



Kuprat

Introduksjon

I dette kurset skal vi introdusere programmeringsspråket Python. Det brukes til det meste. Python brukes av mange av de største og mest kjente organisasjonene i verden som NASA, CERN og andre.

Vi begynner ganske enkelt med å se på hvordan vi kan lage små programmer. Vi vil lage vår egen versjon av et historisk program som heter **Cowsay** visdom.



Steg 1: Hei Verden

For å sjekke at Python virker som det skal begynner vi med å lage et lite program som skriver ut en hilsen på skjermen.



Sjekkliste



Åpne IDLE, editoren som følger med Python. Vi vil bruke denne i resten av kurset.

Windows: Åpne IDLE fra startmenyen.

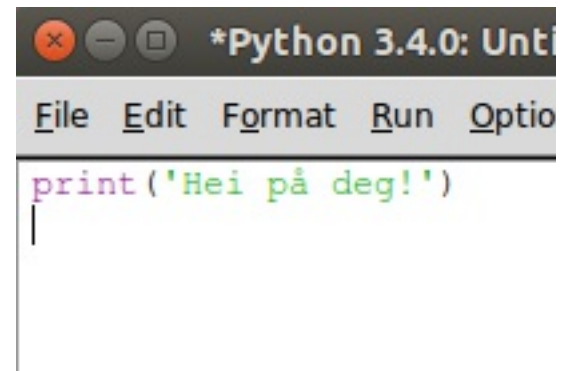
Mac: Åpne terminal.app, skriv `idle` og trykk enter.

Linux: Åpne en terminal, skriv `idle` og trykk enter.

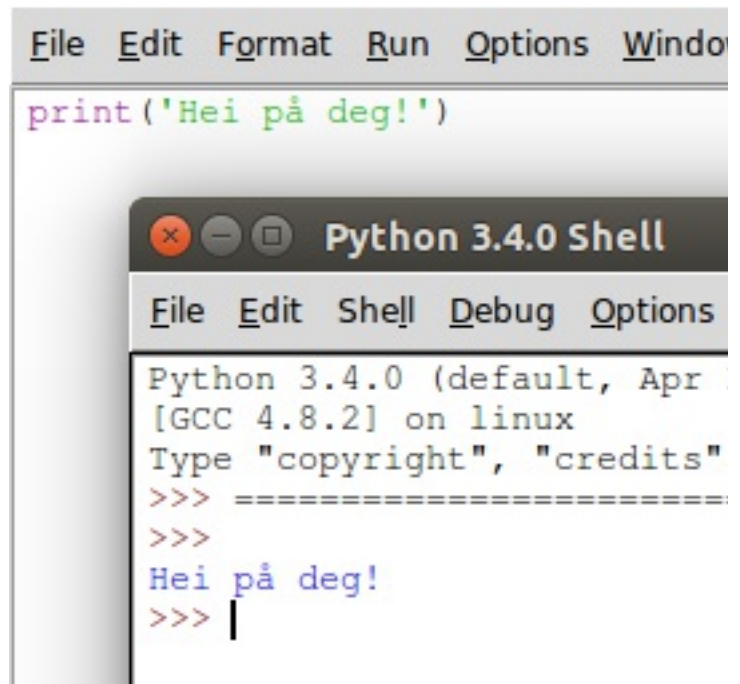
Dette vil åpne et vindu som heter `Python Shell`. Hvis du ikke får det fordi Python ikke er installert. I såfall kan du laste ned siste versjon av Python om nødvendig.

☐ Vinduet `Python Shell` som åpnet seg er der du vil se resultatet av programmet. Vi åpner et programmeringsvindu i tillegg. I menyen, velg `File > New File`.

☐ I dette nye vinduet skal vi nå skrive vårt første Python-program.

A screenshot of a Python 3.4.0 IDLE window. The title bar reads '*Python 3.4.0: Untitled'. The menu bar includes 'File', 'Edit', 'Format', 'Run', and 'Options'. The code editor contains the line `print('Hei på deg!')` with a cursor at the end of the line.

☐ Vi skal nå lagre og kjøre dette programmet. Velg først `File > Save` for å lagre programmet. Deretter kan du kjøre programmet ved å klikke `Run > Run Module`. Du skal se resultatet i terminalvinduet.



The image shows a screenshot of a Python IDE. The main window has a menu bar with 'File', 'Edit', 'Format', 'Run', 'Options', and 'Window'. Below the menu bar, the code `print('Hei på deg!')` is visible. A smaller window titled 'Python 3.4.0 Shell' is open in front of it. This shell window has its own menu bar with 'File', 'Edit', 'Shell', 'Debug', and 'Options'. The shell's output area shows the following text: 'Python 3.4.0 (default, Apr [GCC 4.8.2] on linux', 'Type "copyright", "credits"', a separator line '=====', and the prompt '>>>'. The output 'Hei på deg!' has been printed, and the prompt '>>>' is followed by a cursor.

Gratulerer! Du har nå skrevet og kjørt ditt første Python-program

Feilmeldinger

Python krever at du er veldig nøyaktig når du programmerer, og hvis du får en feilmelding til deg når du velger **Run > Run Module**. Du har kanskje allerede prøvd å bytte ut `print` med `pint` i koden din, og prøve å kjøre programmet ditt.

Når du får en feilmelding må du gå tilbake til koden, og sjekke eksterne feilmeldinger. Dette blir enklere etterhvert som du får mer trening i å programmere.

