



UiT Norges arktiske universitet

Funksjoner– Forelesning 6

BED-1304 (Python-lab), 7.5 ECTS

Markus J. Aase

markus.j.aase@uit.no, kontor 02.411

Universitetslektor i matematikk og statistikk

Handelshøgskolen, UiT

Økonomi og administrasjon og samfunnsøkonomi med datavitenskap



Basert på notater av: Charles Severance, University of Michigan
School of Information

Endringer/tilpasninger: Markus J. Aase, Universitetet i Tromsø

Logikk og løkker

- Essensen av programmering er logikk
- Ofte vil vi at koden skal:
 - Repeteres en rekke ganger
 - Repeteres basert på en betingelse
 - Også videre



If/else setninger

- I Python brukes **if/else**-setninger for å ta beslutninger.
- Eksempel:

```
if betingelse:  
    # Kode som kjøres hvis betingelsen er sann  
else:  
    # Kode som kjøres hvis betingelsen er usann
```

Legg merke til kolon



Legg merke til innrykk



If/else setninger

- I Python brukes if/else for å ta beslutninger.
- Eksempel:

```
x = 10
if x > 5:
    print('x er større enn 5')
else:
    print('x er mindre eller lik 5')
```

«Hvis ikke» – altså, når x ikke er større enn 5. Skal koden gjøre...

Dette!

«Hvis x > 5» – altså, når dette er sant. Da skal koden gjøre...

Dette!

Hva med elif?

- elif brukes for flere betingelser.
- Eksempel:

Tilordning!

```
x = 10
```

```
if x > 10:
```

```
    print('x er større enn 10')
```

```
elif x == 10:
```

```
    print('x er lik 10')
```

```
else:
```

```
    print('x er mindre enn 10')
```

Sammenligning!

Hvis $x > 10$: gjør dette

Hvis $x = 10$: gjør dette

Hvis $x < 10$: gjør dette

Logikk og løkker

- **Betingelser:** styrer flyten i koden
- **Sammenligninger:**
`==, <, >, !=, >=, <=`
- **Logiske operatører:** and, or, not
- **Brukes for å ta beslutninger i koden**

```
# Tilordning (=)
x = 5

# Sammenligning (==)
if x == 5:
    print("x er lik 5")
else:
    print("x er ikke 5")
```

Logikk og løkker

- **Betingelser:** styrer flyten i koden
- **Sammenligninger:**
==, <, >, !=, >=, <=
- **Logiske operatoren:** **and**, **or**, **not**
- **Brukes for å ta beslutninger i koden**

Tilordning!

Logisk operator!

```
x = 7  
y = 3
```

```
if x > 5 and y < 5:  
    print("Begge betingelser er sanne")  
elif x > 5 or y > 5:  
    print("Minst én betingelse er sann")  
else:  
    print("Ingen betingelser er sanne")
```

For-løkke

- **For**-løkker brukes til å iterere over sekvenser.
- Eksempel:

```
for i in range(5):  
    print(i)
```

- Dette skriver ut tallene 0 til 4.

For-løkke

- For-løkker brukes til å iterere over sekvenser.
- Eksempel:

```
for i in range(1, 10):  
    print(i)
```

- Dette skriver ut tallene 1 til 10.

For-løkke

- For-løkker brukes til å iterere over sekvenser.
- Eksempel:

```
liste = [1, 2, 3, 4, 5, 6]
for i in liste
    print(i)
```

- Dette skriver ut tallene i *liste*.




range()
Liste
Tuple
String

Ser nærmere på **in**...

- **Iterasjonsvariabelen** “itererer” gjennom **sekvensen** (f.eks. en liste)
- **Blokken (body)** til koden blir kjørt én gang for hver verdi **in sekvensen**.
- **Iterasjonsvariabelen** beveger seg gjennom alle elementene **in sekvensen**.

Iterasjonsvariabel sekvens



```
for i in [1, 2, 3, 4, 5] :  
    print(i)
```

Enda et eksempel

```
venner = ['Markus', 'Guro', 'Øyvind']  
for friend in friends :  
    print('God jul:', friend)  
print('Ferdig!')
```

God jul: Markus
God jul: Guro
God jul: Øyvind

Ferdig!

Hva er det største tallet?

3 41 12 9 74 15

Hva er det største tallet?

