

Bli smartere i problemløsning

Forstanden og datamaskinen - Problemløsning i informasjonsalderen

Forfatter

David Moursund
International Society for
Technology in Education
1787 Agate Street
Eugene, Oregon 97403
Copyright © David Moursund 1990, 1993, 2004

Originaltittel

Boken er skrevet på engelsk (USA), og har originaltittelen "Getting Smarter At Solving Problems". Boken deles på Internett under CC-lisens for ikke-kommersiell bruk. Boken er opprinnelig skrevet i 1990, men bearbeidet i 2004 av forfatteren, for å være bedre tilpasset endringene i utvikling.

Om forfatteren

Dave Moursund har undervist og skrevet om datamaskiner og utdanning siden 1963. Han er professor ved Universitetet i Oregon ved "College of Education". Han er leder av både et mastergradprogram og et doktorgradsprogram i "Computers in Education".

Noe av Dr. Moursunds viktigste arbeid:

- Forfatter eller medforfatter av 25 bøker og en rekke artikler.
- Leder av "Department of Computer Science", Universitetet i Oregon, 1969-1975.
- Leder av "Association for Computing Machinery's Elementary and Secondary School Subcommittee", 1978 - 1982.
- Grunnlegger av "International Council for Computers in Education (ICCE)" i 1979. Denne organisasjonen skiftet navn til "International Society for Technology in Education (ISTE)" in 1989
- Chief Executive Officer, ICCE, 1979 - 1989.
- Executive Officer, ISTE, 1989 – d.d.

Om boken

Denne boken er skrevet som en lærebok i problemløsning. Ved å arbeide seg gjennom de ulike kapitlene, får elever gradvis bedre kompetanse i å løse problemer. Det er viktig å være klar over at boken fokuserer vel så mye på tenkning som på bruk av datamaskiner.

Du kan ha god kompetanse i å bruke datamaskiner, uten å være god til å løse problemer. Boken tar derfor elevene gjennom en reise i problemløsningens verden, og presenterer en rekke praktiske øvelser for å bli bedre problemløsere.

Innholdet i denne boken kan gjennomføres som et kurs, hvor elevene leser og deretter gjør arbeidsoppgaver alene og sammen. Gjennom hele boken presenteres skriveaktiviteter knyttet til lærestoffet. De er viktig at elevene gjennomfører disse aktivitetene og lager seg

egne notater fortløpende. Egne notater er en av hovedideene i boken, for å bli en smartere person.

Boken er omfattende, og ikke knyttet til noe bestemt fag. Det vil kreve mye tid å arbeide seg gjennom den. Den egner seg imidlertid godt til selvstudium, men det anbefales at 2- 4 elever arbeider parallelt med stoffet, slik at felles drøftinger som foreslås i boken, kan gjennomføres i grupper på minimum 2 elever. Det anbefales at lærere med ulike fag samarbeider på tvers og følger opp elevenes arbeid med boken. Elevene vil ha nytte av boken i alle fag.

Derfor kan en hensiktsmessig strategi være at ulike lærere som har de samme elevene gir lekser i å arbeide med boken over en periode på 1 – 2 måneder eller mer. Dette krever at disse lærerne samarbeider om oppfølging og lekser. Det bør også gjøres noen periodevise oppsummeringer i klassen, slik at ikke alt overlates til selvstudium.

Oversettelse og bearbeidelse

David Moursund har gitt NDLA tillatelse til å oversette og bruke boken, også bearbeidet. Innholdet er for det meste oversatt slik det er skrevet av Moursund, men bearbeidelser er likevel gjort av Sigurd Alnæs. Dette er gjort for å tilpasse innholdet bedre til norske forhold. I denne norske utgaven er det også utarbeidet noe nytt stoff, blant annet knyttet til kapittel 10.

Fullversjon og kortversjon

Vi er klar over at mange vil synes at det blir krevende å arbeide seg gjennom fullversjonen. Det er nesten umulig å styrke sin egen kompetanse i problemløsning uten å arbeide hardt. Derfor anbefales det å arbeide med fullversjonen. Hensikten med boken er å modnes og lære å tenke på sine egne tenkevaner. Det er påkrevet at elever legger ned mye arbeid og gjennomfører skrive- og tenkeøvelsene som presenteres underveis. Etter hvert som elevene leser og gjør aktiviteter, vil de finne ut at det faktisk også er ganske morsomt.

Likevel vil det bli laget en kortversjon på norsk, for å imøtekomme de som vil legge ned mindre arbeid. Kortversjonen vil gi dårlige muligheter for å skape en god forståelse av lærestoffet.

Fullversjonen vil bli publisert både som et komplett dokument for nedlasting, og som et sett av nettsider som elevene kan bla seg gjennom med navigasjonspiler. Da kan elever og lærere velge om de vil lese på skjerm eller papir. Å lese boken på skjerm er absolutt mest miljøvennlig.

Kortversjonen vil kun bli publisert som nettsider.

Innhold

Bli smartere i problemløsning	1
Forstanden og datamaskinen - Problemløsning i informasjonsalderen	1
Forfatter	1
Originaltittel	1
Om forfatteren	1
Om boken	1
Oversettelse og bearbeidelse.....	2
Fullversjon og kortversjon.....	2
Innhold	3
Forord	7
Fokus	7
Datamaskinen er et hjelpemiddel for tenking	7
Annen hjelp til forståelse	7
Du vil trenge en notatbok	8
Hva handler denne boken om?	8
Oppsummering.....	8
Kapittel 1- Introduksjon til denne boken	9
Fokus	9
Noen hjelpemidler for læring.....	9
Skriv i notatboken	9
Skriving hjelper for læring	10
Verden er i endring.....	10
Lær å tenke.....	10
Problemløsning.....	12
Målet med denne boken	13
Forskjellige typer kunnskap.....	13
Oppsummering av kapittel 1	14
Aktiviteter til kapittel 1	14
Kapittel 2 – Du er en smart person	15
Fokus	15
Du er smart.....	16
Hva vil det si å være smart?	16
En smart person	16
Ulike problemtyper	17

Tenke om egen tenkning.....	18
Oppsummering av kapittel 2.....	18
Aktiviteter til kapittel 2	18
Kapittel 3 – Hva er et problem?	21
Fokus	21
Hva legger du i ordet "problem"?	21
Hva er din definisjon av et problem?	21
Halloween.....	22
En definisjon av begrepet "problem"	23
Et godt definert problem.....	24
Brainstorming.....	24
Tydelig kommunikasjon.....	25
Oppsummering av kapittel 3.....	25
Aktiviteter i kapittel 3.....	25
Kapittel 4 – En firetrinnsplan for å løse et problem.....	27
Fokus	28
Introduksjon	28
En plan med fire trinn for å angripe problemet.....	28
Et eksempel på bruk av planen	28
Mer om firetrinnsplanen	30
Ikke finn opp kruttet.....	30
Bygg på tidligere arbeid fra andre eller deg selv.....	31
Skolefagene	31
Oppsummering av kapittel 4.....	32
Aktiviteter i kapittel 4.....	32
Kapittel 5 – Strategier for problemløsning.....	33
Fokus	33
Strategier for å bli bedre til å løse problemer.....	34
Det er mange strategier	34
"Bryt ned" - strategi	35
"Bygg opp" – strategi	36
Primitiver	36
Kalkulatoren som primitiv.	37
Datamaskinen som primitiv	37
Oppsummering av kapittel 5.....	38

Aktiviteter i kapittel 5.....	38
Kapittel 6 – Bli bedre til å tenke.....	40
Fokus	40
Introduksjon	41
Skolen	41
Hva er tenkning?	41
Hjerneteori	42
Utholdenhet i problemløsning	43
Tenkevane 1: Å forholde seg til uklare problemer.....	43
Tenkevane 2: Å se andres perspektiv.....	44
Tenkevane 3: Å verdsette utdanning og god tenkning	44
Tenkevane 4: Innstilling og selvtillit	45
Tenkevane 5: Lete etter mål	45
Tenkevane 6: Se gjennom målene	46
Tenkevane 7: Ressurser.....	46
Tenkevane 8: Begrensninger	47
Tenkevane 9: Vurdere planen	47
Tenkevane 10: Forutse resultater	48
Tenkevane 11: Er problemet løst?	48
Tenkevane 12: Uventede sideeffekter	49
Oppsummering av kapittel 6.....	50
Aktiviteter i kapittel 6.....	50
Kapittel 7 – Overføring av læring	52
Fokus	52
Introduksjon	53
Litt hjerneteori	54
Betydningen av ord	55
Å lære for nær og fjern overføring.....	56
Et eksempel på type 1 og type 2	57
Konsepter og prosesser.....	58
Lære å bli bedre på fjern overføring	58
Oppsummering av kapittel 7.....	60
Aktiviteter i kapittel 7.....	60
Kapittel 8 – Modellering.....	62
Fokus	62

Introduksjon	62
Mentale modeller.....	62
Verbale modeller.....	63
Skriftlige modeller	64
Litt historie om skriving.....	64
Skriftlige modeller hjelper hjernen	65
Skalamodeller.....	66
Matematiske modeller.....	66
Datamaskinmodeller	68
Hvorfor er datamaskiner så viktige?	68
Mer om datamaskinmodeller.....	69
Oppsummering av kapittel 8.....	70
Aktiviteter i kapittel 8.....	70
Kapittel 9 – Vanlige digitale verktøy	72
Fokus	72
Introduksjon	73
Nytte av vanlige dataprogrammer	74
Et tilbakeblikk på hovedideene ved datamaskinmodellen	75
Tekstbehandleren.....	75
Database.....	76
Typer av databaser.....	77
Grafiske programmer	78
Regneark.....	79
Nettverkstjenester	80
Oppsummering av kapittel 9.....	80
Aktiviteter i kapittel 9.....	81
Kapittel 10 – Datamaskinsystemer.....	82
Fokus	82
Introduksjon	83
Hva kan en datamaskin gjøre?	83
Litt historikk om maskinen	84
Dataprogrammer.....	85
Programmeringsspråk	86
Eksempler på dataprogram	87
Å gjøre datamaskiner lettere å bruke	92

Noen tanker om å være programmerer.....	93
Oppsummering av kapittel 10.....	94
Aktiviteter.....	94
Ordliste	96

Forord

Fokus

Hvert kapittel starter med *Fokus*. Hensikten er å gi deg en rask oversikt over hva kapittelet handler om. Bruk noen minutter på å studere punktene. Det vil hjelpe deg å lære innholdet i hvert kapittel bedre. Her er fokus for forordet:

- Hjelpemidler for tenking.
- Forståelse av hvordan du lærer.
- Bruk av datamaskiner for å løse problemer.

Datamaskinen er et hjelpemiddel for tenking

Dette er en bok om hvordan datamaskiner kan hjelpe deg å løse problemer. Datamaskiner er bare verktøy. Du må fremdeles bruke hjernen og tenke ut det meste selv!

Boken handler derfor også mye om hvordan du tenker og lærer, og hvordan du kan bli bedre til å løse problemer. Hvis du følger ideene i denne boken, vil du bli mye flinkere til å løse problemer både i skolearbeid og i andre situasjoner.

Annen hjelp til forståelse

Lesing, skriving og regning er viktige redskaper for å forstå. Disse ferdighetene hjelper for å løse problemer som tenkning ikke kan klare alene.

Lesing er spesielt viktig. Ved å lese kan du lære nye ting. Du kan løse et problem ved å lese hvordan andre har gjort det før.

En annen viktig ferdighet, er å lære hvordan man lærer. Det er mange måter å lære på. Noen lærer best gjennom å se, andre gjennom å høre, mens andre lærer best gjennom å gjøre ting selv.

Siden vi lærer på forskjellige måte, er det vanskelig for en lærer å undervise på en måte som passer alle. Derfor er det viktig at du selv lærer deg hvordan du lærer, og finner ut hvilken læringsstil som passer deg best. Du skal selv ta ansvar for din læring.

Hvilken læringsstil som passer deg best, vil avhenge av hva du skal lære. Tenk på sport, og hvordan du blir bedre i sport. Tenk på venner, og hvordan du lærte å skaffe deg venner, og hvordan du ble bedre på å skaffe deg venner. Tenk på hvordan du lærte å lese, og hvordan du ble bedre til å lese. Tenk på matematikk, og hvordan du ble bedre til å løse matematiske problemer.

Tenk på hva du gjorde akkurat nå. Du startet å tenke på hvordan du lærer. Du er en unik person. Ingen andre er lik deg. Du er den eneste personen som virkelig kan kjenne deg selv. Du er den eneste personen som kan lære hvordan du lærer best.

En av de viktigste tingene du kan lære, er hvordan du lærer. Hvordan arbeider hjernen din når du lærer nye ting? Hva er viktig for å lære ting fort og godt? Hva hjelper deg å huske viktige ideer? Hva hjelper deg å bli en bedre problemløser?

Du lærer nye ting hver eneste dag. Dette betyr at du får masse trening i det å lære. Noe av denne læringen skjer uten at du trenger å anstrenge deg. Du ser for eksempel TV, og lærer om et nytt produkt. Du hører en venn bruke et nytt ord, og du begynner selv automatisk å bruke dette ordet.

Annen læring krever mer anstrengelse.

En av grunnene til at vi har et skolesystem, er for å gjøre det enklere for mennesker å lære. Hensikten med denne boken er å bli bedre til å lære. Det er en enkel måte å bli bedre på: Når du lærer, tenk på læring. Studer din egen **læringsprosess**. Finn ut hva som passer best for deg. Eksperimenter, prøv forskjellige synsvinkler. Prøv forskjellig tilnærminger til lærestoffet. Gradvis vil du bli bedre til å lære.

Du vil trenge en notatbok

Tidligere oppdagere gjorde seg notater og skrev dagbøker på sine reiser. Tenk på deg selv som en oppdager når du leser denne boken. Du oppdager innholdet i boken. Du oppdager hvordan hjernen reagerer på det som står her. Du utforsker datamaskiner og hvordan de kan hjelpe deg å løse problemer.

Mange av aktivitetene i denne boken ber deg å gjøre notater mens du leser. Du kan bruke en kladdebok eller en tekstbehandler. Lag deg et dokument i en tekstbehandler før du starter, eller skaff deg en notatbok.

De som studerer læring, har funnet ut at notatskriving hjelper å lære bedre. Dette gjelder for alle fag, både for voksne og unge.

Hva handler denne boken om?

Dette er en bok om problemløsning. Det legges stor vekt på datamaskinen som et hjelpemiddel i problemløsning. Datamaskiner er kraftfulle verktøy. Datamaskiner er spesielt egnet til å hjelpe deg å løse problemer du arbeider med på skolen. Du kan bli mye bedre til å løse problemer hvis du vet hvordan du skal bruke datamaskiner som hjelpemiddel.

Du skal imidlertid ikke forvente mirakler. Det har tatt deg et helt liv å bli så god som du er nå. Denne boken vil hjelpe deg å bli bedre, men regn med at det krever tid og innsats.

Oppsummering

I slutten av hvert kapittel er det en oppsummering av hovedideene i kapittelet. Her er oppsummeringen av forordet:

1. Datamaskinen er et kraftfullt verktøy for problemløsning.
2. Lesing, skriving og regning er også kraftige verktøy.
3. Det er viktig at du lærer hvordan du best lærer.
4. Gjennom å studere og øve kan du bli bedre i problemløsning.

Kapittel 1- Introduksjon til denne boken

Fokus

Du lever i en kompleks verden som endrer seg hurtig. En av hensiktene med skole er å hjelpe deg å lære hvordan du skal takle problemer du vil møte som voksen. Det betyr at du må:

- Lære å lære
- Bli bedre til å løse problemer

Det finnes mange metoder for å bli bedre til å løse problemer. En metode er å notere mens du leser. Denne metoden skal du benytte i denne boken.

Noen hjelpemidler for læring

Hvis du hoppet over forordet, gå tilbake og les det. Forordet forteller deg hva denne boken handler om. Det er viktig at du tenker gjennom hovedideene i forordet, før du leser resten av boken.

Fokus har til hensikt å sette i gang en tankeprosess hos deg om hovedpunktene i hvert kapittel før du begynner å lese. Forskning sier at beskrivelse av mål for et kapittel hjelper deg til å lære innholdet i kapittelet bedre.

Boken inneholder en **ordliste**. Ordlisten inneholder definisjoner av viktige begreper som benyttes i boken. Det er viktig at du benytter disse begrepene i dine notater og når du snakker med andre. Gjør begrepene til en del av ditt språk. Dette vil hjelpe deg å lære de viktige ideene i denne boken. Bruk ordlisten når du vil ha en forklaring av begreper.

Skriv i notatboken

Denne boken er laget for å hjelpe deg å lære noen viktige prinsipper for problemløsning. Du har kanskje hørt uttrykket:

"Du kan lede en hest til vannet, men du kan ikke bestemme om den skal drikke."

Hva betyr dette? Hva har det med lesing av denne boken å gjøre?

Før du går videre, begynn å skrive i notatet. Lag en overskrift hver gang du noterer.

Skriv: Hesten og vannet

Fant du noe svar på spørsmålet om hesten og vannet? Tenkte du ut flere mulige svar? Hvis ikke, les spørsmålene igjen.

1. Tenk på flere mulige svar.

2. Skriv svarene i notatet.

Skriving hjelper for læring

Du kan lære mye av denne boken, men for å lære må du tenke gjennom det du leser. Du trenger å øve deg på ideer. Det vil hjelpe deg mye å skrive ned tankene dine.

Når du leser, skal du lage deg notater om de tankene som er mest viktige for deg.

Verden er i endring

Den verden du lever i, er ganske annerledes enn den verden dine besteforeldre levde i som unge.

Vår tidsalder endrer seg hurtig. Mange mennesker arbeider i dag med informasjon.

Det blir stadig færre som produserer varer, mens det blir stadig fler som arbeider med tjenester og informasjon.

Når du er voksen, vil det være helt andre tjenester enn i dag. Nye typer underholdning, medisiner, klær, musikk. Det vil være nye jobber som vi ikke kjenner i dag. Mange av disse jobbene vil måtte utføres ved hjelp av datamaskiner.

Skriv: Endringer siste 5 år

Tenk over endringer som har skjedd i verden de siste 5 årene.

1. Lag en liste over endringer.
2. Sett en hake ved de endringene som har påvirket deg sterkt.

Lær å tenke

Skoler har mange mål. To av dem er:

1. Hjelp elever til å lære grunnleggende ferdigheter
2. Hjelp elever til å tenke og løse problemer ved å bruke informasjon og grunnleggende ferdigheter

Det kan se enkelt ut.

For det første må du lære deg noe grunnleggende ferdigheter og informasjon.

Du tenker kanskje

- Hvilken informasjon er viktig å lære?
- Er det viktig å kunne huske navn på alle stater i USA?
- Hva om jeg glemmer noe av det jeg skulle huske?
- Er det akseptert å slå opp i en bok eller på nettet i stedet for å huske?

Videre tenker du kanskje

- Hvilke ferdigheter er egentlig viktige?
- Er det viktig å ha en pen håndskrift?
- Er det viktig å kunne skrive fort på en datamaskin?

For det andre må du lære deg å tenke og løse problemer ved å bruke grunnleggende informasjon og grunnleggende ferdigheter som lese, skrive, regne.

Du tenker kanskje

- Hva innebærer det å tenke?
- Hvordan kan jeg bedømme om jeg er god til å tenke?
- Hvordan henger tenkning og problemløsning sammen?
- Kan datamaskiner tenke?

Det er altså ikke så enkelt likevel. La oss se på noen eksempler.

Ta for eksempel skrijving. Du må huske hvordan ordene staves, og hvilke regler som gjelder for rettskriving.

Skrijving forutsetter også tenking. Du må kunne organisere og uttrykke dine tanker.

Skriv: Bli bedre til å skrive

Hvordan blir man bedre til å skrive? Skal du bruke tid på å lære deg rettskriving? Eller skal du bruke tid på å lære deg å organisere og uttrykke tankene dine? Hva er best for akkurat deg? Tror du svaret er det samme for alle elever? Du har brukt en tekstbehandler. Den har stavekontroll. Tror du bruk av stavekontroll kan ha betydning for hvor mye tid du bruker på å lære deg rettskriving?

Skriv noen tanker om det.

Et annet eksempel er matematikk.

I matematikk har du pugget mange regler og formler. Noe av det viktigste vi pugger, er gangetabellen.

	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9
2	2	4	6	8	10	12	14	16	18
3	3	6	9	12	15	18	21	24	27
4	4	8	12	16	20	24	28	32	36
5	5	10	15	20	25	30	35	40	45
6	6	12	18	24	30	36	42	48	54
7	7	14	21	28	35	42	49	56	63
8	8	16	24	32	40	48	56	64	72
9	9	18	27	36	45	54	63	72	81

Du har pugget regler for å gange, dele, subtrahere, addere. Du har brukt mange år på å trene på disse ferdighetene.

Skriv: Matematikk

En kalkulator kan hjelpe deg med addisjon, subtraksjon, multiplikasjon og divisjon. Det tok lang tid å lære hvordan du skulle regne lang divisjon med papir og blyant. Det tar bare noen minutter å lære hvordan du skal utføre samme divisjon på en kalkulator. Hvorfor skal du bruke skoletid og krefter på å lære lang divisjon med papir og blyant?

Skriv noen tanker om det.

Det er ingen enkle svar.

Andre spørsmål er enda vanskeligere.

En datamaskin kan løse mange av problemene vi jobber med på skolen. Anta at en datamaskin kan løse et problem du arbeider med på skolen. Hva skulle du da lære om å løse dette problemet?

Mens du tenker på dette spørsmålet, skal du være klar over at det samtidig utvikles nye og bedre datamaskiner og dataprogrammer. Bør du bruke tid på skolen til å lære deg noe som datamaskiner kommer til å løse de neste årene?

Disse spørsmålene er veldig viktig å tenke gjennom. Bruk litt tid på å skrive om dem.

Skriv: Problemer som datamaskiner kan løse

Vær oppmerksom på at dette spørsmålet ikke har noe korrekt svar.

"Bør du bruke tid på skolen til å lære deg noe som datamaskiner kommer til å løse de neste årene?"

Skriv din egen mening og dine argumenter.

Problemløsning

Lærere vet at den beste måten å forberede deg på fremtiden, er å hjelpe deg å bli bedre til å lære, tenke og løse problemer.

Når du er god til å lære, vil du kunne lære nye ting som fremtiden vil bringe.

Når du er god til å tenke, vil du kunne tenke klart om ting i fremtiden.

Når du er god til å løse problemer, vil du kunne løse problemer også i fremtiden.

Det er mange veier for å bli en god problemløser.

1. Lær deg å snakke og lytte.
2. Lær å lese og skrive.
3. Lær deg å regne.

I denne boken drøftes to andre veier for å bli en god problemløser.

1. Strategi for å angripe problemer.
 - a. Del opp store problemer i mindre problemer som er lettere å løse.
 - b. Tegn et bilde eller et diagram som hjelper deg å se hva som skjer.
 - c. Bruk "brainstorming" enten alene eller sammen med andre.
 - d. Lag deg notater (slik du gjør når du leser denne boken).
 - e. Lær deg å bruke datamaskinen riktig.
2. Verktøy for å arbeide med problemer.
 - a. Hammer, kniv, mikroskop, plog.
 - b. Telefon, kommunikasjonsradio, tog, fly.
 - c. Datamaskin og nettverk.

Skriv: Verktøy

Tenk på ulike verktøy du kan bruke til å løse forskjellige problemer. En kam og en tannbørste er verktøy.

Lag en liste over noen verktøy og en beskrivelse av hvilke problemer de er nyttige for å løse.

Målet med denne boken

Denne boken har to klare mål:

1. Å hjelpe deg til å bli en bedre problemløser.
2. Å hjelpe deg til å forstå hvordan datamaskinen er et verktøy når du løser problemer.

Dette er store mål. Du trenger å trene på ideene, hvis du skal utvikle deg. Du må trene hver gang du støter på et problem.

Forskjellige typer kunnskap.

Det er stor forskjell på å lære noen grunnleggende fakta, og hvordan man skal bruke fakta.

Forsøk å svare på disse grunnleggende fakta:

1. Columbus oppdaget Amerika i år _____.
2. Jorda går rundt en stjerne som vi kaller _____.
3. Den første presidenten i USA het _____.

Forsøk så å svare på spørsmål som gjør bruk av sammenhengen mellom fakta:

4. Columbus oppdaget Amerika i 1492. Hva ville han oppnå med sin reise, og hvorfor var det viktig å oppnå dette?
5. Jorda går rundt en stjerne vi kaller sola. Hvordan vet vi det?
6. Den første presidenten i USA var George Washington. Hvilken prosess førte til at han ble president, og hvorfor ble han ikke konge?

Spørsmål 1 – 3 spør etter rene fakta, og om du kan dem. De spør ikke etter sammenhenger og forklaringer. Spørsmålene 1 – 3 tester **faktakunnskap**. For å svare må du huske detaljer.

Spørsmål 4 – 6 spør etter sammenhenger. For å svare må du forstå sammenhengen mellom flere underliggende fakta. Spørsmål 4 – 6 tester **forståelse** av sammenhenger mellom fakta.

Noen mennesker har spesielle evner og kan huske en stor mengde fakta. Spør Egil Drillo Olsen om det høyeste fjellet i et hvilket som helst land, og han svarer deg antakelig riktig.

Hjernen har en naturlig tendens til å glemme store mengder fakta. Bøker og datamaskiner glemmer ikke, så lenge dataene er lagret.

Elever må vite hvordan de skal slå opp fakta i datamaskiner og bøker, og må derfor ha ferdigheter i å søke etter informasjon.

Hvis du skal arbeide med spørsmål 4 – 6, kan du slå opp de fakta du trenger og utforske sammenhengen mellom dem. Du arbeider da med mer **komplekse spørsmål** enn faktaspørsmålene i oppgave 1 – 3, og det krever mer tenking og forståelse av sammenhenger.

Det samme gjelder for problemløsning. En person har et problem dersom han ikke kan nå sitt ønskede mål umiddelbart. Hvis den samme problemsituasjonen gjentar seg, kan samme strategi kanskje være til nytte.

Dessverre gjelder ikke det for alle problemer, siden problemer ofte er ulike av natur. Antall ulike problemer er ubegrenset. Å lære seg å løse problemer ville være enkelt dersom alle problemer var like av natur. Slik er det dessverre ikke.

Derfor er det bedre å lære seg gode ferdigheter i å takle nye og uvante situasjoner. Dette er hva problemløsning handler om.

Oppsummering av kapittel 1

Fire metoder for å bli en bedre problemløser

1. Lær lesing, skriving og regning.
2. Lær ulike strategier for å løse problemer.
3. Lær å bruke verktøy, som datamaskiner.
4. Praktiser problemløsning.

Aktiviteter til kapittel 1

Hvert kapittel avsluttes med noen aktiviteter. Hensikten med aktivitetene må du finne ut selv. Å finne ut det, vil gi deg trening i å tenke, og dermed trening i å løse problemer. Hvis du gjør notater, lærer du bedre.

Aktivitet 1.1 – Skriv: Noe jeg tenkte på i kapittel 1

Var det noe spesielt du begynte å tenke på når du leste kapittelet? Kanskje du tenkte på hvordan du lærer, eller på dine egne ferdigheter i å løse problemer?

Skriv noen linjer i notatet om det du tenkte på.

1. Skriv tittel: "Noe jeg tenkte på i kapittel 1"
2. Skriv og forklar hva du tenkte
3. Forklar hvorfor dette interesserer deg. Hvorfor tenkte du på det?
4. Hva har det med livet utenfor skolen å gjøre?
5. Er det noe mer du tenker?

Aktivitet 1.2 – Del ideen

Finn en elev å snakke med. Begge velger en idé i kapittelet som hver synes er viktig. Bruk ett minutt hver til å fortelle den andre om ideen. Tips til hva dere kan snakke om:

- Hvorfor er det viktig for deg?
- Hvilken betydning har det for ditt liv?

Når dere snakker sammen bør dere bruke begreper som

- Grunnleggende ferdigheter
- Faktakunnskap
- Forståelse av sammenhenger
- Strategier

Aktivitet 1.3 – Lag en quiz som tester faktakunnskap

Lag 2 spørsmål. Du skal teste faktakunnskap.

Prøv quizen på noen elever. Det skal være tekstsvar. Bare ett riktig svar til hvert spørsmål.

Aktivitet 1.4 – Lag en quiz som tester forståelse

Lag 2 spørsmål. Presenter noe fakta og still spørsmål som krever tenking og vurdering av disse fakta og andre fakta som kanskje må slås opp i en bok eller ved et søk. Prøv quizen på noen elever. Det skal være tekstsvar, og ingen fasit.

Aktivitet 1.5 – Tenk og skriv

Tenk på følgende spørsmål:

På hvilke måter er en tekstbehandler et bedre verktøy enn penn og papir?

Skriv noen svar.

Tenk så på følgende spørsmål:

På hvilke måter er en tekstbehandler dårligere verktøy enn penn og papir?

Skriv noen svar.

Aktivitet 1.6 – Tenk mer

Kapittelet snakker om faktakunnskap og om forståelse av sammenhenger mellom fakta.

Hva er mest viktig for deg? Forklar hvorfor.

Hva er du best på?

Skriv.

Aktivitet 1.7 – Foreslå tiltak for å bli en bedre problemløser

Det er mange måter å bli en bedre problemløser på. Kapittelet foreslår flere. Her er noen til:

1. Øv deg på å lage problemstillinger.
2. Lær deg mye om et fag. Det vil hjelpe deg å løse problemer i dette faget.
3. Ha en positiv holdning til problemer. Styrk troen på deg selv.
4. Lær deg å arbeide hardt og ikke gi opp.
- 5.

Foreslå flere tiltak på listen. Finn ut hvilke tiltak som er best for deg.

Hvis du skulle plukke ut ett tiltak som viktigst for deg, hvilket ville du velge? Hvorfor?

Hva har andre elever plukket ut som viktigst for dem? Spør!

Aktivitet 1.8 – Tenk på en god tenker

Tenk på en person som er en god tenker. Hvordan vet du at denne personen er en god tenker? Skriv hvorfor. Skriv gjerne hva som gjør personen til en god tenker.

Kapittel 2 – Du er en smart person

Fokus

Å tenke før vi handler, er viktig i problemløsning. De fleste er gode til å se ulike alternativer. I dette kapittelet har vi fokus på:

1. Hva vil det si å være smart?

2. Lære å tenke på din egen tenkning.
3. Lære å bli bevisst på hvilke problemer som er lette for deg.

Du er smart

Denne boken kan hjelpe deg å bli smartere. Vi vil straks presentere en test. Den vil du sikkert gjøre det bra på. Det vil bevise at du er smart nok til å forstå at denne boken kan gjøre deg enda smartere. Her er testen:

Skriv: Hyggelige ting jeg gjør etter skoletid

1. Skriv tittelen i notatet.
2. Skriv en liste over tre hyggelige ting du gjør etter skoletid.
3. Strek under den tingen på lista som du liker minst.
4. Til slutt skriver du ned en ny ting du kunne ha lyst til å gjøre og som du tror vil være hyggeligere enn det du streket under.

Du besto antakelig testen. Var det en vanskelig test? Hva var vanskelig?

For å bestå testen måtte du kunne lese, skrive og følge instruksjoner. Du måtte kunne tenke på hva som kan komme til å skje i framtiden. Du måtte lage et alternativ til hva som kunne skje. En person som kan alt dette, er ganske smart.

Hva vil det si å være smart?

Det er ikke lett å definere ordet smart. Det er enda vanskeligere å lage en god test for å se hvor smart en person er. Kanskje du har en katt eller en hund du synes er ganske smart? Kan et dyr planlegge aktiviteter som skal skje gjennom en dag? Hvordan skulle vi teste det?

Skriv: Er dyr smarte?

1. Skriv noen svar på spørsmålene under.
2. Hvordan kan du avgjøre om en katt er smart?
3. Planlegger katter dagen sin?
4. Er katter smartere enn hunder?
5. Hvordan kan du teste for å bekrefte eller avkrefte svarene dine?

Skriv: Din definisjon av en smart person

- Lag en definisjon av en smart person. Skriv i notatet.

En smart person

I denne boken vil vi bruke en enkel definisjon av en smart person.

Du er en smart person hvis du kan forstå og løse de problemene du møter. Jo flere problemer du klarer å løse, dess smartere er du. En smart person kan løse mange ulike typer av problemer. En smart person kan løse vanskelige problemer, og kan lære å løse nye problemer.

Hvor smart du er, avhenger av hvor god du er til å løse problemer. Du blir smartere dersom du lærer deg å løse ulike **problemtyper**. Det er mange veier.

Noen ganger handler det om å ta avgjørelser.

- "Hva skal jeg ha på meg i dag?"
- "Hva skal jeg spise til middag?"
- "Hvilken bok skal jeg lese for å gjøre det bra på prøva?"

Du vil komme til å tenke på flere mulige avgjørelser. Du vil vurdere følgene av å gjøre et valg. Å vurdere ulike valg og hva de fører til, er én viktig problemtype.

Ulike problemtyper

Det er mange ulike problemtyper som for eksempel:

- Ditt forhold til venner.
- Å finne veien til et helt nytt sted.
- Å bli bedre i en sport.
- Å løse oppgaver i matematikk.
- Å velge riktige trekk i sjakk.
- Å endre innstillinger på mobiltelefonen.
- Å fjerne virus fra en datamaskin.
- Å bruke riktig dataprogram til en arbeidsoppgave.
- Å bli en bedre problemløser.

