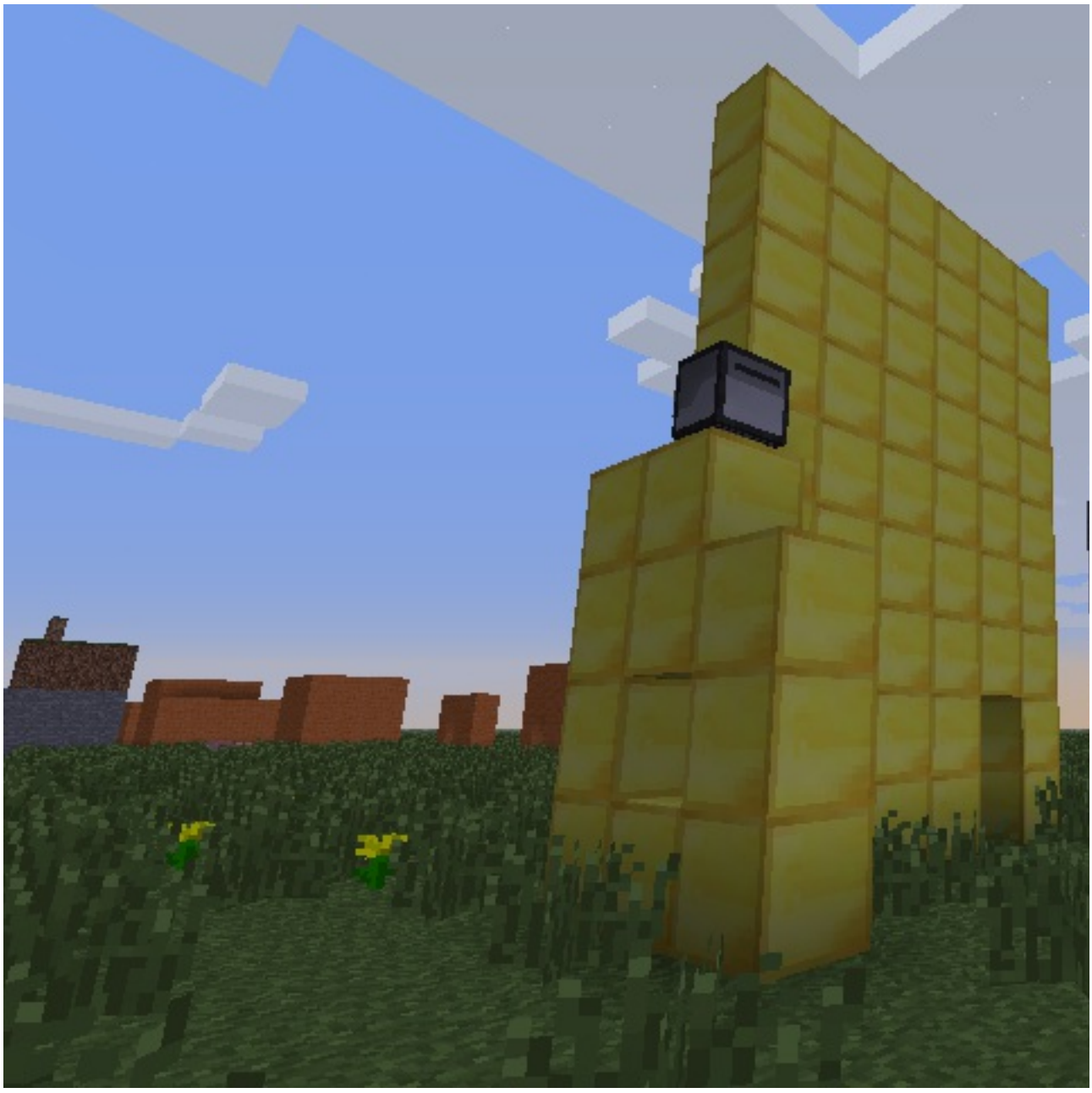
Bygg et Hus

Introduksjon

I denne leksjonen vil vi se litt på hvordan vi kan få en robot til å bygge
bruke løkker og funksjoner for å gjenta ting som gjøres flere ganger.



Steg 1: Prøv selv først

Vi har tidligere lært om hvordan roboter behøver fuel for å kjøre og by
nok fuel mens du gjør oppgavene under.

Tidligere har vi sett hvordan vi bruker `turtle`-biblioteket for å få robo
kommandoene

`turtle.forward()`, `turtle.back()`, `turtle.turnLeft()`, `turtle.tur`

for å flytte roboten. Videre kan vi bruke `turtle.place()` eller `turtle`
at vi kan bruke `for`-løkker for å gjenta ting.

