FØRSTE GRUPPETIME
INF1820

tobiaaa@ulrik.uio.no
andrekaa@ulrik.uio.no

Obligformat

```
1  #!/usr/bin/env python
2  # -*- encoding:utf-8 -*-
3
4  # Oppgave 1.
5
6  svar = 'ok'
7
8  print("oppgave 1 er", svar)
9
10 # Oppgave 2.
11
12 osv = ''
13
14 # Slutt
```

Kommentarer i kode

```
1  #!/usr/bin/env python
2  # -*- encoding:utf-8 -*-
3
4  '''
5  Trenger du flere
6  linjer med kommentarer?
7  '''
8
9  # linjekommentar
```

Python 3

https://www.python.org/

NLTK

http://www.nltk.org/

Variabler & tilordning

- Variabelnavn kan være nesten hva som helst alfanumerisk, men

Variabler & tilordning

- Variabelnavn kan være nesten hva som helst alfanumerisk, men
  - visse ord som *for*, *if* osv er reservert.

Variabler & tilordning

- Variabelnavn kan være nesten hva som helst alfanumerisk, men
  - visse ord som *for*, *if* osv er reservert.
  - kan ikke starte med tall.

Variabler & tilordning

- Variabelnavn kan være nesten hva som helst alfanumerisk, men
  - visse ord som *for*, *if* osv er reservert.
  - kan ikke starte med tall.
- Python evaluerer uttrykk til høyre for $=$, for så å tilordne verdien til variablen til venstre.

```
>>> x = ''streng''
>>> x = 17
>>> x = True
>>> x = 8.1
>>> x = x + 4
>>> x
12.1
>>>
```

Variabler & tilordning

- Variabelnavn kan være nesten hva som helst alfanumerisk, men
  - visse ord som *for*, *if* osv er reservert.
  - kan ikke starte med tall.
- Python evaluerer uttrykk til høyre for $=$, for så å tilordne verdien til variablen til venstre.

```
>>> x = ''streng''
>>> x = 17
>>> x = True
>>> x = 8.1
>>> x = x + 4
>>> x
12.1
>>>
```

- Variabler deklareres ikke for type, men all data har en type.

Variabler & tilordning

- Variabelnavn kan være nesten hva som helst alfanumerisk, men
  - visse ord som *for*, *if* osv er reservert.
  - kan ikke starte med tall.
- Python evaluerer uttrykk til høyre for $=$, for så å tilordne verdien til variablen til venstre.

```
>>> x = ''streng''
>>> x = 17
>>> x = True
>>> x = 8.1
>>> x = x + 4
>>> x
12.1
>>>
```

- Variabler deklareres ikke for type, men all data har en type.
- Variabler er ikke sjølve dataen, men refererer til data.

Datatyper

- Heltall

```
>>> a = 3
```

Datatyper

- Heltall

```
>>> a = 3
```

- Desimaltall

```
>>> b = 7.9
```

Datatyper

- Heltall

```
>>> a = 3
```

- Desimaltall

```
>>> b = 7.9
```

- Strenger

```
>>> c = ''c'' # merk at dette ikke er char
```

Datatyper

- Heltall

```
>>> a = 3
```

- Desimaltall

```
>>> b = 7.9
```

- Strenger

```
>>> c = ''c'' # merk at dette ikke er char
```

- Boolske verdier

```
>>> g = True
>>> h = False
```

Datastrukturer

- Tupler

```
>>> d = (1, 1.2, 'd')
```

Datastrukturer

- Tupler

```
>>> d = (1, 1.2, 'd')
```

- Lister

```
>>> e = [12, 1.3, (1, 1.2, 'd'), 'e']
```

Datastrukturer

- Tupler

```
>>> d = (1, 1.2, 'd')
```

- Lister

```
>>> e = [12, 1.3, (1, 1.2, 'd'), 'e']
```

- Dictionaries/Hashmap

```
>>> f = {1: 3, 2: 7.9, 3: 'c'}
```

Typeavklaring

- type() & isinstance()

```
>>> type(a)
<type 'int'>
>>> isinstance(a, int)
True
>>>
```

Typeavklaring

- type() & isinstance()

```
>>> type(a)
<type 'int'>
>>> isinstance(a, int)
True
>>>
```

```
>>> type(e)
<type 'list'>
>>> isinstance(e, (int,float))
False
>>> isinstance(e, (int, float, list))
True
>>>
```

Pythonaritmetikk

```
>>> c = c*2
>>> print(c)
[1, 2, 'tredje', 4, 1, 2, 'tredje', 4]
>>> d = [5]*5
>>> print(d)
[5, 5, 5, 5, 5]
>>> d = d + [1, 1, 1]
>>> print(d)
[5, 5, 5, 5, 5, 1, 1, 1]
```

Funksjoner

```
>>> print((1.3*74)/(1.80**2.5))
22.1306316292
>>> print((1.3*73)/(1.79**2.5))
22.137757967
>>>
```

Funksjoner

```
>>> print((1.3*74)/(1.80**2.5))
22.1306316292
>>> print((1.3*73)/(1.79**2.5))
22.137757967
>>>
```

```
>>> def bmi(vekt, hoyde):
...     return (1.3*vekt)/(hoyde**2.5)
...
>>> bmi(60, 1.69)
21.007667798746546
>>>
```

bmi.py

```
1  #!/usr/bin/env python
2  # -*- encoding:utf-8 -*-
3
4  def bmi(vekt, hoyde):
5          return (1.3*vekt)/(hoyde**2.5)
```

bmi2.py

```
1  #!/usr/bin/env python
2  # -*- encoding:utf-8 -*-
3
4  import sys
5
6  vekt = int(sys.argv[1])
7  hoyde = float(sys.argv[2])
8
9  print((1.3*vekt)/(hoyde**2.5))
```

Løkker

- if

```
if <betingelse>:
    <kode>
```

Løkker

- if

```
if <betingelse>:
    <kode>
```

One-liner:

```
if <betingelse>: <kode>
```

Løkker

- if

```
if <betingelse>:
    <kode>
```

One-liner:

```
if <betingelse>: <kode>
```

Ulike betingelser:

SYNTAKS	SEMANTIKK
z == w	z er like w.
z != w	z er ikke lik w.
z > w	z er større enn w.
z < w	z er mindre enn w.
z >= w	z er større enn, eller lik w.
z <= w	z er mindre enn, eller lik w.

Løkker forts.

- if

```
>>> g = [5, 6, 7, 8, 9]
>>> 5 in g
True
>>> if g[0] < g[2]: print(g[2])
7
>>> if g[0] < g[2] and g[1] < g[2]:
...:     print(True)
...:else:
...:     print(False)
True
>>> if g[0] > g[2] or g[1] > g[2]:
...:     print(True)
...:else:
...:     print(False)
False
```

Løkker forts.

- for

```
>>> g = [5, 6, 7]
>>> for i in g:
...:     print(i)
5
6
7
>>> for i in range(len(g))
...:     g[i] = g[i]*2
>>> print(g)
[10, 12, 14]
>>> for i in g:
...:     g[i] = g[i]*2
...:
```

Oppgaver

for den erfarne:
`http://www.nltk.org/book/ch01.html`

for alle andre:
`http://openbookproject.net/thinkcs/python/english2e/`
Kap. 1 2-6, kap. 2 1-4,6-7, kap. 3 1-4.