

# Viktige Python-kommandoer

Variabler og datatyper	
a = 4	Heltall
b = 2.718	Flyttal (desimaltall) – punktum brukes i stedet for komma
c = "hei"	Tekst (streng)
d = int(4.2)	Gjør om til heltall (her 4)
e = float(3)	Gjør om til flyttall (her 3.0)
f = string(7)	Gjør om til tekst (her "7") – teksten "7" er ikke det samme som tallet 7

Logiske operatører (anta at a = 4)		
==	Er lik	a == 4 gir True
!=	Ikke lik	a != 4 gir False
<	Mindre enn	a < 8 gir True
>	Større enn	a > 8 gir False
<=	Mindre enn eller lik	a <= 4 gir True
>=	Større enn eller lik	a >= 5 gir False
and	Alle betingelser må være sanne samtidig	a < b and b < c
or	Minst én av flere betingelser må være sann	a < b or b > c

Aritmetiske operatører		
+	Addisjon	a + b
-	Subtraksjon	a - b
*	Multiplikasjon	a * b
/	Divisjon	a / b
**	Eksponent	a ** b
%	Modulus (rest ved divisjon)	17 % 5 gir 2
//	Heltallsdivisjon	17 // 5 gir 3

Valg (if-setninger)
Valg lar oss styre hva som skjer i et program (basert på betingelser).
<pre>if &lt;betingelse&gt;:     # kode som utføres hvis betingelsen er sann (True) elif &lt;betingelse&gt;:     # kode som utføres hvis betingelsen er sann (True) else:     # kode som utføres hvis ingen tidligere betingelse er sann</pre>
Vi kan velge å bare bruke if-delen. Vi kan ha så mange elif-betingelser som vi ønsker. Husk at hvis én betingelse er sann, vil ingen senere betingelser bli tatt i bruk.

while-løkke
En while-løkke gjentas så lenge en gitt betingelse er sann (True).
<pre>while &lt;betingelse&gt;:     # kode som utføres hvis betingelsen er sann</pre>
Eksempel som skriver ut 1, 2, 3, 4, 5, 6, 7:
<pre>x = 1 while x &lt; 8:     print(x)     x = x + 1</pre>

for-løkke
En for-løkke gjentas et forhåndsbestemt antall ganger.
<pre>for i in range(fra og med, til, steglengde):     # kode som gjentas</pre>
Eksempel som skriver ut 1, 2, 3, 4, 5, 6, 7:
<pre>for i in range(1, 8, 1):     print(i)</pre>
Vi kan også skrive range(til), da blir fra og med satt til 0 og steglengde til 1.  Et annet alternativ er range(fra og med, til), da blir steglengde satt til 1.

Graftegning (krever pylab <sup>1</sup> )	
<code>plot(x, y)</code>	Lager en graf med <b>x</b> langs førsteaksen og <b>y</b> langs andreaksen. Det er mulig å tegne flere grafer i samme figur.
<code>show()</code>	Gjør grafen synlig
<code>title("Tittel")</code>	Tittel ovenfor grafen
<code>xlabel("Tekst")</code>	Aksetittel langs førsteaksen (x-aksen)
<code>ylabel("Tekst")</code>	Aksetittel langs andreaksen (y-aksen)
<code>legend()</code>	Viser beskrivelse av grafer – må brukes sammen med egenskapen <b>label</b> :  <code>plot(x, y, label="graf 1")</code> <code>plot(x, z, label="graf 2")</code> <code>legend()</code>
<code>xlim(fra, til)</code>	Definisjonsmengde (verdier som vises langs x-aksen)
<code>ylim(fra, til)</code>	Verdimengde (verdier som vises langs y-aksen)
<code>grid()</code>	Tegner inn et rutenett
<code>axhline(10, color="red")</code>	Setter inn en horisontal linje, her gjennom y = 10
<code>axvline(5, color="blue")</code>	Setter inn en vertikal linje, her gjennom x = 5

Tilfeldige tall (krever pylab <sup>1</sup> )	
<code>randint(a, b)</code>	Tilfeldig heltall fra og med <b>a</b> til, men ikke med, <b>b</b>
<code>uniform(a, b)</code>	Tilfeldig flyttal fra og med <b>a</b> til, men ikke med, <b>b</b>

Lister og arrayer (krever pylab <sup>1</sup> )	
<code>liste = [1, 2, 3]</code>	Lager en liste med verdiene 1, 2 og 3
<code>liste.append(4)</code>	Legger til verdien 4 bakerst i listen
<code>len(liste)</code>	Gir oss listens lengde (antall verdier)
<code>liste[i]</code>	Henter verdi nummer i fra lista (teller fra og med 0)
<code>array(liste)</code>	Gjør om en liste til en array
<code>zeros(10)</code>	Lager en array bestående av 10 nuller: <code>[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]</code>
<code>linspace(a, b, n)</code>	Array med <b>n</b> jevnt fordelte tall fra og med <b>a</b> til og med <b>b</b>
Hovedforskjellen mellom en liste og en array er at vi kan gjøre matematiske operasjoner på <i>enkeltverdier</i> i en array. Vi kan for eksempel gange en array med 2 og doble alle verdiene. Hvis vi ganger en liste med 2, vil verdiene i lista gjentas, og vi får en dobbelt så lang liste i stedet.	

Funksjoner
En funksjon lar oss skrive en avgrenset kodebit som vi kan gjenbruke. Vi kan bruke funksjoner som dem vi kjenner fra matematikken, men også i mange andre sammenhenger.
Eksempel på funksjonsdefinisjon:
<pre>def f(x):     return x**2 - 2x + 3</pre>
Eksempel på bruk (tegner grafen til ...):
<pre>x = linspace(0, 10, 100) # lager en array med x-verdier y = f(x)                  # bruker funksjonen og lager y-verdier  plot(x, y) # lager grafen show()    # gjør grafen synlig</pre>

<sup>1</sup>Første linje i programmet må være: `from pylab import *`