Sjekkliste

- [ ] Lag en robot, gi den fuel og legg en del byggemateriell i roboter
- [ ] Lag et nytt program, `edit byggmitthus`, og skriv den følgende

```
for i = 1, 5 do
    turtle.back()
    turtle.place()
end
```

Kjør programmet. Bygger roboten en liten vegg av klosser?

Prøv selv

Jobb videre med `bygghus`-programmet, og se om du klarer å utvide
vegger og tak. Bruk litt tid på denne oppgaven slik at du får tenkt g

Et par små tips:

- [ ] Det er nok å bruke kommandoene vi har nevnt så langt.

- [ ] Du vil helst bruke flere for-løkker. For at dette skal virke må d
bytte ut `i` men andre bokstaver eller variabelnavn i de andr

- [ ] For å lage dør eller vinduer i huset ditt vil det enkleste være
roboten bygger.

Sjekkliste

- [ ] Når du har laget et enkelt hus så vis det frem til de andre. Hvilk

  Ingen skriver programmer riktig første gangen, og det er veldig
  utfordring, og hvilke metoder som ikke virker.

I de neste stegene vil vi utvikle et mer avansert program for husbygg
kommandoer og konsepter som gjør at programmet blir både kraftige

Steg 2: Flyvende robot

Et problem du kanskje allerede har oppdaget er at roboten innimellom
enn du hadde tenkt. Et eksempel på dette er i den følgende koden, hv

Sjekkliste

- [ ] Skriv programmet `grunnmur` med koden

```
for i = 1, 4 do
    for j = 1, 5 do
        turtle.back()
        turtle.place()
```

```
    end
    turtle.turnLeft()
end
```

Lagre og kjør programmet? Ser du problemet?

På slutten av byggingen av grunnmuren krasjer roboten inn i mu
siste klossen. En smart måte å unngå dette problemet på er å la
under seg med `turtle.placeDown()`. Dette har den ekstra ford
bakover, som jo er mer logisk.

Endre programmet som følger:

```
turtle.up()                                    -- ny linj
for j = 1, 4 do
    for i = 1, 5 do
        turtle.forward()                       -- endret
        turtle.placeDown()                     -- endret
    end
    turtle.turnLeft()
end
```

Når du kjører programmet nå klarer roboten å bygge hele firkan
resten av denne leksjonen.

Steg 3: Bygg en vegg

Når vi skal skrive større programmer (som for eksempel et som bygge
deloppgaver som er relativt enkle. Da kan vi heller kode disse delopp
programmet.

En naturlig deloppgave når vi skal bygge et hus er å bygge en vegg. L
robotprogrammet ditt virker etterhvert som du skriver det inn.

Sjekkliste

Lag et nytt program `bygghus`. Vi begynner med en enkel stripe:

```
turtle.up()
for i = 1, 5 do
    turtle.placeDown()
    turtle.forward()
end
```

For å bygge en vegg vil vi bygge flere slike striper på toppen av
roboten rygge tilbake med `turtle.back()` for å bygge neste str

```
for j = 1, 3 do                                    -- ny linj
    turtle.up()
    for i = 1, 5 do
        turtle.placeDown()
        turtle.forward()
    end

    for i = 1, 5 do                                -- ny linj
        turtle.back()                              -- ny linj
    end                                            -- ny linj
end                                                -- ny linj
```

Før vi fortsetter med programmet vårt vil vi også introdusere et
betyr. Dette vil også gjøre det enklere for oss å endre størrelsen

```
local hoyde = 3                                    -- ny linj
local lengde = 5                                   -- ny linj

for j = 1, hoyde do                                -- endret
    turtle.up()
    for i = 1, lengde do                           -- endret
        turtle.placeDown()
        turtle.forward()
    end
```

```
    for i = 1, lengde do                    -- endret
        turtle.back()
    end
end
```

Legg merke til at vi skriver `hoyde` med `o` og ikke med `ø`. Sider
kan ikke variabler ha navn som inneholder de norske bokstaven

Prøv selv

Prøv å endre verdiene av variablene `hoyde` og `lengde`. Gjør robote

Steg 4: Funksjoner

Vi har nå lært roboten hvordan den lager en vegg. For at vi enkelt ska
funksjon. I praksis betyr det at vi lærer roboten en ny kommando, son
innebygde kommandoene (som for eksempel `turtle.forward()`).

