# Innledning

Dette heftet er laget til kurset «Intro til python programmering». Heftet inneholder forklaringer og eksempler på grunnleggende programmering i python. Det er ikke mening å dekke programmeringspensum i videregående skole etter læreplanene i Fagfornyelsen, bare en introduksjon i python med oppgaver.

All programmering i heftet foregår i språket Python, versjon 3.

# Brukergrensesnitt Thonny

Du trenger en IDE (Interaktiv Development Enviroment) for å skrive kode (programmere) og kjøre kode/programmer i Python. En IDE kan være nettbasert (f.eks. <http://trinket.io>) eller et program du laster ned og kjører lokal, på lik linje som Word.   
Hvis du ikke har noe installert kan du installere et IDE som heter «Thonny». Andre alternativer er for eksempel «Spyder», «Visual Studio Code». Python virker på samme måte i alle disse programmer, så det spiller liten rolle hvilket program du bruker.

Vi installerer Thonny fra fra nettsiden til Thonny (<https://thonny.org/>).

|  |
| --- |
| Et bilde som inneholder tekst  Automatisk generert beskrivelse |

En beskrivelse av hvordan du kommer i gang med Thonny får du i introkurset.

|  |
| --- |
|  |

# Grunnleggende programmering

## Et første program

Vi lager en ny fil i Thonny og lagrer denne programfilen med et passende navn. Thonny fungerer som Word, bare velg File->New for å opprette en ny fil.

På første ledig linje skriver vi koden du finner under (linjer 1-4)

Et bilde som inneholder tekst

Automatisk generert beskrivelse

Vi skal nå kjøre programmet. I Thonny trykker vi på den grønne «Run file» knappen. Første gang du kjører et program må du lagre det. Velg en mappe og gi fil et passende navn.   
  
Når du kjører koden skal teksten «Velkommen til Akademiet NRG» komme til syne i konsollen.

I programmet du skrevet har du bruk en kommentar, en variable og en funksjon for å skrive til konsoll:

Et bilde som inneholder tekst

Automatisk generert beskrivelse

## Kommentar - #

**Kommentar-tegnet i Python er «#».** All tekst etter dette tegnet fram til neste linjeskift blir ikke tolket som kode og dermed ignorert.

Vi bruker kommentarer i koden for at den skal bli lettere å lese for andre. Du kan også skrive lange kommentarer over flere linjer mellom tre anførselstegn. Dette brukes også for å forhindre kjøring av visse deler av kode ved testing.

Et bilde som inneholder tekst

Automatisk generert beskrivelse

## Skrive til skjerm – print()

Når vi bruker kommandoen print() sender vi tekst til konsollen. Vi kaller dette *å skrive til skjerm*. Det er vanlig å kalle teksten som kommer til syne i konsollen for programmets resultat eller output.

I **print** -kommandoen står tall og Python-uttrykk uten anførselstegn, mens tekst står **print** med anførselstegn. Vi lager et program med dette innholdet:

**Et bilde som inneholder tekst

Automatisk generert beskrivelse**

1

Når vi kjører programmet, kommer denne teksten i konsollen:

Et bilde som inneholder tekst

Automatisk generert beskrivelse

Alt som står mellom anførselstegn blir gjengitt akkurat slik det står, mens det som står uten anførselstegn blir erstattet med verdien til uttrykket.

## Variabler

Variabler brukes for å lagre verdier, som kan brukes seiene i programmet. «skole» i første program vi skrev var en variable, som vi brukte igjen i print-setning for å skrive til skjerm navnet på skolen.

### Opprette en variabel

En variabel har et navn og en verdi. Å opprette en variabel i python og tilordner variabel en verdi kaller vi det «å definere en variabel». Du kan ikke bruke en variable før du har definert den. Hvis du gjør det, vil du få en feilmelding «is not defined».

Du tilordner variabler verdi ved å skrive variabelens navn, etterfulgt av et likhetstegn og variabelens nye verdi.

