

Løsningsforslag til Python intensivkurs ved USN Kongsberg

Skrevet av Sirajuddin Asjad og Haytham Ali

SFIP1102 - Ingeniørrollen og prosjektarbeid - Revisjon 1.0

Oppgave 2.2.1)

Hva blir $2 \cdot 3 - 4 / 2 \cdot 2$, $(1-3) \cdot (-3) / (2 \cdot 2)$, $2 \cdot 3^{**}2 / 2$, $1/2^{**}3 - 4 - 8 / 2 \cdot 4$, $2^{**}3 - (4-8) / 2 \cdot 4$?

In []:

```
print(2 * (3-4) / 2 * 2)
print((1-3) * (-3) / (2 * 2))
print(2 * 3 ** (2/2))
print((1/2) ** (3-4-8) / 2 * 4)
print(2 ** 3 - (4-8) / 2 * 4)
```

Oppgave 2.2.2)

Regn ut arealet av en firkant med side a=3 og b=12.

In []:

```
a = 3
b = 12
areal = a * b
print(areal)
```

Oppgave 2.2.3)

Regn ut omkretsen av en firkant med side a=4 og b=5.

In []:

```
a = 4
b = 5
omkrets = 2 * (a+b)
print(omkrets)
```

Oppgave 2.2.4)

Løs $3x - 2 = 13$

In []:

```
x = (13+2) / 3
print(x)
```

Oppgave 2.2.5)

Regn ut omkretsen av en sirkel med radius 3. (Hva er radius? Og pi? Diskuter med sidemannen.)

In []:

```
import math
r = 3
c = 2 * math.pi * r
print(c)
```

Oppgave 2.3.1)

Du har to tekstvariabler, a = '7.3' og b = '30.1'. Bruk korrekt omgjøring av datatyper og finn så differansen (den ene minus den andre), summen og produktet (den ene ganget med den andre).

In []:

```
a = '7.3'
b = '30.1'

a = float(a)
b = float(b)

differanse = a - b
summ = a + b
produkt = a * b

print(differanse)
print(summ)
print(produkt)
```

Oppgave 2.3.2)

Du har c = '7.345'. Er det mulig å runde av tallet til to desimaler på bare en linje?

In []:

```
c = '7.345'
resultat = round(float(c), 2)
print(resultat)
```

Oppgave 2.3.3)

Du har en variabel a = 3.14 og b = "Pi er ". Lag en tekststreng c som kombinerer a og b og gir verdien "Pi er 3.14".

In []:

```
a = 3.14
b = "Pi er "
c = b + str(a)
print(c)
```

Oppgave 2.4.1)

En rett vinkel er 90 grader. Finn sin og cos til en rett vinkel.

In []:

```
import math
grader = 90
radian_verdi = math.radians(grader)
print(math.sin(radian_verdi))
print(math.cos(radian_verdi))
```

Oppgave 2.4.2)

Finn et tilfeldig desimaltall mellom 0 og 1, mellom -5 og 5, mellom -2 og 18.

In []:

```
import random
print(random.random())
print(random.uniform(-5,5))
print(random.uniform(-2,18))
```

Oppgave 2.4.3)

Finn et tilfeldig heltall mellom 0 og 9, mellom -5 og 5, mellom -2 og 18. Grensene skal være med.

In []:

```
import random
print(random.randint(0,9))
print(random.randint(-5,5))
print(random.randint(-2,18))
```

Oppgave 2.4.4)

Finn cos og sin til 35 grader. Regn også ut kvadratet av sin og av cos, og ta summen.

In []:

```
import math
grader = 35
radian_verdi = math.radians(grader)
sin_verdi = math.sin(grader)
cos_verdi = math.cos(grader)
sin_opphoyd = sin_verdi**2
cos_opphoyd = cos_verdi**2
print(sin_opphoyd + cos_opphoyd)
```

Oppgave 2.4.5)

Finn cos og sin til 2.3 radianer.

In []:

```
import math
print(math.cos(math.radians(2.3)))
```

Oppgave 2.4.6)

Finn kvadratroten av 256.

In []:

```
import math
print(math.sqrt(256))
```

Oppgave 2.5.1)

Be om et tall med input(). Gjør den innleste verdien om til float, og finn til slutt kvadratroten og multipliser med pi.

In []:

```
import math
verdi = input("Skriv et tilfeldig tall\n")
float_verdi = float(verdi)
b = 3.14
print(math.sqrt(float_verdi) * math.pi)
```

Oppgave 2.6.1)

Lag et program som leser inn fornavn, etternavn og alder i passende variabler. Omtrentlig fødselsår skal så regnes ut, og når du fyller 50 år. Skriv til slutt ut alle opplysningene.