3 41 12 9 74 15

største_tall

74

Hvordan kan vi kode dette?

```
største_tall = -1
print('Før løkka er det største tallet',
      største_tall)
for tall in [9, 41, 12, 3, 74, 15] :
    if tall > største_tall:
        største_tall = tall
    print(største_tall, tall)

print('Etter løkka er det største
tallet:', største_tall)
```

Før løkka er det største
tallet -1

9 9

41 41

41 12

41 3

74 74

74 15

Etter løkka er det største
tallet: 74

While-løkke

- **While**-løkker kjører så lenge en betingelse er sann.
- Eksempel:

```
while betingelse:  
    # Kode vil kjøre så lenge  
    betingelse er True
```


While-løkke

- While-løkker kjører så lenge en betingelse er sann.
- Eksempel:

I while-løkker, er det viktig å initiere løkka med en variabel, f.eks. sånn her

```
i = 5
```

```
while i > 0:
```

```
    print(i)
```

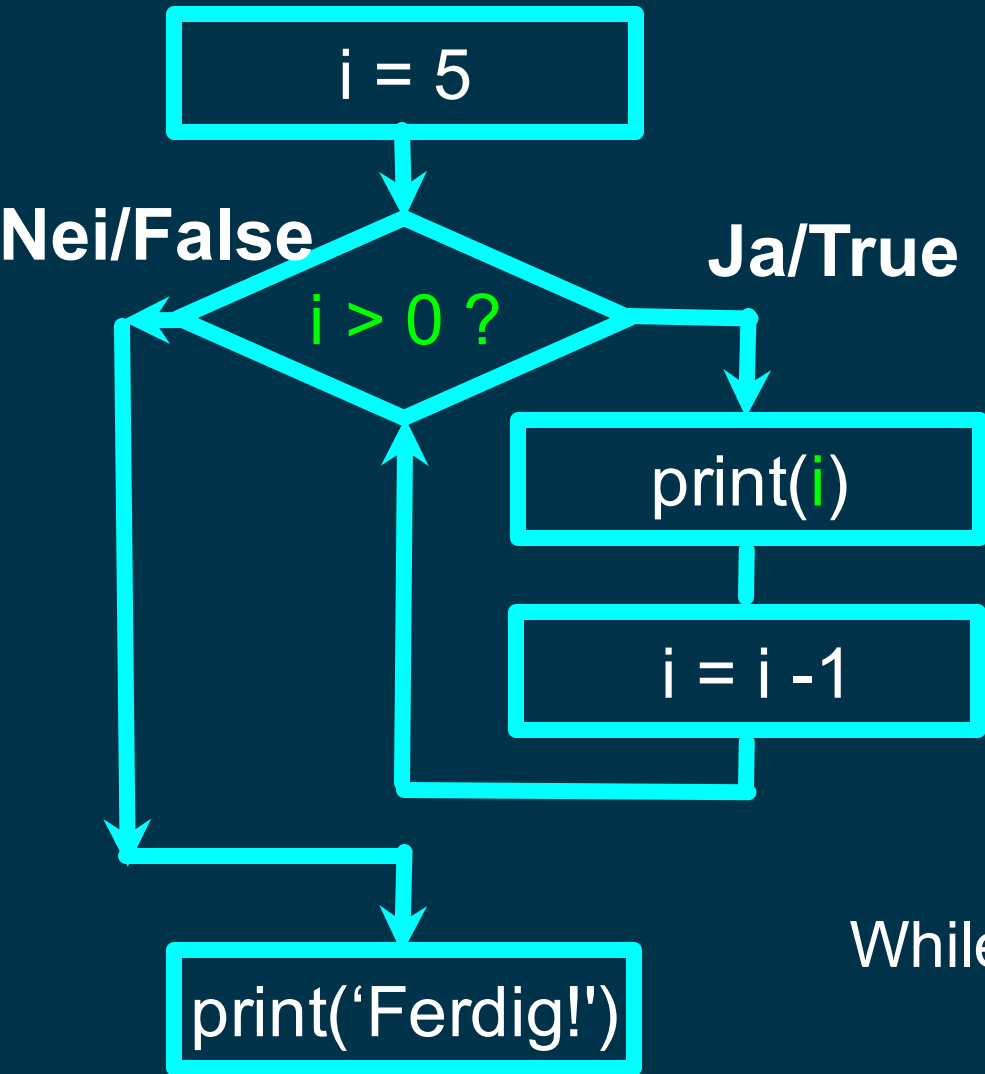
```
    i = i - 1
```

Så lenge dette er sant (True). Gjør koden...

Dette!

Dette er viktig, for å unngå uendelige løkker («infinite loops»)

While-løkker



Kode:

```
i = 5
while i > 0 :
    print(i)
    i = i - 1
print('Ferdig!')
```

Output:

5
4
3
2
1
Ferdig!

While løkker har **iterasjonsvariabel** som endrer seg hver gang gjennom løkka. Ofte iterer disse **iterasjonsvariabelene** gjennom en sekvens av tall.

Arbeidsoppgaver

- Gå igjennom forelesningsnotatene i Jupyter Notebook
 - Tekst, teori, oppgaver
- Seminaroppgaver
 - Oppgaver dere kan jobbe med på seminar, og egenstudium
- Repetisjonsark
 - Masse oppgaver med løsningsforslag

Spørsmål?