Et bilde som inneholder tekst

Automatisk generert beskrivelse

Merk at når verdi er et tall skriver du det rett fram, mens tekst må ha enkle eller doble anførselstegn rundt seg. Du kan velge om du vil bruke enkle eller doble tegn, bare du bruker samme tegn til å åpne og avslutte teksten.

Du kan kalle variabel hva du vil, men det kan være lurt å vende seg til å bruke variabelnavn som gir mening

|  |
| --- |
| Variabelnavn i python må oppfylle noen regler. |
| * + - 1. Variabler skal begynne med en vanlig bokstav.       2. Variabler kan inneholde bokstaver, tall og understrek, \_.       3. Variabler kan ikke inneholde mellomrom eller noe annet.       4. Noen ord er opptatt. Hvis du skriver inn et ord og det endrer farge, betyr det at ordet har spesiell betydning i programmet og at det derfor ikke kan brukes som variabelnavn. |

## Datatyper

## Matematiske operatorer

For de vanlig regneoperasjoner (addisjon, subtraksjon, multiplikasjon og divisjon) bruker vi samme tegn som vi pleier på digitale verktøy. Det betyr at vi kan skrive

4 + 3, 54 - 3, 5 \* 6 og 24/3, og resultatet blir som forventet. Potens skriver vi med to gangetegn: 2\*\*3 betyr 23 og blir 8.

Et bilde som inneholder tekst, bord

Automatisk generert beskrivelse

Et bilde som inneholder tekst

Automatisk generert beskrivelse

Gir resultat:

Et bilde som inneholder tekst

Automatisk generert beskrivelse

* Desimalkommaet er punktum. Siden vi har en divisjon operasjon får vi resultat som desimaltall (float), med punktum.
* Heltallsdivisjon er divisjon uten rest. Dette betyr at for eksempel både 24//8

og 25//8 blir 3

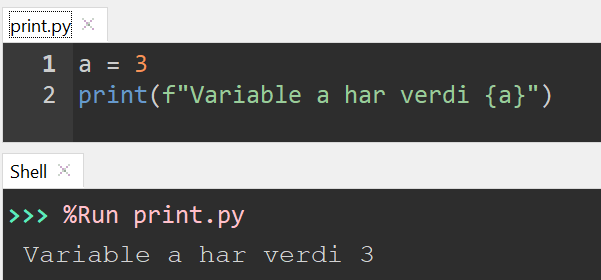
3

* Modulo-operatoren gir resten ved en divisjon. Dette betyr at for eksempel et 5%2 blir 1 (oddetall), mens 10%2 blir 0 (partall)

|  |
| --- |
| Modulo-operatoren egner seg godt til å teste delelighet. Dersom et tall er delelig med et annet, så gir modulo-operatoren 0. Altså har vi at 24 % 8 gir 0, siden 24 er delelig med 8.  ≡ |

## Utskrift av tekst og variabler sammen - f-strings

Når det vi skal skrive inneholder både tekst og verdien av variabler, bruker vi et såkalt f-string. Se eksempel. Mer f-bokstaven foran anførselstegn:



Vi bruker en «f» rett før anførselstegnet for å fortelle at det kommer et f-string. Alt som står inni krøllparenteser blir tolket som pythonkode. Det betyr at {a} blir tolket som verdien av variabelen *a*.

## Oppdatere en variabel

Likhetstegnet vi bruker til å tilordne variabler verdi må forstås som en tilord- ningsoperator, ikke som likhet. Når vi skriver a = 3, så blir verdien av *a* satt til 3, uansett hva verdien var før dette. Hvis vi derimot skriver a = a + 4, så blir verdien av *a* satt til summen av 4 og den verdien *a* hadde tidligere. Dette siste kan også skrives som a += 4. Denne hurtigskrivemåten virker med alle operatorer.

**EKSEMPEL 4**

1 a = 3 # a har nå verdien 3

2 a = a + 4 # a har nå verdien 7

3 a += 5 # a har nå verdien 12

4 a \*= 3 # a har nå verdien 36