In []:

```
fornavn = input("Hva er fornavnet ditt? ")
etternavn = input("Hva er etternavnet ditt? ")
alder = input("Hvor gammel er du? ")
aarstall = 2019

cirka_bursdag = aastall - int(alder)
print("Du ble født i %s" % cirka_bursdag)

if int(alder) < 50:
    antall = 50 - int(alder)
    print("Du har %s år igjen før du blir 50 år gammel." % antall)
elif int(alder) > 50:
    print("Du er allerede over 50 år gammel.")
    print("Ditt navn er %s %s, du er %s år gammel og du ble født i %s." % (forna
vn, etternavn, alder, cirka_bursdag))
```

Oppgave 2.6.2)

Lag et program som leser inn navnet ditt, og så finner et tilfeldig tall mellom 15 og 35, og påstår at det er alderen din.

In []:

```
import random
navn = input("Hva er navnet ditt? ")
random_alder = random.randint(15, 35)
print("Ditt navn er", navn, "og du er", random_alder, "år gammel.")
```

Oppgave 2.7.1)

Lag et program som leser inn tre tall, gjør de om til float, så sjekker hvilket av de tre som er størst og skriver til skjerm om det var det første, andre eller tredje som ble skrevet inn.

In []:

```
tall1 = float(input("Skriv tall 1: "))
tall2 = float(input("Skriv tall 2: "))
tall3 = float(input("Skriv tall 3: "))
maxverdi = max(tall1, tall2, tall3)

if maxverdi == tall1:
    vinner = "tall1"
elif maxverdi == tall2:
    vinner = "tall2"
elif maxverdi == tall3:
    vinner = "tall3"

print("Vinneren er", vinner, "med verdien", maxverdi)
```

Oppgave 2.7.2)

Lag et program som finner et tilfeldig tall mellom 1 og 10 (grenser inkludert), og så ber deg gjette tallet. Programmet sier til slutt ifrå om du gjettet for høyt, for lavt eller riktig.

In []:

```
import random
random_tall = random.randint(0,10)
verdi = int(input("Gjett et tall mellom 0 og 10: "))

if verdi > random_tall:
    print("Du gjettet for høyt.")
elif verdi < random_tall:
    print("Du gjettet for lavt.")
else:
    print("Du gjettet riktig.")
```

Oppgave 2.7.3)

Lag et program som simulerer tre terningkast og skriver ut hvor mange like det ble (3 like, 2 like eller ingen like).

In []:

```
import random
kast1 = random.randint(1, 6)
kast2 = random.randint(1, 6)
kast3 = random.randint(1,6)

if kast1 == kast2 == kast3:
    print("Alle kastene er like!")
elif tall1 == tall2 or tall1 == tall3 or tall2 == tall3:
    print("To terningkast er like!")
else:
    print("Ingen terningkast er like!")
```

Oppgave 2.7.4)

Lag et program som plukker to tall, x og y, mellom f.eks 10 og 100. Brukeren blir så spurt om å svare på følgende regnestykker, x+y, x-y, x*y (etc?). For hvert regnestykke kontrollerer programmet svaret og skriver en kommentar.

In []:

```
min_verdi = 10
max_verdi = 100
x = random.randint(min_verdi, max_verdi)
y = random.randint(min_verdi, max_verdi)

def sjekk_svar(svar, verdi):
    if svar == verdi:
        print("Bra jobba! Du har svart riktig!")
    else:
        print("Svaret ditt er feil. Riktig svar er", verdi)

print("Svar på følgende beregninger:")
opg = str(x) + "+" + str(y)
verdi = eval(opg)
svar = input("Hva er " + opg + "? ")
sjekk_svar(int(svar), verdi)

opg = str(x) + "-" + str(y)
verdi = eval(opg)
svar = input("Hva er " + opg + "? ")
sjekk_svar(int(svar), verdi)

opg = str(x) + "*" + str(y)
verdi = eval(opg)
svar = input("Hva er " + opg + "? ")
sjekk_svar(int(svar), verdi)
```

Oppgave 2.8.1)

Lag et program som leser inn tre navn og legger de i en liste, sorterer listen alfabetisk og skriver den sortert ut til skjerm, navn for navn.

