

Programming



Innledning

Kursmål DEL 2, programmering i skolen

- Grunnleggende programmering
- Matematiske operatorer
- Konsollen
- Variabler
- Datatyper
- Hente opplysninger fra kunde

Et første program

Vi lager en ny fil i Spyder og lagrer denne programfilen med et passende navn, for eksempel «test_01.py». For enkelhets skyld lar vi filnavnet slutte på «.py». Har du nå klargjort en fil som ligger på skrivebordet eller i en mappe du lett finner frem til og har et navn som viser innholdet?

På første ledige linje skriver vi nå

```
print('Matematikk')
```

Vi har skrevet vårt første program. Programmet er på én linje. Vi skal nå kjøre programmet.

I Spyder trykker vi på den grønne «Run file»-knappen. Nå skal teksten «Matematikk» komme til syne i konsollen.

```
Python 3.8.1 (default, Jan 8 2020, 16:15:59)
Type "copyright", "credits" or "license" for more information.

IPython 7.19.0 -- An enhanced Interactive Python.

In [1]: runfile('/Users/gunnarknutzen/.spyder-py3/eksempe_36.py', wdir='/Users/gunnarknutzen/.spyder-py3')
Matematikk

In [2]:
```

2 Grunnleggende programmering

2.1 Skrive til skjerm

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 med anførselstegn.

Vi lager et program med dette innholdet:

```
print('Regnestykket 3 + 5 er lik ')  
print(3+5)
```

Når vi kjører programmet, kommer denne teksten i konsollen:

```
In [2]: runfile('/Users/gunnarknutzen/.spyder-py3/eksempel_36.py', wdir='/Users/gunnarknutzen/.spyder-py3')  
Regnestykket 3 + 5 er lik  
8  
  
In [3]:
```

```
Regnestykket 3 + 5 er lik  
8
```

Programmering

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

2.2 Matematiske operatører

Operator	Betydning	Eksempel	Resultat
+	Addisjon	3+5	8
-	Subtraksjon	3-5	-2
*	Multiplikasjon	3 * 5	15
/	Divisjon	14 / 3	4,6667
**	Potens	2 ** 4	16
//	Heltallsdivisjon	14 // 3	4
%	modulo	14 % 3	2

De vanlige regneartene fungerer slik vi er vant til. Vi bruker addisjon, subtraksjon, multiplikasjon og divisjon slik 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.

MERK.

- Desimalkommaet er punktum. Potens skriver vi med to gangetegn: $2^{**}3$ betyr 2^3 og blir 8.
- Heltallsdivisjon er divisjon uten rest. Dette betyr at for eksempel både $24 // 8$ og $25 // 8$ blir 3, siden $24 : 8 = 3$ og $25 : 8 = 3 + 1/8$
- Modulo operatoren gir resten ved en divisjon. Så $5 \% 2$ gir 1, siden $5 : 2 = 2$ og $1/2$, altså 1 i rest. Videre får vi at $257 \% 11$ gir 4, siden $257 : 11$ har 4 i rest.
For $257 = 23 \cdot 11 + 4$
- Modulo operatoren egner seg godt til å teste delelighet. Dersom et tall er delelig med et annet, så gir modulo-operatoren 0.
 $24 \% 8$ gir 0, siden 24 er delelig med 8.
- Med vanlig matematisk notasjon skriver vi $24 \equiv 0 \pmod{8}$ og leser det som «tjuefire er ekvivalent med null modulo åtte».

2.3 Konsollen

Resultatet fra print kommandoen kommer i konsollen. Konsollen kan også brukes til å finne verdien av små Python-uttrykk direkte, uten å bruke print.

Regn ut: $3 + 4 \cdot 5^3 - 4.53 + 3^{2-1} / 2^3 + 1$

Løsning:

Vi skriver `print(3 + 4 * 5**3 - 4.53 + (3**2 -1) / (2**3 +1))` og kjører programmet.

