

Hangman

Skrevet av: Omsett frå Code Club UK ([//codeclub.org.uk](http://codeclub.org.uk))

Oversatt av: Stein Olav Romslo

Kurs: Python

Tema: Tekstbasert, Spill

Fag: Programmering

Klassetrinn: 5.-7. klasse, 8.-10. klasse

Introduksjon

La oss lage eit spel: Hangman! Datamaskina vil velje eit ord, og du kan gjette det bokstav for bokstav. Viss du gjettar feil for mange gonger tapar du.

Steg 1: Vel eit ord

Fyrst må me få datamaskina til å velje eit tilfeldig ord.

Sjekkliste

☐ Åpne IDLE, og åpne eit nytt vindauge.

☐ Skriv inn følgjande kode:

```
from random import choice

word = choice(["kode", "kurs"])

print(word)
```

☐ Lagre programmet ditt og køyr det. Kva ord blir skrive ut?

☐ Køyr programmet ein gong til. Skriv det ut eit anna ord?

Kvar gong du køyrer dette programmet vil det velje eit tilfeldig ord frå lista `["kode", "kurs"]` ved hjelp av `choice`-funksjonen.

Steg 2: Gjett ein bokstav

No har me valt eit ord, la oss finne ut korleis me gjettar ein bokstav.

Sjekkliste

- ☐ I den same fila, endre koden så den ser slik ut:

```
from random import choice

word = choice(["kode", "kurs"])

out = ""

for letter in word:
    out = out + "_"

print("Gjett ein bokstav i ordet:", out)
```

- ☐ Lagre og køyr programmet.
- ☐ Du burde sjå Gjett ein bokstav i ordet: _____, i output-vindauget (det andre vindauget, ikkje vindauget du har skrive programmet ditt i).

Me brukar ei `for`-løkke til å byggje ein tekst der kvar bokstav i ordet er bytta med ein understrek `_`. Ordet `kode` vil til dømes skrivast som _____ til skjermen.
- ☐ La oss gjette ein bokstav! Endre koden så den ser ut som dette

```
from random import choice

word = choice(["kode", "kurs"])

out = ""

for letter in word:
    out = out + "_"

print("Gjett ein bokstav i ordet, avslutt med enter:", out)

guess = input()

if guess in word:
    print("Yay")
else:
    print("Nope")
```

Me brukar ein ny prosedyre `input()` for å finne ut kva bokstav spelaren skriv. Me brukar `if` for å sjekke om bokstaven er i ordet.

Då har me gjort det viktigaste. La oss fortsetje vidare.

Python 2 tips:

Bruk `raw_input` i staden for `input` viss du brukar ein gamal versjon av python.

Steg 3: Hugs bokstavane som er gjetta

No skal me bruke to nye komponentar i python, lister og `while`-løkker.

Sjekkliste

☐ I den same fil, endre koden så den ser slik ut:

```

from random import choice

word = choice(["kode", "kurs"])

guessed = []

while True:
    out = ""
    for letter in word:
        if letter in guessed:
            out = out + letter
        else:
            out = out + "_"

    if out == word:
        print("Du gjetta", word)
        break

    print("Gjett ein bokstav i ordet:", out)
    guess = input()

    if guess in guessed:
        print("Du har allereie gjetta denne bokstaven:", guess)
    elif guess in word:
        print("Yay")
        guessed.append(guess)
    else:
        print("Nope")

    print()

```

☐ Køyr koden og prøv å gjette bokstavane.

Me har laga ei `while True`-løkke, tilsvarande for alltid i Scratch. Denne vil i utgangspunktet fortsetje å spørje spelaren om å gjette bokstavar for alltid. For å kome ut av løkka brukar me kommandoen `break` når ordet har blitt gjetta.

Me brukar òg ei liste, `guessed`, der me legg til bokstavane som er riktige for å hugse dei til seinare.

Steg 4: Tel feil

For at Hangman skal halde oversikt over alle bokstavane som er gjetta på må me òg hugse når spelaren gjettar feil.



Sjekkliste



Endre fila du jobbar med slik at den blir sjåande ut som dette:

```
from random import choice

word = choice(["kode", "kurs"])

guessed = []
wrong = []

while True:
    out = ""
    for letter in word:
        if letter in guessed:
            out = out + letter
        else:
            out = out + "_"

    if out == word:
        print("Du gjetta", word)
        break

    print("Gjett ein bokstav i ordet:", out)

    guess = input()

    if guess in guessed or guess in wrong:
        print("Du har allereie gjetta denne bokstaven:", guess)
    elif guess in word:
        print("Yay")
        guessed.append(guess)
    else:
        print("Nope")
        wrong.append(guess)

    print()
```

Me brukar ei ny liste `wrong` som tek vare på alle bokstavane me har gjetta som er feil.

Steg 5: Berre nokre få forsøk

Berre ein ting står att før spelet er ferdig, me vil avgrense kor mange forsøk spelaren har til å gjette.



Sjekkliste



Endre fila for å leggje til ein ny variabel, `tries` :

```

from random import choice

word = choice(["kode", "kurs"])

guessed = []
wrong = []

tries = 7

while tries > 0:
    out = ""
    for letter in word:
        if letter in guessed:
            out = out + letter
        else:
            out = out + "_"

    if out == word:
        break

    print("Gjett ein bokstav i ordet:", out)
    print(tries, "forsøk igjen")

    guess = input()

    if guess in guessed or guess in wrong:
        print("Du har allereie gjetta denne bokstaven:", guess)
    elif guess in word:
        print("Yay")
        guessed.append(guess)
    else:
        print("Nope")
        tries = tries - 1
        wrong.append(guess)

    print()

if tries:
    print("Du gjetta", word)
else:
    print("Du klarte ikkje å gjette", word)

```



Køyr programmet, og sjå kva som skjer når du gjettar feil bokstavar.

Legg merke til at me endra `while`-løkka ved å leggje inn ein føresetnad, `while tries > 0`. Dette tyder at løkka berre køyrer så lenge variabelen `tries` er større enn 0. Ser du litt rundt i koden ser du at `tries` startar med verdien 7, og blir 1 mindre for kvar feil bokstav som blir gjetta. Altså vil spelaren kunne gjette opp til 7 bokstavar feil før spelet er slutt.

Steg 6: Legg til nye ord

Sjekkliste

- ☐ Finn linja i programkoden som seier:

```
word = choice(["kode", "kurs"])
```

- ☐ Me kan endre denne linja for å leggje til fleire ord i spelet. Prøv til dømes

```
word = choice(["kode", "kurs", "robot", "klubb"])
```

Hugs at orda må stå i hermeteikn og at det må vere komma mellom orda for å lage ei liste. Legg til fleire ord som du finn på sjølv.

Lisens: Code Club World Limited Terms of Service

(<https://github.com/CodeClub/scratch-curriculum/blob/master/LICENSE.md>)