In []:

```
navn1 = input("Skriv inn navn 1: ")
navn2 = input("Skriv inn navn 2: ")
navn3 = input("Skriv inn navn 3: ")
names = [navn1, navn2, navn3]
names.sort()

print("Sorterte navn: ")
for name in names:
    print(name)
```

Oppgave 2.8.2)

Som ovenfor, men nå skal det sorteres motsatt vei.

In []:

```
navn1 = input("Skriv inn navn 1: ")
navn2 = input("Skriv inn navn 2: ")
navn3 = input("Skriv inn navn 3: ")
names = [navn1, navn2, navn3]
names.reverse()

print("Sorterte navn: ")
for name in names:
    print(name)
```

Oppgave 2.8.3)

Lag ved hjelp av range() følgende lister: [0,1,2,3,4,5,6,7,8,9], [0,2,4,6], [2,4,6], [5,8,11,14,17]

In []:

```
print(list(range(0, 10)))
print(list(range(0,7,2)))
print(list(range(2,7,2)))
print(list(range(5,18,3)))
```

Oppgave 2.8.4)

Fjern element med indeks 3 fra listen lst1 = [0,1,2,3,4,5]. Fjern så elementet med verdi 2. Stikk så inn i listen et element 5 slik at det får indeks 2. Still så inn i listen et element 'fire' slik at det får indeks 2. Finn så hvilken indeks elementet 5 har. (NB: du har to like elementer i listen. Forstå hva funksjonen returnerer.) Fjern så elementet 5 fra listen. (Igjen, du har to elementer med verdi 5. Se hvilken som fjernes.)

In []:

```
lst1 = [0,1,2,3,4,5]
lst1.pop(3) # Fjern element med index 3
lst1.remove(2) # Fjern element med verdi 2
lst1.insert(2, 5) # Sett inn element 5 i index 2
lst1.index(5) # Viser index til element 5

forste_index = lst1.index(5)
andre_index = lst1[forste_index+1:].index(5) + forste_index+1

print(forste_index)
print(andre_index)
```

Oppgave 2.8.5)

Du har `lst2 = [11,22,33,44,55,66,77]`. Lag en ny liste som inneholder de tre midterste elementene. (Det kan gjøres på mange måter. Hva er den raskeste?)

In []:

```
# Raskeste metoden:
lst2 = [11,22,33,44,55,66,77]
lengde = 3
beggesider_lengde = len(lst2) - lengde
side_lengde = int(beggesider_lengde / 2)
svar = lst2[side_lengde : -side_lengde]
print(svar)

# Annen metode:
for counter in range(side_lengde):
    lst2.pop(0) # Popper første verdi
    lst2.pop(-1) # Popper siste verdi
print(lst2)
```

Oppgave 2.9.1)

Hva om du brukte `range(1,101)` i for-linja. Hva ville endret seg? (Noen ganger bruker du tellevariabelen (iterasjonsvariabelen) inne i løkka, andre ganger er den ikke i bruk til annet enn å telle f.eks 100 ganger.)

In []:

```
# Da hadde vi endret kast nummeret i print fra (kast+1) til (kast).
# Tallet vil ikke endre siden det har samme lengde som range(100), men hvis man
  printer ut
# iteration verdien så vil den gå fra 1 til 100 istedenfor 0 til 99.
```

Oppgave 2.9.2)

Tell også opp antall ganger terningen ga 1, 2, 3 og 4, og skriv ut. Forsikre deg også om at det hele summerer til 100, skriv evt. ut feilmelding.

In []:

```
import random
n1 = 0 # Antall enere
n2 = 0 # Antall toere
n3 = 0 # Antall treere
n4 = 0 # Antall firere
n5 = 0 # Antall femmere
n6 = 0 # Antall seksere

counter = 0
for kast in range(100):
    counter = counter + 1
    tall = random.randint(1,6)
    print("Kast nr", kast+1, "ble", tall)
    if tall == 1:
        n1 = n1 + 1
    elif tall == 2:
        n2 = n2 + 1
    elif tall == 3:
        n3 = n3 + 1
    elif tall == 4:
        n4 = n4 + 1
    elif tall == 5:
        n5 = n5 + 1
    elif tall == 6:
        n6 = n6 + 1
    else:
        raise ValueError("Verdien er utenfor terning rekkevidde.")

print("Antall enere:", n1, "og antall toere:", n2, "og antall treere:", n3, "og
    antall firere:", n4)
print("Det ble totalt", n6, "seksere og", n5, "femmere.")
print("Counter:", counter)
```

Oppgave 2.9.3)

Som ovenfor, men i stedet for å bruke variablene n1, n2, ..., n6, bruk heller en liste antall som er initialisert slik: antall = [0,0,0,0,0,0]. NB: antall[0] vil da typisk inneholde antall 1-ere. NB: Kan du nå forenkle testingen? Bruke bare én test-linje i stedet for seks?