En person kan være veldig flink til å løse én problemtype, men ikke så flink med andre typer. For eksempel kan en person være mester i sjakk, men ganske dårlig til å omgås andre mennesker. En person kan være flink med skolefag, men ikke så flink med ting utenfor skolen.

Noen er gode i sport, men kanskje ikke så gode i musikk.

Tenk på deg selv. Hvilke problemtyper er du god til å løse? Husk at det å skrive mens du leser, er en viktig strategi for å lære bedre.

Skriv: Problemer som er lette for meg.

1. Skriv ned en problemtype som er lett for deg.

Tips: Tenk på et problem som du har løst lett. Kanskje du klarte å spare penger til noen klær du hadde lyst på. Altså er du god på å løse økonomiske problemer.

Tenk på en annen problemtype som du løser greit. Velg en type som er helt forskjellig fra det første.

Tips: Du tenker kanskje på en ny venn du fikk i sommer. Altså er du god til å skaffe venner. Å være god til å skaffe venner er en helt annen problemtype enn å være økonomisk.

Skriv det du har valgt.

Tenke om egen tenkning

Pedagoger har tenkt mye på hvilke metoder som kan hjelpe elever å bli bedre til å løse problemer. En av de beste metodene er å **tenke om sin egen tenkning**.

Du husker testen i begynnelsen av kapittel 2. I testen måtte du tenke på hva du likte å gjøre etter skoletid. Du måtte tenke på hvilken aktivitet som var minst hyggelig. Du tenkte om din egen tenkning!

Å tenke om sin egen tenkning er så viktig, at det er laget et eget ord for det. Det kalles **metakognisjon**.

Hvis du øver på å tenke om din egen tenkning, praktiserer du metakognisjon og blir bedre til å tenke. Når du blir bedre til å tenke, blir du bedre til å løse problemer. Du blir smart.

I notatet skrev du to problemtyper som du er god til å løse. Du tenkte på dine egne ferdigheter i å løse problemer. Du praktiserte metakognisjon! Se på det du skrev. Tenk på det du skrev. Tren mer på metakognisjon.

Når læreren underviser i klassen, hender det at du drømmer deg bort. Du begynner å tenke på noe helt annet, som en venn du skal treffe etter skoletid.

Så tenker du: "Jeg lurer på hva læreren sa, som fikk meg til å begynne å tenke på at jeg skal treffe min venn etterpå?"

Du tenker på din egen tenkning. Du praktiserer metakognisjon.

Skriv: Praktisering av metakognisjon

1. Les i notatet ditt hva du skrev under overskriften "Problemer som er lette for meg."
2. Hva er felles for disse to problemene?
3. Hva er ulikt for disse to problemene?
4. Skriv noen setninger om hvorfor disse to problemene er lette å løse for deg.

Oppsummering av kapittel 2

Metakognisjon, eller å tenke om egen tenkning, er veldig viktig. Det er en viktig vei for å bli bedre til problemløsning.

- Du vet nå noen problemtyper du er god til å løse.
- Du har fått ferdigheter i å tenke på hvorfor noen problemtyper er enklere å løse for deg enn andre.

Aktiviteter til kapittel 2

Aktivitet 2.1 – Noe jeg tenkte på i kapittel 2

Var det noe spesielt du begynte å tenke på da du leste kapittelet? Kanskje du tenkte på hvorfor du er god til noen ting, og mindre god til andre ting? Kanskje du tenkte på dagdrømming og om dagdrømming er metakognisjon? Kanskje du tenkte at å tenke om egen tenkning gjør deg forvirra?

1. Skriv tittel: "Noe jeg tenkte på i kapittel 2"

2. Skriv og forklar hva du tenkte.
3. Forklar hvorfor dette interesserer deg.
4. Hvorfor tenkte du på det?
5. Hva har det med livet utenfor skolen å gjøre?
6. Er det noe mer du tenker på?

Aktivitet 2.2 – Del ideen

Finn en elev å snakke med. Begge velger en idé i kapittelet som hver synes er viktig. Bruk ett minutt hver til å fortelle den andre om ideen. Tips til hva dere kan snakke om:

- Hvorfor er ideen viktig for deg?
- Hvilken betydning har den for ditt liv?

Når dere snakker sammen bør dere bruke begreper fra kapittelet:

- Metakognisjon
- Problemtipe
- Smart person

Aktivitet 2.3 – Lag en quiz som tester faktakunnskap fra kapittelet

Lag 2 spørsmål. Du skal teste faktakunnskap fra dette kapittelet.

Det skal være tekstsvar. Bare ett riktig svar til hvert spørsmål.

Spørsmålene skal være så enkle at en som ikke har lest kapittelet kan svare.

Prøv quizen på noen elever.

Aktivitet 2.4 – Lag en quiz som tester forståelse av kapittelet

Lag 2 spørsmål. Du skal teste forståelse av kapittelet.

Presenter noe fakta fra kapittelet og still spørsmål som krever tenkning og vurdering av fakta.

Testpersonene kan slå opp i en bok, gjøre et søk eller lese i dette kapittelet for å lage et svar.

En quiz som tester forståelse, har alltid flere korrekte svar. Du lager ingen svaralternativer.

Spørsmålene skal være så vanskelige at de bare kan besvares av en person som har en god forståelse av innholdet i dette kapittelet.

Prøv quizen på noen elever. Det skal være tekstsvar, og ingen fasit. Diskuter svarene sammen til slutt.

Aktivitet 2.5 – Tenk og skriv

Tenk på følgende spørsmål:

Er en bok smart?

En bok "vet" en masse fakta. En ordbok "vet" hvordan ord skrives.

Skriv noen setninger under overskriften: Er en bok smart?

1. Skriv en setning som argumenterer for at en bok er smart.
2. Skriv en setning som argumenterer for at en bok ikke er smart.
3. Skriv en setning som inneholder din egen mening om spørsmålet.

Aktivitet 2.6 – Tenk mer

Tenk på følgende spørsmål:

Er en maskin "smart"?

Tenk på en heis. Du går inn i heisen, trykker på en knapp til 4.etg. Heisen kjører til 4.etg. Eller hva med en kalkulator? Du vil multiplisere to tall. Du skriver inn ett tall, trykker stjerne, skriver det andre tallet, trykker utfør, og riktig svar vises. Kalkulatoren regner hurtig og uten feil.

Hvis en person regner multiplikasjon hurtig og uten feil, sier vi gjerne at personen er smart eller flink i multiplikasjon.

Skriv noen setninger i notatet under overskriften: Er en maskin smart?

1. Skriv en setning som argumenterer for at en maskin er smart.
2. Skriv en setning som argumenterer for at en maskin ikke er smart.
3. Skriv en setning som inneholder din mening om spørsmålet.

Aktivitet 2.7 – Problemer som er vanskelige for meg

I notatet ditt har du allerede skrevet om "Problemer som er lette for meg". Du ga noen eksempler på problemer som du løser lett.

Denne gangen skal du tenke på noen problemer som er vanskelige for deg.

1. Skriv en overskrift: Problemer som er vanskelige for meg
2. Skriv en problemtype som du ikke er god til å løse. For å finne en problemtype, kan du tenke på et konkret problem som du ikke har klart å løse. Eksempel: Du kjøpte deg en hamburger og en kinobillett for penger du skulle spare til en ny genser. Så hva slags problemtype er dette? Kanskje problemet er at du ikke er så god til å spare penger? Altså et økonomisk problem.
3. Skriv en annen type problem som du ikke er god til å løse. Velg en problemtype som er helt forskjellig fra det første. Kanskje problemet er at du ikke klarer å synge rent?

Aktivitet 2.8 – Bruk av metakognisjon for å løse vanskelige problemer

Denne aktiviteten forutsetter at du har gjort Aktivitet 2.7

Du skulle tenke på to problemtyper som var vanskelig for deg. Her skal du imidlertid tenke gjennom hva det er som gjør disse problemtypene vanskelige for deg.

1. Skriv overskrift: Bruk av metakognisjon for å løse vanskelige problemer.
2. Hva er likheten mellom de to problemene? Lag en liste.
3. Hva er ulikt mellom de to problemene? Lag en liste.

4. Skriv flere setninger som forteller hvorfor disse problemene er vanskelige å løse.

Aktivitet 2.9 – Blir smartere av å gå på skole

Hensikten med skole er å bli smartere. Du skal argumentere for og mot dette.

1. Skriv en liste med argumenter som viser at vi blir smartere av å gå på skolen.
2. Skriv en liste med argumenter som viser at vi ikke blir smartere av å gå på skolen.
3. Oppsummer din personlige mening. Ta utgangspunkt i det du skrev i punkt 1 og 2.

Ekstra:

1. Skriv hva man kunne gjøre for å bekrefte at argumentene i punkt 1 er riktige.
2. Skriv hva man kunne gjøre for å bekrefte at argumentene i punkt 2 er riktige.

Aktivitet 2.10 – Argumenter for og imot

En god tenker kan argumentere begge sider av en sak. En god tenker kan forstå en annen persons oppfatning og synsvinkel. Du kan bli bedre til å tenke ved å argumentere for begge sider av en sak.

1. Tenk på forrige gang du hadde en diskusjon med en annen person.
2. Lag en liste over argumentene fra begge personer
3. Jobb ekstra mye med å lage en god liste over den andre personens argumenter

Du kan gjøre dette i notatet, eller på et løst ark som du kaster etterpå.

Kapittel 3 – Hva er et problem?

Fokus

Dette kapittelet inneholder en firedelt definisjon av begrepet "problem". Definisjonen brukes gjennom resten av boken. Dette kapittelet vil gi deg trening i:

- Metakognisjon. Å forstå meningen ved begrepet "*problem*".
- Hvordan forholde seg til problemsituasjoner hvor problemet ikke er avklart.
- Brainstorming. En viktig strategi for å løse problemer.

Hva legger du i ordet "problem"?

Du bruker sikkert ordet "problem" ofte. Og du har ikke noe problem med å forstå hva det er. Men tror du at alle mennesker har samme oppfatning av hva "problem" betyr?

Anta at du legger noe annet i ordet enn denne boken gjør. Da ville vi ikke forstå hverandre.

Hva er din definisjon av et problem?

Anne er elev som deg. Hun støter på en rekke problemer hver dag. Men hun vet ikke hvordan hun skal løse problemene.

1. Anne våkner og skal på skolen, Hun tenker: "Alle klærne mine er skitne. Hva skal jeg gjøre?"
2. Anne setter på nyhetene og hører om barn som sulter. Hun mister matlysten til frokost.

3. Anne må rekke skolebussen. Hun tenker: "Hvis jeg skal sminke meg, rekker jeg kanskje ikke bussen. Hva skal jeg gjøre?"
4. Læreren ber klassen arbeide med oppgaver på side 98. Anne tenker: "Jeg forstår ikke problemet her? Hva skal jeg gjøre?"
5. Hun snakker med en venninne. Anne sier: "Roger vil treffe meg i kveld. Skal jeg gå ut med han?"
6. Læreren snakker om fyllekjøring. Han sier: "I fjor døde 200 mennesker på grunn av promillekjøring." Anne tenker på kammeraten til broren hennes som ofte kjører i fylla. Hun blir redd.

Tenk på noen problemtyper du møter gjennom en dag. Gjør litt metakognisjon. Prøv å tenke over på hvilken måte dine problemtyper er like og ulike.

Skriv: Min definisjon av ordet problem.

1. Skriv hva begrepet "problem" betyr for deg.
2. Gå sammen med en elev. Begge leser sin definisjon for den andre.
3. Lag en felles definisjon av begrepet "problem" som begge er enige om. Skriv denne felles definisjonen i notatet ditt.

Du synes kanskje øvelsen var dum, men den har tre viktige hensikter:

1. Du klargjør for deg selv hva du legger i begrepet "problem".
 2. Du trener deg på å lytte til andres oppfatninger.
 3. Du må samarbeide for å lage en felles løsning.
- Å klargjøre for seg selv er metakognisjon. Det er viktig i problemløsning.
 - Å lytte til andres oppfatninger gir deg flere **perspektiver**. Det er nødvendig i problemløsning.
 - Å jobbe sammen gir **samarbeidslæring**. Det er nyttig i problemløsning.

Du ser sikkert at det ikke er lett å gi noen god definisjon av begrepet "problem".

Halloween

Her er et eksempel på en firedelt metode for å løse et problem. Metoden kan brukes på alle problemer.

1. Problemsituasjon

Neste torsdag er det Halloween. Anne skal på fest. Men hun har ikke noe kostyme.

2. Målbeskrivelse

Hun bestemmer seg for å skaffe et kostyme før torsdag.

3. Ressurser og begrensninger

Hun lager en liste over **ressurser**:

1. Vi har noen gamle klær hjemme som jeg kan bruke.
2. Jeg har 500 kroner jeg kan kjøpe meg et kostyme for.

3. Jeg har et kostyme fra i fjor.
4. Jeg kan spørre noen venner om de har noe å låne meg.
5. Jeg kan lage meg et kostyme selv.
6. Jeg kan kombinere noen av alternativene i punkt 1 – 5.

Hun lager også en liste over **begrensninger**:

1. Jeg gidder ikke legge for mye arbeid i kostymet.
2. Jeg har ikke mye tid til å jobbe med det.

4. Viktighet

Å ha et kostyme som jeg er fornøyd med, er veldig viktig for meg. Jeg vil ikke skille meg ut.

Skrivingen løste ikke problemet, men organiserte det i fire deler. Problemet er tydelig definert. Dette kalles derfor et **godt definert problem**.

Anne stiller opp problemet i fire kjente deler som gjelder ved all problemløsning:

- Problemsituasjon
- Målbeskrivelse
- Ressurser og begrensninger
- Viktighet for meg

Dette er et godt utgangspunkt for å starte med å løse selve problemet. Dette kalles **problemoppstilling**.

Problemoppstilling krever nøye tenkning. Alle som skal til Halloweenfesten må stille opp problemet med kostymet. For noen er det kanskje ikke noe problem, hvis de allerede har et kostyme de skal bruke. Alle andre stiller opp problemet på sin måte i hodet, eller kanskje på papiret?

Skriv: Hvordan oppstår problemer?

1. Skriv eksempler på problemer du har stilt opp i dag.
2. Har du klart å løse alle disse problemene?
3. Er det noen problemer som ikke lar seg løse?
4. Hvilket problem ville du helst jobbe med av følgende to:
 - a. Et problem læreren gir deg på skolen
 - b. Et problem du stiller opp selv

En definisjon av begrepet "problem"

De fleste er enige om at et problem kan defineres i fire punkter:

1. **Forutsetninger** (Problemsituasjon). Beskriver hvordan ting er, hva som skjer, hva som er kjent. Det er altså alle fakta.
2. **Mål**. Dette er hva vi ønsker å oppnå når vi har funnet en løsning.

3. **Ressurser og begrensninger.** Hvilke ressurser har du? Hva kan du gjøre for å nå målet? Hva kan du ikke gjøre? Hvilke ferdigheter har du? Hvor mye tid? Verktøy?
4. **Eierskap** (Viktighet for meg). Hvilken interesse har du i å løse problemet? Hvilken betydning har det for deg? Du må ha et ønske om å løse problemet.

Skriv: Jeg er god til å addere tall i hodet

Et eksempel på en problemoppstilling er slik:

1. **Forutsetninger:** Jeg har to positive hele tall som ikke er mer enn 3 sifre lange.
2. **Mål:** Å finne summen raskt.
3. **Ressurser og begrensninger:** Jeg må bruke hoderegning, hjelpemidler er ikke tillatt.
4. **Eierskap:** Jeg er god i hoderegning. Jeg trives med å addere korrekt og raskt i hodet. Det imponerer venner. Det er nyttig i mange situasjoner.

Lag slik problemoppstilling for to **Problemer som er lette for meg** fra kapittel 2. Se i dine notater, og velg to problemer du beskrev som lette for deg.

Et godt definert problem

For det første: For å løse et problem, må du ha interesse av det.

For det andre: Du må forstå problemet. Du må derfor stille det opp som beskrevet for addisjonen.

Når dette er på plass, har du et **godt definert problem**.

Hvis dette ikke er på plass, har du et **dårlig definert problem**.

De fleste daglige problemsituasjoner er dårlige definerte problemer. For å løse dem, må du stille opp problemene slik at de blir godt definerte. Du kan gjøre det i hodet, på papir med en tekstbehandler, eller på annen måte.

Brainstorming

Når vi skal stille opp et problem, kan det være nyttig å bruke brainstorming. Skriv ned alle ideer du kommer på, helt **ukritisk**:

1. Lag deg en liste over mulige mål.
2. Lag deg en liste over mulige ressurser.
3. Lag deg en liste over mulige begrensninger.

Dette kan du gjøre alene, eller sammen med andre. Flere kan arbeide sammen med brainstorming, slik at du får opp forslag du kanskje ikke hadde tenkt på selv.

NB! Det er viktig at du ikke tar stilling til forslagene som kommer. Det skal du gjøre etterpå. Alt skal godtas i første omgang. Målet er å lage en lang liste over det du kommer på i farta. Du skal bare storme hjernen, og skrive alt hjernen foreslår.

Skriv: Promillekjøring - brainstorming

Annes lærer snakker om promillekjøring og alle som dør fordi noen kjører i fylla. Anne tenker på kameraten til broren som ofte kjører med promille. Hun blir redd.

Bruk brainstorming:

1. Arbeid først alene. Skriv en liste over alle mulige mål som Anne kan ha for å løse dette problemet. Prøv å lage 10 mulige mål.
2. Arbeid sammen med en elev. Sammenlikne målene dere har notert. Dere skal velge ett mål sammen, som dere kan arbeide for å nå. Velg ett mål.
3. Sammen bruker dere brainstorming for å lage en **ukritisk liste** over alle ressurser og begrensninger knyttet til målet dere valgte. Ikke vær kritisk! Alle forslag skal med.

Tydelig kommunikasjon

Når vi arbeider sammen med store problemer, kan vi dele opp og fordele arbeidet på mange. Hvis problemet ikke er tydelig kommunisert, kan vi risikere at ulike deltakere ikke har samme forståelse av problemet, og dermed jobber mot ulike mål.

Når et problem er tydelig kommunisert, kan mange jobbe mot det samme målet. Vi praktiserer **samarbeidslæring**. Når vi samarbeider om å lære, har vi ikke bare ansvar for egen læring, men også for å hjelpe de vi samarbeider med til å lære.

Et godt eksempel er miljøproblemene i verden, som krever at alle land jobber i samme retning og mot samme mål.

Dette vil bli drøftet mer i kapittel 10, når vi snakker om å programmere datamaskiner slik at de kan hjelpe oss å løse problemer. Hvis vi programmerer feil, vil ikke datamaskinen gi oss den hjelpen vi forventer. Presis og tydelig kommunikasjon er derfor helt nødvendig.

Oppsummering av kapittel 3

Et problem har fire deler:

1. Forutsetninger
2. Mål
3. Ressurser og begrensninger
4. Eierskap

De fleste problemsituasjoner er **dårlig definerte problemer**.

Det krever litt arbeid å omforme et problem til et **godt definert problem**.

Brainstorming er en nyttig strategi for **problemoppstilling**.

Problemoppstillingen må **kommunisere tydelig** til alle hva vi skal løse.

Aktiviteter i kapittel 3

Aktivitet 3.1 – Noe jeg tenkte på i kapittel 3

Var det noe spesielt du begynte å tenke på da du leste kapittelet? Kanskje du tenkte på hvordan skoleproblemer er forskjellige fra problemer utenfor skolen? Kanskje du tenkte på at det er nyttig å snakke med andre elever om skolefag og problemløsning?

1. Skriv tittel: "Noe jeg tenkte på i kapittel 3"
2. Skriv og forklar hva du tenkte.
3. Forklar hvorfor dette interesserer deg.
4. Hvorfor tenkte du på det?
5. Hva har det med livet utenfor skolen å gjøre?

6. Er det noe mer du tenker på?

Aktivitet 3.2 – Del ideen

Finn en elev å snakke med. Begge velger en idé i kapittelet som hver synes er viktig. Bruk ett minutt hver til å fortelle den andre om ideen. Tips til hva dere kan snakke om:

- Hvorfor er ideen viktig for deg?
- Hvilken betydning har den for ditt liv?

Når dere snakker sammen bør dere bruke begreper fra kapittelet:

- Godt definerte problemer
- Dårlig definerte problemer
- Brainstorming
- Forutsetninger
- Mål
- Ressurser og begrensninger
- Eierskap
- Problemoppstilling
- Samarbeidslæring

Aktivitet 3.3 – Lag en quiz som tester faktakunnskap fra kapittelet

Lag 2 spørsmål. Du skal teste faktakunnskap fra dette kapittelet.

Det skal være tekstsvar. Bare ett riktig svar til hvert spørsmål.

Spørsmålene skal være så enkle at en som ikke har lest kapittelet kan svare.

Prøv quizen på noen elever.

Aktivitet 3.4 – Lag en quiz som tester forståelse av kapittelet

Lag 2 spørsmål. Du skal teste forståelse av kapittelet.

Presenter noe fakta fra kapittelet og still spørsmål som krever tenkning og vurdering av fakta.

Testpersonene kan slå opp i en bok, gjøre et søk eller lese i dette kapittelet for å lage et svar.

En quiz som tester forståelse, har alltid flere korrekte svar. Du lager ingen svaralternativer.

Spørsmålene skal være så vanskelige at de bare kan besvares av en person som har en god forståelse av innholdet i dette kapittelet.

Prøv quizen på noen elever. Det skal være tekstsvar, og ingen fasit. Diskuter svarene sammen til slutt.

Aktivitet 3.5 – Brainstorm "om å kjede seg"

Du snakker med en venn. Han eller hun sier: "Jeg kjeder meg. Jeg vet ikke hva jeg skal gjøre i kveld."

Brainstorm en liste med forslag til din venn.

Aktivitet 3.6 – Brainstorm "om å ha for mye lekser"

Du vet hva det vil si å få for mye lekser. Anta at det vil skje i morgen.

1. Lag en liste med mulige mål som kan være aktuelle i dette tilfellet.
2. Velg ut ett mål fra listen.
3. Lag en liste med mulige ressurser og begrensninger knyttet til målet.
4. Velg et annet mål fra listen over forslag til mål.
5. Lag en ny liste med mulige ressurser og begrensninger knyttet til dette målet.

Aktivitet 3.7 – Sur nedbør

Anta at sur nedbør er et problem du er opptatt av.

1. Du skal lage to helt forskjellige godt definerte problemer som har med sur nedbør å gjøre.
2. Derfor skal du tenke på to helt forskjellige mål.
3. For hvert mål skal du lage en liste over ressurser og begrensninger.

Aktivitet 3.8 – Global oppvarming

Anta at du er opptatt av global oppvarming og nedsmelting av polar is.

1. Lag to helt forskjellige godt definerte problemer som har med global oppvarming å gjøre. Husk de 4 punktene: Forutsetninger, mål, ressurser og begrensninger, eierskap.

Aktivitet 3.9 – Godt definert problem uten løsning

Er det mulig at en person kan ha et godt definert problem som ikke lar seg løse? Gi gode argumenter for ditt svar. (Et godt argument for den ene siden av spørsmålet vil være å gi et eksempel på et godt definert problem som ikke har noen løsning.)

Aktivitet 3.10 – Ulike definisjoner av begrepet "problem"

Ulike ordbøker definerer et "problem" på forskjellige måter. Du skal undersøke ulike definisjoner av ordet "problem".

1. Skriv ned definisjonen av begrepet "problem" fra to eller tre ordbøker.
2. Skriv et par setninger om hva som er likt mellom definisjonene.
3. Skriv et par setninger om hva som er ulikt mellom definisjonene.
4. Skriv din mening om hvorfor det er forskjellige definisjoner av "problem".

Aktivitet 3.11 – Hvorfor har vi lover?

Tenk på noen typer av lover. For eksempel vegtrafikkloven. Forklar hvilke problemer denne loven regulerer. For å gjøre dette må du kanskje gi to forskjellige eksempler på godt definerte trafikkproblemer. Forklar hvordan trafikkreglene hjelper til med å løse disse problemene.

Kapittel 4 – En firetrinnsplan for å løse et problem

Hvordan starter du med å løse et problem som du ikke vet hvordan du skal løse? En måte er å skaffe seg en god forståelse av problemet. Deretter lager du en handlingsplan, gjennomfører tiltakene og sjekker om dette løste problemet.

Fokus

Noen viktige ideer:

- Ikke finn opp kruttet på nytt. Bruk arbeid andre har gjort før deg.
- Hvert skolefag fokuserer på hvordan fagets problemer kan løses

Introduksjon

Kapittel 3 snakket om å stille opp et godt definert problem. Men hva er neste skritt?

Det er dessverre ingen enkel vei. Det finnes også problemer som ikke lar seg løse. Prøv å finne to positive tall hvor summen er negativ. Dette er et godt definert matematikkproblem, men har ingen løsning.

Likevel er det mange problemer som lar seg løse. Dette kapittelet foreslår to måter:

1. Lær noen generelle metoder for å angripe problemet.
2. Lær å bygge på det andre har funnet ut.

En plan med fire trinn for å angripe problemet

Det finnes ingen felles metode du kan bruke for å løse alle problemer. Men det finnes en generell plan som er nyttig for nesten alle problemer.

- A. Forstå problemet.
- B. Lag en tiltaksplan.
- C. Gjennomfør tiltakene.
- D. Undersøk om problemet er løst.

Et eksempel på bruk av planen

Vetle er akkurat ferdig med ungdomsskolen. Han skal begynne på videregående, og gleder seg til å få nye venner. Men rommet hans trenger en forandring. Tapetet er litt for barnslig og plakaten har hengt der siden 6.klasse. Det frister ikke å ta med folk hjem. Vetle innser at rommet må pusses opp, men han må ha hjelp. Han kan også trenge en sofa.

Han bestemmer seg for å lage en firetrinns plan for å angripe problemet.

A – Forstå problemet.

Vetle må først lage et **godt definert problem**(se kapittel 3). Han gjennomfører de 4 punktene fra kapittel 3:

1. Forutsetninger:
 - Barnslige tapeter og plakater.
 - Frister ikke å ta med nye venner hjem.
 - Mange gamle saker og ting.
 - Mangler en sofa.
2. Mål:
 - Lage et kult ungdomsrom.

3. Ressurser og begrensninger(Vetles brainstorm):

- Jeg rydder bort gamle ting. (Ressurs)
- Mangler tomkasser. (Begrensning)
- Broren min kan hjelpe meg å tømme rommet. (Ressurs)
- Jeg kan sove hos broren min så lenge. (Ressurs)
- Foreldrene mine har dårlig tid. (Begrensning)
- Foreldrene mine betaler tapet og sofa. (Ressurs)
- Vi har en fin hylle jeg kan få. (Ressurs)
- Faren min er flink til å tapetsere og male.(Ressurs)
- Det er bare 14 dager til skolestart (Begrensning)

4. Eierskap:

- Vetle trives ikke med å ha et barnslig og stygt rom. Han vil ikke ta med folk hjem før rommet er fikset. Så Vetle har en sterk interesse av å løse problemet. Han har absolutt eierskap.

Merk! Han har nå stilt opp problemet og har derfor et godt definert problem.

B – Planlegge og bestemme tiltak

Vetle tenker på flere måter å angripe problemet på. Til slutt bestemmer han seg for å lage en plakat med alle oppgaver som må gjennomføres, og henge den opp på kjøkkenet. Plakaten er en liste over **tiltak**, hvem som skal gjøre dem, og frist for hvert tiltak. Han forteller broren og foreldrene om plakaten, og ber dem se hva de skal gjøre.

Utdrag av plakaten:

Tiltak	Ansvar	Frist	Ferdig
Skaffe tomkasser og søplesekker	Mor	01.08.09	
Rive ned plakater	Bror	05.08.09	
Rydde bort ting jeg ikke vil ha på rommet lenger	Vetle	03.08.09	
Flytte ut seng, skrivebord og andre møbler	Far, Vetle, bror, mor	05.08.09	

C – Gjennomføre tiltak

Vetle tar et raskt møte med familien, og går gjennom alle tiltakene på plakaten. Sammen blir de enige om frister og ansvar. Dette skrives på plakaten. Arbeidet starter og tiltakene gjennomføres.

D – Undersøke om problemet er løst

Vetle er nøye på å sjekke underveis om hver og en har gjennomført de tiltakene de har ansvar for innenfor fristen. Far er ganske opptatt, og klarer ikke å tapetsere ferdig til fristen som er satt. De blir enige om å forskyve fristen, og til slutt er rommet tapetsert ferdig.

Men de oppdager at gardinene også må fornyes, og at det ikke er så lett å finne en sofa som passer. Problemet er nesten løst. Eller er det løst?

Skriv: Oppussingsproblemet

Anta at far ikke rekker sin del av avtalen. Han er opptatt med møter på kveldene. Anta videre at butikken er utsolgt på akkurat den sofaen Vetle ville ha.

1. Skriv tittel "Oppussingsproblemet".
2. Skriv mulige løsninger på at far ikke rekker å tapetsere.
3. Skriv mulige løsninger på at sofaen er utsolgt.
4. Skriv mulige løsninger på hvordan Vetle skal skaffe nye gardiner.
5. Tenk ut en helt annen tiltaksplan for Vetle. Vetle kan for eksempel tapetsere selv, eller kan moren hans gjøre det?
6. Finn en annen elev i klassen, og diskuter mulige tiltaksplaner.

Mer om firetrinnsplanen

En slik plan er til hjelp for å komme i gang med å løse et problem.

Du må ikke føle deg bundet av den, hvis den ikke er til hjelp.

Kanskje du oppdager underveis at du ikke har forstått problemet. Da må du antakelig droppe planen, og starte på nytt. Kanskje du må lage en annen plan.

Første trinn - Forstå problemet

Du stiller opp problemet og lager et **godt definert problem**.

Andre trinn - Planlegge og bestemme tiltak

Du tenker på hvordan du kan anvende ressurser og unngå problemer som har med begrensninger å gjøre. På bakgrunn av dette lager du en **tiltaksplan**.

Tredje trinn - Gjennomføre tiltak

Her gjennomfører du tiltakene du har planlagt.

Fjerde trinn - Undersøke om problemet er løst

I siste trinn undersøker du resultatet og sjekker om målet ditt er nådd. Det er et tenketrinn, men også et vurderingstrinn. Dette trinnet har to deler:

1. Sjekk om problemet er løst.
2. Sjekk om nye problemer har oppstått.

Skriv: Problemer skaper nye problemer

Skriv noen eksempler på hvordan problemer du har arbeidet med har skapt nye problemer som måtte løses.

Ikke finn opp kruttet

La oss si at matematikklæreren gir deg et vanskelig problem du skal løse. Det er ord og symboler du ikke har sett før. Du klarer ikke engang å forstå problemet. Du har ingen aning om verken forutsetninger eller mål. Du har ikke de nødvendige kunnskapene for å løse problemet.

Den samme situasjonen kan oppstå i et hvilket som helst fag på skolen. Hva skal du gjøre?

Skriv: Finne opp kruttet?

Hva tror du uttrykket "Ikke finn opp kruttet" har med problemløsning å gjøre? Hvorfor passer uttrykket på problemer som er ekstra vanskelige og som andre mennesker allerede har løst?

Skriv noen tanker om det.

Bygg på tidligere arbeid fra andre eller deg selv

1. Anta at du skal løse et problem som du har løst før. Prøv å huske hvordan du gjorde det. Hvis det var vellykket, bruk samme metode. Hvis det var mislykket, prøv en ny metode.
2. Anta at du skal løse et problem som andre har løst før. Prøv å finne ut hvordan andre har gjort det. Lær deg å bygge på andres arbeid. Hvis det var vellykket, bruk samme metode.
3. Snakk med andre som har løst problemet.
4. Snakke med andre for å få hjelp til å løse problemet.
5. Samarbeid med andre om å løse problemet.
6. Lær deg å slå opp i kilder. (Internett, bøker)
7. Lær deg å bruke fagstoff i lærebøker eller andre læringsressurser.

Skriv: Metoder for å bygge på andres arbeid

Det er mange metoder for å bygge på andres arbeid. Noen er beskrevet ovenfor. Skriv flere metoder du tenker ut selv. Sett en strek under de metodene som fungerer best for deg.

Skriv: Bruke grunnleggende ferdigheter og kunnskaper i problemløsning

Lesing, skriving og regning er grunnleggende ferdigheter. Kan du lese, kan du skaffe deg informasjon og kunnskap. En av skolens viktigste oppgaver er å gi deg grunnleggende ferdigheter.

1. Gi et eksempel på et godt definert problem du har løst og hvor du trengte å kunne lese
2. Gi et eksempel på et godt definert problem du har løst og hvor du trengte å kunne skrive
3. Gi et eksempel på et godt definert problem du har løst og hvor du trengte å kunne regne
4. Gi et eksempel på et godt definert problem du har løst og hvor du trengte å kunne bruke datamaskin

Skolefagene

I naturfag lærer du å arbeide og tenke som en forsker. Du lærer å angripe problemer slik forskere angriper dem.

Forskere bruker vitenskapelige metoder. De utvikler teorier som beskriver hva som skjer. Albert Einstein utviklet en teori som handlet om bevegelsene til objekter, relativitetsteorien.