Sjekkliste

- [ ] Vi definerer en funksjon ved hjelp av den innebygde kommando

```
function byggVegg()                         -- ny linj
    local hoyde = 3
    local lengde = 5

    for j = 1, hoyde do
        turtle.up()
        for i = 1, lengde do
            turtle.placeDown()
            turtle.forward()
```

```
        end

        for i = 1, lengde do
            turtle.back()
        end
    end
end                                                    -- ny linj
```

Om du kjører programmet slik det er nå vil du se at roboten ikke
funksjonen, det vil si vi har fortalt roboten hvordan den kan byg
skal bygge veggen. Legg til den følgende linjen helt nederst i pr

```
byggVegg()
```

Nå sier vi at roboten også skal bygge veggen.

En veldig nyttig ting med funksjoner er at vi kan la variablene v
bestemmer verdien på variablene utenfor selve funksjonen. End
på at du sletter de to linjene som pleide å gi verdi til `hoyde` og

```
function byggVegg(hoyde, lengde)                    -- endret
    for j = 1, hoyde do
        turtle.up()
        for i = 1, lengde do
            turtle.placeDown()
            turtle.forward()
        end

        for i = 1, lengde do
            turtle.back()
        end
    end
end

byggVegg(3, 5)                                      -- endret
```

Prøv selv

Endre tallene `3` og `5` i den siste linjen. Bygger roboten vegger av

Steg 5: Bygg et hus

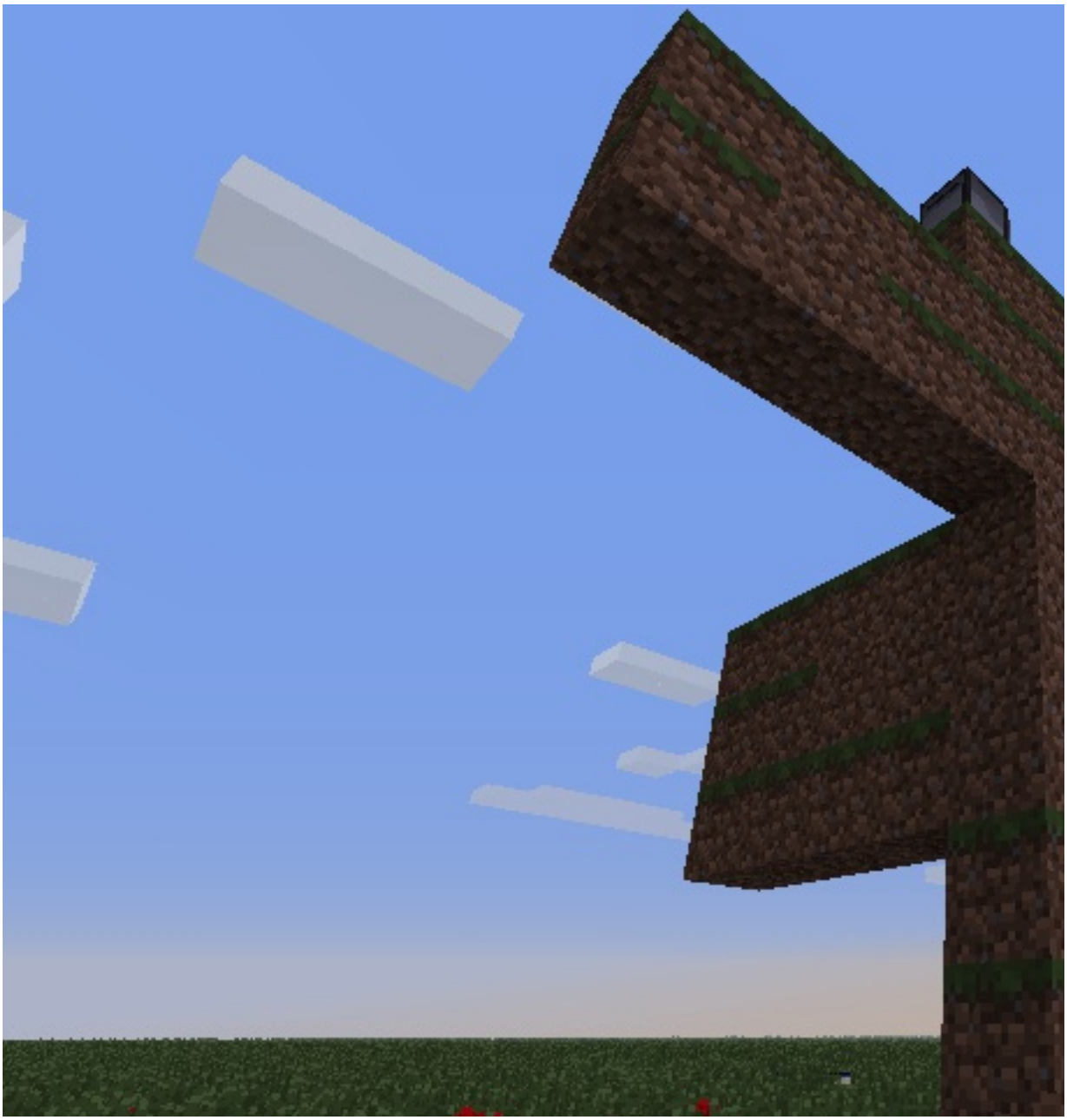
Nå som vi vet hvordan vi bygger en vegg er vi ikke veldig langt unna fire vegger!