Sjekkliste

- ☐ Vårt første program bestod av bare en kommando, nemlig funksjonen `print` for å skrive noe ut på skjermen. Det vi vil skrive til skjermen putter vi i funksjonen `print`. Vi skulle skrive teksten `Hei på deg!`. For å fortelle Python at `Hei på deg!` er en Python-kommando må vi skrive teksten mellom fnutter, `'Hei på deg!'`.

- ☐ Vi kan enkelt endre på hva Python skriver til skjermen. Prøv å endre

```
print('Hei alle sammen!')
```

Lagre filen på nytt, og kjør programmet.

- ☐ Prøv å endre teksten til noe du finner på selv, og kjør programmet

Python-filer

Vi har nettopp lagd et Python-program som heter `hei` og som vi har kjørt. Python kaller en slik tekstfil for en `module`. Du kan selv bestemme filene hvor du lagrer programmene må ha etternavnet `.py` for at Python skal kunne finne dem.

Det er også en god ide å unngå å bruke norske tegn som `æ`, `ø` og `å`. Mellomrom kan du bruke understrek, `_`, for eksempel `mitt_programm`.

Steg 2: Hva heter du?

Vi skal nå se hvordan vi kan få Python til å stille oss spørsmål. For å gjøre dette bruker vi funksjonen `input`.

Når du bruker funksjonen `input`, venter datamaskinen din med å kjøre videre til du trykker på enter-tasten på tastaturet.

Sjekkliste

- ☐ Endre programmet ditt slik at det nå ser slik ut:

```
navn = input('Hva heter du? ')
print('Hei ' + navn)
```

Lagre og kjør programmet. Skriv inn navnet ditt når du blir spur
navn?

- ☐ For at teksten skal se bra ut må du passe på at du bruker mellomrom og `'` i input-funksjonen, og også et mellomrom mellom `Hei` og
- ☐ Legg merke til at i programmet bruker vi en variabel `navn` som hele tiden når vi programmerer. Variabler lages automatisk når skal hete, og det er lurt å velge navn som beskriver det variabe
- ☐ Prøv å legg inn flere linjer i programmet ditt. Kanskje Python kan din eller kanskje hva som er favorittfargen din? Bruk variabler fo tilbake på skjermen etterpå.

Hurtigtaster

Når vi programmerer kan det være greit å slippe å lete inne i meny vi bruke hurtigtaster. Hvis du ser etter vil du se i menyen at til høy For eksempel kan du trykke `Ctrl` og `S` for å lagre (`Cmd` og `S` på Mac).

Steg 3: Kuprat

La oss lage en enkel versjon av et klassisk program som heter Cowsay Monroe. Med dette programmet kan man få en stilig ku til å si omtren

Sjekkliste

- ☐ Vi begynner med å tegne kua. Start et nytt IDLE-vindu ved å vel

```

print('^__^')
print('(oo)\_____')
print('(__)\_____')
print('      ||----W |')
print('      ||      ||')

```

Lagre programmet som `kuprat.py` og kjør det. En ganske stilig

- Men nå må vi få kua til å si noe. Legg til og endre kodelinjene di

```

print(' _____')
print('< Python er morsomt! >')
print(' -----')
print('      \   ^__^')
print('      \  (oo)\_____')
print('         (__)\_____')
print('                ||----W |')
print('                ||      ||')

```

- Nå kan vi bruke det vi har lært tidligere for å enkelt endre på m spørre om hva kua skal si. Endre programmet slik at det ser ut s

```

melding = input('Hva skal kua si? ')

print(' _____')
print('< ' + melding + ' >')
print(' -----')
print('      \   ^__^')
print('      \  (oo)\_____')
print('         (__)\_____')
print('                ||----W |')
print('                ||      ||')

```

- Hvordan virker programmet når du kjører det nå? Prøv med fors

- Snakkeboblen til kua er ikke tilpasset lengden av meldingen, slik den for liten. For å fikse dette skal vi bruke en ny funksjon som lengde). Denne kan finne lengden til en tekst. For å teste denne `input`-linjen i programmet ditt:

```
print(len(melding))
```

Dette vil skrive ut lengden av meldingen før kua skrives ut.

- Vi kan bruke denne lengden av meldingen til å regne ut hvor lar begge sider av meldingen bør snakkeboblen være 2 tegn lengre

```
boblelengde = len(melding) + 2
```

- For å tegne snakkeboblen kan vi bruke et Pythontriks som kan r sammen tekst ved å bruke `+`, for eksempel `'Hei ' + navn`. For eksempel vil `'hei' * 3` bli til `'heiheihei'`. Vi kan derfor gange snakkeboblen i riktig størrelse.

- Endre programmet ditt slik at det blir seende ut som følger:

```
melding = input('Hva skal kua si? ')
boblelengde = len(melding) + 2

print(' ' + '_' * boblelengde)
print('< ' + melding + ' >')
print(' ' + '-' * boblelengde)
print('      \      ^__^')
print('      \    (oo)\_____')
print('          (__)\       )\/ ')
print('              ||----w |')
print('              ||     ||')
```

Lagre og kjør programmet. Blir snakkeboblen riktig størrelse?

Prøv selv

Kan du tegne andre dyr eller figurer som også kan snakke? Prøv ev
for eksempel kan du forandre øynene (med `--` vil det se ut som or

Du kan også legge til flere forskjellige figurer i samme program, sli
deg fram!

Lisens: [CC BY-SA 4.0](#)