Metodene vi bruker i et fag for å løse problemer, kan kanskje også brukes i et annet fag. Derfor er det nyttig å kunne litt fra hvert fag. Bare en liten bit av kunnskap i et fag, kan hjelpe deg å komme i gang med å løse et problem i et annet fag.

Oppsummering av kapittel 4

Mange problemer kan løses med en firetrinns plan:

1. Forstå problemet.
2. Planlegge og bestemme tiltak.
3. Gjennomføre tiltak.
4. Undersøke løsningen.

Når du lager en plan, er det nyttig å bygge på ting du allerede vet, eller på andres kunnskap og erfaringer.

Hvert skolefag fokuserer på forståelse og problemløsning innenfor sitt eget område. Det er nyttig å ha kunnskaper fra flere fag.

Aktiviteter i kapittel 4

Aktivitet 4.1 – Noe jeg tenkte på i kapittel 4

Var det noe spesielt du begynte å tenke på da du leste kapittelet? Kanskje du tenkte på hvordan du kan bygge på egne erfaringer? Kanskje du tenkte på hvorfor noen fag er morsommere enn andre fag?

1. Skriv tittel: "Noe jeg tenkte på i kapittel 4"
2. Skriv og forklar hva du tenkte.
3. Forklar hvorfor dette interesserer deg.
4. Hvorfor tenkte du på det?
5. Hva har det med livet utenfor skolen å gjøre?
6. Er det noe mer du tenker på?

Aktivitet 4.2 – Del ideen

Finn en elev å snakke med. Begge velger en idé i kapittelet som hver synes er viktig. Bruk ett minutt hver til å fortelle den andre om ideen. Tips til hva dere kan snakke om:

- Hvorfor er ideen viktig for deg?
- Hvilken betydning har den for ditt liv?

Når dere snakker sammen bør dere bruke begreper fra kapittelet:

- Firetrinns plan for å løse problemer
- Uløselige problemer

Aktivitet 4.3 – Lag en quiz som tester faktakunnskap fra kapittelet

Lag 2 spørsmål. Du skal teste faktakunnskap fra dette kapittelet. Det skal være tekstsvar. Bare ett riktig svar til hvert spørsmål.

Spørsmålene skal være så enkle at en som ikke har lest kapittelet kan svare. Prøv quizen på noen elever.

Aktivitet 4.4 – Lag en quiz som tester forståelse av kapittelet

Lag 2 spørsmål. Du skal teste forståelse av kapittelet.

Presenter noe fakta fra kapittelet og still spørsmål som krever tenkning og vurdering av fakta.

Testpersonene kan slå opp i en bok, gjøre et søk eller lese i dette kapittelet for å lage et svar.

En quiz som tester forståelse, har alltid flere korrekte svar. Du lager ingen svaralternativer. Spørsmålene skal være så vanskelige at de bare kan besvares av en person som har en god forståelse av innholdet i dette kapittelet.

Prøv quizen på noen elever. Det skal være tekstsvar, og ingen fasit. Diskuter svarene sammen til slutt.

Aktivitet 4.5 – Løse problemer ved hjelp av matematikk

1. Hva er matematikk? Lag en enkel definisjon som de andre elevene forstår.
2. Skriv eksempler på problemtyper mennesker lærer å løse ved å studere matematikk.
3. Skriv eksempler på problemtyper i andre fag som matematikk kan hjelpe til å løse.
4. Skriv eksempler på to problemer utenfor skolen hvor matematikk er nyttig?

Aktivitet 4.6 – Løse problemer ved hjelp av andre fag

Gjør aktiviteten 4.5 for et annet fag du har på skolen, for eksempel norsk.

Aktivitet 4.7 – Transportproblemet

For mange år siden ble mennesker stilt ovenfor transportproblemet. De utviklet bilen. Gi eksempler på problemer som bilen har skapt.

Aktivitet 4.8 – Brainstorm TV-problemer

1. Hvilke problemer løser TV?
2. Hvilke problemer skaper TV?

Aktivitet 4.9 – Brainstorm skoleproblemer

1. Hvilke problemer hjelper skolegang med å løse?
2. Hvilke problemer skaper skolegang?

Aktivitet 4.10 – Brainstorm muligheter

Du kan spørre en lærer om du har løst et problem riktig. Lag en liste over andre måter du kan sjekke om du har løst et problem riktig. Gjør gjerne denne aktiviteten sammen.

Kapittel 5 – Strategier for problemløsning

En god strategi kan løse et problem i naturfag. Den samme strategien løser kanskje ikke et problem i kroppsøving.

En **strategi** er en handlingsplan for å angripe et problem.

Fokus

Dette kapittelet har fokus på:

- Å bruke riktig strategi.
- Å kunne bruke mange strategier.
- Hvilken rolle datamaskiner har.

Strategier for å bli bedre til å løse problemer

Firetrinnsplanen i kapittel 4 er en strategi som kan benyttes på en rekke problemtyper. Men den løser ikke et hvert problem for deg. Imidlertid hjelper den deg å komme i gang. Kan du komme i gang, har du en mulighet til å løse problemet.

Det er mange andre strategier i problemløsning.

For eksempel kan det å være en god lytter, være en god strategi i møte med andre mennesker.

Har du funnet en god strategi for en problemtype, vil hjernen din prøve å benytte en liknende strategi neste gang du møter samme problemtype.

Noen ganger vil hjernen utvikle en strategi som ikke fungerer godt. Du vil bruke samme dårlige strategi om og om igjen, selv om det fungerer dårlig.

Det rare er at hjernen er fornøyd med det. Det er viktig å bryte dette mønsteret. Prøv noe annet.

Skriv: Dårlige strategier

Tenk på minst to dårlige strategier som du ofte bruker, og som ikke virker bra.

Du har kanskje en spesiell strategi når du forhandler med foreldrene dine, og som ofte mislykkes? Eller du har kanskje en dårlig strategi når du leser til prøver?

Skriv: Strategi for å lese til prøve

Du skal ha en stor prøve neste uke. Tenk på tre forskjellige strategier du kan bruke for å gjøre det bra på prøva. En strategi kan for eksempel være å pugge. Hvilken strategi er best for deg? Er denne strategien god for alle fag?

Det er mange strategier

Forskere har funnet ut at de fleste studenter benytter bare et fåtall strategier for å løse problemer. De bruker de samme strategiene om og om igjen.

Noen ganger virker det. Andre ganger ikke.

Du kan bli mye bedre til å løse problemer dersom du har mange strategier å velge i.

La oss si at du arbeider med et problem, men sliter med å lage et mål. Her er noen strategier for å lage mål:

1. Brainstorm alene.
2. Brainstorm sammen med andre eller med en venn.
3. Spør læreren for mulige mål.
4. Les en bok om emnet. Finn mål foreslått av forfatteren.
5. Søk på Internett om emnet. Finn mål i artikler på nettet.
6. Kjenn på dine egne følelser for problemet. Har du noe eierskap til det? Betyr problemet noe for deg? Er du interessert i å løse det?
7. Tenk på mål som vil gi deg et bedre eierskap til problemet.

La oss si du har en prøve. Strategi nr 1 kan da benyttes. Er det en prøve med hjelpemidler, kan du bruke strategi nr 4 og 5 også.

Hvordan kan du vite hvilken strategi du skal velge?

Det finnes ikke noe fasitsvar, men trening og erfaring kan hjelpe. Du må faktisk lære deg selv. Måten å gjøre dette på, er å benytte metakognisjon(tenk på din egen tekning).

Tips til metakognisjon om strategier

Når du løser problemer

1. Tenk på hva du gjør.
2. Tenk på hvordan du gjør det.
3. Tenk på hvilken strategi du bruker.
4. Tenk på hvilke strategier du ikke bruker.

Skriv: Strategier jeg bruker for å huske ting

Du bruker mye tid på å pugge. Du skal huske hvordan du skriver ord, viktige navn og årstall, formler i matematikk og fysikk, for å nevne noe.

1. Lag en liste over tre ulike strategier du bruker for å huske ting.
2. Skriv hvorfor en eller to av dem er best.
3. Finn en elev å samarbeide med. Del strategiene dine med den andre. Den andre deler sine strategier med deg. Får du nye ideer, skriv det i notatet.
4. Har noen elever i din klasse lest noe om hvordan man blir bedre til å huske? Skriv og les.

"Bryt ned" - strategi

Dette er en generell strategi som passer for større problemer.

Hensikten er å dele opp et problem i mindre problemer. Små problemer kan være lettere å løse. Du starter på "toppen" med hovedproblemet, og deler opp problemet i flere mindre problemer. Gjerne problemer som du lett kan løse.

Du bruker denne strategien hele tiden, uten å tenke på det. Her er to eksempler:

Problem: Å komme seg på skolen.

1. Du står opp.
2. Går på badet og steller deg.
3. Kler på deg og grer håret.
4. Spiser frokost.
5. Pakker sekken.
6. Går til bussen eller sykler til skolen.

Problem: Lage en innlevering om et emne i et fag.

1. Du velger et emne.
2. Finner kilder og stoff.
3. Leser fagstoff og skriver notater.
4. Disponerer besvarelsens innhold.
5. Lager besvarelsens innhold.
6. Leverer.

Skriv: Betydningen av rekkefølge

1. Se på problemeksemplene over. Har det noen betydning i hvilken rekkefølge du løser de mindre problemene?
2. Lag et eksempel på et problem du deler opp i mindre problemer. Det skal ikke ha noen betydning i hvilken rekkefølge du løser hvert enkelt problem. Kan du finne et slikt eksempel? Skriv.
3. Lag et eksempel på et problem du deler opp i mindre problemer. I dette eksempelet er det helt avgjørende at problemene løses i en bestemt rekkefølge.

"Bygg opp" – strategi

Dette er også en strategi du bruker ofte, kanskje uten å tenke på det.

Et lite barn som bygger med klosser, bruker en slik strategi. Det kan se planløst ut, men er tårnet høyt nok, er barnet fornøyd med sitt mål.

Ideen med denne strategien er å begynne med det vi kan og det vi vet hvordan vi skal gjøre. Mange artister jobber slik. Det er veldig kreativt. Målet former seg etter hvert.

Et band kan lage en ny låt på denne måten. Musikere starter med et "komp" eller en "groove". De bruker kanskje 3 - 5 kjente akkorder. Etter hvert former det seg en melodi over "kompet". I beste fall bygger det seg opp til en "hit".

"Bygg opp" – strategien brukes på to forskjellige måter:

1. Du har ikke noe mål når du starter. Målet former seg etter hvert.
2. Du har et mål når du starter.

Målet ditt kan være å bli mett, fordi du er sulten etter skolen. Du kikker i kjøleskapet for å finne noe å spise, men finner ikke noe du har lyst på. Du ser i matskapet. Samme der. Du går i kiosken, og bestemmer deg for en pizzabit. Du har nettopp benyttet en "bygg opp" – strategi for å bli mett, som var målet ditt.

Skriv: "Bygg opp" – strategier.

Tenk på to ganger du har benyttet "bygg opp" – strategien.

1. Skriv et eksempel der du hadde et mål før du begynte å bygge.
2. Skriv et eksempel der du ikke hadde noe mål ved start.
3. Hvilket av de to eksemplene er lettest å komme på?
4. Hvorfor?

Primitiver

Enten du bruker "bryt ned"- eller "bygg opp" – strategier, benytter du primitiver.

En **primitiv** er i denne sammenhengen et mindre problem som du vet at du kan løse.

Kan du gangetabellen? Da kan du, uten å tenke, gi svaret på 8×7 .

Du har da et sett av primitiver som er nyttig i en rekke matematikkproblemer.

Du kan tenke på primitiver som byggesteiner.

Ordet primitiv, betyr i denne boken **en grunnleggende oppgave du kan**. I fotball må du kunne "slå" en pasning eller skyte et skudd. Dette er primitiver i fotball. Er du fotballspiller, har du allerede disse primitivene.

Vi kunne benyttet ordet "byggstein" videre i denne boken, men vi har valgt å bruke begrepet primitiv.

Skriv: Mine primitiver.

Tenk på noe du kan godt. Kanskje du er god i ballett eller til å spille et instrument. Lag en liste over dine primitiver på ditt område. Husk at dette er ting du allerede kan godt.

Kalkulatoren som primitiv.

En kalkulator er et godt eksempel på en maskin som sparer oss for arbeidet med å lære oss en rekke primitiver i matematikk.

En billig kalkulator til under 100 kroner kan regne addisjon, subtraksjon, multiplikasjon, divisjon og mer til. Det er en liten datamaskin.

De fleste kalkulatorer kan regne roten av et tall. Taster du inn 144 og rottegnet, får du 12. (-12 er også roten av 144). Å forstå konseptet eller prinsippet for kvadratroten er ikke så arbeidskrevende, men kalkulatoren viser deg ikke dette konseptet.

Å finne roten av et tall er derimot ganske arbeidskrevende. Dette arbeidet gjør kalkulatoren for deg, uten at du vet hvordan det skjer. Det å utføre beregningen av kvadratroten, kaller vi **prosess**. Det å vite hvordan det skjer, kaller vi **konsept**.

Før det ble vanlig med kalkulator i skolen, måtte elever lære hvordan de regnet ut roten av et tall for hånd. Prosessen med å regne ut roten av et tall for hånd, er ganske komplisert. Nå bruker elevene kalkulator til å regne ut roten av tall. Kalkulatoren blir elevenes primitiv for å regne kvadratroten. Men læreren kan sikkert vise elevene konseptet for å finne kvadratroten av et tall.

Med en kalkulator som verktøy, sparer elevene tid, for det er mye enklere å lære seg å bruke noen taster enn å lære seg å kalkulere roten med papir og blyant.

Skriv: Kvadratrotenproblemet.

Kvadratrotenproblemet er todelt. På den ene siden har det konseptet eller prinsippet kvadratroten som gjelder for tall. På den andre siden har det prosessen for hvordan man beregner kvadratroten av et bestemt tall. Alle problemer har disse to delene.

1. Tenk på noen problemer som du vet hvordan du skal løse.
2. For hvert problem skal du skille konsept fra prosess.

Som eksempel på dette, kan du tenke på det å slå opp et ord i en ordliste, eller finne et emne på et bibliotek. I hvert tilfelle er konseptet og prosessen to forskjellige ting.

Datamaskinen som primitiv

Et dataprogram har en rekke mulige definerte handlinger. Det er en fastlagt plan, som forteller maskinen hva den skal gjøre. Et dataprogram kan fortelle datamaskinen hvordan et spesielt problem skal løses. Det kan være en rutine som forteller maskinen hvordan den skal utføre en bestemt oppgave.

En datamaskin med mange dataprogrammer, gir deg et stort antall primitiver. Husk at hver primitiv har både et konsept og en prosess. Datamaskinen kan raskt og korrekt utføre en prosess for deg. Men dette har ingen verdi for deg, hvis du ikke forstår konseptet.

Dette er kanskje den viktigste ideen å forstå om datamaskiner.

En datamaskin kan gi deg tilgang til et stort antall primitiver. Det er mange tusen dataprogrammer som løser større og mindre problemer. Hvert problem har et konsept. Datamaskinen kan utføre prosessen med å løse problemet. Hvis du lærer deg konseptene, vil du ha tilgang til primitivene. Da har du et kraftig verktøy for å løse problemer.

Skriv: Hva en datamaskin kan.

Du har sett plakater og tegninger laget av grafiske kunstnere. Tekst og grafikk på slike plakater er formet pent og nøyaktig. En grafisk kunstner er flink til å lage design for plakater. Kunstneren er også som regel flink til å tegne og skrive pent for hånd.

Datamaskiner har programmer for grafisk design. Det er lett å lære hvordan man skal skrive tekst elegant og presist med et grafisk program. Det er mye vanskeligere å lære konseptet med å designe en plakat.

1. Skriv eksempel på en prosess du vet en datamaskin kan gjøre.
2. Skriv konseptet en person må kunne for å gjøre god bruk av denne prosessen.
3. Skriv flere slike eksempler på prosesser og tilhørende konsepter.

Oppsummering av kapittel 5

En strategi er en plan for hvordan du skal angripe et problem. Det finnes mange måter å angripe problemer på. Noen strategier er spesialiserte og passer bare for noen problemtyper. Andre strategier er mer generelle og kan brukes på mange ulike problemtyper.

1. "Bryt - ned" – og "bygg – opp" – strategier er nyttige for mange problemtyper.
2. Primitiver er mindre problemer som du løser lett og riktig, uten anstrengelser.
3. En primitiv består av både et konsept og en prosess.
4. En datamaskin har mange primitiver og kan utføre mange prosesser for deg når du arbeider med problemløsning.
5. Hvis du lærer konseptene for disse prosessene, har du skaffet deg mange primitiver du kan bruke i problemløsning.

Aktiviteter i kapittel 5

Aktivitet 5.1 – Noe jeg tenkte på i kapittel 5

Var det noe spesielt du begynte å tenke på da du leste kapittelet? Kanskje du tenkte på noen strategier du bruker ofte? Kanskje du tenkte på strategier som ikke er nevnt i kapittelet? Kanskje du tenkte på at skolen bruker mye tid på å lære deg ting som en datamaskin kunne gjøre for deg?

1. Skriv tittel: "Noe jeg tenkte på i kapittel 5"

2. Skriv og forklar hva du tenkte.
3. Forklar hvorfor dette interesserer deg.
4. Hvorfor tenkte du på det?
5. Hva har det med livet utenfor skolen å gjøre?
6. Er det noe mer du tenker på?

Aktivitet 5.2 – Del ideen

Finn en elev å snakke med. Begge velger en idé i kapittelet som hver synes er viktig. Bruk ett minutt hver til å fortelle den andre om ideen. Tips til hva dere kan snakke om:

- Hvorfor er ideen viktig for deg?
- Hvilken betydning har den for ditt liv?

Når dere snakker sammen bør dere bruke begreper fra kapittelet:

- "Bryt ned"- strategi
- "Bygg opp"- strategi
- Primitiver i kalkulatoren
- Primitiver i datamaskinen
- Primitiver i andre sammenhenger
- Prosess
- Konsept

Aktivitet 5.3 – Lag en quiz som tester faktakunnskap fra kapittelet

Lag 2 spørsmål. Du skal teste faktakunnskap fra dette kapittelet.

Det skal være tekstsvar. Bare ett riktig svar til hvert spørsmål.

Spørsmålene skal være så enkle at en som ikke har lest kapittelet kan svare.

Prøv quizen på noen elever.

Aktivitet 5.4 – Lag en quiz som tester forståelse av kapittelet

Lag 2 spørsmål. Du skal teste forståelse av kapittelet.

Presenter noe fakta fra kapittelet og still spørsmål som krever tenkning og vurdering av fakta.

Testpersonene kan slå opp i en bok, gjøre et søk eller lese i dette kapittelet for å lage et svar.

En quiz som tester forståelse, har alltid flere korrekte svar. Du lager ingen svaralternativer.

Spørsmålene skal være så vanskelige at de bare kan besvares av en person som har en god forståelse av innholdet i dette kapittelet.

Prøv quizen på noen elever. Det skal være tekstsvar, og ingen fasit. Diskuter svarene sammen til slutt.

Aktivitet 5.5 – Strategi for å omgå mennesker

Tenk på en strategi du bruker når du har problem med en venn. Du har kanskje benyttet den samme strategien når du omgås flere venner. Skriv et eksempel på hvordan denne strategien kan brukes på mennesker som ikke er dine venner.

Aktivitet 5.6 – Primitiver i fag

Velg et fag du har på skolen. Skriv en liste over primitiver du lærer i dette faget.

Velg et annet fag. Skriv en liste over primitiver du lærer i dette faget.

Hvilket fag har flest primitiver?

Hvorfor tror du det ene faget har flere primitiver enn det andre?

Husk: En primitiv er et mindre problem eller rutine som har både et konsept og en prosess.

Aktivitet 5.7 – Primitiver for datamaskinen

Bruk de to listene du laget i aktivitet 5.6.

For hver primitiv skal du tenke på prosessen for å løse oppgaven. Hvilke av disse prosessene tror du en datamaskin kan utføre?

Har det ene faget flere prosesser som en datamaskin kan gjøre, enn det andre faget?

Aktivitet 5.8 – Hva er du best på?

En primitiv består av både et konsept og en prosess. Noen mennesker er best til å lære seg konsepter. Andre er best til å lære seg prosesser. Hva er du best på?

Når du svarer på dette, kan du tenke på eksemplene du laget i aktivitet 5.6.

Aktivitet 5.9 – Primitiver for både hånd og datamaskin

Det er mange strategier for å løse problemer. For noen primitiver vil du ønske å vite både hvordan du kan utføre prosessen for hånd og med en datamaskin.

Stavekontroll er et godt eksempel. Noen ganger vil du rette tekst for hånd, andre ganger vil du bruke stavekontrollen i tekstbehandleren.

Skriv noen andre eksempler på primitiver hvor du synes det er nyttig å kunne utføre prosessen både for hånd og med datamaskinen.

Aktivitet 5.10 – Mine beste og mine dårligste fag

Lag en liste over strategier du vil bruke for å takle:

1. Faget du er best i.
2. Faget du er dårligst i.

Sammenlikn disse strategiene, og se etter motsetninger mellom dem.

Lag en konklusjon hvis det er mulig.

Kapittel 6 – Bli bedre til å tenke.

Fokus

Kapitlet diskuterer flere ting du kan gjøre for å bli bedre til å tenke. Når du blir bedre til å tenke, blir du bedre til å løse problemer.

For å bli bedre til å tenke, må du:

- Lære hva effektive tenkere gjør.
- Bli bevisst på dine egne effektive tenkemåter.

- Bli bevisst på dine egne mindre effektive tenkemåter.
- Utvikle mer effektive tenkemåter og praktisere dem.

Introduksjon

Anta at det samme problemet stadig dukker opp. Du lærer deg å løse dette problemet hurtig og riktig. Dette er effektivt. Neste gang det dukker opp, bygger du på det du allerede kan.

Da du var liten, lærte du å knyte lisser. Nå knytter du lissene uten å tenke på hvordan du gjør det. Det er automatisert.

Tenk på en idrettsutøver. Utøveren må kunne reglene for sin egen sport. Under en håndballkamp kan ikke en utøver stoppe og tenke på regler. Utøveren må kunne skyte et skudd uten å tenke på skuddteknikk.

Poenget er at hjernen blir veldig god på å løse problemer som dukker opp ofte. Men det krever tid og trening å lære seg alle detaljene og automatisere dem.

Vi har nå kommet til kanskje et av de viktigste spørsmålene for utdanning:

Hvordan skal vi bruke skoletiden best?

Vi kunne fordele skoletiden på følgende læring:

1. Lære grunnleggende informasjon og grunnleggende ferdigheter.(Primitiver)
2. Lære å løse spesifikke problemer som kommer ofte og er viktige.
3. Utvikle evne til å kunne angripe et hvilket som helst problem som dukker opp.

Skriv: Hvordan bør skoletiden benyttes?

Tenk på de tre forslagene ovenfor. En mulighet er å bruke en tredjedel av tiden på hvert punkt. Hva synes du om det? Skriv.

Hvordan ville du fordele det hvis du var rektor og kunne bestemme?

Ville fordelingen bli lik for hver elev?

Skolen

Det er nok slik at skolen bruker ganske mye tid på å hjelpe elever til å takle og løse problemer som dukker opp. Problemer elevene ikke har møtt tidligere.

En strategi er å hjelpe elever til å bli mer effektive tenkere.

Du har en veldig god hjerne. Med trening og erfaring kan du bli mer effektiv til å tenke.

Dette kapittelet vil gi deg litt hjelp til å bli en bedre tenker.

Hva er tenkning?

Ordet "tenke" har flere betydninger. Hver person har en oppfatning av hva det betyr.

Skriv: Hva betyr ordet "tenke"?

1. Brainstorm en liste over hva "å tenke" betyr for deg. Skriv lista i notatet.
2. Gå sammen med en elev og del ideene med hverandre.
3. Sammen skriver dere en god definisjon av "å tenke".

Se på definisjonen dere skrev. Betyr denne definisjonen at mennesker kan bli bedre til å tenke? Foreslår deres definisjon at ulike mennesker tenker på ulike måter?

Gjør litt metakognisjon om spørsmålene ovenfor.

- Dukker det opp noen bilder i hodet ditt når du tenker på det?
- Snakker du til deg selv inne i hodet når du tenker på det?
- Får du **assosiasjoner**(kommer du på noe)?
- Bruker du **intuisjon**(fornemmer du bare det du bestemmer)?
- Oppfatter du følelsene dine når du tenker på det?

Skriv:

Det er mange måter å tenke på. Noen tenker i bilder. Andre tenker i ord. Begge måter er nyttige.

1. Hva er du best på? Skriv.
2. Hvordan vet du det? Skriv.

Prøv følgende eksperiment:

Du skal tenke på to ulike problemer du har løst i dag.

1. Det første problemet løste du ved hjelp av bilder du hadde i hodet.
2. Det andre problemet løste du ved hjelp av ord du hadde i hodet.

Gjør litt metakognisjon om oppgaven du akkurat fikk. Brukte du bilder eller ord i hodet mens du tenkte på eksperimentet?

Skriv noen notater om det.

Hjerneteori

Hjernen er kompleks. Det forskes mye på hjernen, men vi har mye å lære om den enda. Å forstå hvordan hjernen virker, er viktig for å forstå Alzheimers sykdom.

Hjernen har 30 - 100 billioner neuroner. 30000 neuroner får plass i et knappenålshode. Neuroner lagrer informasjon og behandler informasjon.

Forskere kan ta bilder av de delene av hjernen som utfører ulike oppgaver. Slike bilder leder til teorier:

1. Hjerneaktivitet er i stor grad ubevisst aktivitet.
2. Gode tenkere bruker hjernen mer effektivt enn dårlige tenkere. Det betyr at det er mindre hjerneaktivitet hos en god tenker som løser et problem, enn hos en dårlig tenker som løser det samme problemet.
3. Trening og øvelse gjør hjernen mer effektiv. Dette forklarer hvorfor noen blir gode idrettsutøvere.

Du har sett gode idrettsutøvere. De har lært å bruke hjernen og kroppen effektivt. De blir bedre gjennom hardt arbeid.

Det samme gjelder for å bli en effektiv tenker. Gjennom trening kan du bli en god tenker.

Utholdenhet i problemløsning

Mange problemsituasjoner er kompliserte og arbeidskrevende. Det er ingen enkle løsninger. Effektive tenkere er villige til å holde ut og arbeide hardt.

Thomas Edison var en slik tenker. Han sa at han oppnådde suksess fordi han som oppfinner baserte sitt arbeid på 98% utholdenhet og 2% inspirasjon.

Han utvikler lyspæra, eller glødelampen. Han prøvde hundrevis av ulike gasser i pæra. Gassen måtte gi godt lys og vare lenge. Bare gjennom utholdenhet, klarte han å finne en vellykket løsning.

Dårlige problemløsere gir opp fortere. De prøver noen få strategier, og gir opp hvis de ikke lykkes raskt.

Skriv: Min utholdenhet

Tenk på din egen utholdenhet i problemløsning.

1. Gir du opp lett, eller holder du ut? Skriv.
2. Hvordan vil du sammenlikne din egen utholdenhet med dine venners utholdenhet? Skriv.
3. Gi et eksempel på et problem hvor du holdt ut og løste et vanskelig problem. Skriv.
4. Gi et eksempel på et problem hvor du ga opp fort. Skriv.
5. Anta at du skal løse et nytt problem. Kan du ut fra beskrivelse av problemet vite om du vil gi opp fort eller holde ut? Skriv.
6. Hvilke problemtyper mener du selv at du er mest utholdende med? Skriv. (For eksempel: Trening på ferdigheter i idrett, løse likninger, løse problemer med venner)

Å utføre metakognisjon og skrive i notatet, er veldig viktige øvelser for å bli en bedre tenker. Du kan studere dine egne arbeidsvaner og tenkevaner.

I resten av kapittelet skal vi se på noen **tenkevaner** for å bli en bedre tenker.

Det er viktig at du selv finner ut dine egne gode og dårlige tenkevaner. Kanskje du har noen vaner som ikke er så effektive? Å oppdage det, er et godt utgangspunkt for å forbedre seg selv.

Men det er ikke lett å endre sine tenkevaner. Det krever avlæring av dårlige vaner. Det er vanskelig å praktisere avlæring, fordi dårlige tenkevaner er så innarbeidet. Skal du klare det, må du bli bevisst på dine egne tenkevaner. Du må også være villig. Du må eie problemet. Ha interesse av å bli en bedre tenker. Da kan du løse det.

Tenkevane 1: Å forholde seg til uklare problemer

Anta at du støter på et problem som opptar deg.

Til å begynne med er situasjonen uklar.

- Det er mange mulige mål.
- Det er mange mulige ressurser og begrensinger.

Hva skal du gjøre?

Den effektive tenkerens vaner

Effektive tenkere liker utfordringer som dårlige definerte problemer er. De liker uavklarte spørsmål. De liker å gjøre problemet godt definert. De kan definere mange forskjellige problemsituasjoner.

Den mindre effektive tenkerens vaner

Mindre effektive tenkere leter etter allerede avklarte spørsmål. De har vanskeligheter med å forholde seg til dårlig definerte problemer. De ønsker seg et problem som allerede er godt definert. De er fornøyd med å lage bare én problemsituasjon.

Tenkevane 2: Å se andres perspektiv

De fleste komplekse problemer kan ses fra ulike perspektiver. Det samme problemet kan lede til flere forskjellige mål. Mennesker har ulike interesser og perspektiver. Ofte er det mange mulige angrepsvinkler.

Anta at du spiller på midtbanen i fotball. Skal du skyte på mål? Skal du spille en pasning? Skal du dribble og gå helt inn til målet selv? Hva synes de andre spillerne du skal gjøre? Hva synes treneren? Hva synes publikum?

Du skyter fra 40 meter. Du synes det er et riktig valg, men publikum rister på hodet. Du så at keeper sto langt ute, og valgte å prøve et langskudd. For deg var det et riktig valg. For publikum som så keeper ta ballen i fast grep, var det et håpløst valg. Treneren tar deg av banen.

Den effektive tenkerens vaner

Den effektive tenkeren har evne til å finne og tenke på mange ulike **perspektiver**. Denne tenkeren kan se styrke og svakheter i hvert perspektiv. Tenkeren kan argumentere for og imot flere sider av en sak.

Den mindre effektive tenkerens vaner

Mindre effektive tenkere sliter med å vurdere alternativer. De har problemer med å forstå andres perspektiv. De gjør et valg, og holder seg til det. De er ikke åpne for alternative muligheter. De kan ikke tenke på argumenter som støtter ulike synsvinkler.

Du kunne spilt ballen til venstrekanten, hvor en annen spiller sto godt plassert for å lage mål. Keeper ville vært utspilt. Spilleren på venstrekanten var helt alene, og hadde åpent mål. En god pasning hadde skapt en 100% - sjanse for å lage mål.

Hadde du spilt pasningen, ville publikum jublet. Treneren hadde ikke tatt deg av banen. Du hadde fått skryt i garderoben. Folk hadde sagt: "Han har god fotballforståelse."

Tenkevane 3: Å verdsette utdanning og god tenkning

Du kan bli bedre til å løse problemer ved å studere og praktisere fagene. Dette er nøkkelideen. Studier av problemløsning viser at du blir bedre ved å lære mange forskjellige fag.

Den effektive tenkerens vaner

Den effektive tenkeren forstår verdien av kunnskap. Jo mer du lærer og vet, dess bedre blir du til å løse problemer.

Den mindre effektive tenkerens vaner

Mindre effektive tenkere overvurderer bruk av **intuisjon**(antar hva som er riktig). De føler at skolen ikke er så viktig. De tror kanskje ikke at tenkning hjelper dem å forstå hvordan de skal løse problemer.

Tenkevane 4: Innstilling og selvtillit

Hvis du tror at du kan løse et problem, så kan du mest sannsynlig løse det. Å ha troen på seg selv og en positiv innstilling er veldig viktig. Forskning viser at disse to faktorene antakelig er de viktigste.

Den effektive tenkerens vaner

Den effektive tenkeren har en positiv holdning. Denne tenkeren tror på verdien av å være en effektiv tenker og har tro på seg selv og at det nytter.

Den mindre effektive tenkerens vaner

Den mindre effektive tenkeren har en negativ holdning til problemet og mangler tro på at det nytter.

Skriv: Hva slags tenker er jeg?

Gjør litt metakognisjon. Ta for deg ideene du akkurat leste. For hver idé skal du vurdere om du er en effektiv eller en mindre effektiv tenker.

1. Tenkevane 1: Å forholde seg til uavklarte problemer

- a. Er du effektiv tenker? Skriv ja/nei/kanskje
- b. Er du mindre effektiv tenker? Skriv ja/nei/kanskje

2. Tenkevane 2: Å se andres perspektiv

- a. Er du effektiv tenker? Skriv ja/nei/kanskje
- b. Er du mindre effektiv tenker? Skriv ja/nei/kanskje

3. Tenkevane 3: Å verdsette utdanning og god tenkning

- a. Er du effektiv tenker? Skriv ja/nei/kanskje
- b. Er du mindre effektiv tenker? Skriv ja/nei/kanskje

4. Tenkevane 4: Innstilling og selvtillit

- a. Er du effektiv tenker? Skriv ja/nei/kanskje
- b. Er du mindre effektiv tenker? Skriv ja/nei/kanskje

- Hva er du best på av de fire ideene? Skriv.
- Hva er du dårligst på? Skriv
- Hvordan kan du vite det?
- Hva gjør du for å bli en mer effektiv tenker?

