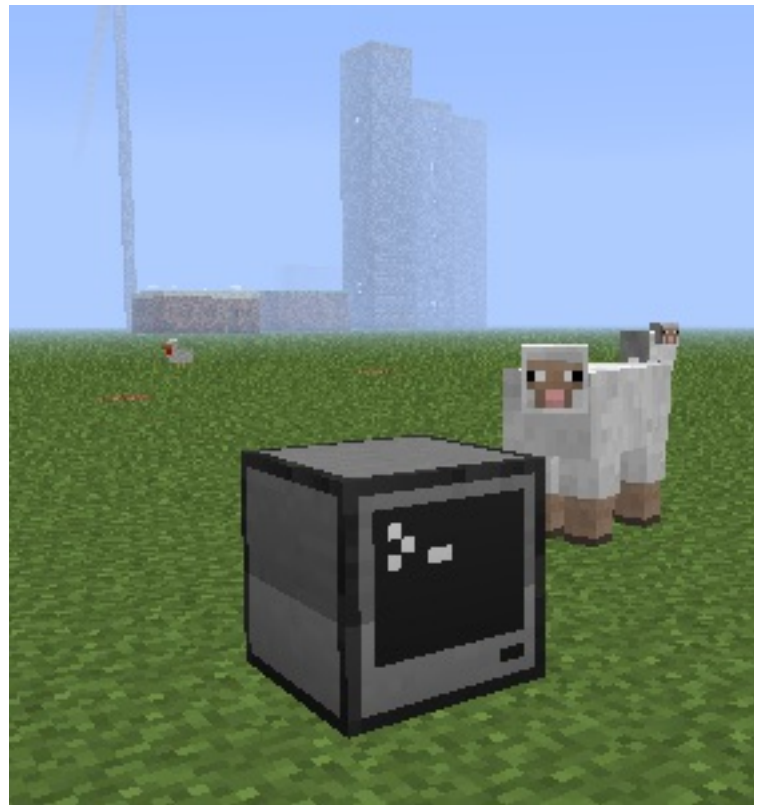


# Bli Kjent med Datama

## Introduksjon

Vi begynner med å bygge en enkel datamaskin. Etter å ha brukt litt tid bruker vi den til å låse opp en dør ved hjelp av passord.



## Steg 1: Vår første datamas

*Vi begynner med å lage en datamaskin.*

I utgangspunktet er det mye enklere å starte i *Creative Mode* når vi skal bruke tid på å samle materiale og slåss mot monstre. Vi vil derfor i for akkurat det du trenger i inventory'et ditt.

Men først, for å vise at en datamaskin kan bygges på vanlig måte i Mi

## ✓ Sjekkliste

- ☐ Trykk **E** for å åpne inventory'et ditt. Finn frem 7 **Stone**, 1 **Reds**
- ☐ Åpne et **Crafting table**, og legg ut materialet slik:



- ☐ Legg den nye datamaskinen i den nederste raden i inventory'et inventory'et.
- ☐ Bruk talltastene til å velge datamaskinen, og høyreklikk for å lag

I *Creative Mode* kan du også få tak i datamaskiner ved å trykke **E**, de fanen med datamaskinsymbolet.

## ComputerCraft datamaskiner

Høyreklikk en datamaskin for å starte den opp. Den vil åpne en svarer datamaskinens *kommandolinje*, og vi vil bruke den til å styre da

## Sjekkliste

- ☐ Prøv å skriv `help` og trykk enter.

Du får nå se noen tips om hvordan du kan finne ut mer om data se en liste over hvilke programmer som er på datamaskinen, eller man programmerer datamaskinen.

- ☐ Skriv `programs` og trykk enter.

Mange av disse programmene er enkle programmer som lar deg se av dem i en senere leksjon.

## Steg 2: Vårt første program

*Det er nå på tide at vi skriver vårt første program.*

Det er en lang tradisjon blant programmerere at det første programmet skriver en trivelig melding til skjermen. Vi følger den tradisjonen og begynner med

## Sjekkliste

- ☐ Start en datamaskin.
- ☐ Skriv `edit heiverden` for å begynne å skrive på et nytt program.
- ☐ I det nye vinduet, skriv

```
print('Hei verden!')
```

Etter at du har skrevet dette så trykker du på *Ctrl*-tasten og velger `Exit`.

- ☐ Vi har nå laget vårt første program. Prøv å skriv `programs` og d
- ☐ For å kjøre programmet vi har laget, skriver vi `heiverden` og try

## Prøv selv

Klarer du å endre på programmet slik at det for eksempel sier hei t deg?