Sjekkliste

- [ ] Vi begynner enkelt. La oss prøve å bygge en vegg, snu roboten, linjen `byggVegg(3, 5)` i programmet ditt med den følgende løkk

```
for i = 1, 4 do
    byggVegg(3, 5)
    turtle.turnLeft()
end
```

Hva skjer når du kjører programmet ditt?



Hmm ... det ble jo litt stilig, men et hus er det definitivt ikke. Pro
etter at den er ferdig med å bygge en vegg.

Vi burde legge til litt kode i `byggVegg()` slik at roboten er klar ti
nedover i stedet for bakover etter at vi har bygd ferdig veggen.

```
function byggVegg(hoyde, lengde)
    for j = 1, hoyde do
        turtle.up()
        for i = 1, lengde do
            turtle.placeDown()
```

```
            turtle.forward()
        end

        if j < hoyde then                          -- ny linj
            for i = 1, lengde do
                turtle.back()
            end
        end                                        -- ny linj
    end

    for j = 1, hoyde do                            -- ny linj
        turtle.down()                              -- ny linj
    end                                            -- ny linj
end
```

Programmet vårt klarer nesten å bygge et hus av fire vegger. M
inn i ting den har bygd tidligere. Vi må rett og slett være litt me
trenger.

```
function byggVegg(hoyde, lengde)
    for j = 1, hoyde do
        turtle.up()
        for i = 1, lengde do
            turtle.placeDown()
            if i < lengde then                     -- ny linj
                turtle.forward()
            end                                    -- ny linj
        end

        if j < hoyde then
            for i = 1, lengde - 1 do               -- endret
                turtle.back()
            end
        end
    end

    turtle.forward()                               -- ny linj
    for j = 1, hoyde do
```

```
        turtle.down()
    end
end
```

Nå har huset vårt fått fire vegger. Vi skal snart se på hvordan vi kan b
passe på at roboten har materiale å bygge med.

Steg 6: Mer byggemateria

Et problem du helt sikkert har oppdaget nå, er at roboten stadig går t
for materiale, vil den bare bruke en av dem. Ved hjelp av funksjonene
`turtle`-biblioteket kan vi gjøre noe med dette.

Sjekkliste

- [ ] Vi vil nå lage en ny funksjon som sjekker at det er materiale tilg
byttet ut `turtle.placeDown()` med et kall på vår egen funksjon

```
function byggVegg(hoyde, lengde)
    for j = 1, hoyde do
        turtle.up()
        for i = 1, lengde do
            plasser()                          -- endret
            if i < lengde then
                turtle.forward()
            end
        end

        if j < hoyde then
            for i = 1, lengde - 1 do
                turtle.back()
            end
        end
    end
```

```
    turtle.forward()
    for j = 1, hoyde do
        turtle.down()
    end
end

function plasser()                                  -- ny linj
    while turtle.getItemCount() == 0 do             -- ny linj
        slot = turtle.getSelectedSlot()             -- ny linj
        if slot < 16 then                           -- ny linj
            turtle.select(slot + 1)                 -- ny linj
        else                                        -- ny linj
            turtle.select(1)                        -- ny linj
        end                                         -- ny linj
    end                                             -- ny linj

    turtle.placeDown()                              -- ny linj
end                                                 -- ny linj

for i = 1, 4 do
    byggVegg(3, 5)
    turtle.turnLeft()
end
```

Les nøye gjennom den nye funksjonen `plasser()`. Skjønner du

Det siste funksjonen gjør er å plassere ut en kloss med `turtle.` vi `turtle.getItemCount()` til å sjekke om det finnes tilgjengelig bruker, går vi inn i `while`-løkken hvor vi velger en annen slot. V `turtle.getSelectedSlot()`. Hvis dette ikke er den siste sloten neste sloten. Hvis vi allerede er på den siste sloten velger vi hel

- [ ] Kjør programmet. Hva skjer når roboten går helt tom for bygger skal fylle den opp med nytt materiale.