Tenkevane 5: Lete etter mål

Mange problemsituasjoner mangler klare mål. Problemsituasjonen kan lede til mange mål. Valg av mål er veldig viktig. Du har verken tid eller krefter til å jobbe mot et hvert mulig mål.

Den effektive tenkerens vaner

Den effektive tenkeren leter etter mange alternative mål, undersøker målene og velger det beste målet.

Den mindre effektive tenkerens vaner

Den mindre effektive tenkeren klarer ikke å finne alternative mål. Denne tenkeren bruker lite tid på å vurdere om det er et godt mål som er valgt. Ofte ønsker denne tenkeren at noen andre skal bestemme målet. "Det er det samme for meg."

Tenkevane 6: Se gjennom målene

Når du arbeider med å løse et problem, får du ofte en bedre forståelse av problemet etter hvert som du arbeider. Du vil kanskje oppdage at din innledende forståelse av problemsituasjonen ikke var riktig. Du vil kanskje oppdage at målet du har satt, ikke kan nås, er feil eller er dårlig formulert.

Den effektive tenkerens vaner

Den effektive tenkeren revurderer målet sitt flere ganger etter hvert som arbeidet skrider frem. Det er kanskje nødvendig å velge et annet mål, eller omformulere målet, slik at det passer med ressurser og begrensninger som er tilgjengelig.

Den mindre effektive tenkerens vaner

Den mindre effektive tenkeren fortsetter å arbeide mot det målet som er valgt, selv om målet er dårlig. Hvis målet ikke kan nås, gir denne tenkeren opp, i stedet for å omdefinere målet eller velge et annet mål.

Skriv: Hva slags tenker er jeg?

Gjør litt metakognisjon. Ta for deg ideene du akkurat leste. For hver idé skal du vurdere om du er en effektiv eller en mindre effektiv tenker. Her er ideene:

5. Tenkevane 5: Lete etter mål

- a. Er du effektiv tenker? Skriv ja/nei/kanskje
- b. Er du mindre effektiv tenker? Skriv ja/nei/kanskje

6. Tenkevane 6: Se gjennom målene

- c. Er du effektiv tenker? Skriv ja/nei/kanskje
- d. Er du mindre effektiv tenker? Skriv ja/nei/kanskje

Tenk på noen problemsituasjoner du har møtt i det siste.

1. Hvordan bestemte du deg for målet?
2. Skiftet du mål etter hvert som du arbeidet med problemet?
3. Eller endret du målet?

Tenkevane 7: Ressurser

Du kan ikke løse et problem uten ressurser. Men hvilke ressurser har du tilgjengelig? Dette er kanskje det vanskeligste spørsmålet å besvare i problemløsning. Du må vite dine styrker og svakheter. Du må lære hvilke typer av ressurser du trenger for å løse et problem.

Den effektive tenkerens vaner

Den effektive tenkeren er god på brainstorming. Denne tenkeren lager seg en stor liste over mulige ressurser. Ideer fra andre tas gjerne imot. Tenkeren søker etter nye ressurser. Ingen ideer forkastes i første omgang.

Den mindre effektive tenkerens vaner

Den mindre effektive tenkeren er dårlig på brainstorming. Denne tenkeren er fornøyd med et begrenset utvalg av ressurser og benytter seg lite av forslag fra andre. Tenkeren er heller ikke åpen for å lytte til andres innspill og ber heller ikke om hjelp. "Jeg klarer meg med det jeg har".

Tenkevane 8: Begrensninger

All problemløsning har begrensninger. Ofte er disse dårlig definert. Noen ganger tror vi at det ikke er noen begrensninger.

Den effektive tenkerens vaner

Den effektive tenkeren identifiserer en rekke begrensninger som vil gjelde for et hvert problem som skal løses. For eksempel: Ikke lyv, ikke juks eller ikke bryt loven. Men den effektive tenkeren vet også at det noen ganger er nødvendig å tøyne begrensningene eller kanskje også bryte dem. (Et eksempel: Er det greit å kopiere den teksten du trenger? Når er det i så fall greit?)

Den mindre effektive tenkerens vaner

Den mindre effektive tenkeren kjenner kanskje ikke til begrensninger for problemløsningen. (Eksempel: Andres tekst kopieres kanskje uten tillatelse. "Jeg visste ikke at jeg måtte spørre om tillatelse.")

Den mindre effektive tenkeren kan også feilaktig sette begrensninger som ikke er nødvendig eller riktig. ("Jeg trodde ikke det fantes en bok om dette." "Jeg trodde vi måtte levere denne uka.")

Skriv: Hva slags tenker er jeg?

Gjør litt metakognisjon. Ta for deg ideene du akkurat leste. For hver idé skal du vurdere om du er en effektiv eller en mindre effektiv tenker. Her er ideene:

7. Tenkevane 7: Ressurser

- a. Er du effektiv tenker? Skriv ja/nei/kanskje
- b. Er du mindre effektiv tenker? Skriv ja/nei/kanskje

8. Tenkevane 8: Begrensninger

- c. Er du effektiv tenker? Skriv ja/nei/kanskje
- d. Er du mindre effektiv tenker? Skriv ja/nei/kanskje

Tenk på noen problemsituasjoner du har møtt i det siste.

1. Gi noen eksempler på situasjoner hvor du var en effektiv tenker.

Tenkevane 9: Vurdere planen

Du bør tenke ut en plan for å løse et problem. Dette er en handlingsplan hvor du bruker ressurser og begrensninger du allerede har. Forskjellige mennesker vil lage forskjellige planer. Hvilken plan skal du bruke?

Den effektive tenkerens vaner

Den effektive tenkeren søker etter svar som utfordrer sin egen plan. Hvilke deler av planen min er dårlig? Hva kan gå galt? Hvilke resultater vil være dårlige? Den effektive tenkeren stiller seg slike spørsmål for å forbedre sin egen plan.

Den mindre effektive tenkerens vaner

Den mindre effektive tenkeren stiller seg ikke spørsmål underveis om planen er god eller dårlig. Denne tenkeren tar ikke hensyn noe som beviser at planen er dårlig eller feil. Denne tenkeren bryr seg ikke om kritiske innspill til planen fra andre.

Tenkevane 10: Forutse resultater

Når du gjennomfører planen, lager du endringer i verden. Dette tar noe av din tid og andre ressurser. Det kan skape nye problemer.

La oss si du skal på trening, men de røde fotballstrømpene er skitne. Du hiver dem inn i maskinen sammen med andre klær som allerede ligger der. Når du er ferdig med å vaske, har alle klærne fått et rosa skjær. Du har fått et nytt problem som er vanskelig å løse.

Den effektive tenkerens vaner

Den effektive tenkeren kan forutse konsekvensene av å gjennomføre en plan. Denne tenkeren kan se sideeffekter og nye problemer som kan oppstå som følge av sine handlinger.

Den mindre effektive tenkerens vaner

Den mindre effektive tenkeren tar ikke hensyn til mulige konsekvenser av handlingene. Denne tenkeren handler først og tenker etterpå, og virker overrasket over uventede resultater.

Skriv: Hva slags tenker er jeg?

Gjør litt metakognisjon. Ta for deg ideene du akkurat leste. For hver idé skal du vurdere om du er en effektiv eller en mindre effektiv tenker. Her er ideene:

9. Tenkevane 9: Vurdere planen

- a. Er du effektiv tenker? Skriv ja/nei/kanskje
- b. Er du mindre effektiv tenker? Skriv ja/nei/kanskje

10. Tenkevane 10: Forutse resultater

- c. Er du effektiv tenker? Skriv ja/nei/kanskje
- d. Er du mindre effektiv tenker? Skriv ja/nei/kanskje

Tenk på noen problemsituasjoner du har møtt i det siste.

1. Gi noen eksempler på situasjoner hvor du var en effektiv tenker

Tenkevane 11: Er problemet løst?

Du gjennomfører planen, men problemet er ikke løst. Det var kanskje en dårlig plan. Eller du gjør noen feil når du utfører handlinger i planen. Hvordan kan du si om problemet er løst?

Den effektive tenkerens vaner

Den effektive tenkeren stiller seg et spørsmål allerede når målet velges.

- Tenkeren spør: "Er dette målet laget slik at jeg kan se om jeg når det?" "Er det målbart?"

Eksempel: Målet mitt er at jeg skal bli bedre til å spille gitar. Hvordan skal jeg kunne sjekke det? Kanskje jeg synes selv at jeg har blitt bedre, mens andre ikke kan høre noen fremgang? "Å bli bedre" er vanskelig å måle.

Et målbart mål må kunne måles når vi har fått et resultat.

Jeg kunne velge et annet mål: "Jeg skal lære meg gitarsoloen på Stairway to heaven."

Det er langt enklere å sjekke om dette målet er nådd. Det er målbart. "Jøss! Du kan jo spille soloen på Stairway to heaven."

Den effektive tenkeren stiller seg også et spørsmål allerede når planen utarbeides.

- Tenkeren spør: "Er denne planen laget slik at jeg kan se om jeg gjør noen feil underveis?"

Etter at planen er gjennomført vil den effektive tenkeren undersøke resultatet for å se om målet er nådd.

Den mindre effektive tenkerens vaner

Den mindre effektive tenkeren gjennomfører planen sin og aksepterer resultatet. Denne tenkeren bryr seg mindre om resultatet er riktig eller om problemet er løst.

Dette er en viktig side ved problemløsning.

Det er fort gjort å lage seg en feil i sin egen plan. Det er fort gjort å gjøre en feil når man gjennomfører planen. Derfor er det nødvendig å utvikle gode ferdigheter i å undersøke resultatet opp mot både målet som er satt, og planen som er utarbeidet.

Mange elever dropper dette steget i problemløsning. De lager løsninger som helt åpenbart er feil. De glemmer å tenke nøye over sin egen løsning. Derfor oppdager de heller ikke åpenbare feil.

Tenkebane 12: Uventede sideeffekter

De stegene du gjennomfører for å løse et problem, kan ha noen dårlige sideeffekter. Nye problemer kan oppstå. Det kan skje selv om du har en god plan. Straks problemet er løst, oppstår det en ny problemsituasjon.

Den effektive tenkerens vaner

Den effektive tenkeren oppdager nye problemer som vil oppstå mens problemløsningen foregår. Denne tenkeren undersøker den nye problemsituasjonen for å se om det fører til nye viktige problemer.

Den mindre effektive tenkerens vaner

Den mindre effektive tenkeren bryr seg ikke om sideeffekter. Straks en plan er utarbeidet, glemmer denne tenkeren å vurdere om nye problemsituasjoner kan oppstå.

Skriv: Hva slags tenker er jeg?

Gjør litt metakognisjon. Ta for deg ideene du akkurat leste. For hver idé skal du vurdere om du er en effektiv eller en mindre effektiv tenker. Her er ideene:

11. Tenkevane 11: Er problemet løst?

- a. Er du effektiv tenker? Skriv ja/nei/kanskje
- b. Er du mindre effektiv tenker? Skriv ja/nei/kanskje

12. Tenkevane 12: Uventede sideeffekter

- a. Er du effektiv tenker? Skriv ja/nei/kanskje
- b. Er du mindre effektiv tenker? Skriv ja/nei/kanskje

Tenk på et problem du har løst, hvor resultatet skapte et nytt problem som var vanskeligere å løse enn det opprinnelige problemet.

1. Skriv noen ord om dette.

Oppsummering av kapittel 6

En metode for å bli bedre til å løse problemer, er å bli bedre til å tenke.

Effektive tenkere bruker hjernen mer effektivt. De har en rekke gode tenkevaner knyttet til de kjente stegene du må gå gjennom for å løse problemer.

Noen gode vaner er:

1. Verdsetting av utdanning og arbeid for å lære fag.
2. Gode ferdigheter i å sette seg mål og revurdere og **evaluere** mål.
3. Gode ferdigheter i å finne fram ressurser og kjenne til begrensninger.
4. Gode ferdigheter til å tenke på mulige resultater som kan produseres ved hjelp av en plan.

Du blir en mer effektiv tenker dersom du kan finne ut hvilke dårlige tenkevaner du har, og er villig og motivert til å arbeide for å bedre dine tenkevaner.

Aktiviteter i kapittel 6

Aktivitet 6.1 – Noe jeg tenkte på i kapittel 6

Var det noe spesielt du begynte å tenke på da du leste kapittelet? Kanskje du tenkte på noen av dine mest effektive tenkevaner? Kanskje du tenkte på at du har noen mindre effektive tenkevaner? Kanskje du tenkte på et problem hvor du brukte gode tenkevaner da du løste det?

1. Skriv tittel: "Noe jeg tenkte på i kapittel 6"
2. Skriv og forklar hva du tenkte.
3. Forklar hvorfor dette interesserer deg.
4. Hvorfor tenkte du på det?
5. Hva har det med livet utenfor skolen å gjøre?
6. Er det noe mer du tenker på?

Aktivitet 6.2 – Del ideen

Finn en elev å snakke med. Begge velger en idé i kapittelet som hver synes er viktig. Bruk ett minutt hver til å fortelle den andre om ideen. Tips til hva dere kan snakke om:

- Hvorfor er ideen viktig for deg?
- Hvilken betydning har den for ditt liv?

Når dere snakker sammen bør dere bruke begreper fra boken.

Aktivitet 6.3 – Lag en quiz som tester faktakunnskap fra kapittelet

Lag 2 spørsmål. Du skal teste faktakunnskap fra dette kapittelet.

Det skal være tekstsvar. Bare ett riktig svar til hvert spørsmål.

Spørsmålene skal være så enkle at en som ikke har lest kapittelet kan svare.

Prøv quizen på noen elever.

Aktivitet 6.4 – Lag en quiz som tester forståelse av kapittelet

Lag 2 spørsmål. Du skal teste forståelse av kapittelet.

Presenter noe fakta fra kapittelet og still spørsmål som krever tenkning og vurdering av fakta.

Testpersonene kan slå opp i en bok, gjøre et søk eller lese i dette kapittelet for å lage et svar.

En quiz som tester forståelse, har alltid flere korrekte svar. Du lager ingen svaralternativer.

Spørsmålene skal være så vanskelige at de bare kan besvares av en person som har en god forståelse av innholdet i dette kapittelet.

Prøv quizen på noen elever. Det skal være tekstsvar, og ingen fasit. Diskuter svarene sammen til slutt.

Aktivitet 6.5 – Friminuttets lengde

Tenk deg at skolen skal endre lengden på friminuttene. Lag en liste over mulige alternativer, som å øke eller minske friminuttet med ett eller flere minutter.

Gi argumenter for og imot hvert forslag.

Aktivitet 6.6 – Skoleårets lengde

Tenk deg at skoleårets lengde skal endres. Feriene blir enten kortere eller lengre. Du skal se dette problemet fra ulike synsvinkler og gi argumenter for og imot.

1. Lag en liste over forslag til skoleårets lengde. Noen kortere og noen lengre enn i dag.
2. Gi argumenter for og imot sett fra forskjellige perspektiver som
 - a. Lærere
 - b. Foreldre
 - c. Elever

Aktivitet 6.7 – Ranger dine tenkevaner

Dette kapittelet presenterer 12 tenkevaner. Din oppgave er nå å rangere dem fra best til dårligst. Den første tenkevanen du skriver opp, skal være din beste vane. Den siste du skriver opp, skal være din dårligste tenkevane.

En måte å gjøre dette på er å bryte ned problemet(oppgaven) i tre mindre problemer.

Lag følgende tre kategorier:

- A. Mine beste tenkevaner.
- B. Mine middels gode tenkevaner.
- C. Mine dårligste tenkevaner.

Legg 2 – 3 tenkevaner i kategori A.

Legg 5 – 6 tenkevaner i kategori B.

Legg 2 – 3 tenkevaner i kategori C.

Nå fortsetter du med å rangere hver tenkevane innenfor hver kategori.

Når du gjør dette, oppdager du kanskje at en tenkevane har havnet i feil kategori og må flyttes.

Når du løser problemet på denne måten, har du også gjort en øvelse i å bruke ”Bryt ned” – strategien. Denne strategien bryter større problemer ned i mindre deler som er enklere å løse. Det var vel ikke så vanskelig?

Aktivitet 6.8 – Forbedre en dårlig tenkevane

Se på tenkevanene du plasserte i kategori C i aktivitet 6.7. Altså dine dårligste tenkevaner. Velg ut en av dine dårligste tenkevaner og gjør en vurdering av om du kunne tenke deg å bli bedre på denne type tenkevane. Hvis ikke, velger du en annen tenkevane fra kategori C. Velg en som er viktig for deg.

Prøv å tenke ut hvordan du kan begynne å bli en mer effektiv tenker når det gjelder den tenkevanen du har valgt.

Lag en liten liste.

Tenk på noen situasjoner som vil inntreffe i dag eller senere denne uken, hvor du vil få mulighet til å praktisere ideene på listen.

Når du får mulighet til å praktisere ideen, skal du gjøre det.

Tenkt da også på det faktum at du øver deg på viktige tenkeferdigheter. Du blir smartere. Du praktiserer metakognisjon.

Kapittel 7 – Overføring av læring

Fokus

Å lære er en viktig forberedelse til framtiden. Du kan lære om en problemtype og en bestemt problemsituasjon i dag. Du kan bruke denne kunnskapen til å løse et tilsvarende problem senere.

Når to problemsituasjoner likner hverandre, kalles dette **nær overføring**.

Når to problemsituasjoner er veldig ulike, kalles det **fjern overføring**.

- Nær overføring er lett. Hjernen gjør det automatisk og ubevisst.
- Fjern overføring er vanskeligere. Du kan bli bedre på fjern overføring ved å studere og øve. Du lærer mer effektiv dersom du blir bedre på fjern overføring.

Introduksjon

En av de viktigste sidene ved problemløsning er å bygge på tidligere arbeid. Du har lært masse. Når et nytt problem dukker opp, vil du gjerne bruke ideer og løsninger du har lært tidligere. Men hva skal du gjøre dersom problemet er litt annerledes enn det du har løst før?

Da du var ung brukte du mye tid på å lære deg å knyte skolissene. Nå er det lett å lage en pen knute.

Men er det like lett å lage en pen knute på en presang du pakker inn? Er det samme problem som skolissene, eller er det litt annerledes?

Eller hvordan går det når du skal knyte et slips? Det er på en måte det samme, men likevel litt annerledes. Ikke sant?

Hvordan er det å lage en båtsmannsknop for å skjøte to like tau?

Det likner litt på å knyte lisser det også, men det er likevel forskjellig.

Noen ganger er en problemsituasjon ganske lik noe du har erfart tidligere. Hjernen din gjenkjenner situasjonen automatisk. Du husker hvordan du gjorde det før og hvor godt det fungerte. Det er en god hjelp når du skal løse nye problemer.

Andre ganger virker en problemsituasjon helt ny og ulik alt du har lært før. Hjernen din klarer ikke å finne noe som likner en tidligere situasjon.

Forskere har studert dette grundig. De har laget to begreper som dekker dette.

I det første tilfellet oppdager hjernen din at problemsituasjonen er ganske lik noe du har erfart før. Dette kalles **nær overføring**.

I det andre tilfellet klarer ikke hjernen å finne en liknende problemsituasjon du har møtt og løst før. Det ser ut som et helt nytt og ukjent problem. Dette kalles **fjern overføring**.

Hver persons hjerne er unik. Hver persons erfaringer er unike. Derfor kan en problemsituasjon være fjern overføring for meg, mens det for deg kan være nær overføring.

Skriv: Tenk på noen ulike problemsituasjoner

Tenk på flere "nær overføring" – problemsituasjoner du har møtt i det siste. Kanskje du brukte en ny type gaffel for å spise mat på et spisested. Du har kanskje aldri brukt en slik gaffel før, og kanskje heller ikke spist akkurat det spesielle måltidet før. Du brukte "nær overføring" for å løse situasjonen.

Tenk på noen "fjern overføring" – problemsituasjoner du har møtt i det siste. Kanskje du brukte en metode du har lært i norsk for å løse et problem du møtte hjemme. Kanskje du brukte en metode du lært i matematikk for å løse et problem i samfunnsfag.

Skriv opp problemsituasjonene du tenkte på, og sorter dem etter nær- eller fjern overføring.

Litt hjerneteori

Tenk på hva som skjer når du kommer inn i et rom fullt av mennesker. Du kikker rundt for å se om du kjenner noen. Uten å bruke hjernen særlig mye, oppdager du fort hvilke personer du kjenner. Ditt visuelle system fungerer godt for å oppdage kjente.

Hjernen din er god på å gjenkjenne mønstre, selv om det ikke er akkurat det samme mønsteret som er lagret i hjernen. Du kan kjenne igjen en venn selv om vennen din har helt ny hårfrisyrer eller helt nye klær. Du kan kjenne igjen en person du ikke har sett på mange år, selv om denne personen har blitt mye eldre og har endret kroppsvekten. Dette skjer fort og automatisk. Det er nær overføring.

Hjernen er kompleks. Den har billioner av **neuroner**. Hvert neuron har koblinger til tusenvis av andre neuroner. Neuronene og koblingene mellom dem, lagrer mønstre. Hjernen din lagrer visuelle mønstre, lydmønstre, luktmønstre og så videre.

Hjernen er god på å koble sammen lagrete mønstre med nye mønstre. Du kjenner igjen en venn fordi hjernen har lagret et mønster av hvordan ansiktet til vennen din ser ut. Når du ser på en person, begynner hjernen automatisk å søke etter et lagret mønster som stemmer med det du ser. Hvis hjernen får et treff, kjenner du igjen personen. Noen ganger gjør hjernen en feil, og du sier "Hei!" til en person du ikke kjenner.

Å gjenkjenne ansiktet til en person er ikke det samme som å huske hva personen heter. Mønstre for ansikter er lagret på et helt annet sted enn mønstre for navn. De fleste mennesker synes selv at det er lettere å huske et ansikt enn et navn.

Skriv: Eksperimenter med ansikter og navn

Prøv følgende eksperiment:

1. Tenk på et ansikt og prøv deretter å huske navnet til personen.
2. Tenk på et navn og prøv deretter å huske hvordan ansiktet ser ut.
3. Hva er du best på? Skriv.

Prøv nå dette eksperimentet:

1. Prøv å se et bilde av en venn i hodet uten å tenke på hva vennen heter.
2. Klarte du det? Skriv.
3. Prøv å tenke på navnet til en venn uten lage et bilde av ansiktet i hodet.
4. Klarte du det? Skriv.

Gå sammen med et par elever i klassen. Diskuter resultatet av eksperimentet med de andre. Er det samme resultat for alle? Var det noen store forskjeller?

Hjerneforskere studerer slike ting. De har lært mye om hjernen. Det kan hjelpe deg. De har lært at hjernen "lærer" ved å bygge nye koblinger mellom neuroner. Hjernen blir "flinkere" ved stadig å bygge koblinger mellom neuroner og stadig øve på å bruke disse koblingene. Jo mer varierte koblingene er, dess bedre blir hjernen til å takle nye varierte situasjoner.

Dette hjelper oss til å forstå hvordan vi lærer. Når du lærer noe nytt, bygger hjernen nye koblinger mellom neuroner. Jo mer hjernen øver på å bruke disse koblingene, dess bedre blir den til å bruke dem.

Betydningen av ord

Ord er veldig viktig for å lære bedre og for å overføre læring til andre situasjoner og problemer. Her er et eksempel:

I ungdomsskolen lærte du at data kan representeres eller vises som et kakediagram. Et kakediagram likner en kake delt opp i biter.

Når du tenker på ordet *kake*, tenker du på mat, dessert, kake, bli sulten?

Kakediagram er ikke et spesielt godt begrep. Det er kanskje bedre med sektordiagram. Det finnes flere typer av diagrammer, som stolpediagram og linjediagram.

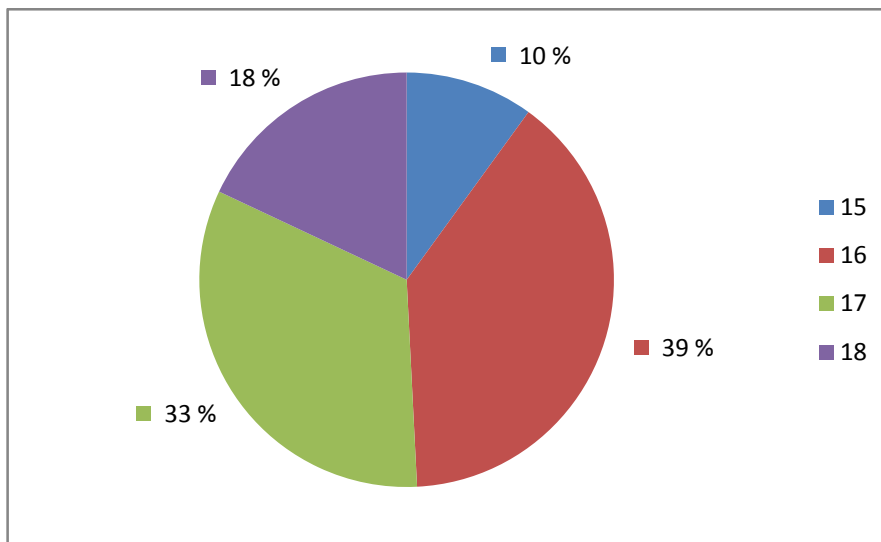
Det hjelper å ha navn som likner hverandre. Hvis du vet hva et sektordiagram er, forstår du lettere hva et stolpediagram er, siden begge har navnet *diagram* i seg. Du overfører kunnskap fra det ene til det andre.

Et sektordiagram er nyttig for å vise en helhet som er delt i flere deler. Tenk på en skole som har 250 elever. De fleste er 16 - og 17 år gamle, men noen er eldre og noen er yngre. Hvor stor prosent av elevene faller inn i hver aldersgruppe?

Et sektordiagram kan gi et bilde av dette. Vårt visuelle system er godt egnet til å forstå bildeinformasjon. Derfor er et sektordiagram egnet for å vise deler av en helhet.

Alder	Antall elever
15	25
16	98
17	82
<u>18</u>	<u>45</u>
Til sammen	250

Prosentvis fordeling av elever med ulik alder



Å lære for nær og fjern overføring

Nå skal vi snakke litt om læringsteori. Hvordan arbeider hjernen når den lærer nye ting? Du kan tenke på hvordan du lærer. Du kan lære hvordan du best lærer nye ting. Det er veldig viktig å forstå hvordan du kan hjelpe deg selv til å lære. Her er to forskjellige måter for å lære noe:

Type 1- Læring for nær overføring

Når du skal lære noe nytt, konsentrer deg bare om det nye stoffet. Arbeid med å bygge bare noen få koblinger i hjernen. Øv deg på å bruke disse få koblingene om og om igjen, slik at du kan gjøre det nye du lærte fort og nøyaktig. Bygg deg noen få, veldig sterke **neurale koblinger**.

Type 2 – Læring for fjern overføring

Når du skal lære noe nytt, konsentrer deg både om det nye og mange forskjellige ting som du synes henger sammen med det nye. Arbeid for å bygge mange koblinger. Koble det nye til mange ting du allerede kan. Bygg deg mange sett av neurale koblinger.

Type 1 er viktig hvis du skal gjøre noe fort og nøyaktig, kanskje uten å tenke så mye. Du øver om og om igjen. Til slutt blir du god på det. Neste gang du får bruk for det, går det av seg selv. Det blir en primitiv for deg.

Type 2 er viktig for fjern overføring. Når hjernen oppdager en ny problemsituasjon, leter den etter mønstre som likner den nye situasjonen. Jo flere koblinger du har til den gamle problemsituasjonen, dess bedre blir hjernen på å finne mønstre som er **relaterte**. Men det tar tid å lage mange neurale koblinger i hjernen.

I sport kan det være nyttig med type 1. For å gjøre det bra, må du trene hjernen til å reagere hurtig og nøyaktig i visse situasjoner som gjentar seg hele tiden.

Men i de fleste læringssituasjoner er det viktig å kunne bruke begge typer. Du blir bedre i problemløsning ved å bygge deg et stort og variert antall av neurale koblinger i hjernen.

Skriv: Hva er jeg best på?

Tenk på type 1 og 2.

1. Hvilken type læring er du best på?
2. Hvorfor?
3. Hvilken type er mest nyttig i skolearbeidet?
4. Hvorfor?
5. Hvilken type er mest nyttig utenfor skolen?
6. Hvorfor?

Et eksempel på type 1 og type 2

Matematikklæreren vet at sektordiagrammet er viktig. Men mange elever synes det er vanskelig å lære hvordan de skal tegne et sektordiagram for hånd. Du må arbeide med prosent. Dette krever divisjon. Du må vite at sirkelen har 360 grader. Du må for eksempel regne ut hvor mange grader 18% er. Dette krever multiplikasjon og arbeid med desimaler. Du må kunne bruke en transportør og måle grader, og en passer til å tegne sirkelen.

På grunn av alt dette, bruker mattelæreren mye tid på å lære deg å lage et sektordiagram (sirkeldiagram). Dette er type 1 – læring. Du vet at du blir testet i dette. Derfor bruker du mye energi på type 1 – læring. Ditt mål er å klare dette godt på neste prøve.

Noen uker senere lærer du om inntektsnivåer i samfunnet i samfunnsfag. Kanskje klassen studerer hvor mange mennesker som tjener mindre enn 300.000 kroner i året, hvor mange som tjener mellom 3 – 400.000 kroner og hvor mange som tjener over 400.000 kroner. Oppdager du da at det ville være nyttig å bruke sektordiagrammet igjen?

For noen elever er svaret ”nei”. For disse elevene vil inntektsproblemet i samfunnsfag være fjernt fra det de lærte om sektordiagrammer i matematikk for to uker siden. De var så opptatt av type 1 – læring, at de glemte å bruke tid på type 2 – læring. De brukte ikke tid på å bygge opp et variert sett av neurale koblinger i hjernen. Derfor klarer ikke hjernen å oppdage mønsteret de lærte i matematikk. Derfor er hjernen ikke i stand til å innse at sektordiagrammet kunne være nyttig i samfunnsfaget.

Skriv: Bruk av sektordiagram

Det kan vær nyttig å kunne lage et sektordiagram i mange fag og i mange ulike emner innenfor fagene.

1. Lag en liste over situasjoner hvor sektordiagrammet er nyttig.
2. Er det å kunne lage et sektordiagram en primitiv du har?

Hvor lang tid tror du det tar for deg å lage et sektordiagram med følgende syv sektorer:

1. Sove
2. Skoletid
3. Se på TV, høre radio, lytte på musikk
4. Spise
5. Snakke med venner
6. Reise mellom steder
7. Andre aktiviteter

Gjør først et overslag over hvor mange timer i uka du bruker på hvert punkt. Totalt antall timer på en uke er $7 \times 24 = 168$.

Lag deretter et sektordiagram med dine data. Ta tiden på arbeidet med å lage det.

Konsepter og prosesser

Sektordiagrammet illustrerer en veldig viktig side ved læring. Det er ideen om primitiver. Dette leste du i kapittel 5.

Det er to helt forskjellige ting å lære om sektordiagrammet:

1. **Konseptet** sektordiagram. Konseptet er dette; Et sektordiagram gir et godt *bilde* av alle delene som utgjør helheten. Hvert kakestykke, som til sammen blir en hel kake, om du vil. En person kan lett se hvor stor hver enkelt del er av hele sirkelen.
2. **Proessen** eller stegene du må utføre for å lage et sektordiagram. Å lage et sektordiagram krever at du kan ganske mye matematikk.

Hjernen er i stand til å lære konseptet sektordiagram og alle prosessene som kreves for å lage diagrammet for hånd.

En datamaskin kan hurtig omdanne noen tall i et regneark til et sektordiagram.

Datamaskinen trenger bare noen tall, og kan deretter gjennomføre prosessen med å lage et helt korrekt diagram for deg.

Dette betyr at å lage et sektordiagram med en datamaskin, kan bli en primitiv for deg, hvis du lærer deg konseptet.

Altså: Du må forstå hva et sektordiagram brukes til og når det er lurt å bruke det.

Når du har en datamaskin med et program som kan lage sektordiagrammer, har du tilgang til en primitiv på datamaskinen.

Skriv: Stolpediagram

Er det lettere for deg å lage et sektordiagram enn et stolpediagram?

På hvilken måte er et sektordiagram og et stolpediagram nesten det samme for deg?

På hvilken måte er de forskjellige?

Har du brukt et dataprogram som kan tegne et sektordiagram? Da kan det antakelig også tegne stolpediagram. Med noen klikk kan du skifte fra sektor- til stolpediagram med de samme dataene. Hvordan kan det hjelpe deg å løse problemer?

Lære å bli bedre på fjern overføring

Hvor mange ganger har du sittet i klassen og tenkt:

"Jeg lurer på hvorfor jeg skal lære dette? Det ser ikke ut som det har mye verdi for meg."

Slik tenkning er en god start for å bli bedre på fjern overføring. Din hjerne kan koble sammen kunnskap akkurat som du vil. Det er opp til deg å fortelle hjernen hvilke neurale koblinger den skal bygge. Men det krever at du er bevisst på hva hjernen skal koble.