In []:

```
import random
liste = [0,0,0,0,0,0]
counter = 0

for kast in range(100):
    counter = counter + 1
    tall = random.randint(1,6)
    print("Kast nr", kast+1, "ble", tall)
    if tall == 1:
        liste[0] = liste[0] + 1
    elif tall == 2:
        liste[1] = liste[1] + 1
    elif tall == 3:
        liste[2] = liste[2] + 1
    elif tall == 4:
        liste[3] = liste[3] + 1
    elif tall == 5:
        liste[4] = liste[4] + 1
    elif tall == 6:
        liste[5] = liste[5] + 1
    else:
        raise ValueError("Verdien er utenfor terning rekkevidde.")

print("Antall enere: ", liste[0], "og antall toere:", liste[1], "og antall treere: ", liste[2], "og antall firere: ", liste[3])
print("Det ble totalt", liste[4], "seksere og", liste[5], "femmere.")
print("Counter:", counter)
```

Oppgave 2.10.1)

Lag et gangetabelltestingsprogram som først tar følgende input fra brukeren: ...

In []:

```
import random
opg_antall = int(input("Hvor mange oppgaver ønsker du å gjøre? "))
poeng_antall = int(input("Hvor mange poeng vil du gi per oppgave? "))
minuspoeng_antall = int(input("Hvor mange poeng skal trekkes ved feil? "))
laveste_tall = int(input("Laveste tall fra gangetabellen: "))
hoyeste_tall = int(input("Høyeste tall fra gangetabellen: "))
print("Du får %s poeng for riktige svar og -%s for feil svar." % (poeng_antall,
minuspoeng_antall))

feil_antall = 0
riktig_antall = 0
opgnr = 0

while True:
    a = random.randint(laveste_tall, hoyeste_tall)
    b = random.randint(laveste_tall, hoyeste_tall)
    svar = a * b
    print("Hva er %s * %s ?" % (a, b))
    bruker_svar = int(input("Hva er svaret? "))

    if svar == bruker_svar:
        riktig_antall += 1
        opgnr += 1
    else:
        feil_antall += 1

    if opgnr == (opg_antall):
        break

total_poeng = (riktig_antall * poeng_antall) - (feil_antall * minuspoeng_antall)
print("Du har prøvd %s ganger. Du har %s feil svar og %s riktige svar." % ((riktig_antall + feil_antall), feil_antall, riktig_antall))
print("Totalt poengsum: %s" % total_poeng)
```

Oppgave 2.10.2)

For å gjøre det litt gøyere, kan du også legge til tidtaking. Samme oppgave som over.

In []:

```
import random, time
opg_antall = int(input("Hvor mange oppgaver ønsker du å gjøre? "))
poeng_antall = int(input("Hvor mange poeng vil du gi per oppgave? "))
minuspoeng_antall = int(input("Hvor mange poeng skal trekkes ved feil? "))
laveste_tall = int(input("Laveste tall fra gangetabellen: "))
hoyeste_tall = int(input("Høyeste tall fra gangetabellen: "))
print("Du får %s poeng for riktige svar og -%s for feil svar." % (poeng_antall,
minuspoeng_antall))

start_tid = time.time()
feil_antall = 0
riktig_antall = 0
opgnr = 0

while True:
    a = random.randint(laveste_tall, hoyeste_tall)
    b = random.randint(laveste_tall, hoyeste_tall)
    svar = a * b
    print("Hva er %s * %s ?" % (a, b))
    bruker_svar = int(input("Hva er svaret? "))

    if svar == bruker_svar:
        riktig_antall += 1
        opgnr += 1
    else:
        feil_antall += 1

    if opgnr == (opg_antall):
        break

total_poeng = (riktig_antall * poeng_antall) - (feil_antall * minuspoeng_antall)
print("Du har prøvd %s ganger. Du har %s feil svar og %s riktige svar." % ((riktig_antall + feil_antall), feil_antall, riktig_antall))
print("Totalt poengsum: %s" % total_poeng)

slutt_tid = time.time()
total_tid = slutt_tid - start_tid
print("Du har brukt %s sekunder totalt." % total_tid)
```

Oppgave 2.10.3)

Mulige utvidelser:

- Legge til mulighet for addisjon, subtraksjon og/eller divisjon.
- Legge til tidskrav per spørsmål. Rett svar på spørsmålet blir ikke godkjent hvis for lang tid er brukt.