I konsollen står det da 499.358888888888894.

```
Python 3.10.6 (v3.10.6:9c7b4bd164, Aug 1 2022, 17:13:48) [Clang 13.0.0 (clang-1300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>> ( 3 + 4 * 5**3 - 4.53 + ( 3**2 -1) / (2**3 +1 ))
499.358888888888894
>>>
```

Programmering

Regn ut: $3 + 4.2 - 5 * 11 + 10.46 / 2$


Løsning: Vi skriver rett inn i konsollen og trykker Enter.

```
In [1]: 3 + 4.2 - 5 * 11 + 10.46 / 2
```

```
Out[1]: -42.569999999999999
```

```
Python 3.10.6 (v3.10.6:9c7b4bd164, Aug 1 2022, 17:13:48) [Clang 13.0.0 (clang-1300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>>
>>> 3 + 4.2 - 5 * 11 + 10.46 / 2
-42.569999999999999
>>>
```

NB! Legges dette inn i PyCharm eller Spyder må vi huske kommandolinjene:

Print($3 + 4.2 - 5 * 11 + 10.46 / 2$) og så trykke på play .

Oppgaver

1. skriv ut «Jeg skal nå lære litt programmering»
2. Regn ut:
 - a) $3 + 4^2 - 3 / 2^4 + 1$
 - b) $43,2 : 11$
3. Løs oppgavene
 - a) Hvor mange ganger går 6 opp i 50?
 - b) Hva blir resten når du deler 50 på 6?
4. Løs oppgavene
 - a) Avgjør om 167 335 er delelig med 3.
 - b) Avgjør om 64 er et partall.
5. Løs oppgavene
 - a) Det har gått 115 469 timer siden klokka var midnatt. Hva er klokka nå?

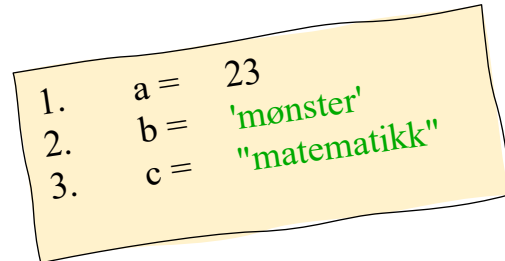
2.4 Variabler

2.4.1 Opprette en variabel

Du tilordner variabler verdi ved å skrive variabelens navn. Variabelens navn bør gi mening, dvs. hva er det den skal benyttes til. Navnet du velger etterfulgt av et likhetstegn og variabelens nye verdi. Tall skriver du 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.

Her setter vi:

- a. til verdien 23
- b. får verdien «mønster»
- c. får verdien «matematikk».



```
1. a = 23
2. b = 'mønster'
3. c = "matematikk"
```

Variabelnavn i Python må oppfylle noen regler. Disse er:

- Variabler skal begynne med en vanlig bokstav.
- Variabler kan inneholde bokstaver, tall og understrek, _.
- Variabler kan ikke inneholde mellomrom eller andre tegn.
- 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.

Eksempler på akseptable variabler:

a, avvik, gjennomsnittlig_avvik, gjennomsnittlig_Avvik, avvik1, Avvik_2.

Det kan være lurt å vende seg til å bruke variabelnavn som gir mening, for eksempel «gjennomsnitt» i stedet for «x». Da blir det lettere å lese koden og ikke minst å redigere, endre eller overta kodingen fra andre.

2.4.2 Oppdatere en variabel

Likhetstegnet vi bruker til å tilordne variabler verdi må forstås som en **tilordningsoperator**, ikke som likhetstegn.

- 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. Denne siste kan også skrives som $a += 4$. Denne hurtige skrivemåten virker med alle operatører.

Eksempel, satt i rekkefølge

```
a = 3      #a har nå verdien 3
a = a + 4  #a har nå verdien 7
a += 5     #a har nå verdien 12
a = 3      #a har nå verdien 3
```

Vi skriver variabelens verdi til skjerm med print uten anførselstegn.

Eksempel

Skriv et program der du oppretter en variabel m med verdien 26. Lag en ny variabel n som er det dobbelte av m. Skriv så verdien av n til skjermen.

```
m = 26
n = 2 * m
print(n)
```

Programmet kjører, og vi får svaret 52.