Du forteller hjernen hvordan den skal bygge koblinger mellom ulike kunnskaper gjennom å tenke på hvordan ting henger sammen. Hvis du vil at hjernen skal koble sektordiagrammer og inntektsfordeling i samfunnet, må du bevisst tenke på at det kan være en kobling.

Bare du kan gi et fornuftig svar på spørsmålet: "Hvorfor skal jeg lære dette?"

Alt du lærer kan relateres til andre ting du allerede vet. Den effektive tenkeren lager bevisst hele tiden koblinger mellom ulike kunnskaper. Den effektive tenkeren finner mange svar på spørsmålet: "Hvorfor skal jeg lære dette?"

Skriv: Katt og hund

La oss øve litt.

1. Tenk på hvilke måter en katt og en hund er relatert til hverandre.
2. Tenk så på hvilke måter en katt og et tre er relatert til hverandre.
3. Tenk til slutt på hvilke måter en hund og et tre er relatert til hverandre.

Anta at du hadde startet med punkt 3 (På hvilke måter er en hund og et tre relatert til hverandre.)

1. Ville du ha laget de samme koblingene da?
2. Hvorfor?
3. Hvorfor ikke?

La oss gå tilbake til sektordiagrammet. Mattelæreren påpeker at det er flere viktige prinsipper for sektordiagrammer:

1. For mange problemer er det viktig å tenke på "deler av en helhet".
2. Et sektordiagram er en god visning av delene av en helhet.

Du kan høre på hva læreren sier. Du kan pugge uttrykket "deler av en helhet". Du kan bruke kreftene på type 1 – læring som vil hjelpe deg til å klare den neste prøva.

Det er flere ting som er galt med denne måten å lære på:

1. Det vil skje lite type 2 – læring.
2. Du vil ikke utvikle et rikt og variert utvalg av neurale koblinger i hjernen, noe som er nødvendig for å bli bedre på fjern overføring av læring til nye situasjoner.
3. Du vil bruke tiden til å lære noe en datamaskin kan gjøre for deg.

Husk spørsmålet vi stilte tidligere i boken:

"Hvis en datamaskin kan løse eller hjelpe deg å løse et problem, hva skal da en elev lære om å løse dette problemet?"

Vi er i ferd med å utvikle et svar.

I eksempelet med sektordiagrammet er det to forskjellige typer av læring som skjer.

1. Type 1 – læring hjelper hjernen til å arbeide som en datamaskin. Du lærer bare en prosedyre for hvordan du skal lage sektordiagrammet. Det hjelper hjernen til å utføre

prosedyren raskt og riktig. Men ofte kan en datamaskin utføre denne prosedyren mye bedre enn et menneske.

2. Type 2 – læring hjelper hjernen til å løse nye og varierte problemsituasjoner. Du lærer deg konseptene i stedet for prosessene. Type 2 – læring er opptatt av hvordan de generelle ideene kan brukes på mange ulike problemsituasjoner. Mennesker er gode på dette. Et menneske kan lære at et dikt og et tre kan ha noe med hverandre å gjøre ved at begge er vakre. Et menneske kan forstå hva det menes at noe er vakkert. Det er datamaskiner dårlig til.

Oppsummering av kapittel 7

Du kan bli mye bedre på fjern overføring ved å studere og praktisere det. Når du lærer deg å løse et nytt problem, skal du tenke:

1. Hva er de underliggende konseptene?
2. Hvordan er disse konseptene relatert til andre ting jeg allerede kan?
3. Hvilke prosesser er nødvendige for å løse problemet?
4. Kan en datamaskin eller en annen maskin utføre disse prosessene for meg?

Aktiviteter i kapittel 7

Aktivitet 7.1 – Noe jeg tenkte på i kapittel 7

Var det noe spesielt du begynte å tenke på da du leste kapittelet? Kanskje du tenkte på at "fjern overføring" er en interessant idé? Kanskje du tenkte på hvorfor noen lærere legger så mye vekt på "nær overføring"? Kanskje du tenkte på hvilke måter et dikt og et tre er relatert til hverandre?

1. Skriv tittel: "Noe jeg tenkte på i kapittel 7"
2. Skriv og forklar hva du tenkte.
3. Forklar hvorfor dette interesserer deg.
4. Hvorfor tenkte du på det?
5. Hva har det med livet utenfor skolen å gjøre?
6. Er det noe mer du tenker på?

Aktivitet 7.2 – Del ideen

Finn en elev å snakke med. Begge velger en idé i kapittelet som hver synes er viktig. Bruk ett minutt hver til å fortelle den andre om ideen. Tips til hva dere kan snakke om:

- Hvorfor er ideen viktig for deg?
- Hvilken betydning har den for ditt liv?

Når dere snakker sammen bør dere bruke begreper fra kapittelet:

- Fjern overføring
- Nær overføring
- Konsepter
- Prosesser

Aktivitet 7.3 – Lag en quiz som tester faktakunnskap fra kapittelet

Lag 2 spørsmål. Du skal teste faktakunnskap fra dette kapittelet.

Det skal være tekstsvar. Bare ett riktig svar til hvert spørsmål.

Spørsmålene skal være så enkle at en som ikke har lest kapittelet kan svare.

Prøv quizen på noen elever.

Aktivitet 7.4 – Lag en quiz som tester forståelse av kapittelet

Lag 2 spørsmål. Du skal teste forståelse av kapittelet.

Presenter noe fakta fra kapittelet og still spørsmål som krever tenkning og vurdering av fakta.

Testpersonene kan slå opp i en bok, gjøre et søk eller lese i dette kapittelet for å lage et svar.

En quiz som tester forståelse, har alltid flere korrekte svar. Du lager ingen svaralternativer.

Spørsmålene skal være så vanskelige at de bare kan besvares av en person som har en god forståelse av innholdet i dette kapittelet.

Prøv quizen på noen elever. Det skal være tekstsvar, og ingen fasit. Diskuter svarene sammen til slutt.

Aktivitet 7.5 – Konsepter og prosesser i fire fag

1. Velg fire fag du har på skolen.
2. For hvert fag skal du gi ett eksempel på et konsept og ett eksempel på en prosess(prosedyre) som er viktig å lære i faget.
3. Hvilket av fagene bruker mest tid på konsepter?
4. Hvorfor tror du at dette faget bruker mest tid på konsepter?
5. Hvilket fag bruker mest tid på prosesser(prosedyrer)?
6. Hvorfor tror du at dette faget bruker mest tid på prosesser?

Aktivitet 7.6 – Overføring i multiplikasjon

Du kan sikkert utføre multiplikasjonen $5,293 \times 718$.

Du vet også at multiplikasjon er en hurtig måte å utføre repetert addisjon på. Skulle du regne ut dette stykket med addisjon, ble det 718 linjer med det samme tallet(5,293).

Du har sikkert trent mye på å multiplisere slike tall på barneskolen og kanskje også ungdomsskolen.

1. Diskuter hvordan ting du har lært om multiplikasjon av hele tall kan overføres til å lære multiplikasjon av desimaltall som for eksempel: $3,73 \times 89,413$
2. Diskusjonen skal snakke om både konsepter og prosesser

Aktivitet 7.7 – Verdien av selvtillit

En viktig ting for å løse problemer er selvtillit. Noen mennesker vet at de er gode til å løse problemer. Når de møter et nytt problem, er en av styrkene deres at de tror de kan klare å løse det. Dette er en ressurs som kan overføres til en hver problemsituasjon.

1. Tenk på din egen grad av selvtillit som problemløser.
2. Er den så god som den burde være?

3. Hva kan gjøre selvtilliten bedre?
4. Tenk på hvordan du kan øke selvtilliten din som problemløser.
5. Praktiser det hver eneste dag.

Kapittel 8 – Modellering

Fokus

En modell kan tjene som en erstatning for ”det virkelige” for å hjelpe deg å løse problemer. Det er flere måter man kan lage modeller på. Modellen skal representere problemet. En måte er en mental modell du lager i ditt hode. En annen er en verbal modell som du lager ved å forklare problemet for andre. En tredje modell er å bruke skriving og bilder. Til slutt nevner vi det å lage en modell med en datamaskin.

Du kan bli bedre til å løse problemer ved å:

- Lære ulike måter å lage modeller på
- Teste ut ulike modeller og lære når en modell er bedre enn en annen.
- Lære å lage modeller med datamaskinen.

Introduksjon

For å løse et problem, må du forstå problemet. Du må ha et godt definert problem.

En representasjon av problemet kalles en **modell** av problemet. Det er mange måter å lage modeller på. Du kan for eksempel lage en miniatyr av problemet eller tegne et bilde av problemet. Du kan tenke på en arkitekts tegning av et hus som en modell av huset. Du kan lage en modell med ord eller matematikk. Hver modell har sine fordeler. En type modell som er nyttig for ett spesielt problem, passer kanskje ikke for et annet problem.

I dette kapittelet vil vi drøfte ulike modeller. Hver type har sine spesielle særtrekk og egenskaper. Vi vil snakke om hvordan datamaskiner kan brukes til å lage modeller. Bruk av datamaskiner er veldig kraftig verktøy for å løse mange ulike problemtyper.

Mentale modeller

Du kan løse mange problemer i hodet. Du tenker på et problem. Du vurderer ulike ting du kan gjøre for å løse problemet. I tankene kan du se hvilke steg du må gjennomføre for å nå målet. Du forestiller deg hvordan du kan utføre ulike oppgaver og oppnå det ønskede resultatet. Mens du tenker på det, får du nye ideer. Du endrer planen din.

Tenk på et problem du jobbet med nylig. Klargjør hele problemet i hjernen din.

Lat som du skal forklare problemet for deg selv. Fremstill problemet for deg selv ved å bruke ord og bilder i hodet. Hvilke følelser får du for problemet? Kanskje du til og med kan lukte eller smake noe?

Du har akkurat laget en **mental modell** av problemet. Det er et indre bilde, en mental representasjon for deg selv. En modell er ikke det samme som et objekt eller ide som blir

modellert. En mental modell er ikke det samme som et problem. Men en modell kan representere noen viktige deler av et problem. Det kan hjelpe deg å løse problemet.

Du kan bruke en mental modell for å tenke om et problem. Du kan bruke modellen til å vurdere ulike tiltak. Du kan forestille deg hva som skjer dersom du velger noen spesielle tiltak du vil gjennomføre. En mental modell er en del av tenkningen. Det er veldig viktig i problemløsning.

Skriv: Litt metakognisjon

Tenk på en mental modell du utformet tidligere i boken. Var modellen hovedsakelig bilder? Var modellen hovedsakelig ord og lyder? Noen ganger er en bildemodell mest nyttig. Andre ganger er ord og lyder nyttige.

Tenk på en ku. Når du tenker på en ku, får du et bilde i hodet? Hører du rauting? Kjenner du lukten av kumøkk? Den mentale modellen du får er avhengig av din erfaring med kuer.

Tenk på en kald og vindfull dag. Gjør metakognisjon på den mentale modellen.
Tenk på å gå seg vill i skogen. Gjør metakognisjon på den mentale modellen.

--

Din hjerne er kraftfull. Den kan lage en mental modell av hva du enn måtte tenke på. Hvis du tenker på en sky, lager du en mental modell av en sky. Hvis du tenker på en flekkete ku, vil du lage en mental modell av en flekkete ku. Med litt innsats klarer du til og med å lage en mental modell av flekkete melk fra en slik ku!

Verbale modeller

Det er nyttig å dele sine mentale modeller med andre mennesker. En måte å gjøre det på er å snakke. Du kan bruke ord og andre lyder for å representere en **verbal modell**. Du kan også kommunisere din verbale modell med kroppsspråk. En stor del av din kommunikasjon formidles også med ansiktsuttrykk, håndbevegelser og andre kroppsuttrykk. Noen ganger kaller vi derfor en verbal modell for verbal/kroppsspråk – modell, for å vise at både lyder og kroppsspråk er en del av kommunikasjonen.

Du er antakelig god til å snakke. Men noen ganger er det vanskelig å finne de rette ordene. Hvordan kan du finne ord som beskriver smaken av god mat? Kan du finne ord som beskriver hva en god venn er? Noen ganger er det ikke lett å lage en god verbal modell av det som skjer inne i hodet. Det er derfor til stor nytte å ha et stort ordforråd.

Tenkning og språk henger sammen. Når du tenker, snakker du inne i deg selv. Når du snakker høyt, hører du bedre hva du tenker, og det hjelper deg derfor å organisere tankene dine. Det er en metode for å praktisere tenkning. Du kan bli bedre til å tenke ved å bli bedre til å snakke. Ord er en støtte for tenkning. Dette er grunnen til at et stort ordforråd og tale blir vektlagt i denne boken.

Når mennesker blir eldre kan de si: " Nå merker jeg at jeg har blitt eldre. Jeg snakker høyt med meg selv." Er det et tegn på at de begynner å bli senile? Kanskje det heller er et tegn på at de har blitt gode på å lage seg verbale modeller? De har forstått gjennom et langt liv,

kanskje ubevisst, at språk er intelligens og intelligens er språk. De har blitt mer intelligente, ikke senile!

Forskere har funnet ut at det å snakke om det du lærer, er til stor hjelp for å lære. Lærere har kjent til dette i mange år. De lærer selvsagt mye om et fag selv, ved å fortelle om faget til elevene. Du kan bli bedre til å lære ved å forklare nye ideer for venner.

Skriv: Lag en verbal modell

Dette er et spill som kan utføres i små grupper. Lag små grupper på 2 - 3 elever.

En person er "avsender". De andre to er "mottakere".

1. Avsender lager en mental modell i hodet av en person, et sted eller en ting.
2. Avsender gir en verbal beskrivelse av objektet, uten å si hva det er.
3. Straks mottaker vet hva det er, skrives dette ned.
4. Spillet avsluttes når begge mottakere har skrevet ned hva det er.
5. Sjekk til slutt for å se om svaret er korrekt.
6. Snakk sammen om hvilke ledetråder som var mest nyttige for å finne svaret.
7. Bytt avsender og gjenta spillet.

Alternativt kan det samme spillet gjennomføres ved at avsender lager en tegning i stedet for å forklare verbalt. Det kan også gjennomføres bare med kroppsspråk. Ingen tale.

Skriftlige modeller

Oppfinnelsen av lesing og skrijving er utvilsomt den viktigste hendelsen i utdanningens historie. Skrijving, bruk av bokstaver, tall, tegnsetting, diagrammer og liknende, er en måte å lage en modell av talespråket. Vi kaller dette en *skrevet symbolisk modell*, eller bare en **skriftlig modell**. Når du skriver, lager du en skriftlig modell. Når du tegner, lager du en skriftlig modell av et objekt.

Å skrive er sterkt knyttet til det å tenke. Skrijving hjelper hjernen å tenke. Skrijving hjelper deg å organisere tankene. Skrijving er en veldig viktig hjelp i problemløsning. For å være en god skriver, må du være en god tenker.

Skrijving har i tillegg andre fordeler. Det hjelper deg å huske. Derfor kan du lage en skriftlig modell som er langt mer komplisert enn det du klarer å holde orden på bare i hodet. Denne skriftlige modellen kan lagres for senere bruk. Kopier kan lages og gis til andre mennesker.

Litt historie om skrijving

Forskere har funnet tegninger på hulevegger som er mer enn 15.000 år gamle. På den tiden skaffet folk seg mat ved å jakte og samle ville vekster, frukt, grønnsaker og korn. Tegningene på huleveggene er skrevne modeller. Kanskje de hjalp mennesker å huske vellykket jakt eller andre viktige begivenheter.

For mer enn 10.000 år siden begynte noen mennesker å dyrke jorda. De lærte seg å så korn og lage avlinger. De lærte seg å holde husdyr. Men dette skapte mange problemer.

Når skal vi plante? Når skal vi så? Når skal vi høste? Hvor mange husdyr klarer jeg å fø på i vinter? Er det nok mat til vinteren kommer?

Tall og skrift ble utviklet for å løse disse problemene. De tidligste skriftspråkene brukte bilder som sto for ideer og ord. En tegning av en sau, en enkel strektegning av en person, og så videre.

Men hvordan tegner vi en gammel person, et barn, en baby, ei jente og så videre? Hvordan skal vi beskrive med tegninger hvem som er mor, far, tante, venner?

Det er lett å se hvorfor språkene har så mange ord.

Utviklingen av skriftspråket førte til et viktig vendepunkt for tidlige bønder. Et godt eksempel kommer fra Nilen i Afrika. Hvert år førte flom i Nilen til at åkrene ble oversvømt av vann og fjernet grensemerkene til bøndene. Ordet geometri betyr "å måle jorden". Geometri var påkrevet for å kartlegge landet og dermed fastsette grensene.

Har du noen gang tenkt over hvorfor det er så mange forskjellige fag på skolen? Hvorfor har vi kunst- og håndverk, musikk, kroppsøving, matematikk, naturfag og så videre? Hvert fag forholder seg til ulike typer av problemer. Problemene i kunst- og håndverk er ganske forskjellige fra problemene i naturfag eller engelsk.

Mennesker som arbeider innenfor et fag, ønsker å lage presise beskrivelser av problemene i faget. De ønsker å lage nøyaktige beskrivelser av hvordan de prøver å løse problemene i faget. De ønsker å gi eksempler på problemer som de vet hvordan man kan løse. Dette fører til at hvert enkelt fag utvikler sine egne ord og spesielle symboler. Dette hjelper fagpersoner til å lage veldig presise skrevne modeller innenfor sitt eget fag. Hvert fag utvikler sin egen **terminologi**, eller fagspråk.

Et hovedmål i hvert fag er å lære hvordan man lager skrevne modeller for problemene i faget. Det betyr at det er veldig viktig å lære fagspråket. Det er veldig viktig å øve seg på å snakke og skrive om ideene innenfor hvert eneste fag. Det er viktig å lære terminologien.

Skriv: Noter og musikk

Tenk på musikere. De kommuniserer med skrevne modeller som består av noter. Tenk på et annet fag som bruker spesielle tegn og ord. Skriv noen eksempler på spesielle tegn og spesielle faguttrykk som brukes i dette faget.

Skriftlige modeller hjelper hjernen

Neuronene i hjernen gjør to forskjellige ting. De *lagrer* informasjon og de **prosesserer**(behandler) informasjon. Hjernen din er både en lagringsenhet og en **prossessor**(tenkeenhet).

Dette er nøkkelideen. Ideen foreslår to typer av hjelp for hjernen. Den ene typen er hjelp til å lagre. Den andre typen er hjelp til å prosessere. En ordbok hjelper til med å lagre informasjon. En kalkulator hjelper deg å prosessere, siden den regner for deg.

Det er mulig å få hjelp til både lagring og prosessering i ett og samme hjelpemiddel. Skrivning med papir og blyant er lagringshjelp, men også prosesseringshjelp. Skrivning hjelper deg nemlig med å organisere og omorganisere tankene dine. Det er prosessering.

Å regne med papir og blyant er et annet eksempel. Tenk på å utføre en større divisjon på papiret. Du bruker papiret som midlertidig lagring av tall hver gang du deler. Du bruker også skriving for å hjelpe deg å prosessere tall og tegn.

Skriv: Huskelister

1. Tenk på noen situasjoner hvor du har brukt blyant og papir som hjelp til å huske ting. Kanskje du har notert deg telefonnummer og adresser som du ikke må glemme?
2. Tenk på noen problemer hvor du har brukt hjelpemidler for å prosessere informasjon. Kanskje du har brukt en kalkulator?

Skalamodeller

Du har sikkert sett modeller av fly, hus, biler og så videre. En **skalamodell** av en bygning kan brukes som hjelp til å lage bygningen. Hvordan vil bygget se ut? Hvordan vil det passe inn i landskapet? Hvordan passer det sammen med andre bygninger som står der fra før? Alt dette kan testes ut ved å lage en modell, enten todimensjonalt på papir eller tredimensjonalt som en miniatyrbygning du kan plassere på et bord.

Flyingeniører og flydesignere lager miniatyrmodeller av fly og tester dem ut i en vindtunnel. Hvor mye luftmotstand vil det bli? Det er mye billigere enn å lage et fullt ferdig fly for testing. Og tryggere.

Skriv: Eksempler på bruk av ulike modeller

1. Gi et eksempel på en gang du brukte en mental modell for å løse et problem.
2. Gjør det samme for en verbal modell.
3. Gjør det samme for en skriftlig modell.
4. Gjør det samme for en skalamodell.
5. Tenk på andre typer av modeller du har brukt for å løse problemer. For hver modell, gi eksempel på problemet du arbeidet med.

Eksempler: Et budsjett er en modell for å løse et pengeproblem. Et kart er en modell for å finne fram til et sted. En oppskrift er en modell for å lage en matrett.

Matematiske modeller

Du har kanskje tenkt på hvorfor du bør lære så mye matematikk på skolen? Grunnen er at matematikk er en veldig god hjelp for å lage modeller for mange problemer.

Tenk på en snekker som skal lage et bord. Bordet vil ha en rektangulær bordplate og en pen list rundt kantene. Snekkeren planlegger hvor mye materialer han må skaffe til bordplata. Han trenger både en plate og en list. Her er en skriftlig modell av dette:

- Arealet av bordplata er lengde ganger bredde.
- Omkretsen av bordplata er to lengder pluss to bredder.

Tenk på en bonde som planlegger å sette opp et gjerde og dermed lage et rektangulært beite.

Her er en skriftlig modell for hans problem:

- Arealet av beitet er lengde ganger bredde.
- Lengden av gjerdet som trengs for å komme rundt er to lengder pluss to bredder.

Matematikere har brukt mye tid på å tenke ut slike skrevne **matematikkmodeller**. De oppdaget at de to modellene vi akkurat har beskrevet, er omtrent like. Se på følgende formler:

$$\text{Areal} = \text{Lengde} \times \text{Bredde}$$

$$\text{Omkrets} = 2 \times \text{lengde} + 2 \times \text{bredde}$$

Disse formlene er matematiske modeller for både snekkerens og bondens problem. Men selv slike enkle formler inneholder for mange ord hvis vi skal skrive dem om og om igjen. Derfor har matematikere utviklet forkortelser for dem:

$$A = l \cdot b$$

$$O = 2(l+b)$$

Disse to eksemplene illustrerer to viktige ting:

1. Ekspertene på et område utvikler spesielle fagbegreper, spesiell skrivemåte og spesielle forkortelser for å representere skrevne modeller i faget sitt. Hvis du ønsker å bruke modeller fra et fag, må du lære fagbegrepene, skrivemåten (**notasjon**) og forkortelsene.
2. Den samme modellen kan brukes på en rekke forskjellige problemer. Et sett av matematiske formler kan tjene som modeller for å hjelpe til med å løse problemer for både en snekker og en bonde. Dette hjelper med å overføre kunnskap og ferdigheter i problemløsning på tvers av fag.

Men det er enda en viktig ting ved dette eksempelet.

Matematikkmodellen for rektanglet sier ingen ting om hvilke materialer som kan velges. Den sier ingen ting om hvilket treslag snekkeren kan bruke i bordplata. Den sier ingen ting om hva slags gjerde bonden skal velge. Formlene er bare modeller for to forhold ved et rektangel; Areal og omkrets.

Dette er veldig viktig å tenke litt på. En modell er en framstilling av spesielle forhold ved et objekt. Hvis du vil studere disse spesielle forholdene, kan modellen hjelpe deg. Men objektet du modellerer har mange andre egenskaper som ikke dekkes av modellen. Hvis du vil undersøke disse egenskapene, vil ikke modellen hjelpe deg.

Skriv: Bruk av formler for rektangel

Gi flere eksempler på bruk av formler for et rektangel. For hvert problem du beskriver skal du forklare hvilke deler av problemet formelen ikke løser.

Eksempler:

- Arealformelen kan brukes når du skal male en vegg. Men formelen sier ingen ting om hvilken farge du skal velge, eller hva slags maling du skal bruke.
- Formelen for omkrets kan brukes når det skal settes opp et gjerde rundt en parkeringsplass. Men formelen sier ingen ting om høyden på gjerdet, hvilket materiale som bør brukes eller hvor pent gjerdet vil bli.

Datamaskinmodeller

Datamaskinen ble oppfunnet for å hjelpe til med å løse regneoppgaver. Ideen var at flere matematikkformler kunne lagres i datamaskinen. Ved å mate maskinen med tall, kunne den raskt beregne svar ved å bruke formlene den kjente. Målet var å gjøre det langt enklere å løse problemer som krever store kompliserte og arbeidskrevende utregninger.

Vi snakket tidligere i kapittelet om å lage en skalamodell av et fly. Vi snakket om å bruke en vindtunnel for å teste designet på flyet. I de senere årene har utviklere laget nye måter å teste fly. De lager formler som beskriver formen på flyet. De lager formler som beskriver hvordan et fly vil oppføre seg i ulike hastigheter. Formlene bruker matematikksymboler. Disse symbolene kan legges inn i en datamaskin. Dette kalles en *symbolisk datamaskinmodell*. Vi kaller det bare en **datamaskinmodell**.

Med en datamaskinmodell kan forskere gjøre de samme testene for fly som de kan gjøre med en skalamodell i en vindtunnel. På mange måter er en datamaskinmodell bedre enn en skalamodell. Det er lettere å endre noen formler enn å bygge en helt ny skalamodell.

Straks de første datamaskinene kom på markedet, begynte mennesker å tenke på mange andre måter å bruke en datamaskin på. De tenkte ikke bare på datamaskinen som en avansert regnemaskin. Folk forsto snart at datamaskinmodellen kunne brukes på mange ulike problemer. Etter hvert har datamaskiner blitt billigere og lett tilgjengelige, og flere bruksområder har blitt utviklet og utvikles stadig.

Men en enkel idé gjelder enda. En datamaskin er et verktøy for å *lagre og prosessere* symboliske modeller.

Hvorfor er datamaskiner så viktige?

En mental modell er meget nyttig for å løse problemer. En mental modell er lett å endre, du tenker bare litt annerledes. De endrer bildene i hodet ditt. Da kan du lett tenke på å løse problemet med en annen mental modell. Du buker hjernen både til å lagre og prosessere.

Imidlertid har hjernen din begrensninger som datalager, fordi den lett kan glemme viktige ting. Hjernen har også begrensninger som prosessor, fordi hjernen blir trøtt, kanskje ukonsentrert og lett gjør feil. Likevel er det viktig å bruke hjernen.

En skriftlig modell er vanskeligere å lage enn en mental modell. Den er også vanskeligere å endre enn en mental modell. Men en skriftlig modell er en god hjelp til å lagre og prosessere informasjon. En skriftlig modell er mer permanent, og det er mulig å ta kopier. Dermed kan modellen deles med andre, og andre kan komme med innspill.

Det ville vært kjekt å ha noe som kombinerer de gode egenskapene fra en mental modell med de gode egenskapene fra en skriftlig modell. Du trenger en modell som er lett å endre, men samtidig tar vare på informasjon og kan utføre de samme prosessene nøyaktig på samme måte hver gang.

Datamaskinen er svaret på dette. Den kan lagre informasjon og prosessere informasjon. En datamaskinmodell er lett å endre. En datamaskin kan prosessere informasjon som er i datamaskinmodellen. Informasjonen i en datamaskin kan lett deles med andre. Disse egenskapene gjør datamaskinen til et veldig nyttig verktøy i problemløsning.

La oss si at du arbeider med å skrive en rapport. Målet er å gjøre en veldig god skrivejobb. Det betyr at du vil lese gjennom rapporten din nøye. Du vil være interessert i å finpusse på setningene, slik at den sier akkurat det du vil si, og på en måte som du er fornøyd med. Du vil også rette alle stavefeil og grammatiske feil du kan finne.

La oss anta at du har en datamaskin med et tekstbehandlingsprogram som du benytter til skrivejobben. Med en tekstbehandler er det enkelt å gjøre endringer i rapporten. Det er lett å sette in nye setninger og gjøre rapporten klarere. Det er lett å flytte setninger hvis det er nødvendig. Det er lett å fjerne tekst du ikke er fornøyd med. Det er lett å rette stavefeil og grammatikk.

En tekstbehandler har sikkert stavekontroll og grammatikkontroll. Stavekontrollen slår opp alle ord i en ordliste. Den gjør det fort og automatisk for deg. Slik prosessering kan en datamaskin gjøre veldig bra.

Nå begynner du å se verdien av en datamaskinmodell. En datamaskin er lett å endre. Datamaskinen kan hjelpe deg med å bestemme hvilke endringer du skal gjøre!

Mer om datamaskinmodeller

Resten av denne boken handler om datamaskinmodeller. Du vil lære om noen av problemene som kan representeres med datamaskinmodeller. Her er noen:

1. Det er lett å lære hvordan vi kan bruke en datamaskin for å arbeide med grafiske problemer.
2. Du kan lære hvordan du kan bruke en datamaskin for å arbeide med databaser. En database kan brukes i hvert eneste fag du studerer på skolen.
3. Noen datamaskinmodeller er spesialiserte. Det finnes programmer som hjelper en person å skrive noter eller andre musikk-symboler.
4. En datamaskin kan prosessere mange forskjellige matematikkproblemer.

En datamaskin kan være nyttig for å løse mange forskjellige problemer. Det er imidlertid viktig å forstå datamaskinens begrensinger, og at det er en rekke problemer den ikke kan løse. Du må fremdeles bruke hjernen.

Nøkkelen til å forstå hvordan datamaskinen kan hjelpe til å løse et problem, er å forstå datamaskinmodeller.

Det er lett å se hvordan man skal lage en datamaskinmodell for en ordliste. Vi laster bare opp alle ordene til maskinens minne.

Det er lett å se hvordan man skal lage en datamaskinmodell for å sjekke stavefeil. Du bare får maskinen til å slå opp hvert ord i ordlista.

Men her kan du allerede nå begynne å se datamaskinens begrensninger. Se på setningen:

”Jeg gikk på bussen”.

Stavekontrollen vil ikke fortelle deg at ”på” kanskje skulle vært ”til”. Men kanskje du gikk ”på” bussen og ikke ”til” den. Ja, da var det jo riktig. Men sikkert ikke behagelig.

Imidlertid ser vi at vi fremdeles må prosessere også med vår egen hjerne. Vi kan ikke overlate alt til stavekontrollen.

Det er andre viktigere problemer hvor datamaskinmodellen har sine begrensninger.

- Er det mulig å lage en datamaskinmodell for menneskelige følelser?
- Hvordan skal vi modellere opplevelsen av en vakker solnedgang?
- Hvordan skal vi modellere følelsen av å være forelsket?

Det er mange ting som ikke kan modelleres særlig bra i en datamaskin. En datamaskin er ikke særlig brukbar for å modellere menneskelige verdier, følelser og skjønnhet.

Datamaskinmodeller er likevel viktig. Det er et kraftig verktøy i problemløsning. Det er mange problemer som kan løses ved hjelp av en datamaskin, men fremdeles er de mentale modellene viktigst. Problemløsning begynner alltid i din hjerne.

Oppsummering av kapittel 8

Modellering er en viktig del av problemløsning. Mentale modeller er viktigst. Likevel er det flere typer av modeller som er nyttige når du arbeider med ulike problemer eller deler av problemer. Et komplekst problem kan kreve at du bruker mange typer av modeller.

Mange modeller kan bygges inn i en datamaskin. Datamaskinmodeller utnytter datamaskinens styrke:

1. Lagringskapasitet
2. Prosesseringskraft

En godt laget datamaskinmodell er lett å endre. Dette hjelper når vi arbeider med vanskelige problemer.

Aktiviteter i kapittel 8

Aktivitet 8.1 – Noe jeg tenkte på i kapittel 8

Var det noe spesielt du begynte å tenke på da du leste kapittelet? Kanskje du tenkte på hvordan hjernen er god til å lage mentale modeller? Kanskje du innså at du ikke hadde tenkt så mye på modeller før?

1. Skriv tittel: "Noe jeg tenkte på i kapittel 8"
2. Skriv og forklar hva du tenkte.
3. Forklar hvorfor dette interesserer deg.
4. Hvorfor tenkte du på det?
5. Hva har det med livet utenfor skolen å gjøre?
6. Er det noe mer du tenker på?

Aktivitet 8.2 – Del ideen

Finn en elev å snakke med. Begge velger en idé i kapittelet som hver synes er viktig. Bruk ett minutt hver til å fortelle den andre om ideen. Tips til hva dere kan snakke om:

- Hvorfor er ideen viktig for deg?
- Hvilken betydning har den for ditt liv?

Når dere snakker sammen bør dere bruke begreper fra kapittelet:

- Datamaskinmodell
- Matematisk modell
- Mental modell
- Skalamodell
- Verbal modell
- Skriftlig modell
- Kroppsspråk

Aktivitet 8.3 – Lag en quiz som tester faktakunnskap fra kapittelet

Lag 2 spørsmål. Du skal teste faktakunnskap fra dette kapittelet.

Det skal være tekstsvar. Bare ett riktig svar til hvert spørsmål.

Spørsmålene skal være så enkle at en som ikke har lest kapittelet kan svare.

Prøv quizen på noen elever.

Aktivitet 8.4 – Lag en quiz som tester forståelse av kapittelet

Lag 2 spørsmål. Du skal teste forståelse av kapittelet.

Presenter noe fakta fra kapittelet og still spørsmål som krever tenkning og vurdering av fakta.

Testpersonene kan slå opp i en bok, gjøre et søk eller lese i dette kapittelet for å lage et svar.

En quiz som tester forståelse, har alltid flere korrekte svar. Du lager ingen svaralternativer.

