Viktige Python- kommandoer

|  |  |
| --- | --- |
| **print("tekst", variabel, "tekst")**  **(kommaseparert)** | Skriver ut de elementene med mellomrom |
| **print(f"tekst {variabel} more tekst")** | Alt mellom { } tolkes som kode, for eksempel som en variable. |
| **svar = input("Skriv noe")** | Be bruker om noe tekst informasjon, som blir verdi til variabel **svar** |
| **svar = int(input("Skriv et ­­­tall"))** | Be bruker om et heltall, som blir verdi til variabel **svar** |
| **svar = float(input(("Skriv et tall"))** | Be bruker om et desimaltall, som blir verdi til variabel **svar** |
|  |  |
|  |  |
|  |  |

|  |  |  |
| --- | --- | --- |
| Logiske operatorer (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 |

|  |  |
| --- | --- |
| Variabler og datatyper | |
| **a = 4** | Heltall |
| **b = 2.718** | Flyttal (desimaltall) – punktum brukes i stedet for komma |
| **c = "hei"** | Tekst (**streng**) - kan bruke også ‘hei’ |
| **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** |



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

|  |  |  |
| --- | --- | --- |
| Aritmetiske operatorer | | |
| **+** | 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** |

|  |
| --- |
| for-løkke |
| En **for**-løkke gjentas et forhåndsbestemt antall ganger. |
| **for i in range(fra og med, til, steglengde): # kode som gjentas** |
| Eksempel som skriver ut 1, 2, 3, 4, 5, 6, 7: |
| **for i in range(1, 8, 1):**  **print(i)** |
| 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. |

|  |
| --- |
| Valg (if-setninger) |
| Valg lar oss styre hva som skjer i et program (basert på betingelser). |
| **if <betingelse>:**  **# kode som utføres hvis betingelsen er sann (True)**  **elif <betingelse>:**  **# kode som utføres hvis betingelsen er sann (True)**  **else:**  **# kode som utføres hvis ingen tidligere betingelse er sann** |
| 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. |

|  |  |
| --- | --- |
| Tilfeldige tall (krever matplotlib.pyplot)2 | |
| **randint(a, b)** | Tilfeldig heltall fra og med **a** til, men ikke med, **b** |
| **uniform(a, b)** | Tilfeldig flyttal fra og med **a** til, men ikke med, **b** |

|  |
| --- |
| 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: |
| **def f(x):**  **return x\*\*2 - 2x + 3** |
| Eksempel på bruk (tegner grafen til ...): |
| **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** |

|  |  |
| --- | --- |
| Graftegning (krever matplotlib.pyplot2) | |
| **plot(x, y)** | Lager en graf med **x** langs førsteaksen og **y** langs andreaksen. Det er mulig å tegne flere grafer i samme figur. |
| **show()** | Gjør grafen synlig |
| **title("Tittel")** | Tittel ovenfor grafen |
| **xlabel("Tekst")** | Aksetittel langs førsteaksen (*x*-aksen) |
| **ylabel("Tekst")** | Aksetittel langs andreaksen (*y*-aksen) |
| **legend()** | Viser beskrivelse av grafer – må brukes sammen med egenskapen **label**:  **plot(x, y, label="graf 1") plot(x, z, label="graf 2") legend()** |
| **xlim(fra, til)** | Definisjonsmengde (verdier som vises langs *x*-aksen) |
| **ylim(fra, til)** | Verdimengde (verdier som vises langs *y*-aksen) |
| **grid()** | Tegner inn et rutenett |
| **axhline(10, color="red")** | Setter inn en horisontal linje, her gjennom *y* = 10 |
| **axvline(5, color="blue")** | Setter inn en vertikal linje, her gjennom *x* = 5 |

|  |  |
| --- | --- |
| Lister og arrayer (krever numpy) | |
| **liste = [1, 2, 3]** | Lager en liste med verdiene 1, 2 og 3 |
| **liste.append(4)** | Legger til verdien 4 bakerst i listen |
| **len(liste)** | Gir oss listens lengde (antall verdier) |
| **liste[i]** | Henter verdi nummer i fra lista (teller fra og med 0) |
| **array(liste)** | Gjør om en liste til en array |
| **zeros(10)** | Lager en array bestående av 10 nuller:  **[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]** |
| **linspace(a, b, n)** | Array med **n** jevnt fordelte tall fra og med **a** til og med **b** |
| Hovedforskjellen mellom en liste og en array er at vi kan gjøre matematiske operasjoner på *enkeltverdier* 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. | |

1 Første linje i programmet må være: **from numpy import \* (du kan også bruke from pylab import \* siden pylab inkludere numpy)**

2 Første linje i programmet må være: **from matplotlib.pyplot import \* (du kan også bruke from pylab import \*)**