Prøv selv

Nå som roboten bruker flere slotter kan du eksperimentere med å l
På den måten kan du for eksempel få roboten til å bygge et fargeri

Steg 7: Vi trenger et tak!

Nå er det på tide å lære roboten hvordan den bygger tak på huset vår

Sjekkliste

- [ ] Før vi begynner å bygge taket vil vi flytte koden som bygger hu
med `byggVegg()` tidligere. Bytt den nederste løkken i koden dir
følgende koden:

```
function byggHus(hoyde, bredde, dybde)
    byggVegg(hoyde, bredde - 1)
    turtle.turnLeft()
    byggVegg(hoyde, dybde - 1)
    turtle.turnLeft()
    byggVegg(hoyde, bredde - 1)
    turtle.turnLeft()
    byggVegg(hoyde, dybde - 1)
    turtle.turnLeft()
end

byggHus(3, 5, 4)
```

Ser du hvorfor vi bruker `bredde - 1` i stedet for `bredde`? Sjekk

- [ ] Nå kan vi lage en ny funksjon `byggTak()`. Legg først til denne li

```
byggTak(bredde, dybde)
```

- [ ] Selve funksjonen for å bygge taket kan være ganske lik funksjo
står oppover, vil taket ligge flatt. Legg til funksjonen `byggTak` i

```
function byggTak(bredde, dybde)
    turtle.up()
    for j = 1, dybde do
        for i = 1, bredde do
            plasser()
```

```
            turtle.forward()
        end

        for i = 1, bredde do
            turtle.back()
        end
        turtle.turnLeft()
        turtle.forward()
        turtle.turnRight()
    end
end
```

Steg 8: Dører og vinduer

Da er vi nesten ferdig med programmet vårt. Det som mangler på at

Sjekkliste

- [ ] Vi velger en relativt enkel løsning denne gangen, og lar roboten være. Siden vi ikke vil ha dører på alle veggene, og vil ha litt mu nye parametre til `byggVegg()`-funksjonen vår. Endre denne funk

```
function byggVegg(hoyde, lengde, dor, vindu)     -- endre
    for j = 1, hoyde do
        turtle.up()
        for i = 1, lengde do
            if not (j <= 2 and i == dor or       -- ny li
                    j == 2 and i == vindu) then  -- ny li
                plasser()
            end                                  -- ny li
            if i < lengde then
                turtle.forward()
            end
        end
```

```
        if j < hoyde then
            for i = 1, lengde - 1 do
                turtle.back()
            end
        end
    end

    turtle.forward()
        for j = 1, hoyde do
            turtle.down()
    end
end
```

Med den nye `if`-testen lager vi en dør ved å ikke bygge de to r

- [ ] Vi må til slutt også endre koden i `byggHus()` som kaller `byggVe`

```
function byggHus(hoyde, bredde, dybde)
    byggVegg(hoyde, bredde - 1, 2, nil)
    turtle.turnLeft()
    byggVegg(hoyde, dybde - 1, nil, math.floor(dybde / 2
    turtle.turnLeft()
    byggVegg(hoyde, bredde - 1, nil, bredde - 2)
    turtle.turnLeft()
    byggVegg(hoyde, dybde - 1, nil, 2)
    turtle.turnLeft()

    byggTak(bredde, dybde)
end
```

Legg merke til at vi bruker det spesielle ordet `nil` hvis vi ikke s
du hvor vinduene på de forskjellige veggene plasseres?

Steg 9: Forskjellige hus

Vi har nå laget et program som gjør at roboten vår er en husbyggerob
oss helt ferdige!

Nå har vi kodet hvor stort huset skal være inn i den siste linjen i progr
bygge et større eller mindre hus, må vi forandre på programmet. Dett
hvor stort huset skal være når vi starter programmet.

Sjekkliste

- [ ] Bytt ut linjen `byggHus(3, 5, 4)` med det følgende:

```
local tArgs = { ... }
if #tArgs ~= 3 then
    print('Skriv: bygghus <høyde> <bredde> <dybde>')
    print('F.eks. bygghus 3 5 4')
    return
end

byggHus(tonumber(tArgs[1]), tonumber(tArgs[2]),
        tonumber(tArgs[3]))
```

  Dette kan virke litt mystisk, og vi skal ikke forklare alt som skjer
  leksjoner.

- [ ] Om du prøver å kjøre programmet ditt ved å bare skrive `bygghu`
  høyde, bredde og dybde. Prøv for eksempel å skrive `bygghus 3`