In []:

```
import random, time
opg_antall = int(input("Hvor mange oppgaver ønsker du å gjøre? "))
tidskrav = int(input("Tidskrav per oppgave (i sekunder): "))
poeng_antall = int(input("Hvor mange poeng vil du gi per oppgave? "))
minuspoeng_antall = int(input("Hvor mange poeng skal trekkes ved feil? "))
laveste_tall = int(input("Laveste tall fra gangetabellen: "))
hoyeste_tall = int(input("Høyeste tall fra gangetabellen: "))

while True:
    operasjon = input("Hvilke operasjoner ønsker du å bruke? (Velg mellom: *, +, /, -) ")
    if(operasjon is "/" or (operasjon is "-") or (operasjon is "+") or (operasjon is "*")):
        break
    else:
        print("Operasjonen er ikke tillatt, vennligst velg noe annet.")

print("Du får %s poeng for riktige svar og -%s for feil svar." % (poeng_antall, minuspoeng_antall))

start_tid = time.time()
feil_antall = 0
riktig_antall = 0
opgnr = 0

while True:
    opg_start_tid = time.time()
    a = random.randint(laveste_tall, hoyeste_tall)
    b = random.randint(laveste_tall, hoyeste_tall)
    operasjon_somString = ("a %s b" % operasjon)
    svar = eval(operasjon_somString)

    print("Hva er %s %s %s ?" % (a, operasjon, b))
    bruker_svar = int(input("Hva er svaret? "))
    opg_slutt_tid = time.time()
    tid_differanse = opg_slutt_tid - opg_start_tid

    if svar == bruker_svar:
        if tid_differanse > tidskrav:
            print("Du klarte ikke tidskravet. Du får dessverre ikke poeng for denne oppgaven!")
            opgnr += 1
        else:
            riktig_antall += 1
            opgnr += 1
    else:
        feil_antall += 1

    if opgnr == (opg_antall):
        break

total_poeng = (riktig_antall * poeng_antall) - (feil_antall * minuspoeng_antall)
print("Du har prøvd %s ganger. Du har %s feil svar og %s riktige svar." % ((riktig_antall + feil_antall), feil_antall, riktig_antall))
print("Totalt poengsum: %s" % total_poeng)

slutt_tid = time.time()
total_tid = slutt_tid - start_tid
print("Du har brukt %s sekunder totalt." % total_tid)
```

Oppgave 2.10.4)

Sjekke (omtrentlig) om `time.time()` faktisk angir antall sekunder fra. 1.1.1970.

In []:

```
import time
gammeltid = (1970, 1, 1, 0, 0, 0, 0, 0, 0)
gammeltid = time.mktime(gammeltid)
nytid = time.time()
print(nytid - gammeltid)
```

Oppgave 2.10.5)

La Python tenke på et tilfeldig heltall mellom 1 og 100 (grenser inkludert). Lag så en løkke der du via `input()` gjetter på tallet til du har klart det. For hver gang du gjetter skal programmet si ifra om du gjetter for høyt eller for lavt. Når du gjetter riktig, avsluttes programmet og det skrives ut hvor mange forsøk du brukte.

In []:

```
random_verdi = random.randint(1,100)
antall = 0

while True:
    gjett_verdi = int(input("Gjett tallet: "))
    antall = antall + 1

    if random_verdi < gjett_verdi:
        print("Tallet er for høyt. Gå lavere!")
    elif random_verdi > gjett_verdi:
        print("Tallet er for lavt. Gå høyere!")
    elif random_verdi == gjett_verdi:
        print("Riktig! Antall forsøk: %s" % antall)
        break
```

Oppgave 2.10.6)

Som ovenfor, men la nå programmet gjette på tallet du tenker på.

NB: Løsning må fikses på!

