

Oppgavesett til TENK Tech Camp

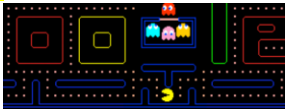
Hva er python?

1. Et av de mest menneskespråk-lignende (engelsk-lignende) programmeringsspråkene som finnes. Python er oppkalt etter Monty Python's Flying Circus som var et populært humorprogram for lenge siden, så mye eksempler og slikt i språket var referanser til det programmet i starten.

Hva brukes python til?

2. Tensor flow (#statistikk)
<https://www.tensorflow.org/js/demos>

3. Pacman:



<https://storage.googleapis.com/tfjs-examples/webcam-transfer-learning/dist/index.html>

4. Piano:



https://magenta.tensorflow.org/demos/performance_rnn/index.html#2|2,0,1,0,1,1,0,1,0,1,0,1|1,1,1,1,1,1,1,1,1,1,1,1|1,1,1,1,1,1,1,1,1,1|false

5. Så har vi også Netflix, Instagram, Google, Spotify, Uber, Dropbox, Pinterest, Reddit osv. som bruker Python.

I dag skal vi lage et spill for å få en skilpadde fra start til slutt på færrest mulig terningkast.

Oppgave 1: Tegn en skilpadde

Vi bruker Turtle, et bibliotek i Python. Det første vi trenger er en skilpadde.

Et bibliotek er en utvidelsespakke til programmeringsspråket. På en måte kan vi si at et programmeringsspråk er som et verktøyskrin med hammer og sag. Du får gjort mye, men av og til trenger du f.eks et skrujern for å gjøre noen flere greier, og turtle-biblioteket er skrujernet i vårt tiffel. I Turtle så skriver man bare `turtle.Turtle()` for å lage ny og 90-talls retro-looking skilpadde.

1. Åpne lenken repl.it i nettleseren din
2. Her har de laget en fil **main.py**. At filen slutter på **.py** betyr at dette er python-kode.
3. I main.py skriv følgende og trykk "Run"

```
main.py
1  import turtle
2
3  skilpadde = turtle.Turtle()
4
```

4. Vi har fremdeles ikke en skilpadde, så nå skal vi sørge for at "greia" vi lagde får en ny look. Skriv inn `skilpadde.shape("turtle")` nederst i main.py og kjør koden.

```
main.py
1  import turtle
2
3  skilpadde = turtle.Turtle()
4
5  skilpadde.shape("turtle")
~
```

Det vi har gjort nå er å hente "biblioteket" Turtle. Så har vi opprettet en "variabel" for en Skilpadde og gitt den et fint utseende.

En variabel er et navn, eller en slags merkelapp. Når man har merkelappen går det an å flytte den mellom ulike ting, eller å gjøre noe med den tingen som merkelappen sitter på. Merkelappen min i virkeligheten, som kanskje er navnet og personnummeret mitt, kan en ikke bare flytte videre til for eksempel Bendik, men det er jo bare fordi noen fra gamle dager har bestemt at det ikke er lov. Tenk så gøy om man bare kunne bytte navn med hverandre!

Men, her har vi et eksempel: i kodesnutten under lager vi en person to ganger, med ulike merkelapper. Så skriver vi `"Kjersti.vink()"`, og da vil kun Kjersti vinke, siden vi bare bruker variabelnavnet hennes foran `vink()`.

```
Kjersti = person()
Bendik = person()
Kjersti.vink()
```

Bonus:

1. Du kan skifte farge på skilpadden. Skriv:
`skilpadde.color("lightgreen", "green")`
2. Eller du kan skifte til en annen form ved å bytte ut innholdet i `shape()`.
 - i. `square`
 - ii. `arrow`
 - iii. `circle`
 - iv. `turtle`
 - v. `triangle`
 - vi. `classic`

Hva skjer hvis du endrer til en annen form uten å fjerne `skilpadde.shape("turtle")`?

Oppgave 2: Lag en terning

Et terningkast kan gi en verdi mellom 1 og 6. En måte å gjøre dette i kode er å først lage en liste. Lister brukes mye i programmering og skrives som oftest med klammeparantes eller krøllparantes. I Python brukes klammeparantes. Et eksempel på en liste i python er:

```
liste = ["eple", "appelsin"]
```

A) Lag en variabel terning som er en liste med tallene på terningen.

Bonusoppgave:

Noen terninger har flere sider. Lag flere sider på terningen din!



Oppgave 3A: Kast terningen

Å kaste en terning i Python betyr å velge et tilfeldig tall fra terningen vi lagde. For å gjøre dette bruker vi et nytt bibliotek "random".

Du har kanskje lagt merke til at vi av og til skriver et ord og med to paranteser bak f.eks: `skilpadde.color()`. `color()` er en funksjon som vi bruker for å fargelegge skilpadden vår. Hvis vi tenker på et bibliotek som et skrujern igjen, så har skrujernet noen funksjoner vi kan bruke. Står det noe inne i parantesen, så brukes det inne i funksjonen. For eksempel `skrujern.skru_inn(skru)`.