Oppgaver

6. Skriv et program som gjør følgende:
 - Opprett en variabel a med verdien 5
 - Legg til verdien 4
 - Legg til verdien 6
 - Legg til verdien 5
 - Skriv verdien av a til skjerm og kontroller at variabelen har verdien 20.

7. Opprett to variabler, variablene a og b med verdiene 2 og 3. Bytt verdi, slik at a får b sin verdi og b får a sin verdi, uten å skrive tallene på nytt.

8. Regn ut verdien av uttrykket $V_0 t + 1/2 at^2$
 - $V_0 = 5$
 - $a = 9,81$
 - $t = 4$

2.4.3 Utskrift av tekst og variabler sammen

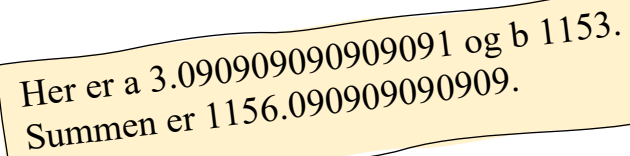
Når det vi skal skrive inneholder både tekst og verdien av variabler, bruker vi et såkalt f-literal. Vi bruker en «f» rett før anførelstegnet for å fortelle at det kommer et f-literal. Alt som står inni krøll-parenteser, blir tolket som Python kode. Det betyr at {a} blir tolket som verdien av variabelen a.

Eksempel

```
a = 34 / 11
```

```
b = 1153
```

```
print(f"Her er a {a} og b {b}. Summen er {a + b}.")
```



Her er a 3.090909090909091 og b 1153.
Summen er 1156.090909090909.

2.4.4 Avrunding og formatering i utskrift

F-literalene bruker vi også når vi vil runde av et tall i en print kommando. Rett før høyre mengdeparentes legger vi inn et kolon.

Kolonet gjør at vi kan styre hvordan utskriften formateres.

Vi ønsker å runde av til tre desimaler med formatdefinisjonen «.3f». Her betyr bokstaven «f» at variabelen skal skrives ut med et **fiksert** antall desimaler og tretallet betyr at det skal være tre desimaler. Hvis du i stedet skriver «.2f», får du to desimaler. Avrundingen følger vanlige regler for avrunding.

$a = 34 / 11$

$b = 1153$

print (f' Her er a = {a:.3f} og b = {b}. Summen er {a + b:.2f}.')

Her er $a = 3.091$ og $b = 1153$.
Summen er 1156.09.

Programmering

```
a = 3445 / 11
```

```
print(f" Tallet a er {a:.2e}")
```

Hvis du skriver «.2e», så får du tallet på standardform, med to desimaler:

Tallet a er $3.13e + 02$

Svaret betyr at variabelen a har verdien $3,13 \cdot 10^2$.

Oppgaver

9. Skriv et program der du oppretter en variabel tall med verdien 56 og skriv ut verdien av variabelen med svar setning. Resultatet skal bli:

Verdien av variablene tall er 56.

10. Skriv et program som legger verdien $11 / 7$ i en variabel. Skriv så ut verdien av variabelen med 2 desimaler.
11. Lagre tallet 1000 i en variabel. Bruk heltallsdivisjon og del variabelen på 23 og legg dette inn i en ny variabel. Resultatet av dette ganger du med 23 og legger i en tredje variabel. Skriv ut verdien av siste variabel.
12. Et rett prisme har sidekantene $l = 3$, $b = 4$ og $h = 5$. Skriv et program hvor du først setter verdien av l , b og h og deretter bruker variablene til å regne ut arealet av overflaten og volumet av prismet. Programmet skal kun inneholde tallene 3, 4 og 5 en gang. Til slutt skal programmet skrive ut svarene, med forklarende tekst

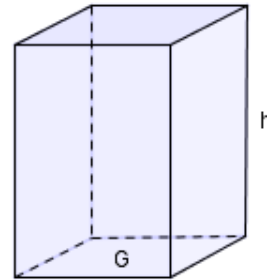
Programmering

Formelen:

