



# Skilpadder

## Introduksjon:

I denne modulen skal vi lære et programmeringsspråk som heter Python. Python er favorittprogrammet på TV: Monthly Pythons Flyvende Cirkus. Python brukes til mange forskjellige ting. Python brukes av YouTube, NASA, CERN og andre. Hvorfor Python til å programmere den? Mange elsker Python fordi de synes det er vanskeligere å lese). Her er meningene mange, når du lærer flere språk. Å være i stand til å lese kode er viktig for en programmerer, kanskje like

## Steg 1: Hei, Skilpadde!

Nå skal vi ha det litt gøy med skilpadder. En skilpadde er en liten robot som kan bevege seg rundt med Python-kommandoer.

### Sjekkliste

- ☐ Åpne et nytt kodevindu og skriv dette:

```
from turtle import *  
  
forward(100)
```

- ☐ Lagre programmet ditt som myturtle.py og velg **Run -> Run Module** punkter fremover på skjermen? Skilpadden har en penn festet til

## Tips

Python-filer skal alltid ha filnavn som slutter med `.py`.

## ✓ Sjekkliste

- ☐ La oss få skilpadden til å bevege seg rundt på skjermen! Forsøk ved å bruke `right(angle)` og `left(angle)`. Instruksjonen `back` skal bevege seg bakover 20 pixler, og `right(90)` forteller den å mer enn en instruksjon om gangen, de blir utført i rekkefølge.

```
from turtle import *

speed(11)
shape("turtle")

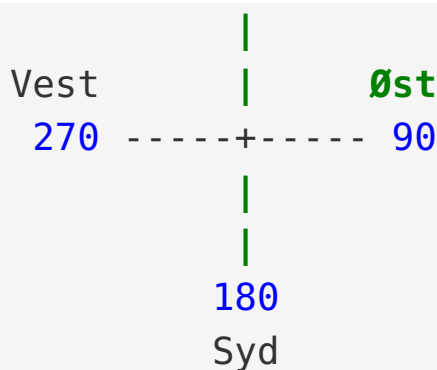
forward(100)
right(120)
forward(100)
left(90)
backward(100)
left(90)
forward(50)
```

## Vinkler og grader

Lek deg litt med å lage dine egne figurer ved å bruke `forward`, `backward` bruker pixler, mens `left` og `right` bruker grader. La oss undersøke

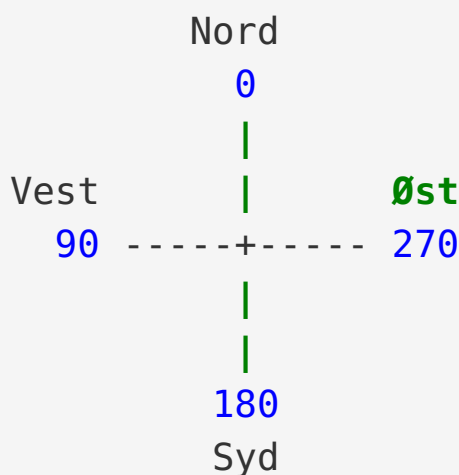
Nord

0



Når skilpadden ser nordover og du ber den snu seg 90 grader til høyre ser den sydover, og snur du den 270 grader fra nord ser den vestover. Kanskje er det lettere å tenke på som snowboard-triks?

Hva med å snu mot venstre?



Når skilpadden ser nordover og du ber den snu seg 90 grader mot venstre ser den østover, og om du ber den snu seg 180 grader mot venstre ser den sydover, og om du ber den snu seg 270 grader mot venstre ser den vestover. Snur du 360 grader er du tilbake der du starter, 360 grader.

## Hva gjør koden på starten av programmet?

- ☐ `from turtle import *` forteller Python at vi vil bruke skilpadden for å tegne på skjermen. Å bruke et ferdig bibliotek gjør det lettere.
- ☐ `speed()` bestemmer farten til skilpadden. Vi må gi inn en verdi for å sette farten til det treigeste.
- ☐ `shape()` Vi bruker formen(shape) "turtle" (skilpadde), men vi kan også bruke andre former som "square", "circle", etc.

"arrow" (pil), "circle" (sirkel), "square" (kvadrat), "triangle" (t

Vi kommer til å bruke disse instruksjonene på toppen av alle programmer. Vi forsøke å gi skilpadden en av de andre formene, som pil, og få den

## Steg 2: Tegne figurer!

La oss lage et kvadrat ved å fortelle skilpadden hvordan den skal bevege seg.

### ✓ Sjekkliste

- ☐ Åpne en ny fil i IDLE og skriv inn følgende kode:

```
from turtle import *\n\nspeed(11)\nshape("turtle")\n\nforward(100)\nright(90)\nforward(100)\nright(90)\nforward(100)\nright(90)\nforward(100)\nright(90)
```

Lagre programmet ditt og velg **Run -> Run Module**. Ser du en firkant? Den har fire hjørner, så vi må snu tre ganger. Den snur 90 grader hver gang, så den snur totalt 360 grader.

Hva med en trekant? En trekant har tre hjørner, så vi må snu tre ganger. Den snur 120 grader hver gang, så den snur totalt 360 grader, akkurat som med firkanten. Derfor snur vi 120 grader.



Endre koden din til å se ut som koden under for å få den til å tegne

```
from turtle import *  
  
speed(11)  
shape("turtle")  
  
forward(100)  
right(120)  
forward(100)  
right(120)  
forward(100)  
right(120)
```



Kjør koden. Ser du en trekant?