Du kan nå lage en terning og så vise tallet på terningen. Prøv:

A) Importer biblioteket `random` i `main.py`

```
main.py
1 import turtle
2 import random
~
```

B) Kast terningen ved å bruke funksjonen `random.choice(terning)` og lagre det i en variabel som du kaller `terningkast`

C) Du kan skrive til vinduet ved å bruke Python sin *print*-funksjon slik: *print(terningkast)*. *print()* brukes for å skrive ut tekst på skjermen. Slik snakker alle programmerere sammen når de ber hverandre ut på dater osv: *print("jeg liker deg, vil du holde hender?")*

Skriv ut hvilket terningkast det ble med *print()*.

```
8  liste = [1,2,3,4,5,6,7]
9  terningkast = random.choice(liste)
10 print(terningkast)
```

Bonus 1:

Random inneholder også en funksjon "*randrange*": *random.randrange(start, stop, step)*. Her tar funksjonen inn 3 tall som den bruker for å gi oss et tilfeldig tall. Start er det laveste tallet, stop det høyeste, og step er da mellomrommet mellom alternativene.

Prøv å løse oppgave 2 og 3 med den funksjonen i stedet.

Bonus 2:

Prøv nå å lage en terning med mange sider, f.eks 100. Ser du hvor enkelt det er nå?

Oppgave 3B: Lag en terningkast-funksjon

Til nå har vi brukt funksjoner laget av andre. Nå skal vi lage vår egen funksjon for å slippe å skrive *random.choice(terning)* for hver gang vi skal kaste terningen. Vi lager funksjonen sammen.

```
liste = [1,2,3,4,5,6]
def kast_terning():
    terningkast = random.choice(liste)
    print("du kastet " + str(terningkast))
    return terningkast
```

Oppgave 4: Lag målstreken

Vi skal nå lage målstreken skilpadden må krysse for å vinne:

Vi lager en ny turtle-variabel som heter *målstrek*.

```
målstrek = turtle.Turtle()
målstrek.color("red")
målstrek.penup()
målstrek.goto(150,-300)
målstrek.pendown()
målstrek.goto(150,300)
målstrek.penup()
```

Oppgave 5: Flytt skilpadden til startposisjonen sin

For å slippe å starte *midt på brettet*, som er startposisjonen, så må vi flytte skilpadden til venstre. Skilpadden lager en strek mens den flytter seg, så vi må også "løfte" den ned *skilpadde.penup()* for å unngå å klusse til brettet.

A) Flytt skilpadden til (-150, 0).

B) X-koordinaten skal endres på når vi beveger skilpadden. Lag en variabel for x- koordinaten og bruk den i goto-funksjonen.

Oppgave 6: Lag spillvariabler

I et spill må vi holde styr på antall poeng og antall runder. Vi lager to nye variabler.

```
poeng = 100
runde = 1
```

Oppgave 7: Lag en while-løkke

“While” betyr “så lenge”, og så lenge vi ikke har kommet i mål ennå, vil vi fortsette å kaste terning. Vi kaster terning en gang hver runde, og om vi ikke har klart det på hundre runder er spillet over.

```
while runde < 100:
    print("Runde " + str(runde) )
    runde += 1
    poeng -= 1
    print("Halla, Aurora!")
```

Oppgave 8: Lag en if-setning for å sjekke om vi har vunnet

If-setninger bruker vi programmerere heeele tiden.

Hvis du prøver å skyte noe i Fortnite, for eksempel, så kan du bare det hvis du har mer enn 0 ammo. Det kan skrives slik:

```
if ammo > 0:
    skyt()
    ammo -= 1
```

Da sjekker vi først om vi har ammo, så skyter vi og fjerner en ammo med “minus-erlik 1”.

Tilbake til spillet vårt: “hvis posisjonen til skilpadden er forbi målstreken på posisjon 150” skrives slik:

```
if skilpadde.pos() > (150,0):
    print("Du har vunnet")
    print("Du fikk " + str(poeng) + " poeng")
    break
```

... dette kan vi putte inn nederst i *while*-løkken fra oppgave 7.

Oppgave 9: Kast terning for å flytte skilpadden

Nå vil vi også bestemme hva som skjer når man ikke har vunnet ennå.

Dette er litt bakvendt, men det gir mening når vi har skrevet det hele ferdig.

“if” bruker vi til å si hva som skal skje hvis skilpadden har kommet forbi posisjon 150, og “else” kan brukes for å si hva som skal skje ellers. Formen på teksten er viktig her også, med riktig innrykk på alle linjene.