In []:

```
mingrense = 1
maxgrense = 100
random_verdi = random.randint(mingrense, maxgrense)
antall = 0
gjett_verdi = int(input("Skriv inn tallet som skal gjettes på: "))

while True:
    antall = antall + 1
    print("Du gjettet på tallet %s." % gjett_verdi)

    if random_verdi < gjett_verdi:
        print("Tallet er for høyt. Gå lavere!")
        maxgrense = gjett_verdi
    elif random_verdi > gjett_verdi:
        print("Tallet er for lavt. Gå høyere!")
        mingrense = gjett_verdi
    elif random_verdi == gjett_verdi:
        print("Riktig! Antall forsøk: %s" % antall)
        break

    gjett_verdi = random.randint(mingrense, maxgrense)
```

Oppgave 2.11.1)

Lag Trumps tvitre-funksjon. Den tar ingen argumenter, og la det i funksjonen tilfeldig velges mellom tre mulige utsagn. I programmet skal tvitre-funksjonen kalles 7 ganger. Prøv også å kalle funksjonen før den er definert. Og prøv å kalle funksjonen med et argument, f.eks. et tall. (Du skal få feilmeldinger.)

In []:

```
import random
trumps_twitter() # Kaller på funksjonen altfor tidlig. Vil vise error.

def trumps_twitter():
    arg1 = "første argument"
    arg2 = "andre argument"
    arg3 = "tredje argument"
    args = [arg1, arg2, arg3]
    random_arg = random.choice(args)
    return random_arg

# Kaller funksjonen syv ganger
print(trumps_twitter())
print(trumps_twitter())
print(trumps_twitter())
print(trumps_twitter())
print(trumps_twitter())
print(trumps_twitter())
print(trumps_twitter())

# Kaller funksjonen med arguments (vil vise error)
trumps_twitter(1)
```

Oppgave 2.11.2)

Lag en funksjon som tar argumentene t , v_0 og a (tid, startfart og akselerasjon). La både v_0 og a ha default-verdi lik 0 mens t ikke har default-verdi. Funksjonen skal regne ut og returnere $s = v_0 * t + 0.5 * a * t^2$. Sett opp en for-loop slik at en tidsvariabel går fra 0, 0.5, 1, ..., 10 (sekunder). For hver t -verdi skal funksjonen kalles med $v_0=0$ og $g=9.81$, og tid og strekning skal printes ut. (Hvor langt faller man på 10 sekunder?)

In []:

```
def distanse(t, v0 = 0, a = 0):
    s = v0 * t + 0.5 * a * (t ** 2)
    return s

tidsliste = []

for antall in range(21):
    tidsliste.append(antall/2)

g = 9.81
for tidsverdi in tidsliste:
    print(distanse(t = tidsverdi, a = g))
```

Oppgave 2.14.1)

Legg til noen linjer i programmet ovenfor. Prøv deg litt frem med linje-for-linjedebugging. Prøv å gå gjennom loopen, og prøv å utføre den i et tastetrykk (Debug/Step Return). Og prøv breakpoint-debugging: sett noen breakpoints og forsikre deg om at du forstår.

In []:

```
sko = ['Adidas', 'Puma', 'Nike']
kopi = []
for i in range(len(sko)):
    print(i, sko[i])
    kopi.append(sko[i])
```

Oppgave 2.16.1)

Lag et program som trekker 10 tilfeldige desimaltall mellom -2000 og 2000 og skriver ut tallene med tre desimaler i en fin vertikal liste.

In []:

```
import random
for antall in range(10):
    verdi = random.uniform(-2000, 2000)
    print("%s %3f"%(antall+1, verdi))
```

Oppgave 2.16.2)

Som ovenfor, men bruk nå scientific notasjon med 2 desimaler (eks: 1.25e+05)

In []:

```
for antall in range(10):
    verdi = random.uniform(-2000,2000)
    print("Verdien er %.2e"%(verdi))
```

Oppgave 2.16.3)

Lag et program som skriver ut navn, alder og fødselsmåned i fine kolonner. La navn være venstrejustert, og la fødselsmåned være høyrejustert.

In []:

```
navn = ['Newton', 'Descartes', 'Euler']
alder = [84, 53, 76],
mnd = ['january', 'mars', 'april']

for antall in range(len(navn)):
    print(" %s | %s "%(navn[antall], mnd[antall]))
```

Oppgave 2.17.1)

Lag en tuple (3,6,77) og skriv ut første, andre og tredje element, dvs. 3, 6 og 77. Prøv å endre første elementet. (Du får feilmelding.)

In []:

```
a = (3,6,77)
print(a[0])
print(a[1])
print(a[2])
a[0] = 1 # Denne vil vise en error
```

Oppgave 2.17.2)

Lag en tuple ('00','11','22','33','44','55'). Lag fra denne en ny tuple som har med elementene '11','22','33'.

