



ComputerCraft

[ComputerCraft](#) er en mod til [Minecraft](#), som gir deg muligheten til å bygge og programmere datamaskiner og roboter inne i Minecraft-verdenen.

Installering av ComputerCraft

Se [Installering av ComputerCraft](#).

Kom i gang

Vi er nå klare til å slå oss opp som programmerere i Minecraft-verdenen. Mens vi utforsker datamaskinene er det enklest med et rolig og enkelt spill:

1. Velg **Singleplayer**, og klikk **Create New World**.
2. Gi verdenen din et navn, og sett **Game Mode** til **Creative**. Klikk **Create New World**.
3. Du kan nå lage din første datamaskin. Trykk **E** for å gå til Inventory-listen. Øverst kan du nå velge å klikke **>** for å gå til neste side. Trykk deretter på datamaskin-symbolet (en grå kloss/skjerm), og gi deg selv en **Advanced Computer**.
4. Sett ut en **Advanced Computer** ved å høyre-klikke. Høyre-klikk så en gang til på datamaskinen du nettopp satte ut. En svart skjerm skal dukke opp, med teksten **CraftOS** og et versjonsnummer i øverste linje.

Gratulerer! Du har startet en datamaskin inne i Minecraft-verdenen.

Oppgaver

Se [Kodeklubbens oppgaver](#).

Leksjoner

Materialet her er i utgangspunktet tilrettelagt for et kurs med rundt 8 leksjoner på rundt en time. Vi antar lite eller ingen programmeringserfaring, men det vil gjøre leksjonene lettere, og ikke minst morsommere, om man har litt kjennskap til og interesse for Minecraft.

Leksjonen

- [Introduksjon til ComputerCraft](#)

er lagt opp til å brukes i et enkeltstående introduksjonskurs (som ikke nødvendigvis går videre med de andre leksjonene), og dekker det meste materialet fra leksjon 1 og 2 nedenfor. Forskjellen er at man har litt mer fokus på å vise frem *action* på bekostning av en noe grundigere innføring i enkelte begreper.

Anbefalt rekkefølge for leksjonene er

1. [Bli Kjent Med Datamaskinen](#)
2. [Robotinvasjon](#)
3. [Hendelser](#)
 - Hvordan bevege seg rundt i filsystemet
 - Kopiering av filer
 - Hendelser, enkel bruk av os.pullevent
 - Et bedre passord-program: auto-startup / kan ikke termineres med ctrl-T
 - Hvordan skrive kode i tekstbehandlere utenfor Minecraft
4. [Bygg et Hus](#)
 - Bruk en robot til å bygge hus
 - Hvordan flytte rundt
 - Bruk av funksjoner for enklere kode
 - Sikre at vi får nok materiale
 - Kommandolinjeargumenter for å bygge forskjellige typer / størrelser hus
5. [Sprettball](#)

- Introduksjon til peripherals
- Koble en skjerm til datamaskinen
- Skrive tekst til skjermen
- En ball faller over skjermen
- Hvor stor er skjermen
- Sprettball
- Kanskje også ticker?

6. Agricola

- Bruk av bonderoboter, som kan så / høste og essensielt ta vare på en åker på egen hånd
- Plukke opp og levere materiale

7. Kommunikasjon mellom Datamaskiner

- Bruk av disketter for å flytte programmer mellom datamaskiner (aside: hva er en diskett? :))
- Bruk av redstone for kommunikasjon
- Enda bedre passord program: datamaskinen trenger ikke stå ved siden av døren
- Trådløst nett mellom datamaskiner

8. GPS-Roboter

- Hvordan fungerer GPS / triangulering
- Sette ut baseroboter
- Hvordan kan en robot finne ut hvor den er
- Bruke lokasjon til å forbedre f.eks. bonderoboten?

Leksjonene er fortsatt under utarbeidelse (med store muligheter for endringer).

Læringsmål

Tabellen under er en oversikt over læringsmål i hver leksjon fra et programmeringsperspektiv. Symbolet / brukes for leksjoner som kommer innom læringsmålet, mens X brukes der læringsmålet er en sentral del av leksjonen.

| Mål \ Leksjon | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|--------------------------------------|---|---|---|---|---|---|---|---|---|
| Kommandolinjen | X | X | X | | | | | | |
| Interaktiv lua-tolker | X | X | X | | | | | | |
| Editering av programmer | X | X | X | | | | | | |
| Filsystemet | | | | X | | | | | |
| Variabler | / | / | / | | | | | | |
| If-tester | / | / | / | | | X | | | |
| For-løkker | / | / | / | | X | | | | |
| While-løkker | / | / | / | | | X | | | |
| Funksjoner | | | | | X | | | | |
| Hendelser (os.pullEvent) | | | | / | | | | | |
| Typer (strings, numbers, bools, etc) | | | | | | | | | |
| Matematiske operasjoner | | | | | | | | | |
| Relasjonelle og logiske operatorer | | | | | | | | | |
| Iteratorer | | | | | | | | | |
| Lokal og global scope | | | | | | | | | |
| Rekursjon | | | | | | | | | |
| Data-strukturer (tables) | | | | | | | | | |
| Debugging | | | | | | | | | |

Også denne tabellen er for tiden under utarbeidelse. Sannsynligvis vil ikke alle disse målene bli dekt i dette introduksjonskurset.

Det vil også dukke opp andre mål som for øyeblikket er glemt.

Lisens: [CC BY-SA 4.0](#)