Introduksjon til programmering

Scratch Lærerveiledning

Programmering

Scratch er et programmeringsspråk som gjør det enkelt å lage dine egne interaktive eventyr, spill og animasjoner - og så dele det med andre på internet. Dette dokumentet er en liten introduksjon til hva dataprogrammering egentlig er og litt om de grunnleggende konseptene innen dataprogrammering. Disse grunnleggende konseptene er ikke bare grunnleggende i Scratch, men faktisk i alle andre programmeringsspråk også.

Du benytter deg sannsynligvis av dataprogrammer hver dag, hverken du vet det eller ikke. Noen finner du på datamaskinen din, som for eksempel Word eller Minecraft og noen ser du ikke i det hele tatt. Eksempler på skjulte dataprogrammer er programmer som styrer trafikklys, autopilot til fly, sanntidsystemet på bussen, betalingsautomater for kort, navigasjonsystem i bilen og mye, mye mer. Kort sagt så kan et dataprogram gjøre akkurat det du vil gjøre, for bak hvert eneste dataprogram som finnes så er det et menneske som har skrevet det.

Når et menneske skriver et program til en datamaskin, så kan hun jo ikke skrive på norsk. Heller ikke på engelsk. Datamaskiner snakker jo ikke norsk! De språkene datamaskiner kan heter **programmeringsspråk**, og det er disse språkene dataprogrammer er skrevet i. Scratch er et eksempel på et programmeringsspråk. Andre kjente programmeringsspråk er Java, Python og C.

Men hva er egentlig et dataprogram? Et dataprogram er en **mengde med instrukser** som programmereren ønsker at datamaskinen skal følge til punkt og prikke. Man kan se på det som en slags oppskrift som datamaskinen skal følge. Men siden datamaskinen ikke kan tenke så må oppskriften være så nøyaktig at det ikke *kan* gå galt. Jobben til en programmerer er å skrive oppskrifter som datamaskiner må følge til punkt å prikke. For selvom datamaskiner ikke kan tenke selv, så er de utrolig flinke til å gjøre akkurat det vi mennesker har bedt dem om å gjøre.

La oss se litt på noen av de grunnleggende byggeklossene i et dataprogram.

Variable

I de aller fleste programmer så har man bruk for å lagre informasjon, for så å ha den lett tilgjengelig for senere bruk. Da bruker man en **variabel** for å holde på den informasjonen. Hvis jeg for eksempel vil skrive et biblioteksprogram, så kan det være en god idé å lage en variabel for hver bok i biblioteket. Man kan gi variabler et **navn** og et **innhold**. I biblioteksekseMpelet kunne vi da ha gitt en variabel navnet “Ringenes Herre” og innholdet ville da vært all teksten i Ringenes Herre bøkene. Sånn sett kan man også se på variabler som skuffer for å oppbevare data du bruker i programmet ditt. I Scratch så finner du alt som har med variabler å gjøre under `Data`. For å lage en ny variabel klikker du på `Lag en Variabel`. Du kan endre verdien til variablen med blokken `sett [variabel v] til (0)`. I spill kan variable være nyttige for å holde styr på poeng, tid, tur, hastighet, vinner og mye mer.

Løkker

Veldig ofte i programmer har vi bruk for å gjøre noen instrukser flere ganger. La oss si vi vil at datamaskinen vår skal lage (data)pannekaker. Da vil vi at datamaskinen skal hente et egg, knuse det i bollen og røre rundt. Og vi vil at den skal gjøre det mange ganger. Da kan vi bruke det som heter en **løkke** for å be den gjøre den samme oppgaven (blande inn et egg) så mange ganger vi måtte ønske. I Scratch så finner vi alle løkkene under `Styring`-kategorien. Hvis vi vil at noe skal gjentas et bestemt antall ganger, så bruker vi en `gjenta (10) ganger`-blokk. Du legger kanskje merke til at denne blokken ikke er som alle andre blokker? Den har nemlig et lite hulrom hvor du kan knepse inn de instruksene du vil skal gjentas. En anne viktig type løkke heter `for alltid`. I stedet for å gjenta noe et bestemt antall ganger, så gjentar den noe *i evigheten*. Da gjelder det å være forsiktig så man ikke ender opp med en uendelig stor eggerøre!

Tester

Noen ganger så vil vi bare at datamaskinen skal utføre en instruks *hvis* en eller annen betingelse er sann. Vi vil for eksempel bare at datamaskinen skal blande et egg i røren *hvis det ikke er rottent*. Det er derfor vi bruker **tester** i et dataprogram. Betingelsen vi tester etter må være noe som kan være sant eller usant. Enten er et egg rottent, eller så er det ikke det! Eksempler på andre tester vi kunne tenke oss å gjøre er å sjekke om to tall er like, eller om en figur i et spill har blitt spist eller om tiden har gått ut. I Scratch finner vi alt som har med tester å gjøre under `Styring`-kategorien. Den mest vanlige testen vi bruker heter `hvis <>`. Legg merke til at det er et tomrom i denne blokken. Her skal du dra inn betingelsen du ønsker å sjekke om er oppfyllt (husk: noe som kan være sant eller usant!). Du får ofte bruk for blokkene med spisse kanter i `Operatorer`-kategorien i disse tomrommene. Dette er blokker som `( )=( )` og `( ) < ( >` som brukes for å sjekke om noe er likt eller mindre enn noe annet.

Lister

Når man jobber med større mengder data, så kan det være lurt å bruke lister. Dette er like sant for mennesker på handletur som for datamaskiner. Hvis vi ser på en variabel som en slags skuff man kan putte informasjon i, så kan vi se på **lister** som kommoder! Lister er sånn sett en lur måte å behandle mange ting som en eneste ting. I stedet for å holde styr på en variabel per bok i bibliotekprogrammet vårt så kan vi putte alle bøkene (variablene) inn i en liste som heter “AlleBøkene”. Kommandoene i Scratch for å jobbe med lister er ganske like som de for variabler. Du finner dem under `Data` og lager en ny liste ved å trykke på `Lag en Liste`. Da kan du bruke blokker som `legg [MinVariabel] inn i [MinListe v]`, for å legge variablen MinVariabel inn i listen MinListe og `slett (1) fra [MinListe v]` for å slette det første elementet i listen MinListe.

Eksempelkode

Til slutt så viser vi et eksempel på et tenkt program en datamaskin kunne brukt for å lage (data)pannekaker. Legg merke til at programmet hverken er skrevet helt på godt norsk eller i et programmeringsspråk. Dette er for å gjøre det litt mer forståelig.

```
hent 3 egg
hent 2 dl melk
hent 2 dl mel
hent 1 bakebolle

bland mel i bakebolle
bland melk i bakebolle

gjenta 3 ganger:
 hvis egg ikke er rottent
  bland 1 egg i bakebollen

gjenta til bakebollen er tom:
 stek en pannekake*
```