Løsningsforslag oppgave 9

```
while runde < 100:
    print("Runde: " + (str(runde)))
    runde += 1
    poeng -= 1

    if skilpadde.pos() > (150,0):
        print("Du har vunnet")
        print("Du fikk " + str(poeng) + " poeng")
        break
    else:
        kast = input("trykk enter for å kaste terning")
        terningkast = kast_terning()
        posX = posX + terningkast
        skilpadde.goto(posX, 0)
```

Oppgave 10: SPILL!!!

Nå er spillet ferdig, men skilpadden er jo skikkelig treg! Hvordan vil du endre på spillet for å få flere poeng eller få skilpadden til å bevege seg raskere?

Bonus:

- Prøv å skille ut flytting av skilpadden ut i en egen funksjon.

Løsningsforslag for hver oppgave

Løsningsforslag oppgave 1

```
import turtle
skilpadde = turtle.Turtle()
skilpadde.shape("turtle")
```

Bonus 1)

```
skilpadde.color("lightgreen", "green") #bonus 1)
```

Bonus 2)

Prinsippet med prosedyreorientert programmering vises hvis de først setter

```
Skilpadde.shape("turtle")
Skilpadde.shape("circle")
```

Koden kjøres fra topp til bunn. Først ser man at den får skilpaddeform og så skifter den raskt til sirkel.

Løsningsforslag oppgave 2

```
terning = [1,2,3,4,5,6]
```

bonus

```
terning = [1,2,3,4,5,6,7,8,9,10,11,12]
```

Løsningsforslag oppgave 3A

```
import random
terningkast = random.choice(terning)
print("du får flytte " + str(terningkast) + " steg")
```

bonus 1

```
import random
terningkast = random.randrange(1,6,1)
```

bonus 2

```
terningkast = random.randrange(1, 100, 1)
```

Løsningsforslag 3B

```
def kast_terning:
```



```
liste = [1,2,3,4,5,6]
terningkast = random.choice(liste)
print(terningkast)
return terningkast
```

Løsningsforslag oppgave 4

```
målstrek = turtle.Turtle()
målstrek.color("red")
målstrek.penup()
målstrek.goto(150,-300)
målstrek.pendown()
målstrek.goto(150,300)
målstrek.penup()
```

Løsningsforslag oppgave 5

```
skilpadde.penup() # for å ikke tegne en strek fra midten av brettet og til start
posX = -150 # setter den til høyre
posY = 0 # midt på brettet
skilpadde.goto(posX,posY)
```

Løsningsforslag oppgave 6

```
poeng = 100
runde = 1
```

Løsningsforslag oppgave 7

```
while runde < 100
    print("Runde " + str(runde) )
    runde += 1
    poeng -= 1
    print("Halla, Aurora!")
```

Løsningsforslag oppgave 8

(koden skal plasseres inne i whileløkken)

```
while runde < 100:
```

```

print("Runde: " + (str(runde)))
runde += 1
poeng -= 1
if skilpadde.pos() > (150,0):
    print("Du har vunnet")
    print("Du fikk " + str(poeng) + " poeng")
    break

```

Løsningsforslag oppgave 9

```

while runde < 100:
    print("Runde: " + (str(runde)))
    runde += 1
    poeng -= 1
    if skilpadde.pos() > (150,0):
        print("Du har vunnet")
        print("Du fikk " + str(poeng) + " poeng")
        break
    else:
        kast = input("trykk enter for å kaste terning")
        terningkast = kast_terning()
        posX = posX + terningkast
        skilpadde.goto(posX, 0)

```

Totalt løsningsforslag

```

import turtle
#oppgave 1: tegn skilpadden
skilpadde = turtle.Turtle()
skilpadde.shape("turtle")
skilpadde.color("lightgreen", "green")

```

```

#oppgave 2 og 3: terning
terning = [1,2,3,4,5,6]

```

```

import random
def kast_terning():
    terningkast = random.choice(terning)
    print("du kastet " + str(terningkast))
    return terningkast

```

```

#oppgave 4: målstrek
målstrek = turtle.Turtle()
målstrek.color("red")
målstrek.penup()

```

```
målstrek.goto(150,-300)
målstrek.pendown()
målstrek.goto(150,300)
målstrek.penup()
```

```
#oppgave 5: flytt skilpadde
skilpadde.penup()
posX = -150
posY = 0
skilpadde.goto(posX, posY)
```

```
#oppgave 6: lag spillvariabler
poeng = 100
runde = 1
```

```
#oppgave 7, 8 og 9: spillet
while runde < 100:
    print("Runde " + str(runde))
    runde += 1
    poeng -= 1
    if skilpadde.pos() > (150,0):
        print("Du har vunnet")
        print("Du fikk " + str(poeng) + " poeng")
        break
    else:
        kast = input("trykk enter for å kaste terning")
        terningkast = kast_terning()
        posX = posX + terningkast
        skilpadde.goto(posX, 0)

print("Game over")
```