## Velg en farge

Hva er yndlingsfargen din? Du kan endre fargen på linjene ved å bruke `color` (i stedet for `colour` som amerikanerne staver `colour` uten u.). Du kan også endre størrelsen



## Sjekkliste



Endre koden fra eksemplet over til å se ut som det neste eksemplet

```
from turtle import *  
  
speed(11)  
shape("turtle")  
  
pensize(10)  
pencolor("red")  
forward(100)  
right(120)
```

```
pencolor("blue")
forward(100)
right(120)
pencolor("green")
forward(100)
right(120)
```

- ☐ Kjør koden din, hva tegner den på skjermen? Denne koden tegn
- ☐ Forsøk å endre fargene i koden din, kjør den og se hva som skje ikke bare blå, rød og grønn. Forsøk med din yndlingsfarge! Du kan med CSS før. Istedenfor å bruke `pencolor("red")` kan du bruke

## Steg 3: Gjenta deg selv (m

Det siste programmet var de samme kommandoene igjen og igjen. Ist gjenta dem for oss. Du har vært borti *iterasjon* i Scratch ved å bruke brukes **for-løkker** når du har kode som du vil gjenta n ganger. I dette 4 ganger fordi en firkant har 4 sider.

### Sjekkliste

- ☐ Åpne en ny fil og skriv inn følgende:

```
from turtle import *

speed(11)
shape("turtle")

for count in range(4):
    forward(100)
    right(90)
```

- ☐ Lagre programmet og velg: Run -> Run module.

Legg merke til at koden er skjøvet inn, *indentert*, eller dyttet til vite hvilke kommandoer som skal gjentas. Du kan bruke Tab-tas å ta dem bort.

- ☐ La oss se hva som skjer hvis vi bare indenterer (skyver inn) `for` dette:

```
from turtle import *  
  
speed(11)  
shape("turtle")  
  
for count in range(4):  
    forward(100)  
    right(90)
```

- ☐ Legg merke til at `forward` er indendert og `right` ikke er det. H og finn det ut.

Fikk du en rett linje? Python vil gjenta `forward` fire ganger, og c gruppere kommandoer sammen, akkurat som Scratch bruker bl mellomrommene riktig.

- ☐ La oss endre programmet tilbake slik at det lager en firkant igje navn. Dette gjør det lettere å se hva programmet gjør, og gjør a Endre filen så den ser slik ut:

```
from turtle import *  
  
speed(11)  
shape("turtle")
```

```
sides = 4
length = 100
angle = 90
for count in range(sides):
    forward(length)
    right(angle)
```

☐ Lagre programmet og velg: `Run -> Run module`.

## Oppgave: Tegn de andre former

Kan du tegne noen av figurene under bare ved å endre verdiene?

- ☐ En trekant? (tre sider)
- ☐ Et pentagram? (fem sider)
- ☐ Et hexagram? (seks sider)
- ☐ Et oktagram? (åtte sider)

Husk, en trekant har tre sider og vi må derfor snu 120 grader i hver tilsammen. For en firkant må vi snu 90 grader i hvert hjørne, som c

Hvis du snur seks ganger, hvor mange ganger må du snu for at de se hva som skjer.

## Steg 4: Snu, snu, snu

Istedenfor å regne ut vinklene, kan vi ikke heller få datamaskinen til å gange og dele. Vi kan skrive `sides = 4 + 1` istedenfor 5, eller `sides` Python `*`, og for divisjon skriver vi `/`. Hvis vi må snu 360 grader tilsa



firkant er  $360 / 4$  lik 90, for trekanten er  $360 / 3$  lik 120.

## Sjekkliste

- ☐ Endre programmet ditt til å regne ut vinkelen.

```
from turtle import *

speed(11)
shape("turtle")

sides = 4
length = 20

angle = 360/sides
for count in range(sides):
    forward(length)
    right(angle)
```

- ☐ Nå kan du endre antall sider, klarer Python å gjøre jobben riktig

## Steg 5: Fylte figurer

## Sjekkliste

- ☐ Vi kan be skilpadden om å fylle figurene med en farge ved å bruke disse kommandoene:

```
from turtle import *

speed(11)
```

```
shape("turtle")

sides = 4
length = 20

fillcolor('red')
pencolor('red')
begin_fill()

angle = 360/sides
for count in range(sides):
    forward(length)
    right(angle)
end_fill()
```

Akkurat som med `pencolor` velger `fillcolor` fargen skilpadde koden tegner en rød firkant med en rød strek rundt.

Du kan bruke `begin_fill()` for å fortelle skilpadden at den skal si at du er ferdig.

- ☐ Forsøk å endre fargene, sidene og lengdene og se hvilke figurer

## Steg 6: Pennen går opp, p

Hvis du vil flytte skilpadden uten at den skal sette spor etter seg, kan skilpadden skal tegne.

### Sjekkliste

- ☐ Forsøk dette i en ny fil:

```
from turtle import *
```

```
speed(11)
shape("turtle")

pencolor('red')

for count in range(20):
    penup()
    forward(10)
    pendown()
    forward(20)
```



Dette burde tegne en stiplet strek over skjermen din. Kjør det og

## Hjem, kjære hjem på skjermen

Et par triks på slutten: `home()` får skilpadden til å gå hjem dit den og `reset()` flytter skilpadden og renser opp skjermen.

# Steg 7: Gjør hva du vil!

Du kan `forward()`, `backward()`, `left()`, `right()`, du kan gjenta ting og til og med fylle figurer!

Kan du tegne et hus, en fugl? En slange? En katt? En hund? En løve? I du tegne en robot?

**Lisens:** [Code Club World Limited Terms of Service](#) **Forfatter:** O  
**Øversetter:** Bjørn Einar Bjartnes