Spørsmålene skal være så vanskelige at de bare kan besvares av en person som har en god forståelse av innholdet i dette kapittelet.

Prøv quizen på noen elever. Det skal være tekstsvar, og ingen fasit. Diskuter svarene sammen til slutt.

Aktivitet 8.5 – Flere betydninger av et ord

Hva tenker du på når du hører ordet "bane"? Tenker du på T-bane?

1. Prøv å finne flere betydninger av ordet "bane". Skriv opp.

2. Når noen bruker ordet "bane", hvordan kan du vite hvilken betydning av ordet de mener? Skriv og forklar.
3. Tror du en datamaskin kan gjøre dette? Begrunn svaret.

Aktivitet 8.6 – Lag en mental modell

Anta at du har et deilig eple i hånden. Eplet er fast, saftig og akkurat passe modent. Tenk deg hvor godt det ville være å ta en bit av det.

1. Lag en mental modell i hodet ditt av dette.
2. Hvilke farge tenkte du på?
3. Hvor stort er eplet du tenker på?
4. Tenkte du på andre fruktslag?
5. Forestill deg selv flere mentale modeller for epler generelt. Eksempel: Epler på 130 – 150 gram er en modell av vekten til vanlige epler. Skriv opp noen enkle eplemodeller.

Aktivitet 8.7 – Muligheter med datamaskinmodeller

Anta at en datamaskinmodell kan utvikles for en datamaskin. Dette gjøres ved å lage et nytt dataprogram. Betyr dette at datamaskinen kan løse problemet? Gi eksempler og argumenter for og mot.

Aktivitet 8.8 – Ulik notasjon for tall

Vi bruker sifrene 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Romerne brukte I, II, III, IV, V, og så videre. Sammenlikn disse to symbolsystemene i forhold til å løse matematiske problemer. For eksempel: Hvilket system er lettest å lære? Hvilket system egner seg best til enkel addisjon? Hvilket system er best for å utføre lang divisjon(f eks $455 : 15 =$)?

Aktivitet 8.9 – Budsjett som modell

På hvilken måte er et budsjett en modell? Gi eksempel på et problem du har løst ved å lage deg et budsjett.

Aktivitet 8.10 – Kart og landområder

En person sier: "Et kart er ikke et landområde".

Tenk på kartet som en modell. En globus er en modell av verden. Forklar noen mulige meninger av uttalelsen til personen.

Kapittel 9 – Vanlige digitale verktøy

Fokus

Læreplanen for grunnskole og videregående opplæring snakker om fem grunnleggende ferdigheter: Uttrykke seg muntlig, uttrykke seg skriftlig, kunne lese, kunne regne og kunne bruke digitale verktøy. Lesing, skriving og regning er verktøy som hjelper deg å løse mange ulike problemer.

Det finnes også digitale verktøy som er nyttige for å løse et stort antall problemer. De mest vanlige er:

- Tekstbehandler
- Databaser

- Grafiske programmer
- Regneark
- Nettverkstjenester

Introduksjon

På en måte er datamaskinen sterkt knyttet til lesing, skriving og regning. Den er til hjelp for hjernen. Den er til generell hjelp i problemløsning.

Du vet at en datamaskin ikke er noen god erstatning for tenkning. Likevel vil den effektive tenkeren vite hvordan en datamaskin kan være til stor hjelp. En smart person vet hvordan ulike digitale verktøy kan brukes i forskjellige situasjoner. En smart person vet hva slags verktøy som passer til jobben. En smart person vet også når digitale verktøy ikke er det beste hjelpemiddelet, og bruker da andre teknikker vi har snakket om i denne boken.

Noen digitale verktøy er utvidelser eller videreutvikling av verktøy som eksisterte før datamaskiner kom i bruk. En **tekstbehandler** er et slikt verktøy. Du kan tenke på en tekstbehandler som en elektronisk skrivemaskin med et minne. På mange måter er det et mer brukervennlig og kraftig verktøy enn papir og blyant. På andre måter er det mindre brukervennlig og mindre nyttig.

Tall, aritmetikk og en håndholdt kalkulator er nyttig i mange problemsituasjoner. Disse verktøyene hjelper deg å utvikle matematikkmodeller når du løser problemer, og utfører aritmetikken i disse modellene. Et **regneark** er et slikt verktøy. Regneark ble først utviklet for å hjelpe til med å løse problemer for næringslivet. Nå har det mange andre bruksområder.

Du vet at det tar tid og krefter å lære å lese, regne og skrive. Men det er verdt det. Du har sikkert funnet ut at mange vanlige digitale verktøy er lette å lære og lette å bruke. Du har også sett at noen er vanskeligere å lære, og krever mye tid foran datamaskinen. Men siden du skal løse så mange problemer med hjelp av digital verktøy, er det verdt å bruke både tid og krefter på å lære seg verktøyene.

Det er lett å lære tekstbehandling. En grunn er at **grensesnittet** mellom deg og programmet er lett å forstå. Hvis du kan skrive på et tastatur, vet du det meste du trenger for å bruke en tekstbehandler. En tekstbehandler er laget for å være **brukervennlig**.

Skriv: Fordeler og ulemper med en tekstbehandler.

På hvilke måter er en tekstbehandler bedre enn papir og blyant? Skriv.

På hvilke måter er papir og blyant bedre enn en tekstbehandler? Skriv.

Tenk på stavekontroll i tekstbehandleren. Sammenlikn dette med å rette på et papir.

Tenk på grammatikkontroll i tekstbehandleren. Vil dette hjelpe deg til å skrive godt?

--

Dette kapittelet er ikke skrevet for å lære deg hvordan du kan bruke vanlige digitale verktøy. Å lære seg å bruke digitale verktøy kalles verktøykompetanse. Skulle vi gi deg verktøykompetanse, måtte vi ta for oss alle tekstbehandlere som finnes i dag. Vi måtte ta for

oss alle regnearkprogrammer som finnes i dag. Vi måtte lage brukeropplæring for alle digitale verktøy som kommer hele tiden, år etter år. Det kan vi ikke.

Du finner hjelp til å lære verktøy i hjelpen som er laget for hvert program. Du finner masse annen hjelp på nettet. Du kan også prøve og feile. Men du må forstå konseptene. Det er viktigst.

Du kan allerede bruke mange verktøy. Er verktøyene laget etter generelle retningslinjer for dataprogrammer, kan du overføre det du allerede kan til nye verktøy du møter.

Du lærer deg selv å bruke et verktøy dersom du bestemmer deg for at det er verdt innsatsen.

Skriv: Nytte av vanlige dataprogrammer

1. Lag en liste over generelle dataprogrammer eller deler av dataprogrammer som du vet hvordan du kan bruke til å løse ulike problemer. Eksempler kan være tekstbehandler, stavekontroll, fletting av brev, regneark, diagramverktøy, formelverktøy. For hvert program eller del av program, skal du angi hvilken problemtype det er til hjelp for.
2. Lag en liste over noen generelle dataprogrammer eller deler av programmer du har hørt om, men som du ikke vet hvordan du skal bruke. For hvert eksempel skriver du hvilken problemtype du tror det kan være til hjelp for. Kanskje du skal skrive ett brev og sende det til mange personer, men ikke vet hvordan du kan flette inn navn og adresser på klistrelapper. Å bruke datamaskinen til å flette inn en lang adresseliste på klistrelapper, kan være til god hjelp når du skal sende ut brev til for eksempel 50 personer.

--

Nytte av vanlige dataprogrammer

Noen programmer er spesiallaget for spesielle problemer. En apotekansatt må kunne bruke spesielle programmer for apotekets medisiner. En som jobber med stålkonstruksjoner må kunne bruke spesielle programmer for å styre maskiner som kan skjære i stål.

Dette kapittelet snakker ikke om slike programmer, men om vanlige dataprogrammer som kan brukes i mange fag. Dette er programmer som har mange generelle bruksområder:

- A. Programmene kan brukes i alle fag på skolen.
- B. Programmene kan brukes til å arbeide med problemer hvor vi stadig må gjøre endringer. Programmene hjelper oss med å prøve forskjellige ideer og se resultatet av dem.
- C. Programmene utnytter kraften i en datamaskin og kan lette arbeidet for deg (For eksempel fletting av adresser). Programmene utnytter lagringsmuligheten for data.
- D. Programmene er lett å lære å bruke. De er brukervennlige.

Husk at dataprogrammer bare er en hjelp til å løse problemer. Du må fortsatt gjøre tenkearbeidet i ditt eget hode.

Et tilbakeblikk på hovedideene ved datamaskinmodellen

I kapittel 8 snakket vi om modeller. Det er viktig at du forstår modellkonseptet og spesielt hva en datamaskinmodell kan være. Verktøyene vi ser på i dette kapittelet er generell hjelp for å lage datamaskinmodeller.

Å lage en datamaskinmodell er viktig i problemløsning fordi:

1. En datamaskin kan brukes til å lage modeller for mange typer av problemer.
2. Datamaskinmodeller er lette å endre. Datamaskinen kan gjøre mye av arbeidet som endringene krever.
3. En datamaskin kan automatisk følge et sett av instruksjoner(prosedyrer).

Tekstbehandleren

Hva slags problemtyper kan tekstbehandleren brukes til?

1. **Organisere tankene.** Du kan lage et utkast. Du kan lett flytte rekkefølgen av ideene. Du kan lett endre ideer, fjerne ideer og sette inn nye ideer. Du kan utvide et utkast til et komplett dokument, en innlevering, en rapport, en besvarelse.
2. **Gjennomsyn og revidering.** Gode forfattere vet at nøkkelen til en god tekst er: "Revidere, revidere, revidere". Tekstbehandleren er et kraftig verktøy for å revidere tekst. Det kan være små endringer, som å omformulere en setning. Det kan være store endringer, som å skrive om hele avsnitt eller kapitler.
3. **Mekaniske operasjoner.** Det er mange mekaniske sider ved skriving, slik som staving, grammatikk, lesbarhet, fonter, overskrifter, innholdsfortegnelser og så videre. Tekstbehandlere kan telle antall ord og antall sider. Tekstbehandlere kan rette skrivefeil, foreslå synonymer og oversette tekst fra et språk til et annet.
4. **Publisering.** Et dokument kan sendes til en blogg, en nettside, en e-postmottaker, eller skrives ut på en skriver eller skrives ut til en pdf - fil. Dokumentet kan lastes opp på en server og bli tilgjengelig som en del av en nettside. Et dokument kan også lagres som et html-dokument og vises som en nettside. Et dokument kan også sendes til trykking, hvis du for eksempel vil lage en skoleavis, en brosjyre, en rapport eller en annen trykksak.
5. **Gjenbruke tekst.** Du kan ta fram et dokument du har laget for en stund siden, og bruke det på nytt som det er, eller med de endringene du lager. Du kan lagre det som et nytt dokument med et annet navn, og på den måten ha flere ulike versjoner som du sammenlikner. Du kan laste opp et dokument og la tekstbehandleren vise deg endringene i forhold til en annen versjon av det samme dokumentet. Du kan klippe ut og lime inn tekst fra ett dokument til et annet. Det er lett å bygge på tidligere arbeid.

Skriv: Emner i fag hvor tekstbehandleren er nyttig

Tenk på fag og emner du studerer på skolen i fagene. For hvert emne skal du tenke på situasjoner hvor tekstbehandleren er nyttig. Lag en liste over fagemner hvor du bruker tekstbehandleren. Ranger listen fra mest nyttig til minst nyttig.

Bruker du tekstbehandleren i alle emner i alle fag? Hvis svaret er nei, diskuter hvorfor. Skriv.

--

Database

De færreste lager seg databaser for å bruke i sitt daglige skolearbeid. Det er synd. Databaser kan nemlig være veldig nyttige for å lære fagene.

En database er en samling av data i tabeller. Tabellene kobles sammen, slik at du kan sette sammen data slik du ønsker.

Kanskje du ønsker å vite om en bok er utlånt eller ikke? Alle bøker på biblioteket ligger registrert i en database. Når du søker etter boken, vet databasen om boken er utlånt eller ikke. Den vet også hvor mange eksemplarer biblioteket har. Den vet hvem som er forfatter. Men forfatteren har skrevet mange bøker, så hvis du søker på forfatter, får du opp alle bøkene som biblioteket har fra denne forfatteren. Alle disse søkemulighetene og den informasjonen du får, er mulig fordi slike data ligger registrert i ulike tabeller i en database på biblioteket. Tabellene kobles sammen og gir deg mange muligheter for å sammenstille data på ulike måter.

Kanskje du ønsker å vite hvilke filmer i samlingen din, som har minst terningkast 5? Du har 200 filmer, og vil gjerne se en film som er god. Med en enkel database kan du lett finne det ut.

Eller kanskje du vil vite hvilke filmer i samlingen din hvor Clint Eastwood spiller hovedrolle? Hvis du kan lage en database, kan du spørre om slike ting, og få svar umiddelbart.

Men hvordan er det nyttig for å lære fagene?

Kanskje du arbeider med brutto nasjonalprodukt(bnp) i samfunnsfag. I en enkel database kan du legge inn land, bnp, folketall og andre opplysninger om landene. Nå kan du lage spøringer som: "Hvilke land har større bnp enn hele Afrika?" Da lærer du også noe om fattige land i forhold til rike land, nettopp fordi du har slike rådata i din egen lille database.

Hva er egenskapene til en database?

1. **Deling av oppdaterte data.** Mange databaser oppdateres veldig ofte. Tenk for eksempel på når du søker på Internett. Du leter etter et godt brukt kamera. Du finner mange på et nettsted som formidler brukte gjenstander. Men samtidig sitter en annen person og gjør akkurat det samme. Du blar litt i søkeresultatet, men bestemmer deg for å gå tilbake tilbake til det første du så på. Kameraet er solgt! Databasen er oppdatert.
2. **Fordelen av tilgang.** Mange databaser er søkbare fra en hvilken som helst datamaskin med nettilgang. Det betyr at informasjonen kommer til deg, i stedet for at du trenger å gå til den. Du slipper å besøke biblioteker, butikker, postkontor, banken, og så videre. Det meste av informasjon du henter, ligger i databaser.
3. **Rask tilgang.** Databaser kan søkes lynraskt. Det tar bare millisekunder å søke gjennom enorme mengder med data og få et søkeresultat som passer til dine søkeord. Tenk bare hvor fort du får opp kanskje 1000 forslag til nettsider som inneholder ditt søkeord etter et søk med for eksempel Google. Samtidig er som regel de mest nyttige treffene de første sidene som listes opp!
4. **Omorganisering av data.** Data i databasetabeller kan omorganiseres hurtig. Med et klikk kan du sortere data stigende eller synkende. Hvis du vil lage en liste over alle

filmene dine, kan du bestemme om du vil se bare tittel og årstall, eller om du vil ha med for eksempel regissør og hovedroller også. Eller kanskje du heller vil se en oversikt over alle filmer som Steven Spielberg har laget. Du bestemmer hvordan data skal organiseres. Men dataene forblir uendret.

5. **Økonomisk.** I mange tilfeller er det billigere å bruke en database enn å få tak i den samme informasjonen på annen måte. Anta at du lager en oppgave på skolen, og du trenger å lese en bestemt artikkel. Denne artikkelen finner du ikke på skolebiblioteket, og heller ikke på biblioteket på det stedet du bor. Hvis artikkelen ligger i en **fulltekstdatabase**, kan du raskt få tilgang til den over Internett. Det er billigere enn om skolen skulle abonnere på alle tidsskrifter elever måtte trenge.

Skriv: Nytte av databaser i fagene

Tenk på alle fagene du har på skolen. Tenk på hvor nyttig en database kunne være i hvert fag. Sorter fagene etter nytteverdi. Det første faget skal være det faget hvor en database over fagressurser ville være mest nyttig.

Hvis databaser ikke benyttes i dine fag, prøv å tenke på hvorfor det er slik? Hvordan kunne databaser endre noen av dine fag?

--

Typer av databaser

Det finnes mange typer av databaser. Vi skal nevne tre.

1. **Filsystemer.** Data lagres som regel i én stor tabell. En rad i en tabell kalles en *post*. Hver post inneholder noen data, som for eksempel navn, adresse, telefonnummer for en person. Du kan tenke på et regneark som en databasefil. Slike databaser har store begrensninger, og brukes lite. Det kalles flate filer.
2. **Relasjonsdatabaser.** Steven Spielberg har laget mange filmer. Hvordan ville det se ut i et regneark? Hvis vi har kolonner for regissør, fødselsdato, land og film, kunne vi gjenta Steven Spielberg over 20 ganger og skrive hans navn, fødselsdato og land hver gang vi skulle registrere en ny filmtittel. Eller vi kunne ha én rad og lage en ny kolonne for hver film, slik at vi fikk én rad med regissør, fødselsdato, land, og deretter laget kolonner for film1, film2, film3, film4, film5 og så videre. Men det ville passe dårlig for Mel Gibson og de 5 filmene han har laget som regissør. Han ville få 15 tomme kolonner. Dessuten har Mel Gibson spilt roller i nesten 50 filmer, og hvordan skulle vi registrere det? Svaret er relasjonsdatabaser. Med relasjonsdatabaser lager vi mange tabeller, og kobler dem sammen. Vi kan lage en tabell vi kaller *person*. Denne tabellen har kolonner personID, navn, fødselsdato, land. Vi registrerer Steven Spielberg(personID = 1) og Mel Gibson(personID = 2) og har da to poster. Så lager vi en tabell vi kaller *film*. Denne har kolonner filmID, filmtittel, årstall, og personID. Vi registrerer Spielbergs 21 filmer og Gibsons 5 filmer som regissør(vi glemmer de andre nå). I stedet for å gjenta Steven Spielberg og hans persondata 21 ganger, skriver vi bare tallet 1 i kolonnen personID. Så registrerer vi Mel Gibsons 5 filmer som regissør, og skriver tallet 2 i kolonnen personID. Nå kan vi lett koble dataene slik at vi kan søke opp Mel Gibson og se hvilke filmer han har regissert. Dette er relasjonsdatabaser. Det er altså relasjoner, eller koblinger, mellom tabeller og data.
3. **Fulltekstdatabaser.** Kapasiteten til datamaskiner har blitt enorm. Det har blitt mulig å lagre hele bøker med bilder og andre illustrasjoner. Mange skoler har tilgang til store

leksika fra en datamaskin. Mange relasjonsdatabaser inneholder komplette tekster av bøker, artikler og annet tekstmateriale. En av de viktigste ideene i problemløsning, er å bygge på andres arbeid. Men har ofte vært vanskelig å få tilgang til det. Nå har det blitt enklere med fulltekstdatabaser.

Grafiske programmer

Du har hørt uttrykket: "Et bilde sier mer enn tusen ord."

Det har lenge vært kjent at et bilde eller en film kan være langt mer effektivt enn ord og tekst for å beskrive en scene. Et bilde er ofte en veldig god modell.

Datamaskiner er spesielt nyttige for å lagre bilder, tegninger og diagrammer. Grafiske dataprogrammer har alle egenskapene i punktene A – D nevnt under avsnittet "Nytte av vanlige programmer".

Her har du noen generelle typer av grafikkprogrammer:

1. **Matematikkgrafikk.** Et eksempel er å lage diagrammer av data ved å bruke et sektor-, stolpe- eller linjediagram. Et annet eksempel er å lage grafer av funksjoner og ligninger.
2. **Ingeniørgrafikk.** Dette dekker et stort område som arkitekttegninger av bygninger eller tegninger av komponenter som lages i et verksted eller i industrien.
3. **Grafisk design.** Dette handler om å lage alt fra et pent bordkort eller elegant webdesign, til en profesjonell video.

Grafiske dataprogrammer har åpnet opp en helt ny verden for problemløsning. Det er lett å bruke et bildebehandlingsprogram for å lage eller manipulere et bilde.

Grafiske programmer har følgende egenskaper:

1. **Lett å endre.** Tenk deg at du lager forside til en innlevering. Forsiden skal ha et bilde, en tittel og noe tekst. Dette er lett å sette opp med en datamaskin, men vanskelig å gjøre for hånd. På datamaskinen er det lett å rette feil, flytte på bildet eller endre størrelsen på tekst og bilde. Du kan se på det du har laget, og vurdere om du liker det eller ikke. Hvis du ikke liker det, kan du lett endre det.
2. **Grafisk fremstilling av data.** Hjernen er ikke så god til å tolke store mengder data i tabeller. Tenk på en usortert tabell som har tall for befolkningstetthet i alle fylker i Norge. Det er vanskelig å se mønstre og avgjøre hvilke deler av landet som har stor eller liten befolkningstetthet. Men en datamaskin kan lett vise slike data som et kart over landet. Den kan bruke ulike farger som representerer forskjellig befolkningstetthet. Da kan du med et lite blick på kartet fort avgjøre hvilke deler av landet som har høy og lav befolkningstetthet.
3. **Farge og bevegelse.** Datamaskinen har tatt over noen av egenskapene ved filmproduksjon. Du kan bruke datamaskinen til å lage animasjoner. Videobilder og lydspor kan lagres på harddisken. Dette kan kombineres med dine egne tegninger og tekst du lager.

Skriv: Nytte av grafiske programmer i fagene

Tenk på fagene du har på skolen. For hvert fag, skal du tenke på hvordan grafiske programmer kan være nyttig. Sett opp en liste over fagene, og start med det faget hvor grafiske programmer er mest nyttig.

Hvis grafiske programmer ikke brukes i dine fag, diskuter hvorfor. Hvordan kunne dine fag bli bedre dersom grafiske programmer ble brukt?

--

Regneark

Et regneark er egentlig et dataprogram for bokføring eller regnskap. Du kan skrive ord og tall i celler, og bruke ulike formler for å regne med tallene. På den måten kan du lett lage deg et budsjett, eller føre opp utgifter og inntekter for å se økonomiske resultater.

Etter hvert har regneark utviklet seg, og kan i dag brukes til mange andre problemtyper enn økonomi og regnskap.

Regneark har alle egenskapene i punktene A – D nevnt under avsnittet "Nytte av vanlige programmer".

Se på eksempelet nedenfor. En lærer har laget seg et ark for å beregne standpunkt karakter for noen elever.

Elev	Prøve 15.09	Prøve 15.10	Innlevering 25.10	Prøve 15.11	Snitt	Standpkt
Ellingsen, Sandra	4	3	5	4	4	4
Gashi, Bessart	5	5	3	4	4,25	4
Knutsen, Erik	4	4	4	4	4	4
Pettersen, Line	4	4	4	3	3,75	4
Ås, Henrik	5	4	4	4	4,25	4
Snitt for gruppa					4,05	4

Anta at læreren oppdager at hun har satt feil karakter på en prøve for en av elevene. Siden hun bruker formler for å beregne gjennomsnitt og standpunkt, vil disse tallene oppdatere seg automatisk når karakteren endres for eleven. Datamaskinen gjør alt arbeidet, men læreren må vite hvordan hun skal bruke formler.

Et regneark er nyttig når man arbeider med en tabell med tall og bruker formler for å beregne for eksempel gjennomsnitt, eller runde av tall til nærmeste hele tall. Dette er vanlig i kontorarbeid, men også i naturfag, matematikk og andre fag. Regneark er derfor et generelt hjelpeverktøy for mange problemtyper.

Det er lett å lære enkle beregninger, men regneark er i dag avanserte verktøy som krever mye trening og opplæring for å utnytte det fullt ut.

Skriv: Nytte av regneark i fagene.

Tenk på fagene du har på skolen. For hvert fag, skal du tenke på hvordan regneark kan være nyttig. Sett opp en liste over fagene, og start med det faget hvor regneark er mest nyttig.

Hvis regneark ikke brukes i dine fag, diskuter hvorfor. Hvordan kunne dine fag bli bedre dersom regneark ble brukt?

--

Nettverkstjenester

Dette er tjenester som muliggjør sammenkobling av datamaskiner. Dette kan gjøres ved hjelp av kabler eller trådløs tilgang. Dette kalles datamaskinnettverk.

Med nettverkstjenester kan mennesker kommunisere med hverandre eller med informasjon som finnes på andre maskiner. Det er tre hovedtyper av slik bruk:

1. **Kommunikasjon mellom mennesker.** Det kan skje på flere måter, som for eksempel e-post, sosiale nettverk, IP- telefoni, meldinger, ulike fora og pratetjenester. I mange tilfeller er kommunikasjonen **asynkron**. Det vil si at du skriver en melding eller et innlegg, og svar kommer når det passer mottakere. I andre tilfeller er kommunikasjonen **synkron**. Det vil si at du får tilbakemelding umiddelbart og du kommuniserer med andre personer i sanntid.
2. **Kommunikasjon med databaser.** Dette er nevnt tidligere i dette kapitlet.
3. **Deling av datamaskinressurser.** Datamaskiner kan dele filer med hverandre gjennom nettverk. Dette er et område som utvikler seg hele tiden, og gir alle nettverksbrukere tilgang til en rekke ressurser, både lovlig og ulovlig. Vi kan også få tilgang til skrivere som er plassert på ulike steder, og vi kan også få tilgang til å fjernstyre en annen datamaskin over et nettverk. På denne måten kan vi logge oss inn på en maskin vi har stående hjemme, når vi er et annet sted. Når vi er koblet opp i et skolenettverk, vil vi ofte ha tilgang til felles sentrale maskiner og tjenester de kjører. Dette kan være filtjenere, læringsplattformer og felles portalløsninger. En sentralt plassert server kan utføre store beregninger for mange brukere.

Skriv: Nytte av nettverkstjenester i fagene.

Tenk på fagene du har på skolen. For hvert fag, skal du tenke på hvordan nettverkstjenester kan være nyttig. Sett opp en liste over fagene, og start med det faget hvor nettverkstjenester er mest nyttig.

Hvis nettverkstjenester ikke brukes i dine fag, diskuter hvorfor. Hvordan kunne dine fag bli bedre dersom nettverkstjenester ble brukt?

--

Oppsummering av kapittel 9

Vanlige digitale verktøy er laget for å hjelpe oss å løse et stort omfang av problemer. Slike dataprogrammer kan brukes til å lage datamaskinmodeller for disse problemene.

Noen nøkkelfunksjoner for slike programmer er:

- A. Programmene kan brukes i alle fag på skolen.
- B. Programmene kan brukes til å arbeide med problemer hvor vi stadig må gjøre endringer.
- C. Programmene utnytter kraften og lagringskapasiteten i en datamaskin.

- D. Programmene er lett å lære å bruke.

Aktiviteter i kapittel 9

Aktivitet 9.1 – Noe jeg tenkte på i kapittel 9

Var det noe spesielt du begynte å tenke på da du leste kapittelet? Kanskje du tenkte på hvordan et bestemt dataprogram hjalp deg å løse et problem? Kanskje du tenkte på noen type dataprogrammer som ikke var nevnt i kapittelet? Kanskje du tenkte på at det kunne være en idé og bruke et spesielt dataprogram innenfor et fag?

1. Skriv tittel: "Noe jeg tenkte på i kapittel 9"
2. Skriv og forklar hva du tenkte.
3. Forklar hvorfor dette interesserer deg.
4. Hvorfor tenkte du på det?
5. Hva har det med livet utenfor skolen å gjøre?
6. Er det noe mer du tenker på?

Aktivitet 9.2 – Del ideen

Finn en elev å snakke med. Begge velger en idé i kapittelet som hver synes er viktig. Bruk ett minutt hver til å fortelle den andre om ideen. Tips til hva dere kan snakke om:

- Hvorfor er ideen viktig for deg?
- Hvilken betydning har den for ditt liv?

Når dere snakker sammen bør dere bruke begreper fra kapittelet:

- Nettverkstjenester
- Database
- Tekstbehandler
- Grafiske programmer
- Regneark
- Brukervennlige programmer
- Grensesnitt

Aktivitet 9.3 – Lag en quiz som tester faktakunnskap fra kapittelet

Lag 2 spørsmål. Du skal teste faktakunnskap fra dette kapittelet. Det skal være tekstsvar. Bare ett riktig svar til hvert spørsmål.

Spørsmålene skal være så enkle at en som ikke har lest kapittelet kan svare. Prøv quizen på noen elever.

Aktivitet 9.4 – Lag en quiz som tester forståelse av kapittelet

Lag 2 spørsmål. Du skal teste forståelse av kapittelet.

Presenter noe fakta fra kapittelet og still spørsmål som krever tenkning og vurdering av fakta.

Testpersonene kan slå opp i en bok, gjøre et søk eller lese i dette kapittelet for å lage et svar.

En quiz som tester forståelse, har alltid flere korrekte svar. Du lager ingen svaralternativer. Spørsmålene skal være så vanskelige at de bare kan besvares av en person som har en god forståelse av innholdet i dette kapittelet.

Prøv quizen på noen elever. Det skal være tekstsvar, og ingen fasit. Diskuter svarene sammen til slutt.

Aktivitet 9.5 – Bruk av tekstbehandleren til regneoppgaver

Tenk på å bruke tekstbehandleren for å utføre vanskelig multiplikasjon eller divisjon. Tenk for eksempel på 973×729 eller $6883073 / 729$. Prøv å utføre dette med en tekstbehandler. Skriv en kort konklusjon om hva du mener.

Aktivitet 9.6 – Lære et nytt program

Velg et vanlig digitalt verktøy du ikke har brukt før. Kanskje du ikke har laget en database før? Lær deg å bruke det til en enkel oppgave. Mens du arbeider med å lære programmet, skal du tenke på hvordan du lærer. Hva gjør du for å lære det? Skriv detaljerte notater om hvordan du lærer.

Aktivitet 9.7 – Sammenlikning av grunnleggende ferdigheter

Skolen skal lære oss de grunnleggende ferdighetene:

1. Uttrykke seg muntlig.
2. Uttrykke seg skriftlig.
3. Lese.
4. Regne.

De vanlige digitale verktøyene er:

1. Tekstbehandler
2. Databaser
3. Grafiske programmer
4. Regneark
5. Nettverkstjenester

Sammenlikne disse to listene. På hvilken måte er de like? På hvilken måte er de forskjellige? Bruk denne sammenlikningen til å forutsi en ny type dataprogram som kan utvikles i fremtiden.

Aktivitet 9.8 – Talebehandler

Noen snakker om et mulig dataprogram som kan skrive tekst du taler inn med en mikrofon. Det er en skrivemaskin du kan snakke til. Du slipper å bruke tastaturet. Anta at yngre skoleelever hadde et slikt hjelpemiddel. Hvordan tror du det ville endre læring og trening av skriveferdighetene?

Kapittel 10 – Datamaskinsystemer

Fokus

En datamaskin består av maskinvare og programvare. Maskinvare er fysisk maskineri og harde komponenter. Programvare er dataprogrammer, såkalte myke komponenter.

Dette forandrer seg hurtig. Hvert år kommer det nye maskiner som er litt bedre. Både maskinvare og programvare blir bedre.

For å bruke et dataprogram, må du kunne kommunisere med maskinen. Du må fortelle den hva den skal gjøre. Nyere maskiner er lettere å bruke enn gamle, fordi det stadig bli lettere å fortelle maskinen hva den skal gjøre. Det foregår en hurtig utvikling på dette området.

Introduksjon

Det er vanlig å tenke på datamaskinen som en "svart boks". Du slipper å tenke på hva som skjer inne i boksen. Du gir beskjeder, og maskinen utfører dem. Du trenger ikke vite hva som egentlig skjer der inne i boksen.

De fleste mennesker liker likevel å forstå litt av hva som skjer. Det er morsomt å forstå maskinens indre virkemåte. Det er også nyttig for å kunne bruke den bedre og smartere. I dette kapittelet vil vi øke din forståelse av hvordan maskinen egentlig fungerer.

Skriv: En svart boks

Er det som skjer under panseret på en bil en "svart boks" for deg? Hva med en elektrisk motor? En lyspære? Tenk på ideen "svart boks". Hvordan føles det å bruke en gjenstand du ikke forstår?

--

Hva kan en datamaskin gjøre?

En datamaskin er egentlig en "dum" maskin. Den kan bare utføre helt enkle ting. Men den utfører disse enkle tingene meget hurtig og nøyaktig. Her er de viktigste tingene den kan gjøre:

1. Datamaskinen kan arbeide med bokstaver, tall og symboler.
2. Datamaskinen kan arbeide med bilder og lyd.
 - a. Bilder er satt sammen av **pikslar** eller små prikker, som egentlig er små firkanter. Datamaskinen kan lagre tallsekvenser som forteller hvilken farge pikslene skal ha.
 - b. Lyd er bølger med ulike **frekvenser**. En lydbølge klippes opp i korte **segmenter**. Datamaskinen måler frekvensen av hvert segment og lagrer tall som representerer frekvensen i hvert segment. Den samme teknologien brukes for å lage en CD med musikk.
3. Datamaskinen kan gjøre enkle enkle regneoperasjoner som addisjon av to tall. Den kan også sammenlikne to tall og se hvilket som er størst.
 - a. Hver datamaskin har en **CPU** (Central Processing Unit), eller prosessor. CPU utfører enkel addisjon og sammenlikning av tall. CPU kan også sjekke om et tall er negativt.
 - b. Datamaskinen har et **minne**, RAM(Random Access Memory). Minnet kan lagre et dataprogram, som består av en rekke detaljerte instruksjoner. Et dataprogram forteller CPU hvordan CPU skal løse en bestemt oppgave eller et problem. Hovedideen er at minnet kan lagre både dataene du mater inn, og programmet som skal gjøre beregningene med dataene dine.