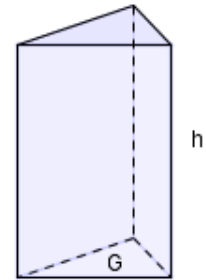
$$lb + lb + lh + lh + hb + hb = \text{overflate}$$

$$\text{Overflate} = 2 \cdot l \cdot b + 2 \cdot l \cdot h + 2 \cdot b \cdot h$$

$$\text{Volum} = l \cdot b \cdot h$$



Rett firkantet prisme



Rett trekantet prisme

2.5 Datatyper

Kode	Datatype	Eksempel
Str	streng (tekst)	'programmering'
Int	heltall	1153
Float	flyttall (desimaltall)	2.7182818
List	liste	['a', 3, 5, 1, 1, 1, 'b']
Bool	boolsk verdi	True, False
np.array	np-array	array([2, 1, 4, 3])

De matematiske operatorene brukes sammen med typene **int** og **float**. Noen av dem kan også brukes med andre datatyper, men da har de en annen betydning. For eksempel betyr «+» sammensetting og «*» gjentakelse når de brukes på strenger og lister. Vi ser hvordan dette virker i konsollen:

Programmering

Eksempel

```
>>> "fem" + "ti"           # tekststreng og operatoren +
    'femti'

>>> 'fem' * 3             # tekststreng og operatoren *
    'femfemfem'

>>> [1, 2, 3] + [4, 5, 6]  # liste og operatoren +
    [1, 2, 3, 4, 5, 6]

>>> [1,2,3] * 2           # liste og operatoren *
    [1, 2, 3, 1, 2, 3]
```

2.6 Hente inn opplysninger fra brukeren

Vi ber om informasjon fra brukeren med kommandoen `input()`.

Før at du skal kunne bruke informasjonen til noe, tilordner du verdien til en variabel når du bruker «`input()`».

Eksempel, skriv kommandoen og angi navnet ditt!

```
navn = input('Skriv inn et navn her: ')
```

```
print(f'Du skrev inn «{navn}».)
```

Nå kjører vi programmet kommer først teksten «Skriv inn et navn her:» fram i konsollen.

Vi skrev inn vårt navn «Fantomet». Til slutt ser konsollen slik ut:

```
Skriv inn et navn her: Fantomet  
Du skrev inn «Fantomet».
```

Det er lurt å legge til et mellomrom før siste anførselstegn i `input-` kommandoen. Da blir det et mellomrom før brukeren skriver inn noe!

Programmering

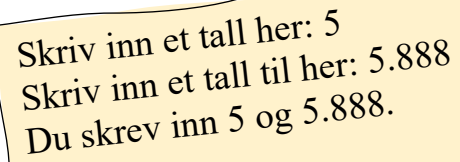
Hvis du skal be brukeren om et tall, må du konvertere til riktig datatype, heltall, int, eller desimaltall, float. Det kan du gjøre direkte, samtidig med inputkommandoen.

Alle kommandoer skal ha argumentet i parenteser, så pass på at du bruker dem systematisk.

```
a = int (input ( 'Skriv inn et tall her: ' ) )
```

```
b = float (input ( 'Skriv inn et tall til her: ' ) )
```

```
print(f'Du skrev inn {a} og {b}.')
```



Skriv inn et tall her: 5
Skriv inn et tall til her: 5.888
Du skrev inn 5 og 5.888.

Her kan du se at 5 oppfattes som et heltall og 5,888 som et desimaltall, siden det står henholdsvis 5 og 5.888.

Oppgaver

13. Be brukeren om fornavn og etternavn. Skriv ut en hilsen som inneholder fullt navn.
14. Lag en variabel der du ber om et tall fra brukeren. Konverter tallet til «float». Skriv ut til skjermen hva brukeren tastet inn.
15. Be om et tall fra brukeren. Regn ut halvparten av tallet. Skriv svaret til skjerm.
16. Be om et tall fra brukeren. Regn ut hvor mange hele ganger det tallet er større enn 5. Skriv svaret til skjerm.

Klar for neste del ?