Prøv å skriv `edit heiverden` en gang til. Da åpner programmet dit noe annet. Som tidligere må du bruke *Ctrl*-tasten for å gå til meny

## Steg 3: Litt enkel matemat

*Vi skal nå bruke datamaskinen til å regne litt matematikk for oss, og k regne.*

Datamaskinene i ComputerCraft bruker et programmeringsspråk som språk som ofte brukes inne i andre programmer. For eksempel kan og programmeres med Lua.

### Lua

Lua ble opprinnelig laget i Brasil på begynnelsen av 1990-tallet. På språk som het *Simple Object Language* (SOL). Ordet *lua* er portugis som passet sammen med SOL.

## Sjekkliste

Vi skal nå prøve å skrive noen kommandoer direkte i Lua. Dette er en

- ☐ Skriv `lua` og trykk enter. Dette starter en *Lua-tolker* som vil utf
- ☐ Vi begynner med kommandoen fra det første programmet vårt. samme som tidligere?
- ☐ Som de aller fleste programmeringsspråk er Lua glad i å regne. Lua kjenner alle de vanlige matematikk-operasjonene. Prøv for igjen hva hver av disse betyr?
- ☐ Skriv `math.random(1, 10)` og trykk enter.  
Dette skriver ut et tilfeldig tall mellom 1 og 10. Dette er et eksempel programmerer. I dette tilfellet heter funksjonen `random` og den

## Funksjoner

Alle programmeringsspråk lar deg lage noe som kalles *funksjoner*. sammen slik at det blir enklere å gjøre vanskelige ting. Lua kommer `print` og `math.random` som eksempler på dette. Senere vil vi også

## Sjekkliste

- ☐ Trykk pil opp-tasten slik at du kan kjøre `math.random(1, 10)` er forskjellige tall tilbake?
- ☐ Avslutt Lua-tolkeren ved å skrive `exit()` og trykk enter.

# Steg 4: En liten matteprøve

Vi skal nå bruke de tilfeldige tallene til å lage et enkelt spill. Underveis

## ✓ Sjekkliste

☐ Start et nytt program ved å skrive `edit mattetest` og trykk ent

☐ Skriv inn følgende program

```
local tall1 = math.random(2, 12)
local tall2 = math.random(2, 12)
print('Hva er ' .. tall1 .. ' ganger ' .. tall2 .. '?')
```

Pass på at du skriver de to punktumene `..` riktig. Disse betyr a

☐ Lagre og avslutt editoren. Kjør programmet ved å skrive `mattet`

Blir du spurt om svaret på et gangestykke? Spør den om et ann  
Hva skjer om du prøver å svare?

## Variabler

Vi har sett det meste i programmet vårt tidligere. `math.random` lag  
skjermen. Det nye er at vi bruker *variabler* til å huske verdiene av  
et tilfeldig tall mellom 2 og 12. Dette tilfeldige tallet husker vi så m  
foran variabelen sier at vi bare skal huske variabelen i dette progr

## ✓ Sjekkliste

- ☐ Vi skal nå jobbe videre med programmet. Skriv [edit mattetest](#)

```
local tall1 = math.random(2, 12)
local tall2 = math.random(2, 12)
print('Hva er ' .. tall1 .. ' ganger ' .. tall2 .. '?')

svar = read() -- ny linje
```

Lagre, avslutt, og kjør programmet på nytt. Får du lov til å svare på tastaturet.

- ☐ Det neste vi vil er at programmet skal sjekke om vi svarer riktig tester. Disse kan teste om noe er sant, og vi vil bruke dem for å faktiske svaret. Legg til en [if](#)-test nederst i programmet ditt slik

```
local tall1 = math.random(2, 12)
local tall2 = math.random(2, 12)
print('Hva er ' .. tall1 .. ' ganger ' .. tall2 .. '?')

svar = read()

if tonumber(svar) == tall1 * tall2 then -- alle linjer
    print('Ja, svaret er ' .. svar) -- og ned er ny
else
    print('Nei, det riktige svaret er ' .. tall1 * tall2)
end
```

Kjør programmet igjen. Hva skjer når du svarer riktig? Hva skjer

- ☐ Det er kjedelig at vi hele tiden må starte programmet på nytt. Vi spørsmål hver gang vi spiller. Her bruker vi en [for](#)-løkke som b

```
for i = 1, 5 do -- ny linje
    local tall1 = math.random(2, 12)
    local tall2 = math.random(2, 12)
    print('Hva er ' .. tall1 .. ' ganger ' .. tall2 .. ' ' .. i .. ' ganger')
```

```

svar = read()

if tonumber(svar) == tall1 * tall2 then
    print('Ja, svaret er ' .. svar)
else
    print('Nei, det riktige svaret er ' .. tall1 * t
end
end
-- ny linje