4. Datamaskinen kan utføre stegene i et dataprogram ekstremt hurtig og veldig nøyaktig. En moderne datamaskin kan utføre millioner av slike steg på ett sekund. Den kan fungere i mange dager uten å gjøre en eneste feil. Den blir aldri trøtt og ukonsentrert.

Litt historikk om maskinen

For omtrent 60 år siden, endret datamaskinen seg langsomt. Den første elektroniske datamaskinen var ferdig i 1945. Etterpå har utviklingen gått raskt. En av grunnene til at vi snakker om datamaskinens historie, er at dagens hurtige endringer får stor betydning for deg.

Datamaskinen ble utviklet for å lage en avansert regnemaskin. **Abakus** ble laget for 5000 år siden, og kan forstås som en fjern slektning til dagens datamaskin. I mange land benytter man fremdeles Abakus som hjelpemiddel til regneproblemer. Abakus er det samme som kuleramme.

På 1800- tallet ble det laget mekaniske regnemaskiner som kunne følge et sett av instruksjoner. Det var egentlig de første maskinene som kan regnes som datamaskiner, fordi de kunne utføre instruksjoner.

I 1930 – årene utviklet ideen om en elektronisk datamaskin seg, noe som ledet til en elektronisk digital datamaskin på 40 – tallet. Disse maskinene var store som hus, og hadde bare en brøkdel av kapasiteten til dagens PC. De var bygget med **vakumrør**, og utviklet enorm varme.

På 50- tallet begynte for alvor utviklingen av en industri. Mange firmaer laget og solgte datamaskiner, og disse ble etter hvert vanlige i næringsliv, offentlig sektor og ved universiteter.

På grunn av problemene med vakumrør og varmeutvikling, ble **transistoren** utviklet. Den kunne gjøre det samme som et vakumrør, og brant seg ikke fast. Transistormaskiner kom på markedet på slutten av 50-tallet.

Nå ble det fart i utviklingen, og **integreerte kretser** ble utviklet. En integrert krets er et kort med mange transistorer og andre elektroniske deler. En integrert krets kalles noen ganger for en **chip**, fordi en liten klatt med silikon, eller "a chip of silicone" ble brukt i produksjonen av disse kortene. En chip er på størrelse med en negl.

I dag tilsvarer et integrert kretskort millioner av vakumrør. Dette har gjort det mulig å bygge inn slike kretskort i kalkulatorer, bærbare datamaskiner og mye annen elektronikk.

Datamaskiner har blitt så kraftige, små og rimelige, at millioner av mennesker har dem hjemme.

Grovt sagt har maskiner gjort et framskritt med en faktor på en million i løpet av de siste 40 årene. Det betyr for eksempel at kostnaden ved å produsere en bestemt mengde datakraft, har blitt redusert med en faktor på en million i løpet av denne tiden. Datamaskiner vil fortsette å utvikle seg hurtigere og hurtigere i årene som kommer.

Denne utviklingen vil få stor betydning for deg. Om 20 år vil folk se tilbake på datamaskinene vi bruker i dag og le av disse primitive klumpene vi drasser på i ryggsekker. Likevel har ikke de grunnleggende ideene om hvordan en datamaskin kan hjelpe oss å løse problemer, endret seg mye i denne tiden. Hvis du lærer deg hvordan du løser problemer ved hjelp av datamaskiner, vil denne kompetansen vare livet ut!

Skriv: Hurtige endringer

1. Lage en liste over ting som endrer seg hurtig. Listen kan inneholde for eksempel populære TV-programmer, medisiner, forbruksvarer, dataprogrammer.
2. For hvert element i lista skal du gjøre en kort vurdering av hvilken betydning disse hurtige endringene har for ditt liv.

--

Dataprogrammer

Et dataprogram er et sett av instruksjoner som forteller datamaskinen hva den skal gjøre.

Hvis oppgaven er ganske enkel, kan programmet være enkelt. Hvis oppgaven er kompleks, må også programmet være komplekst. Noen dataprogrammer inneholder millioner av instruksjoner, og mange av dem er laget for å fange opp ting som kan gå galt. Hvis noe går galt, må programmet vite hva det skal gjøre. Store team av utviklere kan arbeide i årevis for å lage store dataprogrammer som er robuste og sjelden går galt.

La oss ta et eksperiment. Ta en penn eller blyant i hånden. Tegn en sirkel. Tegne en mindre sirkel inne i den første sirkelen. Legg vekk pennen eller blyanten.

Du har akkurat utført noen "steg- for- steg" – instruksjoner. Hvis du ikke har en funksjonshemming som gjør det vanskelig for deg å tegne eller skrive, så var det lett for deg å utføre instruksjonene.

Men da du var liten, hadde du ikke klart å følge disse enkle instruksjonene. Det tok mange år å trene både kropp og sinn for å kunne ta i mot beskjeder, løfte opp en blyant og tegne først en sirkel, og deretter en mindre sirkel inne i den første. Dette er egentlig ganske kompliserte handlinger. Din kropp og din hjerne kan gjøre dette lett nå, fordi du har lært mye og gjort mange øvelser.

Skriv: Å spisse en blyant

Tenk på følgende situasjon i klasserommet:

Du plukker opp blyanten, går til blyantspisseren ved tavla, spisser blyanten og går tilbake til pulten din.

1. Skriv en detaljert liste over alle instruksjoner for situasjonen "å spisse en blyant" slik det er beskrevet ovenfor.
2. Undersøk hvert steg du har beskrevet for å sjekke hvor vanskelig det er å utføre. Sett et merke ved de to vanskeligste stegene.
3. Tok dine steg hensyn til at blyantspissen kunne brekke? Hvis ikke, legg til dette steget på riktig plass i sekvensen.

--

Programmeringsspråk

Et sett av instruksjoner forteller noen eller noe hva som skal gjøres. Et slikt sett av instruksjoner må ta hensyn til den som skal utføre instruksjonene. Hvis denne personen eller tingen er ganske smart, kan instruksjonene være enkle. Du kan for eksempel si:

”Gå til blyantspisseren ved tavla.”

Personen som gjør dette, vil unngå å tråkke på andre personer som sitter i klasserommet. Personen vil uten videre vite at han skal gå der hvor det ikke er pulter, vesker eller annet som står i veien.

En robot er en datamaskin med et kamera som øye og med mekaniske bein og armer. Roboten har et dataprogram som styrer den. Hvordan forteller du en robot at den skal gå til tavla og spise en blyant? Hvordan kan roboten vite om blyanten har blitt spiss? Hvordan instruerer du en robot til ikke å tråkke på andre elever i klasserommet? Dette er meget vanskelige problemer. Utviklere som lager styringsprogrammer for roboter, må slite for å finne løsninger på disse problemene.

En datamaskin er laget for å utføre et sett av instruksjoner. Du kan tenke på hver instruksjon som en primitiv, et veldig enkelt problem som en datamaskin kan løse. En datamaskin ”forstår” et *språk* som inneholder slike primitiver. Dette kalles **maskinspråk**. Et maskinspråk er et sett av instruksjoner som forteller maskinen hva den skal gjøre.

Datamaskiner er ikke like. Ulike merker og modeller har ulike innebygde primitiver. De har altså ulike maskinspråk. Et program som er skrevet for ett maskinspråk kan kanskje ikke brukes for et annet maskinspråk på en annen maskin. For eksempel kan det være programmer som er skrevet for Windows som ikke kan brukes med Linux.

Et annet problem er at stegene i et maskinspråk er for enkle i forhold til hva brukere av datamaskinen ønsker at maskinen skal kunne gjøre. Tenk på hvor mange forskjellige steg som kreves for å slå opp et ord i en ordbok. Et maskinspråkprogram for å gjøre dette, er en lang liste av instruksjoner. Den minste feil i disse instruksjonene vil føre til at programmet stopper opp eller avbrytes plutselig.

Få mennesker er gode til å lage et sett av instruksjoner som aldri vil medføre feil(**bugs**). Det er meget vanskelig å lage et feilfritt program. Det er også vanskelig å teste og oppdage mulige feil(**debug**). En programmerer må ha gode ferdigheter i å oppdage feil og rette feil.

Derfor konsentrerte tidlig forskere seg om å bruke mye tid på hvordan de kunne lære studenter å skrive dataprogrammer. De laget derfor programmeringsspråk som liknet mer på vårt eget språk. (Maskinspråk likner lite på vårt eget språk.) Ideen var å lage et programmeringsspråk som liknet engelsk, tysk, norsk og andre språk, og som igjen kunne oversette videre til maskinspråk som maskinen forsto. Altså et språk som lå mellom menneskespråk og maskinspråk. Dette kalles **høynivåspråk**.

Et av de første høynivåspråkene var Fortran. Det var ferdig i 1957 og skulle kunne brukes av forskere og ingeniører. Det er fremdeles i bruk. Årsaken til at det fremdeles benyttes, er at det ble investert billioner av dollar for å lage dataprogrammer med Fortran. Mange av disse programmene er fremdeles nyttige og kan brukes. Dette betyr at mennesker bygger på arbeidet til andre mennesker, slik vi har understreket betydningen av i problemløsning.

På 60 – tallet ble det utviklet et høynivåspråk kalt BASIC. Det har blitt brukt av studenter og mange andre mennesker.

Mange andre språk ble utviklet for å møte stadig nye behov hos forskjellige mennesker. COBOL ble laget for å skrive programmer for næringslivet. På slutten av 90-tallet ble Java et språk som "alle" måtte lære. Vi kan også nevne C++ og C# som ligger til grunn for mange av programmene vi bruker på en datamaskin i dag.

Semour Papert og flere utviklet allerede på 60-tallet høynivåspråket Logo, for å ha et språk som kunne brukes i skolen fra barnehage til høyskole. Det benyttes fremdeles i dag i mange skoler, helt opp til universitetsnivå. Logo er lett å lære for de fleste, og er fint å bruke for å forstå hvordan vi kan instruere en datamaskin til å utføre alt fra enkle oppgaver til kompliserte dataprogram.

Skriv: Oversetting av språk.

Et dataprogram kan oversette BASIC- eller Logo- programmer til maskinspråk. Kan en datamaskin oversette russisk til norsk? På hvilke måter er de to oversettingsoppgavene like? På hvilke måter er de forskjellige?

--

Eksempler på dataprogram

Vi skal vise deg noen eksempler på enkle dataprogram.

BASIC - program

Her skal du få et par eksempler på veldig enkle dataprogram skrevet i BASIC.

Det første instruerer en datamaskin til å spørre om navnet til den som bruker programmet. Datamaskinen skriver ordet "HEI" på skjermen, etter fulgt av navnet den har fått. Så ber den brukeren taste inn to tall. Datamaskinen regner ut summen av disse to tallene, og skriver svaret på skjermen.

Hver linje i et BASIC-program har et linjenummer. Vi starter med linje 10, og øker med 10 for hver nye linje. Altså: 10, 20, 30, 40 og så videre. Dette gjør det lett å sette inn nye linjer mellom hver hele tier.

Del 1

```
10 PRINT "HVA HETER DU?"  
20 INPUT N$  
30 PRINT "HEI!"; N$  
40 END
```

Del 2

```
10 PRINT "DETTE ER ET PROGRAM SOM KAN ADDERE TO TALL"
20 PRINT "SKRIV INN DET FØRSTE TALLET."
30 INPUT A
40 PRINT "SKRIV INN DET ANDRE TALLET."
50 INPUT B
60 LET C = A + B
70 PRINT "SUMMEN AV TALLENE ER:"; C
80 END
```

Dette var ikke så vanskelig. Du forsto sikkert denne koden.

Logo – program

Her skal du få et eksempel på et enkelt program skrevet i Logo. Det forteller datamaskinen hvordan den skal tegne et enkelt hus. Huset består av et kvadrat med en trekant på toppen. Dette er litt vanskeligere, men prøver du å forstå det, så klarer du det.

```
TO HOUSE
  SQUARE 50
  MOVE.TO.ROOF 50
  TRIANGLE 50
  MOVE.TO.START 50
END
```

```
TO SQUARE :SIZE
  REPEAT 4 [FORWARD :SIZE RIGHT 90]
END
```

```
TO TRIANGLE :SIZE
  REPEAT 3 [FORWARD :SIZE RIGHT 120]
END
```

```
TO MOVE.TO.ROOF :LENGTH
  FORWARD :LENGTH
  RIGHT 30
END
```

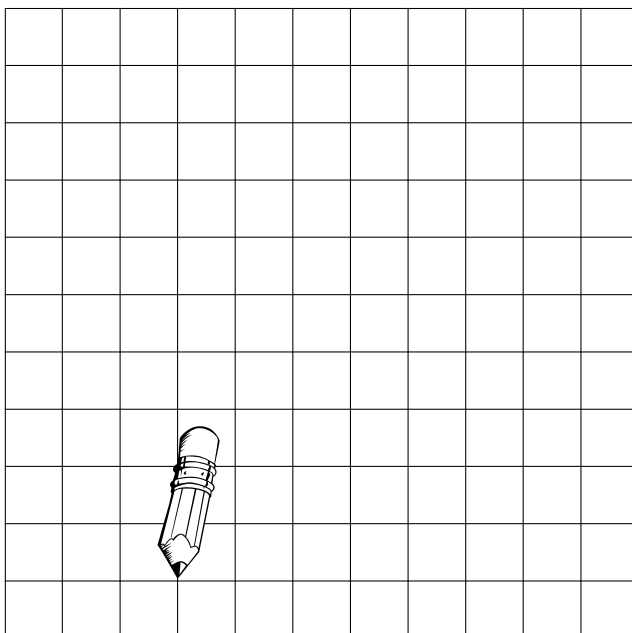
```
TO MOVE.TO.START :DISTANCE
  LEFT 30
  BACK :DISTANCE
END
```

Det fine med Logo er at studenten umiddelbart ser hva som skjer på skjermen når koden testes og kjøres. Skrivemerket(Cursor) beveger seg slik du har instruert, og tegner en linje dit merket flytter seg. I Logo har vi to kommandoer som heter PD og PU. PD betyr Pen Down, og PU betyr Pen Up. Hvis vi skriver PD, vil huset bli tegnet på skjermen, siden cursor beveger seg over skjermen slik vi har instruert. I koden ovenfor har vi ikke skrevet PD, men vi kunne

ha begynt med en linje som hadde instruksjonen PD og avsluttet med PU helt til slutt i programmet.

Du kan studere koden, og prøve å utføre instruksjonene på et ruteark med en blyant, en linjal og en gradskive. La hver rute tilsvare størrelsen 10. Rutene må være kvadratiske. TO HOUSE betyr at vi går til programmet HOUSE. ("Her er instruksjonene for hus") Det første du skal gjøre er å plassere blyantspissen i nederste venstre hjørne av en rute. Dette er startpunktet for huset, som også er nederste venstre hjørne av kvadratet som utgjør veggen. Blyanten står klar til å gå rett oppover arket(vertikalt).

TO HOUSE



Det første du skal gjøre i TO HOUSE er instruksjonen SQUARE 50.

Se på koden for TO SQUARE. Den sier at du skal lage et kvadrat (square) med størrelsen SIZE. SIZE = 50, for du kaller på SQUARE med tallet 50. Hva skjer i TO SQUARE?

Jo vi gjentar 4 ganger den samme handlingen: FORWARD 50(fram 5 ruter) og RIGHT 90 (snu retningen 90 grader til høyre). Altså, vi beveger oss fram 50, snur 90 på stedet, går fram 50 og snur 90, går fram 50 og snur 90, går fram 50 og snur 90. Det vi har gjort er å gå i omkretsen av kvadratet, og ender opp akkurat slik vi sto ved start. Vi har tegnet et kvadrat.

Hva må du gjøre for å tegne et større kvadrat?

Hvis du kaller på SQUARE med 60 kanskje? Det er derfor vi bruker SIZE. SIZE kan være et hvilket som helst tall, slik at du kan tegne akkurat så stort kvadrat du vil, bare ved å endre et tall. Prosedyren er den samme, men med et tall som kan variere. Vi kaller SIZE for **variabel**.

Det er variablene som gjør at vi kan eksperimentere. Variabler brukes i alle programmeringsspråk. Hadde vi ikke brukt variabler, måtte vi laget en ny prosedyre for hvert eneste kvadrat vi ville tegne i forskjellige størrelser. Det ville vært veldig slitsomt, og

tungvint. Hvis vi ikke bruker variabler, bruker vi **hardkoding**. Å hardkode regnes som dårlig programmering. Vi hardkoder ikke verdier som vi kan trenge å endre eller variere.

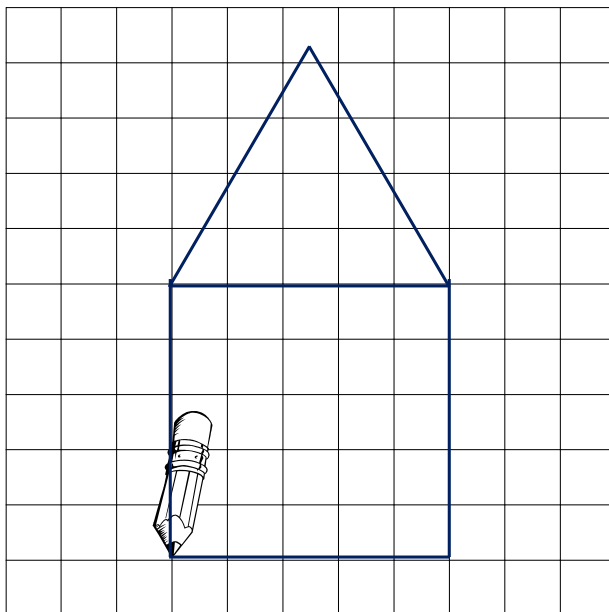
Nå har du sett nok til å kunne forstå resten av logo- koden. Du kan fortsette å tegne taket på huset, ved å bevege deg med blyanten slik resten av koden instruerer. MOVE.TO.ROOF flytter blyanten slik at du står klar til å tegne taket på rett sted.

TRIANGLE tegner opp taket.

MOVE.TO.START flytter deg tilbake til det opprinnelige utgangspunktet.

Når du er ferdig har du kanskje en tegning som denne:

TO HOUSE



Hvis du forstår og klarer dette logoprogrammet på papir, har du forstått mye om programmering med høynivåspråk. Men husk at Logo ble laget i forrige århundre, og dagens programmeringsspråk er mye nærmere vårt eget talespråk, og derfor kanskje enda lettere å bruke.

Likevel er Logo et språk som passer veldig bra for skoleelever, fordi vi kommer fort i gang og forstår snart hva en datamaskin kan gjøre og hva vi må "lære" den. Vi må lage alle primitivene, så kan maskinen gjøre jobben for oss.

Skriv: Programmere en sirkel med Logo

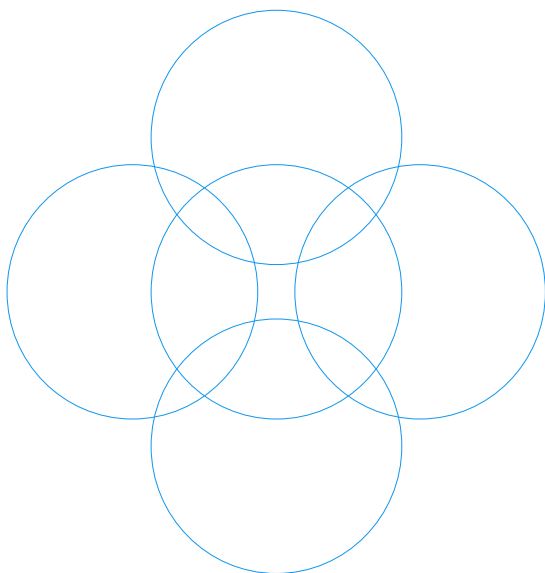
En nøkkelidé med Logo er at vi tenker som vi handler. Skal du tegne et hus, så tenker du: "Hvordan beveger jeg meg rundt et kvadrat? Hvordan beveger jeg meg rundt en trekant? Hvor mye snur jeg meg når vi skal skifte retning?" "Hvordan flytter jeg meg fra nederste venstre hjørne av veggen til øverste venstre hjørne, slik at jeg kan tegne taket?"

1. Bruk denne tankegangen for å finne ut hvordan du ville bevege deg for å trække deg rundt i en sirkel. Tips: Bruk veldig små skritt, og skift retning ofte. Prøv på gulvet selv. Prøv å gå i ring og tenk over hva du egentlig gjør. Bruk metakognisjon.
2. Lag et Logo- program som du kaller TO SIRKEL. **Tips:** Husk at du kan gjenta et mønster med `REPEAT antall ganger [det du skal gjenta]`. Vi brukte dette for å tegne kvadratet, `REPEAT 4[FORWARD :SIZE RIGHT 90]`. Det blir ikke så ulikt med sirkelen! Husk at en sirkel har det til felles med et kvadrat at du beveger deg rundt og kommer tilbake. Jo flere kanter du har, dess nærmere kommer du en sirkel. Er kantene små nok så trenger du bare skifte retning bitte lite grann hver gang! Mange kanter og små retningsendringer vil til slutt bli en sirkel, hvis du kan finne ut hvor mange kanter du trenger og hvor mange grader du må skifte retning hver gang.
3. Koden skal se slik ut:
4. TO SIRKEL
5. REPEAT *ditt tall* [FORWARD :*ditt tall* RIGHT *ditt tall*]
6. END

--

Når vi snakker om å eksperimentere med datamaskinen som verktøy, er Logo ideelt. Vi kan prøve ut ideer. La oss si at du skal dekorere noe. Kanskje du skal lage en fin bakgrunn på en forside til en festsang? Med Logo kan du bruke REPEAT til å gjenta et mønster så mange gange du vil, og med få kodelinjer kan du flytte skrivemerket litt hver gang. Har du først laget kode for en sirkel, et kvadrat og et rektangel, kan du gjenta disse formene i forskjellige størrelser og med forskjellige plasseringer. Du endrer bare noen variabler.

Du kan for eksempel tegne 5 sirkler som er litt forskjøvet i forhold til hverandre, både vertikalt og horisontalt. Du tegner lett en figur som kan minne om en blomst. Er du ikke fornøyd, kan de bare endre litt på variablene.



Hundrevis av høynivåspråk har blitt utviklet. Alle har noen ting felles:

1. Hvert språk er laget for å møte kravene fra ulike typer av programmerere. Et språk for å løse økonomiproblemer er kanskje ikke så godt for vitenskaplige problemer. Et språk for ungdomsskoleelever passer kanskje ikke like bra for universitetsstudenter.

2. Hvert språk kan brukes på flere typer av datamaskiner og **operativsystemer**. Når et nytt språk utvikles, kan det lages program som oversetter det til maskinspråk som er spesielle for ulike datamaskiner. Logo må for eksempel kunne kjøre både på Linux, Mac og Windows.

Å gjøre datamaskiner lettere å bruke

Høynivåspråk gjør det lettere for folk å lage dataprogrammer. Finnes det andre måter en datamaskin kan hjelpe mennesker?

Svaret er "Ja". Utviklere har gjort maskinene enklere å bruke. På et tidspunkt mente man at elever måtte lære å programmere, hvis de skulle kunne bruke datamaskinen. Det stemmer ikke lenger. De fleste kan bruke en datamaskin som en "svart boks" uten å vite hva som skjer der inne.

I stedet bruker elever og lærere vanlige dataprogrammer, eller programpakker. Eksempler er Office-pakken eller CS3-pakken som gir brukere tilgang til vanlige **sluttbrukerprogram**(Du er sluttbruker). Disse er laget slik at alle kan lære å bruke dem. En tekstbehandler er et meget avansert verktøy, men det tar deg bare noen minutter å lære deg å bruke de viktigste funksjonene.

Det samme gjelder for regneark, grafikk og databaser. Alle er kompliserte programmer med tusenvis av kodelinjer du ikke trenger å se, lære eller bekymre deg om. Men programmene er likevel lette å lære. Hvert program er laget for en spesiell type av problemer. Du ville ikke bruke en tekstbehandler for å løse regneproblemer eller lage en database.

En **brukermanual** er ofte på flere hundre sider. Det tar dagevis å studere den og praktisere og øve på alt en tekstbehandler kan gjøre. Det samme gjelder for andre programmer. De fleste vanlige programmer er derfor laget slik at du kan lære dem selv, ofte bare ved å bruke dem, prøve og feile. Derfor følger også slike programmer som regel de samme prinsippene med menyer, valgmuligheter, visninger og redigering. Har du lært deg å lagre en fil i et program, finner du snart hvordan du skal gjøre det i et annet program.

Likevel har programmene ulike konsepter og ulike prosesser, og det er først og fremst konseptene du trenger å lære. Du må forstå konseptet med databaser for å kunne gjøre fornuftig bruk av et databaseverktøy som for eksempel Microsoft Access. Lær deg konseptene, så forstår du hvordan du skal bruke dataprogrammene som verktøy for å løse problemene.

Det er enda et nivå av kompleksitet. Et moderne dataprogram har et innebygget programmeringsspråk. En database er et godt eksempel. Databaseprogrammet har et programmeringsspråk som hjelper deg å løse databaseproblemer. Du kan lett lære deg å bruke en database som hjelp for å løse enkle databaseproblemer. Men det tar hundrevis av timer å studere og praktisere det fullt ut for å mestre hele dataprogrammet.

En tekstbehandler har også et innebygget programmeringsspråk, som du kan bruke til å utføre for eksempel rutiner som repeterer seg ofte. Dette kalles gjerne **makrospråk**. Du kan lage en **makro** som du bruker hver gang du skal gjøre det samme menyvalget. La oss si at du

skal klippe ut tekst. Du kan bruke musen, flytte den til saksen eller til menyen for å redigere. Du velger menyvalget ”klipp ut”, og teksten klippes ut. Men etter en stund finner du ut at det å klippe ut er noe du gjør ofte, og du blir lei av alle museklikkene du må gjøre. Du lærer deg da at du kan holde Ctrl- tasten nede og trykke X. Altså Ctrl+X. Det utfører akkurat det samme; klippe ut det som er markert.

Du har akkurat brukt en innebygget makro. Snart finner du ut at du kan lage deg makroer selv, for rutiner du utfører ofte. Du kan tilordne makroen til en tastaturkommando som ikke er i bruk for en annen makro. Snart har du et sett av tastaturkommandoer som gjør deg til en mer effektiv og fornøyd PC – bruker. Du lærer deg datamaskinens primitiver.

Plutselig finner du også ut at makroen kan redigeres i makrospråk, som er innebygget i tekstbehandleren. Nå blir det litt vanskeligere, men bruker du en del tid på å studere hvordan makrospråket fungerer, så har du et kraftig verktøy for å jobbe smart. Du lærer deg faktisk å programmere i tekstbehandleren! Og det er gøy!

I denne boken har vi ikke lært deg et eneste dataprogram. Det er bevisst valgt slik fordi det er viktigere å gi deg hjelp til å lære å tenke smart. Det er å bli smart, denne boken handler om. Har du jobbet deg gjennom boken, blir du smartere, fordi du begynner å praktisere metakognisjon.

Noen tanker om å være programmerer

Alle elever kan lære seg å skrive enkel kode for å løse enkle problemer. Millioner av elever har lært å skrive enkle programmer i Logo. Særlig i USA. Ved mange studier på høyskoler og universiteter, lærer man å skrive kode for dataprogrammer. Dette gjelder innenfor vitenskap og ingeniørstudier.

Likevel blir vanlige sluttbrukerprogrammer stadig bedre. Dette betyr at det for hvert enkelt fag eller studium antakelig finnes dataprogrammer som er laget spesielt for faget. Du kan raskt lære slike programmer for å løse et stort utvalg av enkle problemer innenfor faget.

Men etter hvert som du prøver å løse stadig mer kompliserte problemer, vil du finne ut at det krever mye lesing og trening å bruke dataprogrammene som verktøy. Oppgaven er nesten det samme som å lære seg å programmere høynivåspråk.

Dette leder til to spørsmål:

1. Hvem bør lære seg å programmere?
2. Hvem bør satse på en karriere som programmerer?

De følgende punktene bør hjelpe deg å besvare disse spørsmålene.

1. Du kan lære å skrive enkel kode i språk som BASIC, Logo, PHP, Java. Dette kan være nyttig for din egen utvikling og utdanning. Måler er egentlig ikke å lære programmering, men hvordan man programmerer, og hvorfor. Lær hva det kreves å løse problemer ved å skrive din egen kode i et programmeringsspråk. Du ønsker å lære hvordan din egen hjerne arbeider når den møter slike utfordringer.
2. På mange fagområder som ingeniørstudier og vitenskap, er det veldig nyttig å kunne skrive programmer. Hvis du har planer om å studere slike fag, kan du ha fordeler av å lære deg programmering.

3. En profesjonell programmerer er en person som har følgende ferdigheter:
 - a. Evnen til å mestre flere programmeringsspråk og mange moderne dataprogrammer for sluttbrukere.
 - b. God forståelse av datamaskiner, operativsystemer og nettverk.
 - c. Evnen til å skrive dataprogrammer som kan løse problemer innenfor et eller flere spesielle fagområder. For å kunne skrive programmer innefor et fagområde, bør du ha gode kunnskaper om selve faget og være god til å løse problemer innenfor faget.
 - d. Gode ferdigheter i å kunne kommunisere og samarbeide med menneskene du lager dataprogrammer for. Det betyr at du må ha meget gode muntlige og skriftlige ferdigheter.
 - e. Gode ferdigheter i å kunne samarbeide med andre programmerere og ulike mennesker som er involvert i teamarbeidet.
 - f. Stort pågangsmot og stor vilje til å lære nye teknologier og løse problemer. Du må være villig til å arbeide hardere enn vanlige arbeidsdager.

En datamaskin er et verktøy for å løse problemer. En programmerer forteller datamaskinen hvordan den skal løse problemer. Programmering krever presis kommunikasjon. En datamaskin gjør akkurat det den er instruert til å gjøre. Den minste feil fra programmereren kan lede til at datamaskinen gir gale svar på et problem.

Noen mennesker elsker å arbeide på et område som krever presis tenkning og nøyaktig kommunikasjon. Andre mennesker liker ikke slikt arbeid. En viktig grunn til å lære litt programmering er at du lærer mer om deg selv. Å lære mer om seg selv er absolutt et mål med din utdanning. Da ser du også hva du kan bli og hva du ikke bør bli.

Oppsummering av kapittel 10

Utviklingen av datamaskiner og dataprogrammer går stadig raskere. Dette skjer på tre områder:

1. Maskinvare blir stadig hurtigere, mer pålitelig og billigere.
2. Programmeringsspråk blir stadig lettere å lære og enklere å bruke.
3. Programvare for sluttbrukere utvikles stadig for problemløsning på de fleste fagområder.

Aktiviteter

Aktivitet 10.1 – Noe jeg tenkte på i kapittel 10

Var det noe spesielt du begynte å tenke på da du leste kapittelet? Kanskje du tenkte på at hjernen din er som en datamaskin og hvordan skolen "programmerer" den? Kanskje du tenkte på hvordan noen fag har detaljerte "steg for steg"-instruksjoner som du må lære, mens andre fag har få slike instruksjoner?

1. Skriv tittel: "Noe jeg tenkte på i kapittel 10"
2. Skriv og forklar hva du tenkte.
3. Forklar hvorfor dette interesserer deg.

4. Hvorfor tenkte du på det?
5. Hva har det med livet utenfor skolen å gjøre?
6. Er det noe mer du tenker på?

Aktivitet 10.2 – Del ideen

Finn en elev å snakke med. Begge velger en idé i kapittelet som hver synes er viktig. Bruk ett minutt hver til å fortelle den andre om ideen. Tips til hva dere kan snakke om:

- Hvorfor er ideen viktig for deg?
- Hvilken betydning har den for ditt liv?

Når dere snakker sammen bør dere bruke begreper fra kapittelet:

- Bug(feil)
- CPU (prosessor)
- Chip(silikonklatt)
- Datamaskinsystemer
- Debug(feilretting)
- Maskinvare
- Programvare
- Høynivåspråk
- Maskinspråk
- Integrerte kretser

Aktivitet 10.3 – Lag en quiz som tester faktakunnskap fra kapittelet

Lag 2 spørsmål. Du skal teste faktakunnskap fra dette kapittelet.

Det skal være tekstsvar. Bare ett riktig svar til hvert spørsmål.

Spørsmålene skal være så enkle at en som ikke har lest kapittelet kan svare.

Prøv quizen på noen elever.

Aktivitet 10.4 – Lag en quiz som tester forståelse av kapittelet

Lag 2 spørsmål. Du skal teste forståelse av kapittelet.

Presenter noe fakta fra kapittelet og still spørsmål som krever tenkning og vurdering av fakta.

Testpersonene kan slå opp i en bok, gjøre et søk eller lese i dette kapittelet for å lage et svar.

En quiz som tester forståelse, har alltid flere korrekte svar. Du lager ingen svaralternativer.

Spørsmålene skal være så vanskelige at de bare kan besvares av en person som har en god forståelse av innholdet i dette kapittelet.

Prøv quizen på noen elever. Det skal være tekstsvar, og ingen fasit. Diskuter svarene sammen til slutt.

Aktivitet 10.5 – Snakke med en datamaskin

På hvilke måter er det å kommunisere med en datamaskin det samme som å snakke med en person? På hvilke måter er det forskjellig? Tror du at en person som er flink til å snakke med andre mennesker også vil være en god programmerer? Hvorfor?