In []:

```
a = ("00", "11", "22", "33", "44", "55")
b = ("11", "22", "33")
print(a + b)
```

Oppgave 2.17.3)

Lag de 100 tuplene som kan lages av to siffer som begge er mellom 0 og 9, dvs. (0,0), (0,1), ..., (0,9), (1,0), (1,1), ..., (1,9), ..., (9,0), (9,1), ..., (9,9) (Tips: Bruk to forløkker, den ene inni den andre.) Legg tuplene inn i en liste som du laget ved start.

In []:

```
tuples = []
for antall in range(100):
    temp_tuple = ()
    for antall in range(2):
        verdi = random.randint(0,9)
        temp_tuple = temp_tuple + (verdi, )
    tuples.append(temp_tuple)
print(tuples)
```

Oppgave 2.18.1)

Lag et program som leser inn et par opplysninger fra tastaturet (navn, yrke etc.), skriver opplysningene til fil, og til slutt leser opplysningene fra filen og skriver de til skjerm. (Bruk gjerne skrivetilfil.py som filnavn.)

In []:

```
navn = input("Hva heter du? ")
yrke = input("Hva jobber du som? ")
tlf = input("Mobilnummer: ")
info = [("navn = %s" % navn), ("yrke = %s" % yrke), ("tlf = %s" % tlf)]

# Skrive til fil:
fill = open('oppg_skriverifil.txt', 'w')
for verdi in info:
    fill.write(verdi+'\n')
fill.close()

# Lese fra fil:
fill = open('oppg_skriverifil.txt', 'r')
linjer = fill.readlines()
for linje in linjer:
    print(linje)
fill.close()
```

Oppgave 2.18.2)

Lag et program som skriver til fil 20 linjer som hver skal inneholde linjenummer til venstre, så 5 tilfeldig valgte tall mellom 0 og 200 oppgitt med 3 desimaler. Tallkolonnene skal være fint justerte.

In []:

```
fil = open('oppg_skriverifil.txt', 'w')
for antall in range(1,21):
    linjer = []
    for _ in range(5):
        nummer = random.uniform(0,200)
        nummer = float("%.2f" % nummer)
        linjer.append(nummer)

    fil.write("%i - %s"%(antall, linjer))
fil.close()
```

Oppgave 2.19.1)

Lag et program som leser inn informasjon om tre personer fra tastaturet. For hver person skal fornavn, etternavn, alder leses inn og legges i en dictionary. Hver persons dictionary legges så inn i en liste. Til slutt skal det loopes gjennom listen og informasjonen skrives ut.

In []:

```
def is_exit(setning):
    if(setning == "exit") or (setning == "quit"):
        return True
    else:
        return False

folkeinfo = []

while True:
    status = input("Skriv 'exit' for å avslutte, eller alt annet for å fortsett
e: ")

    if is_exit(status):
        if len(folkeinfo) > 0:
            for info in folkeinfo:
                print(info)
            break

    fornavn = input("Fornavn: ")
    etternavn = input("Etternavn: ")
    alder = input("Alder: ")

    personinfo = {"fornavn":fornavn, "etternavn":etternavn, "alder":alder}
    folkeinfo.append(personinfo)

print(folkeinfo)
```

Oppgave 2.19.2)

Lag en dictionary dct1 som inneholder navn og alder (som ovenfor). Dictionary'en inneholder også en key 'annet' som har annen dictionary som verdi. Denne dictionary'en har innhold 'hobby':'lego', 'mobiler':4.

In []:

```
dct1={
    "fornavn":fornavn,
    "etternavn":etternavn,
    "alder":alder,
    "annet":{
        "hobby":"lego",
        "mobiler":4
    }
}
print(dct1)
```

Oppgave 2.20.1)

Lag et program som trekker ti desimaltall mellom -10 og 10. For hvert tall skal kvadratroten regnes ut. For negative tall vil imidlertid sqrt gi feilmelding, så skriv da i stedet ut melding om at kvadratroten ikke lar seg regne ut for dette tallet. Bruk try/except-konstruksjon til å gjennomføre. (NB: her kunne vi enkelt testet om tallet var negativt i stedet for å sette opp en try/except-konstruksjon.)

In []:

```
import math
random_verdi = random.uniform(-10, 10)
try:
    kvadr = math.sqrt(random_value)
    print("Kvadratroten til tallet %.4f er %.4f." % (random_verdi, kvadr))
except:
    print("Kvadratroten til tallet %.4f kan ikke regnes ut." % random_verdi)
```

Oppgave 3.2.1)

Lag et array som går fra 0 til 10 (inkludert endepunktene). Steglengden (spacing) skal være 1 (dvs. [0, 1, 2, ..., 10]).

