



Python med Magnus

Grunnleggende programmering i Python



Læringsplan i dag

- Programmering som konsept
- Installering av programmer
- Introduksjon Python
- Jobbe med oppgaver





01

HVORDAN FUNGERER
PROGRAMMERING

Programflyt

- Programmer utføres linje for linje - ei linje av gangen
- Vi snakker gjerne om en **programpeker**
 - Denne flytter seg nedover linje for linje i programmet.
 - Det som står øverst skjer først!

```
slides-rekkefolge.py ×
1 print('5 + 8 =:', 5+8)
2
3 print('Denne skjer etterpå')
4
5 print('5*5 = ', 5*5)
6
7 print('Denne kommer til slutt.')

Assistant ×
The code in slides-rekkefolge.py looks good.
If it is not working as it should, then consider
using some general debugging techniques.
Was it helpful or confusing?

Shell ×
Python 3.7.9 (bundled)
>>> %Run slides-rekkefolge.py
5 + 8 =: 13
Denne skjer etterpå
5*5 = 25
Denne kommer til slutt.
>>> |
```



Uttrykke deg presist

- Grammatikkfeil -

Syntaxfeil

- Datamaskinen nekter å kjøre programmet -> feilmelding

- Logiske feil

- Ting må gjøres i rett rekkefølge

```
Thonny - /Users/terjery/Library/Mobile Documents/com~apple~CloudDocs/ITGK2019/1Forelesninger/Uke 35/2 Onsdag-fredag/Oppg 9.py @ 28 : 1
untitled> Oppg 10-kopi.py Oppg 10.py Logisk.py Oppg 9.py
1 # Oppgave 8.9 - Vowels and consonants.
2 # Brukeren skal skrive inn en setning, og programmet skal rapp
3 # mange vokaler og konsonanter det er i teksten
4
5 def tellVokaler(tekst):
6     antVok = 0
7     vokaler = 'aeiouyæøå'
8     for i in tekst:
9         if i in vokaler:
10             antVok += 1
11     return antVok
12
13 def tellKonsonanter(tekst):
14     antKons = 0
15     konsonanter = 'bcd fghjklmnpqrstvwxyz'
16     for i in tekst:
17         if i in konsonanter:
18             antKons += 1
19     return antKons
20
21
22 tekst = input('Skriv inn en tekst som skal telles: ')
23
24 vokaler = tellVokaler(tekst.lower())
25 konsonanter = tellKonsonanter(tekst.lower())
26
27 print(f'{vokaler} vokaler, og {konsonanter} konsonanter.')
28

Shell
>>>
>>>
>>>
>>>
```



Syntaksfeil

```
1 #Syntaksfeil
2
3 a = 5
4 b = 7
5 print(a+c)
6
```

Shell x

Python 3.7.9 (bundled)

```
>>> %cd 'C:\Repositories\itgk-forberedelser-host-2021\uke34'
```

```
>>> %Run slides-syntaksfeil.py
```

Traceback (most recent call last):

File "C:\Repositories\itgk-forberedelser-host-2021\uke34\slides-syntaksfeil.py", line 5, in <module>

```
    print(a+c)
```

NameError: name 'c' is not defined

```
>>>
```



Logiske feil

```
1 def beregn(a,b):  
2     return a*b+1  
3  
4 print(f'2+2 = {beregn(2,2)}')
```

Shell x

```
>>> %Run Logisk.py
```

```
2+2 = 5
```

```
>>>
```



Matte i Python – praktisk

Matematikk	Python	Merknad
$9 + 4$	<code>9 + 4</code>	Addisjon
$9 - 4$	<code>9 - 4</code>	Subtraksjon
$9 \cdot 4$	<code>9 * 4</code>	Multiplikasjon
$9 : 4$	<code>9 / 4</code>	Divisjon (blir 2.25)
	<code>9 // 4</code>	Heltallsdivisjon (blir 2)
	<code>9 % 4</code>	Modulo, rest (blir 1)
2^3	<code>2 ** 3</code>	Potens
$2a$	<code>2 * a</code>	*-tegn MÅ skrives eksplisitt
$(5 - 2) \cdot 4$	<code>(5 - 2) * 4</code>	Bruk parentes hvis matem. uttrykk har parentes
2^{a+b} ; $\frac{a+b}{c-d}$	<code>2 ** (a+b)</code> ; <code>(a+b)/(c-d)</code>	...pluss hvis uttrykk har "skjulte" parenteser





02

Print og input

Print-statement

- Spørre om input fra bruker:

1 - eksempel-eksamen.py × uke34.py × slides-input.py ×

```
1 a = input('Skriv noe: ')\n2 print('Du skrev: ' + a)
```

Shell ×

```
>>> %Run slides-input.py
```

```
Skriv noe: Hei
```

```
Du skrev: Hei
```

```
>>> |
```

1 - eksempel-eksamen.py × uke34.py × slides-input.py ×

```
1 a = input('Tall 1: ')\n2 b = input('Tall 2: ')\n3 print(a+b)
```

Shell ×

```
>>> %Run slides-input.py
```

```
Tall 1: 5
```

```
Tall 2: 4
```

```
54
```

- Blir alltid lagret som tekst – eksempel:



Variabler

- variabelNavn = 'verdi'
- variabel-navn = 65
- Slik som i matematikk
- En 'boks' som man putter en verdi i



Oppgaver

https://github.com/magnustvilde/python_imc