Aktivitet 10.6 – Å kunne programmere

Kan du litt programmering? Hvis svaret er "ja", skal du gjøre litt metakognisjon på hvordan du lærer. Liker du å programmere? Var det vanskelig å lære? Vil du lære mer om programmering?

Aktivitet 10.7 – Elever som programmerere

Skriv argumenter for og imot følgende spørsmål:

"Bør alle elever i videregående skole lære litt programmering?"

Når du skriver argumentene, skal du bestrebe deg på å se andres syn på saken. For eksempel kan du prøve å bruke lærerens, elevens og IT- personalets argumenter når svaret er "Ja". Deretter kan du bruke de samme personenes argumenter når svaret er "Nei".

Skriv til slutt din egen konklusjon.

Aktivitet 10.8 – Skriv en artikkel

Denne boken stiller spørsmålet:

"Når vi antar at en datamaskin kan løse eller hjelpe til med å løse et problem for elever på skolen, hva bør elever lære om å løse dette problemet?"

Skriv en artikkel eller en blogg om dette.

Aktivitet 10.9 – De viktigste ideene i denne boken

Hva synes du er de viktigste ideene i denne boken? Hvorfor? Begrunn svaret.

Ordliste

Begreper vi har benyttet i denne boken, forklares i hvert kapittel. Ordlisten er en oversikt over disse begrepene.

Begrep	Beskrivelse av begrepet
Abakus	Kuleramme.
Algoritme	En steg for steg prosedyre eller rutine som forteller hvordan man kan løse et spesielt problem. Prosedyren garanterer at problemet løses hver gang under forutsetning av at det ikke er noen feil i stegene. I matematikk brukes begrepet algoritme i multiplikasjon, divisjon, beregning av kvadratroter og når andre databeregninger skal utføres.
Assosiasjon	Du kan få en assosiasjon når du opplever noe eller tenker på noe. Når du tenker på en ting, fører det noen ganger til at du begynner å tenke på noe annet. Det andre er en assosiasjon til det første du tenkte på. Tenk på en sykkel. Får du en assosiasjon til noe annet? Kanskje du tenker på en sykkel tur du har hatt? Eller kanskje du tenker på en pumpe?
Asynkron kommunikasjon	Indirekte kommunikasjon. E-post er asynkron kommunikasjon. Avsender får ikke svar i sanntid, og må vente til mottaker har bestemt seg for å svare.
Binær	Datamaskinen kan bare forstå to verdier; 0 og 1. Datamaskinen benytter 2-tallsystemet, eller det binære tallsystemet som det også kalles. Dette tallsystemet har bare to sifre, 0 og 1. Datamaskinen forstår likevel

Begrep	Beskrivelse av begrepet
	kompliserte tall, tegn og data, ved å arbeide med lengre grupper av sifrene 0 og 1. Eksempel: Bokstaven A kan for datamaskinen hete 1000001.
Brainstorming	Å lage en liste over alle ideer du kommer på. Du skal ikke utelate noen ideer under selve brainstormingen. Alt du kommer på skal skrives ned.
Brukermanual	En brukerveiledning for programvare eller maskinvare. Brukermanualer følger ofte med programvare som egne installerte hjelpefiler. Disse filene utgjør som regel et stort antall sider med informasjon, veiledning og hjelp til hvordan man kan bruke programmene. Brukermanualer for maskinvare kan i noen tilfeller være papirbaserte, og i andre tilfeller digitale dokumenter som kan lastes ned.
Brukervennlig	Beskrivelsen av et grensesnitt mellom menneske og datamaskin som er lett å bruke og lett å lære. En datamaskin kalles brukervennlig dersom den er lett å bruke, hjelper deg med enkle brukerfeil og er lett å lære å bruke.
Bryt ned – strategi	En strategi for å bryte ned problemer i mindre problemer. Målet er å finne alle primitiveene og arbeide med disse. Se <i>Primitiv</i> . Når alle primitiveene er løst og løsningene er smeltet sammen på en hensiktsmessig måte, er det opprinnelige overordnede problemet løst.
Bug	En feil eller en misforståelse. En bug i et dataprogram kan skyldes en feil i planen for å lage programmet. Det kan også være en feil eller misforståelse i prosedyren som programmeres.
Bygg – opp strategi	En strategi for problemløsning som starter med primitiveene. Strategien kan være målrettet, slik at primitiver settes sammen til en helhet med et bestemt mål i sikte. Strategien kan også brukes på en mer tilfeldig måte, uten et klart mål i sikte. Målet kan formes etter hvert som du løser flere av primitiveene.
Byggestein	Et annet ord for <i>primitiv</i> . En byggestein kan være gangetabellen. Du bruker denne byggesteinen når du løser likninger eller divisjon. En byggestein er en rutine som du automatisk kan gjennomføre uten å tenke noe særlig. Du kan sikkert umiddelbart si at $7 * 7 = 49$. Det er en byggestein i matematiske problemer.
Chip	Se <i>Integrerte kretser</i> .
CPU	Central Processing Unit. Dette er datamaskinens prosessor som utfører alle instruksjoner og beregninger i et dataprogram. Noen datamaskiner har bare en CPU. Andre maskiner har flere prosessorer som kan arbeide samtidig med det samme problemet. Dette kan sammenliknes med hvordan neuronene arbeider i hjernen. Se <i>Neuron</i> .
Database	En organisert samling av data, ofte i databasetabeller. En datamaskin er spesielt godt egnet til å lagre databasedata og hente ut data fra databaser. Databaser kan også benyttes til å sammenstille data på mange forskjellige måter. Dette er spesielt nyttig i problemløsning.
Database	Et dataprogram som kjører som en tjeneste på en datamaskin. I databaser lagres data som regel i tabeller som er koblet sammen på forskjellige måter. Databaser har den fordelen at du kan sette sammen data hurtig på mange forskjellige måter, og dermed analysere sammenhenger og få svar på kompliserte spørsmål som det ville være vanskelig å besvare ellers. Databaser kan gjøres tilgjengelig også over nettverk, slik at du kan skaffe deg informasjon fra databaser som ligger langt unna.
Datamaskinmodell	En modell som er lagret i en datamaskin slik at datamaskinen kan gjøre handlinger og beregninger for å løse et bestemt beskrevet problem. En datamaskin kan lagre bokstaver, tall, tegn og spesialsymboler som binær

Begrep	Beskrivelse av begrepet
	kode (kombinasjoner av 0 og 1). Datamaskinen kan også lagre lyd og bilder binært. (Eks: 00100010100) Se også <i>Binær</i> .
Datamaskinnettverk	En samling av datamaskiner og andre datakomponenter som er koblet sammen trådløst eller med kabler. Disse sammenkoblingene gjør det mulig for datamaskinene å kommunisere med hverandre.
Datamaskinprimitiv	En primitiv du har fordi prosessen med å beregne eller løse et problem kan utføres på en datamaskin du har tilgang til.
Datamaskinsystem	En kombinasjon av maskinvare og programvare som arbeider sammen.
Dataprogram	Se <i>Programvare</i>
Debug	Å fjerne feil fra dataprogrammer ved å analysere den overordnede prosessen for å løse et problem.
Desktop publishing	Bruk av programvare for å lage layout for sammensatte tekster med bilder, skifter, diagrammer, illustrasjoner. Dette er den fullstendige prosessen ved å utforme dokumenter layoutmessig og skrive ut eller publisere på annen måte. Ofte trykking.
Dårlig definert problem	En problemsituasjon hvor en av de fire komponentene <i>forutsetninger, mål, ressurser/begrensninger og eierskap</i> er dårlig definert.
Eierskap	En av de fire komponentene ved et problem. Et problem kan bare eksistere dersom en gruppe av mennesker eller en enkelt person er interessert i å løse det, villig til å bruke tid og krefter på det. Eierskap betyr at du er interessert i å løse en gitt problemsituasjon og oppnå målet du har satt.
Evaluerer	Gjøre en vurdering av noe. Er det bra, dårlig, godt nok, har jeg nådd målet, satte jeg meg riktig mål, er problemet løst?
Fagkunnskap	Detaljert kunnskap innenfor et spesielt fag eller fagfelt. Hvert fagområde har samlet en stor mengde informasjon og kunnskap. Hvert fagfelt har sin egen terminologi; et sett av begreper som brukes hovedsakelig innenfor fagfeltet. Se <i>Terminologi</i> . Fagfelt har også ofte sin egen notasjon, som for eksempel matematikk. Notasjon er symboler og tegn som anvendes i faget. Se <i>Notasjon</i> .
Faktakunnskap	Hjernen kan arbeide med enkle fakta og med forståelse av komplekse ideer og sammenheng mellom ideer og fakta. Læring av enkle fakta innebærer at data kan huskes og gjenskapes. På engelsk kalles det "lower-order thinking". Vi kunne oversette det til norsk med "tenking av lavere orden".
Fire- steg plan for problemløsning	Se <i>Fire- steg strategi for problemløsning</i> .
Fire- steg strategi for problemløsning	En strategi for problemløsning som består av fire steg: 1. Forstå problemet 2. Utarbeide en handlingsplan 3. Gjennomføre handlingsplanen 4. Undersøke resultatene for korrekthet og for å se om nye problemer har oppstått
Fjern overføring	Når to problemsituasjoner er så fjernt fra hverandre at du ikke uten videre er i stand til å overføre erfaringer og ferdigheter fra det ene problemet til det andre. Når du er i stand til det, har du evne til fjern overføring.
Formelt problem	Et uttrykk som brukes for å legge vekt på de fire komponentene for problemer generelt: 1. Forutsetningene eller problemsituasjonen 2. Mål 3. Ressurser og begrensninger 4. Eierskap

Begrep	Beskrivelse av begrepet
	Du har forsikret deg om at alle 4 komponentene er tilstede og er klart definert. (Klart definert problem)
Forståelse	Hjernen kan forholde seg både til enkle fakta og kompliserte ideer og relasjoner mellom fakta. Forståelse handler om å se sammenheng mellom fakta og kunne forstå kompliserte ideer og relasjoner mellom ideer og fakta. Hvorfor sier vi at Columbus oppdaget Amerika når det allerede bodde masse mennesker der da han ankom? Hva gjør statsministeren til en god leder av landet vårt? Hvorfor lever noen mennesker i fattigdom, og hva kan vi gjøre med det? Dette klarer vi ikke å besvare uten å ha forståelse. På engelsk kalles det "higher-order thinking". Vi kunne på norsk oversette det med "tenking av høyere orden"
Forutsetninger	Dette er forutsetningene for problemet. Det er tilstanden vi har nå vi begynner å arbeide med problemet. Det er en av de fire bestanddelene for et problem. Det kalles også <i>gitte forutsetninger</i> . Se <i>Problemsituasjon</i> .
Frekvenser (lyd)	All lyd er bølger med forskjellige frekvenser. Disse bølgene kan spre seg i luft. Dette måles i Hz. Lave frekvenser er dype lyder, som bass. Høye frekvenser er diskantlyder som oppleves som høye (ikke nødvendigvis sterke). Gresshopper lager lyder med høye frekvenser.
Fulltekstdatabase	En database som inneholder hele bøker og tidsskrifter. Du kan søke i all tekst, ikke bare titler, forfatter, emner og så videre.
Godt definert problem	Et problem hvor alle de fire komponentene <i>forutsetninger</i> , <i>mål</i> , <i>ressurser</i> og <i>begrensninger</i> og <i>eierskap</i> er godt definert. Målet er tydelig og alle ressurser og begrensninger er godt kjent. Du er også interessert i å løse problemet. Det er viktig for deg.
Grafiske programmer	Dataprogrammer for å lagre, manipulere og vise bilder, diagrammer og annen grafikk.
Grensesnitt	Når vi mennesker bruker et dataprogram på en datamaskin, er det et grensesnitt mellom oss og maskinen. Dette grensesnittet gjør at vi kan kommunisere med datamaskinen og bruke den. Vi snakker ofte om grafiske brukergrensesnitt, fordi dataprogrammer vises grafisk med vinduer, menyer, knapper og liknende. Men tastaturet og mus er også et grensesnitt mot maskinen for oss. Når grensesnittet er lett å bruke, sier vi at det er brukervennlig.
Grunnleggende ferdigheter	Ferdigheter du trenger for å løse mange problemer. I utdanning er dette å uttrykke seg muntlig, skriftlig, kunne lese, regne og bruke digitale verktøy.
Grunnleggende oppgave du kan	Dette kalles også "en primitiv". Se <i>Primitiv</i> . Gangetabellen er en slik grunnleggende oppgave du allerede kan. Du slipper å tenke på hvordan du regner ut $5 * 6$. Du vet umiddelbart svaret. Det er mange grunnleggende oppgaver du allerede kan, og som du slipper å streve med. Dette har du lært gjennom livet.
Hardkoding	Et uttrykk innenfor programmering som betyr at man i kildekoden til dataprogrammet skriver inn faste verdier for data. Når noe er hardkodet i kildekoden, kan det ikke endres av sluttbruker. En programmerer kan hardkode verdier som aldri endrer seg. Et eksempel kan være et konstant tall, som brukes i mange sammenhenger, og hvor tallet alltid er det samme. 3.14 er et tall som brukes i geometri ved beregninger for sirkler. 3.14 har ubegrenset antall desimaler, men rundes ofte av med 2 desimaler for enkelhetsskyld. 3.14 kunne være et eksempel på et tall som kunne hardkodes, og da kanskje med flere desimaler, for større nøyaktighet.
Heuristisk	En tommelfingerregel for å løse et spesifikt problem. Dette er steg for steg

Begrep	Beskrivelse av begrepet
	prosedyrer som ikke er garantert å virke, men som virker egnet for å løse problemet og er laget for å være en mulig prosedyre som kan brukes. Du kan tenke på heuristisk som en algoritme som ikke er garantert å virke. Se <i>Algoritme</i> .
Hjernen	Hjernen har billioner av neuroner. Neuronene lagrer informasjon og prosesserer informasjon. Hvert neurom er som en datamaskins prosessor med et minne. Se <i>Neuron</i> .
Høynivåspråk	Et programmeringsspråk som for eksempel BASIC, COBOL, Logo, Java, C++, C# eller PHP. Alle kodelinjer i språket oversettes av datamaskinen til maskinspråk. Høynivåspråk er lettere å forstå enn maskinspråk, fordi det likner mer på vårt eget talespråk enn maskinspråk gjør.
Informasjonsalderen	Et samfunn hvor store deler av befolkningen arbeider med service og informasjon og har yrker som krever bruk av informasjonsteknologi(datamaskiner, mobiltelefoner, TV). I USA begynte informasjonsalderen i 1956. I dette året ble for første gang antall ansatte i service- og informasjonsyrker høyere enn antall ansatte i industrien.
Integrerte kretser	En elektrisk krets på størrelse med en negl, som inneholder et stort antall transistorer og andre elektroniske komponenter. Det kalles ofte en chip på grunn av en liten klatt(chip) med silikon som brukes for å lage det.
Intuisjon	Noe du antar eller fornemmer, men ikke vet sikkert. Noen ganger kan vi ta en avgjørelse bare ved å bruke vår intuisjon. Vi kan fornemme at en person ikke er til å stole på, men vi vet det ikke. Vi kan bruke intuisjon for å avgjøre om det er smart eller ikke å stole på personen. Intuisjon kan også brukes på mange andre situasjoner, ikke bare i vårt forhold til andre mennesker.
Kalkulatorprimitiv	Dette er en primitiv du har fordi prosessen med å regne ut eller beregne et svar er bygget inn i kalkulatoren. Du har tilgang til en kalkulator.
Klart definert problem	Et problem hvor forutsetninger, mål, ressurser/begrensninger og eierskap er klart definert.
Kognisjon	Tenkning.
Kommunisere tydelig	Når flere skal samarbeide om å løse et problem, må alle ha en felles forståelse av problemet. Det må være godt definert slik at ingen er i tvil om hva som er målet og hvilke tiltak vi planlegger å gjennomføre for å løse problemet. Hvis i fordeler oppgaver, må alle jobbe mot samme mål.
Komplekse spørsmål	Spørsmål som ikke kan besvares på enkle måter. Komplekse spørsmål er ofte sammensatt av flere problemstillinger, noe som krever gode ferdigheter i problemløsning for å arbeide med.
Konsept	En abstrakt idé eller en mental modell. Modellen for å beregne arealet av et rektangel er et konsept. Konseptet kan brukes på alle rektangler. Konseptet er det samme hver gang. Stavekontroll er et konsept. Du kan sjekke en tekst for skrivefeil. En talebehandler er et konsept. Du kan tenke deg at du snakker til datamaskinen, slik at den skriver teksten du taler inn. Det finnes ubegrenset antall konsepter. Du kan gjennomføre et konsept, men da kaller vi det å utføre prosessen. I problemløsning er det viktig å kjenne til og forstå konseptene. Så kan vi bruke hjelpemidler for å utføre prosessene.
Konstruktivisme	Et syn på læring der kunnskap er noe den lærende må konstruere på egen hånd.
Korttidsminne	Et mellomlager eller temporært lager i hjernen. Dette er en aktiv del av hjernen. Det "bevisste" mellomlageret kan hos de fleste mennesker bare lagre noen få bruddstykker av informasjon.

Begrep	Beskrivelse av begrepet
Kroppsspråk	Ansiktsuttrykk, håndbevegelser, kroppsbevegelser, lyder og betoning av ord og uttrykk.
Langtidsminne	En permanent hukommelse, eller ting du husker lenge.
Læringsprosess	Hvordan du lærer. Det som skjer når du lærer.
Læringsteori	En teori om hvordan hjernen arbeider når den lærer nye ting.
Makro	En makro for en datamaskin eller et dataprogram, er et enkelt handlingsmønster eller en sekvens av handlinger som kan gjenta seg og startes med en enkelt kommando. Ofte startes makroen med en tastekommando. Makroer kan brukes for å utføre flere innebygde handlinger etter hverandre. Slike handlinger kan også utføres enkeltvis fra menyer i dataprogram, men når flere menyhandlinger settes sammen til en kombinasjon, og deretter kjøres med en enkel tastekommando, kan du effektivisere arbeid som repeterer seg ofte.
Makrospråk	Et programmeringsspråk eller skriptspråk hvor du kan skrive din egen kode for å lage en makro. Slike skriptspråk likner ofte på høynivåspråk som brukes for å lage fullstendige dataprogrammer, men de er som regel enklere og har færre muligheter for å skrive avansert kode. I vanlige dataprogrammer kan du ofte "ta opp" en makro nesten på samme måte som du tar opp musikk. Du starter opptaket, gjør alle handlingene en gang og i riktig rekkefølge, og avslutter opptaket til slutt. Deretter kan du "spille av" makroen, og den utfører nøyaktig det samme som du akkurat har gjort i dataprogrammet.
Maskinspråk	Hver datamaskin er laget for å "forstå". Det betyr at datamaskinen må være i stand til å utføre et begrenset antall handlinger og operasjoner. Slike operasjoner, eller primitiver, kalles maskinspråk. Ulike merker og modeller har forskjellig maskinspråk.
Maskinvare	Det fysiske maskineriet i et datamaskinsystem. Det omfatter komponenter som skriver, disk, skjerm, tastatur, minne, vifter, hovedkort, prosessor, mus.
Matematikkmodell	En modell som representerer et formelt problem, ved å bruke matematisk notasjon og matematiske begreper. En formel som $A = I * b$, er en matematikkmodell. Se <i>Notasjon</i> .
Mental modell	En modell for et problem laget i hodet. En modell som representerer et formelt problem slik at man forestiller seg modellen i hodet. Se <i>Formelt problem</i> .
Metakognisjon	Å tenke om sin egen tenkning. Se <i>Tenke om egen tenkning</i> .
Minne	Se <i>RAM</i> .
Modell	En modell er en abstrakt representasjon av noe. Modellen viser nøkkelegenskaper ved problemet så godt at informasjon du får ved å arbeide med modellen sannsynligvis også vil gjelde for det virkelige problemet du lager modellen for.
Mål	En av de fire komponentene for et problem. Målet er den ønskete situasjonen vi vil oppnå. Ønsket resultat.
Nettverk	Se <i>Datamaskinnettverk</i> .
Neurale koblinger	Koblinger mellom neuroner i hjernen. Hjernen består av milliarder av neuroner og koblinger mellom dem. Å ha mange neurale koblinger, innebærer å kunne koble ulik kunnskap hensiktsmessig for å løse problemer.
Neuron	Hjerneceller.
Notasjon	En bestemt skrivemåte for et fagområde. I musikk skriver vi noter. I

Begrep	Beskrivelse av begrepet
	matematikk bruker vi matematiske symboler. I naturfag bruker vi egne skrivemåter for fysikk og kjemi. Hvert fag har sin egen notasjon.
Nær overføring	Nær overføring gjelder når to problemsituasjoner likner noe du har arbeidet med før. Da kan du lett overføre kunnskaper om hvordan du skal løse det nye problemet, ved å tenke på hvordan du tidligere løste det andre liknende problemet.
Operativsystem	Et operativsystem er et grensesnitt mellom datamaskinen og brukeren. Det sørger for at alle komponenter som tastatur, skjerm, harddisk, nettverk, lyd og så videre, virker og kan betjenes av brukeren. Det sørger for at brukere kan mate data inn, at data kommer ut igjen til brukeren. Operativsystemet holder alle programmer som blir installert, og tilbyr en rekke tjenester slik at installerte programmer kan vises på skjermen og betjenes. Dette kan være knapper, vinduer, tekstbokser, menyer og andre grafiske elementer som programmene bruker. Operativsystemet oppretter og behandler brukerkontoer med forskjellige rettigheter til å brukes tjenestene i systemet. Eksempler på operativsystemer er Linux, Solaris, Windows XP, Windows Vista, Mac OS X.
Parallell prosessering	En avansert beregnings- og prosesseringsteknikk hvor en datamaskin kan utføre flere prosesser samtidig og dermed parallelt. Hjernen vår fungerer på denne måten, siden hvert neuron har evne til både å lagre informasjon og prosessere informasjon.
Perspektiv	En persons synsvinkel. Det kan også være et syn som gjelder for en hel gruppe av mennesker. Forskjellige mennesker har ulike perspektiver eller synsvinkler på en sak eller et problem. En som er dømt for noe han ikke har gjort, har et helt annet perspektiv enn dommeren som dømte han.
Piksel	Et grafisk element, eller et bildeelement. Forestill deg at skjermen til datamaskinen er delt opp i et stort antall små prikker. En datamaskin kan lagre et bilde ved å lagre informasjon om fargen for hver enkelt prikk i bildet. Alle disse prikkene blir sammen hele bildet du ser på skjermen. Vi kan bestemme hvor mange prikker vi vil vise på en skjerm, ved å gjøre innstillinger for skjermens oppløsning.
Primitiv	Et problem du kan løse enkelt, raskt og nøyaktig. En "byggstein" som du kan bruke når du skal løse mer komplekse problemer. Du deler opp hovedproblemet i mindre primitiver, og løser dem enkelt. Et eksempel på en primitiv er gangetabellen. Har du lært den, slipper du å tenke på hvordan du løser $8 \cdot 7$. Du vet med en gang at det er 56. Se også <i>Bryt ned – strategi</i> og <i>Bygg opp – strategi</i> .
Problem	Et problem består av 4 komponenter: 1. Problemsituasjon eller forutsetninger 2. Mål 3. Ressurser og begrensninger 4. Eierskap Mange mener også at en person som har et problem ikke har noen klar formening om noen umiddelbar eller opplagt prosedyre for å løse det. Dette betyr dessverre at hvis du studerer og arbeider tilstrekkelig med problemet, er det ikke lenger noe problem.
Problempoppstilling	Dette er prosessen du gjennomfører for å lage et godt definert problem. Du bruker for eksempel brainstorming for å finne riktig mål og alle ressurser og begrensninger. Å stille opp et problem innebærer å gå fra en

Begrep	Beskrivelse av begrepet
	situasjon hvor du har et dårlig definert problem til en situasjon der du har et godt definert problem. Se <i>Godt definert problem</i> og <i>Dårlig definert problem</i> .
Problemsituasjon	En situasjon som opptar deg og angår deg. En situasjon hvor du er opplever avvik mellom hvordan ting er og hvordan du skulle ønske at det egentlig burde være. En problemsituasjon mangler en eller flere av de 4 komponentene ved et godt definert problem. Se <i>Godt definert problem</i> .
Problemtype	Kategori av problem. Et økonomisk problem er en type. Et venneproblem er en annen type. Et teknisk problem kan være hvordan du installerer et program eller formaterer en disk. Alle problemer kan kategoriseres med type.
Program	Se <i>Programvare</i> .
Programvare	Et detaljert sett av handlinger og instruksjoner som kan lagres i datamaskinens minne, slik at datamaskinen kan følge disse instruksjonene. Programvare består av programmer som tekstbehandler, databaser, og andre programmer som er laget for å hjelpe til med å løse spesifikke problemer. Systemprogramvare inkluderer programmeringsspråk som BASIC, Logo, Java, C++, PHP og flere.
Prosedyre	Et detaljert sett av instruksjoner eller steg, som kan utføres av en datamaskin. Det kan også være instruksjoner som kan utføres slavisk av en maskin eller en person. Kommandoer som utføres i en bestemt rekkefølge. For eksempel: Lad våpenet. Sikt. Skyt.
Prosess	Detaljerte oppgaver eller systematisk arbeid vi utfører når vi skal bruke et konsept. Hvis konseptet er stavekontroll av en tekst, så er prosessen selve arbeidet med å rette opp alle skrivefeil og grammatiske feil. Denne prosessen kan utføres med en datamaskin, men kan også gjennomføres manuelt av deg.
Prosessere	Behandle data og beregne resultater.
Processor	Tenkeenhet. Se <i>CPU</i> .
RAM	Random Access Memory eller RAM er datamaskinens minne. RAM kan lagre alle programmer du starter opp og data du skriver inn. Når maskinen slås av, tømmes RAM, slik at alt er borte. Hvis du skal ta vare på data du har i RAM, må data lagres enten som filer eller i databaser.
Regneark	Et dataprogram som kan modellere og hjelpe til med å automatisere mange av nøkkelfunksjonene i regnskap og bokføring. Regneark har siden det kom første gang, utviklet seg til et kraftig verktøy for mange andre regneoperasjoner. Regneark kan også benyttes til grafisk representasjon av tall.
Relasjon	En relasjon er et forhold mellom to eller flere ting. Det kan være en relasjon mellom bil og trafikk, mellom bank og sparing. I databaser er det relasjoner mellom ulike tabeller. Biler registrert i en tabell kan ha relasjoner til personer i en annen tabell. Men biler har sikkert ikke relasjon til hus som personene eier. Derimot har personene relasjoner både til bil og hus.
Relatert	Noe som er knyttet til noe annet. Hus er relatert til farger, ved at hus kan ha mange forskjellige farger, og farger kan brukes ved maling av hus. Barn er relatert til sine foreldre. Dette er som regel en sterk relasjon, siden foreldre og barn lever sammen i mange år. Vi snakker om at data kan være relatert til hverandre ofte i forbindelse med databaser. Hvis vi har personer registrert med navn, alder og telefonnummer i en tabell, kan disse personene være relatert med ulike typer av utdanning i en annen tabell.

Begrep	Beskrivelse av begrepet
	Per kan ha en relasjon til Restaurant- og matfag, men det kan også Raja ha.
Ressurser og begrensninger	En av de 4 komponentene ved et problem. Det er godkjente tiltak eller aktiviteter som en person kan benytte seg av når det arbeides med å løse problemet. Vi løser problemet ved å starte med forutsetningene og arbeide oss mot målet vi har satt. Vi bruker ressurser og begrensninger i dette arbeidet. Se <i>Problem</i> .
Samarbeidslæring	En læremetode hvor en mindre gruppe (2 – 4) elever arbeider sammen. Elevene hjelper hverandre å lære. De har alle et personlig ansvar for egen læring, men også et ansvar for å hjelpe medelever å lære.
Segmenter	I denne sammenhengen er det biter av bølger. Vi kan kutte opp en lydbølge i mange små biter, eller segmenter.
Skalamodell	En fysisk modell av et objekt, laget som en miniatyr av det virkelige objektet. En dukke er en skalamodell av et menneske.
Skriftlig modell	En modell laget med skrevne bokstaver, tall, tegn, diagrammer og andre skriftlige symboler.
Sluttbrukerprogram	Dataprogrammer som er laget for vanlige brukere. Det kan være generelle dataprogram som tekstbehandler, regneark, nettleser og databaser, eller de kan være mer spesielle dataprogram som 3D-programmer, musikkstudioprogrammer, videoredigeringsprogrammer eller fagspesifikke programmer. De fleste programmer kan installeres på en lokal datamaskin, men programmer kan også fjernstyres over nettverk eller betjenes ved hjelp av for eksempel en nettleser.
Smart person	Du er en smart person dersom du kan forstå og løse problemene du møter. Jo flere problemer du kan løse, dess smartere er du. En smart person kan løse mange forskjellige problemer. En smart person kan løse vanskelige problemer. En smarte person kan lære seg å løse nye problemer.
Strategi	En overordnet plan for tiltak som kan iverksettes for å fullføre en oppgave eller løse et problem. Det er ingen garanti at strategien vil virke. Det finnes mange forskjellige strategier som kan anvendes for å prøve å løse et problem.
Symbolisk modell	En modell som benytter symboler, slik som bokstaver, tall, diagrammer og andre symboler og tegn. Se <i>Datamaskinmodell</i> og <i>Skriftlig modell</i> .
Symbolmodell	En modell som bruker symboler som bokstaver, tall, tegn, diagrammer eller andre symboler. Se <i>Datamaskinmodell</i> og <i>Skriftlig modell</i> .
Synkron kommunikasjon	Kommunikasjon i sanntid. Telefonen er synkron kommunikasjon. Vi kunne kalle det direkte toveis kommunikasjon.
Tekstbehandler	Et dataprogram som er laget for å støtte skiving. Det kan tenkes på som en elektronisk skrivemaskin med et minne og flere datamaskinfunksjoner.
Tenke om egen tenkning	Dette handler om å bli bevisst på hvordan du tenker og hvorfor du tenker som du gjør. Det kalles metakognisjon. Det er overordnet tenkning. Å tenke om sin egen tenkning er viktig for å bli en bedre problemløser.
Tenkevaner	Hvordan du tenker i ulike problemsituasjoner. Gode tenkevaner kan kreve metakognisjon, pågangsmot, optimisme, vilje til hardt arbeid, vilje til å se ulike perspektiver, evne til å lete etter alternativer, evne til å se nye problemer som kan dukke opp. Dårlige tenkevaner innebærer at man ikke bryr seg så mye om å anstrenge seg, ambisjonene er små, man vil helst bli fort ferdig og er fornøyd bare det har blitt et resultat.
Terminologi	Fagspråk.
Tiltak	En handling vi utfører som et ledd i å løse problemet. For kompliserte

Begrep	Beskrivelse av begrepet
	spørsmål må vi nesten alltid gjennomføre en rekke konkrete tiltak. Det kan være å samle informasjon, lese bøker, søke på Internett, diskutere med venner.
Tiltaksplan	En plan for hvilke tiltak og handlinger vi vil sette i verk for å løse problemet.
Transistor	Transistoren er en av de viktigste komponentene i en datamaskin. Det er for datamaskinen en liten av/på – bryter laget av grunnstoffet silisium. Dette forurenses med stoffer som fosfor og bor, og får da spesielle egenskaper for å lede strøm. På transistoren er det montert tre tilkoblinger som kalles kollektor, emitter og basis. Det hele holdes sammen med silikon. Strøm kan ikke gå mellom kollektor og emitter, men når en liten strøm sendes til basis, kan strøm gå mellom de to andre likevel. Dermed kan transistoren fungere som en bryter, og det er akkurat det datamaskinen trenger, siden den forstår bare to signaler; av og på eller 0 og 1. De minste transistorene er så små, at mer enn en million av dem kan få plass på en kvadratcentimeter. Datamaskinen bruker et stort antall transistorer.
Ukritisk liste	En liste med alle ideer du kommer på når du brainstormer. Ingen ideer utelates, selv om de er dårlige. Dette gjør du etterpå.
Uløselig problem	Dette er et klart definert problem som ikke lar seg løse. Se <i>Klart definert problem</i> . Innenfor matematikk forsøker forskere å bevise at det finnes spesielle matematikkproblemer som ikke kan løses. I fysikk er Heisenberg-prinsippet kjent for at problemet med å spesifisere posisjonen og momentet til en partikkel ikke kan måles samtidig med høy grad av nøyaktighet.
Vakuumbør	En elektronisk komponent som også har blitt kalt radiator. Før menneskene utviklet transistoren, ble vakuumbør benyttet i elektroniske apparater som radioer, datamaskiner, forsterkere og TV. Konstruksjonen består av en lukket glasstube montert på en sokkel. Inne i glasstuben er det elektroder i et vakuum. Vakuumbøret ble benyttet i datamaskiner for å kunne produsere elektriske signaler. Vakuumbør produserte mye varme og tok mye plass. Vakuumbør ble erstattet av transistoren på 60-tallet.
Variabel	En verdi som kan endres eller varieres. Det kan være et tall, en ord, en tekst eller en annen verdi som datamaskinen kan behandle.
Verbal modell	En modell laget med talte ord eller andre lyder. Ofte inkluderer en verbal modell også kroppsspråk. Se <i>Kroppsspråk</i> .