```

Test programmet ditt igjen. Blir du spurt om fem gangestykker? annet antall gangestykker? Kan du bruke en variabel som sier h



Til slutt vil vi at programmet skal telle hvor mange riktige svar v denne gangen vil vi endre verdien av variabelen etterhvert som

```

local ant_stykker = 5 -- ny linje
local ant_riktig = 0 -- ny linje

for i = 1, ant_stykker do -- endret
    local tall1 = math.random(2, 12)
    local tall2 = math.random(2, 12)
    print('Hva er ' .. tall1 .. ' ganger ' .. tall2 .. '

    svar = read()

    if tonumber(svar) == tall1 * tall2 then
        print('Ja, svaret er ' .. svar)
        ant_riktig = ant_riktig + 1 -- ny linje
    else
        print('Nei, det riktige svaret er ' .. tall1 * t
    end
end
-- ny linje nec
print('Du klarte ' .. ant_riktig .. ' av ' .. ant_stykke

```



## Prøv selv

Kan du endre programmet slik at det spør om andre typer mattestykkestykker?

For minusstykker, hvordan kan du lage tilfeldige `tall2` på en slik måte?

For delestykker, hvordan kan du enkelt lage stykker som du vet går opp?

Kan du bruke `math.random` og passende `if`-tester for å tilfeldig velge mellom delestykker og minusstykker?

## Steg 5: Passordlås på en d

*Datamaskinene våre er jo en del av Minecraft. Vi skal nå se et enkelt eksempel på hvordan man kan låse opp datamaskinene i resten av Minecraft-verdenen rundt oss.*

### Sjekkliste

- ☐ Gå ut av datamaskinen ved å trykke *Esc*-knappen.
- ☐ Samle sammen litt **Stone**, en **Iron door** og en **Computer**, og lag en datamaskin rett ved siden av døren. Det skal se omtrent slik ut:



- ☐ Start datamaskinen.
- ☐ Lag et nytt program som heter `passord`:

```
local password = 'kodeklubben'

while true do
  print('Hva er passordet?')
  svar = read()

  if svar == password then
    print 'Riktig'
  else
    print 'Feil'
  end
end
```

Dette programmet ligner ganske mye på `mattetest`-programm  
ikke har sett tidligere, nemlig en `while`-løkke. Slike løkker vil fo

vårt tilfelle er dette *noe* verdien `true` som alltid er sann. Det betyr at vi fortsetter å spørre oss om passordet for alltid!

- ☐ Kjør programmet. Oppfører programmet seg slik du hadde trodd?

## Avslutte programmer

For å avslutte dette programmet holder du inne *Ctrl* og *T* samtidig. Da blir teksten `Terminated` skrevet på skjermen. Dette fungerer både her

## ✓ Sjekkliste

- ☐ Vi vil at døren skal åpne seg når vi skriver riktig passord. Dette betyr at vi overføre energi i Minecraft. Endre programmet `passord` som følger:

```
local password = 'kodeklubben'

while true do
    print('Hva er passordet?')
    svar = read()

    if svar == password then
        redstone.setOutput('left', true)      -- endret
        sleep(5)                                -- endret
        redstone.setOutput('left', false)      -- endret
    end
end
```

Kjør programmet. Hva skjer nå når du skriver riktig passord?

Funksjonen `setOutput` i `redstone`-biblioteket skrur av eller på `left` hvilken side av datamaskinen døren står på. Du kan også bruke funksjonen `sleep` gjør at datamaskinen sover, det vil si gjør ingenting.

sekunder, før energien skrur av igjen og døren lukker seg.



Før vi er helt fornøyde med dette programmet vil vi gjøre noen `term`-biblioteket. Den ene vil rense skjermen, mens den andre som betyr øverst til venstre. I tillegg forteller vi `read` at vi ikke stedet vil vi at `*`-tegn skal vises. Programmet ser da slik ut:

```
local password = 'kodeklubben'

while true do
    term.clear() -- ny linje
    term.setCursorPos(1, 1) -- ny linje
    print('Hva er passordet?')
    svar = read('*') -- endret

    if svar == password then
        redstone.setOutput('left', true)
        sleep(5)
        redstone.setOutput('left', false)
    end
end
```

Gratulerer, du har nå lært ganske mye om hvordan man programmerer og gjerne å forandre noen av programmene vi har laget. Kan du lage der

## Prøv selv

Datamaskinene kan sende ut *redstone*-energi i alle retninger. Prøv datamaskinen ved hjelp av litt *redstone wire*. Aktiver denne fellen

Alle kan vi skrive feil passord innimellom. Det er kanskje bedre å ikke gangen på rad? Innfør en variabel som teller hvor mange feil passord skrives på rad.

