Skilpaddeskolen

Steg 1: Flere firkanter

Sjekkliste

- [ ] Åpne IDLE-editoren, og åpne en ny fil ved å trykke `File > New`

  Husk at du skal ha to vinduer åpne. Det ene er 'Python Shell' og

  Som sist vil den første linjen alltid være `from turtle import *`
  skilpadde-biblioteket!

  ```
  from turtle import *

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

- [ ] Lagre det som en ny fil, og kjør programmet fra menyen ved å t

  Husk at `for n in range(4)` gjentar koden, og at koden må gru
  for-løkken. Bruk *tab* (knappen rett over *caps lock*) for å flytte ko

Steg 2: Forskjellige firkant

La oss bruke variabler for å gjøre programmet vårt lettere å lese og le

Sjekkliste

- [ ] Endre programmet så det ser slik ut:

```
from turtle import *

sides = 4
length = 100
angle = 360/sides

for n in range(sides):
    forward(length)
    right(angle)
```

- [ ] Kjør det ved å trykke `Run > Run Module` fra menyen. Får du der
går videre.

Dette er et litt langt program, men nå kan vi endre det til å tegne hvil
klippe og lime programmet for å få det til. Som tidligere kan vi skrive
anstrenger seg gjerne litt slik at de kan være late etterpå!). Denne ga
måte for å gjenbruke en kodeblokk (eller oppskrift om du vil) mange g
senere bruke.

Steg 3: Vi lager en funksjo

Sjekkliste

- [ ] Vi endrer koden og legger til `def poly():`. `def` betyr definer, a
den nye funksjonen. For å få innrykk på flere linjer kan man me

caps lock). Dersom du vil ha mindre innrykk, bruk *shift + tab*.

```
from turtle import *

def poly(): # vi lager funksjonen
    sides = 4
    length = 100
    angle = 360/sides

    for n in range(sides):
        forward(length)
        right(angle)

pencolor('red')
poly() # vi kaller på funksjonen
right(180)
poly()
```

- [ ] Kjør programmet. Hvis det virker skal to røde firkanter bli tegne

Vi sparte litt tid ved å lage en ny funksjon i Python, og nå kan vi tegne ganger. Den nye funksjonen `poly()` er fin for å slippe å skrive så mye

Steg 4: Hvorfor stoppe me

Vi er ikke ferdige ennå - hva med å endre funksjonen så den kan tegn `right`, kan vi sende verdier inn i funksjonen istedenfor å endre koder

Sjekkliste

- [ ] Endre koden så den ser slik ut:

```
from turtle import *
```

```
def poly(sides, length):
    angle = 360/sides

    for n in range(sides):
        forward(length)
        right(angle)

pencolor('red')
poly(4, 100)
right(180)
pencolor('blue')
poly(3, 150)
```

- [ ] Kjør den og se hva som skjer.

  La oss ta dette litt sakte, for dette er ganske kule greier. Isteder
  funksjonen tar noen verdier som har navn, og så bruker vi verdi

  Vi flyttet noen verdier ut av funksjonen, og flyttet dem til den de
  eneste funksjon, tegne *hvilken som helst* form, med *hvilken son*
  imponerer meg hver gang jeg tenker på det: Vi kan lære datam

Å være i stand til å lage nye funksjoner som kan oppføre seg forskjelli
verktøyene i programmering.

Tips

I python finnes det funksjoner, mens i andre programmeringsspråk
begrepene går litt inn i hverandre, så det er ikke så farlig om dere
returnere noe, og den skal helst ikke gjøre noe annet enn å regne u
returnere det samme når den får samme innputt. Prosedyrer ligner
forskjellige ting avhenging av andre ting enn innputt. I tillegg er de
returnere noe. For eksempel kan en prosedyre tegne på skjermen.
prosedyrer, så det er vanlig å bare kalle begge deler for funksjoner

Steg 5: Skilpaddestreker

Sjekkliste

- [ ] Selv om skilpadden er en liten robot som kan tegne, kan den og
`penup()` og `pendown()` for å slå av og på at skilpadden setter s

```
from turtle import *

length = 200
for num in range(8):
    forward(length/16)
    penup()
    forward(length/16)
    pendown()
```

- [ ] Dette programmet tegner en stiplet linje over skjermen. Kjør de

Steg 6: Tegne figurer

Vi kan koble figur-programmet og stiplet-linje-programmet sammen v
stiplete linjer. Vi bruker koden for å tegne figurer ytterst, og inni der b
streker.

Sjekkliste

- [ ] Endre koden så den ser ut som følgende:

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

def dashpoly(sides, length):
    angle = 360/sides

    for n in range(sides):
        for num in range(8):
            forward(length/16)
            penup()
            forward(length/16)
            pendown()
        right(angle)

pencolor('red')
dashpoly(4, 100)
right(180)
pencolor('blue')
dashpoly(3, 150)
```

- [ ] Kjør koden og se hva den gjør.

Vi har to for-løkker inni hverandre, en ytre og en indre. Den ytre
figuren, og hver gang kjører den indre løkken `for num in range`

Den ytre løkken bruker variabelen `n` for å holde styr på hvor ma
løkken bruker variabelen `num` på tilsvarende måte. Du må bruke
det bare til.

Steg 7: Byggeklosser for fi

Sjekkliste

- [ ] La oss bruke funksjoner igjen for å rydde opp i koden. Endre koc

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

def dashforward(length):
    for num in range(8):
        forward(length/16)
        penup()
        forward(length/16)
        pendown()

def dashpoly(sides, length):
    angle = 360/sides

    for n in range(sides):
        dashforward(length)
        right(angle)

pencolor('red')
dashpoly(4, 100)
right(180)
pencolor('blue')
dashpoly(3, 150)
```

- [ ] Kjør koden og se at den gjør akkurat det samme som før.

Tips

Trikset er at istedenfor å bygge programmer ved å klippe og lime, blir koden kortere og litt lettere å forstå.

Steg 8: Litt tilfeldigheter

Hva om vi gjør litt tilfeldige sprell rett før vi er ferdige? Vi kan be data
oss, litt som om vi kaster terning. Scratch kan dette også, da brukte v

Sjekkliste

- [ ] I en ny fil, skriv inn følgende:

```
from turtle import *
from random import randrange, choice
colors = ['red', 'blue', 'green']


def poly(sides, length):
    angle = 360/sides

    for n in range(sides):
        forward(length)
        right(angle)

for count in range(10):
    pencolor(choice(colors))
    right(randrange(0,360))
    poly(randrange(3,9), randrange(10,30))
```

- [ ] Lagre og kjør koden

Programmet skal tegne ti figurer i forskjellige farger med forskje
`random` henter inn funksjonene `randrange()` og `choice()`.

`randrange()` plukker ut et tall mellom det laveste og det høyes
mellom 1 og 9 (Python begynner med 1, og stopper rett før 10).

`choice()` velger en ting fra listen vi gir inn. En liste er en samli

ovenfor bruker vi listen `colors`, som har verdiene `'red'`, `'blu`

Ved å bruke `choice()` og `randrange()` kan vi be datamaskiner
kommer til å bli forskjellig resultat nesten hver eneste gang du

Flere ting å prøve

Hva med å prøve flere farger, eller å endre tallene? Hva skjer?