In []:

```
import numpy
arr1 = numpy.arange(0, 11, 1)
print(arr1)
```

Oppgave 3.2.2)

Lag et array som går fra 0 til 10 (inkludert endepunktene). Steglengden (spacing) skal være 0.2 (dvs. [0, 0.2, 0.4, ..., 9.8, 10]).

In []:

```
import numpy
arr2 = numpy.arange(0, 10.2, 0.2)
print(arr2)
```

Oppgave 3.3.3)

Sammenlign numpy.linspace(0,10,6) med numpy.linspace(0,10,5). Forstå forskjellen.

In []:

```
import numpy
len1 = len(numpy.linspace (0,10,6))
len2 = len(numpy.linspace (0,10,5))
print("Første lengde er %i and andre lengde er %i." % (len1, len2))
```

Oppgave 3.2.4)

Lag et array med 10 tilfeldige desimaltall mellom 0 og 100 uten å bruke numpy's egne random-funksjoner. Dvs. du må bruke `random.uniform(0,100)`. Gjør så det samme ved hjelp av numpys randomfunksjoner (`numpy.random.uniform`).

In []:

```
randomtall = []

# Uniform section:
for antall in range(10):
    nummer = random.uniform(0,100)
    randomtall.append(nummer)
print(randomtall)

# Numpy section:
import numpy
randomtall = numpy.random.uniform(0,10,10)
print(randomtall)
```

Oppgave 3.2.5)

En bil starter fra ro (startfart lik 0) og har akselerasjon på 3 m/s². Lag et program som for hvert sekund fra 0 til 10 sekunder regner ut farten til bilen og hvor langt den har gått, og skriver det ut til skjerm.

In []:

```
start_tid = 0
slutt_tid = 10
a = 3
tidsliste = range(start_tid, slutt_tid+1)

for t in tidsliste:
    v = a * t
    s = 0.5 * a * t ** 2
    print("dit: %i s fart: %.2f m/s strekning: %.2f m"%(t, v, s))
```

Oppgave 3.3.1)

Tegn grafen $f(x) = x^3 - x$ for $-2 < x < 2$. Sett passende label på x- og y-aksen, og gi grafen en tittel. Tegn grafen både inline og i eget vindu (i to ulike kjøringer).

In []:

```
# Interactive er av
import numpy
import matplotlib.pyplot as plt
x = numpy.linspace(-2,2,11)
y = x ** 3-x
plt.ioff() # interactive er av
plt.xlabel('X')
plt.ylabel('Y')
plt.plot(x,y, 'b*--')
plt.show()

# interactive er på (vil ikke vises i split window)
import numpy
import matplotlib.pyplot as plt
x = numpy.linspace(-2,2,11)
y = x ** 3-x
plt.ion() # interactive er på
plt.xlabel('X')
plt.ylabel('Y')
plt.plot(x,y, 'b*--')
plt.show()
```

Oppgave 4.2.1)

Finn nullpunktene for funksjonen $f = 3x^3 + 82.5x^2 - 640.5x + 765 = 0$.

In []:

```
from sympy import *
x = Symbol('x')
f = 3 * x ** 3 + 82.5 * x ** 2 - 640.5 * x + 765
solve(f)
```

Oppgave 4.2.2)

Finn x fra ligningen $1/(x-a) = 2/(x+a)$.

In []:

```
from sympy import *
x = Symbol('x')
a = Symbol('a')
likning1 = 2* (x-a)
likning2 = (x + a)
losning = likning1 - likning2
solve(losning, x)
```

Oppgave 4.2.3)

Løs likningssettet $2x + a = x + y + 5$ og $3x - y = 2a - 3$ med hensyn på x og y (forvariabel a).
Finn også løsningene når $a = 5$.

In []:

```
from sympy import *
x = Symbol('x')
y = Symbol('y')
a = Symbol('a')
likning1 = 2*x + a - x - y - 5
likning2 = 3*x - y - 2*a - 3
losning, = linsolve( [likning1, likning2], (x, y))
x_losning, y_losning = losning
print("x løsning er %s" % x_losning)
print("y løsning er %s" % y_losning)

x_verdi = x_losning.subs({"a":5})
y_verdi = y_losning.subs({"a":5})
print("x verdi er %s" % x_verdi)
print("y verdi er %s" % y_verdi)
```

Det vil komme flere oppdaterte utgaver av dette løsningsforslaget i fremtiden!
Mvh Sirajuddin