

Fra Python til Java

En introduksjon til
programmeringsspråkenes verden

Dag Langmyhr
dag@ifi.uio.no

Oversikt

- Introduksjon
- Python ⇒ Java
 - Noe er likt
 - Noe bare ser anderledes ut
 - Noe er helt forskjellig
- Et eksempel
- Klasser (neste time ved *Stein Gjessing*)

Det finnes ikke bare ett programmeringsspråk

10 000-er av programmeringsspråk

Hvorfor finnes det så mange?

- Det finnes mange ulike behov; f eks

Python bash Excel

- Nye behov oppstår
- Vi kan lage bedre programmeringsspråk enn for 50 år siden

Ikke veldig vanskelig

Om dere programmerer en del og tar emnene

- INF2100
- INF3110
- INF5110

kan dere lage deres eget programmeringsspråk.

Mitt forsøk:

```

◦ Syntax_expression: { |
  ◦ parts: §[1];

  ◦ width: { |
    ◦ w: 0.max(| @ #.parts: f.width() |);
    ∫ #.parts.n().>(1): w.①+(¶.h_sep.x(2)) ?;
    ¥ w;
  | };

  ◦ height: { |
    ¥ #.parts[1].height()
  | };

  ◦ depth: { |
    ¥ ∫ #.parts.n().=(1):
      #.parts[1].depth()
    | True:
      #.parts[1].depth().
      +(| @ #.parts[2..]: f.height().+(f.depth()) |).
      +(#.parts.n().--().x(2,¶.v_sep))
    ?
  | };
  
```

Hvorfor finnes Python?

Python ble opprinnelig laget i 1989 som et hobbyprosjekt av *Guido van Rossum* med denne filosofien:

- Beautiful is better than ugly
- Explicit is better than implicit
- Simple is better than complex
- Complex is better than complicated
- Readability counts

Hvorfor ble Java laget?

Java ble laget i 1996 av *James Gosling* og andre i SUN Microsystems (nå kjøpt av Oracle) med følgende mål:

- It must be “simple, object-oriented, and familiar”.
- It must be “robust and secure”.
- It must be “architecture-neutral and portable”.
- It must execute with “high performance”.
- It must be “interpreted, threaded, and dynamic”.



Hvis noen er interessert i hva jeg mener ...

Min personlige mening

Python er best til

- nybegynneropplæring
- korte, enkle programmer jeg trenger *nå*
- kobling mot nyttige programpakker i kjemi, fysikk, matematikk, ...

Java er best til

- objektorientert programmering
- store programmer
- trygge programmer
- raske programmer



Noe er helt likt

Uttrykk

Uttrykk er svært like i de to språkene:

Python

```
2 + v*(t+1)
2 * 3.1416 * r
```

Java

```
2 + v*(t+1)
2 * 3.1416 * r
```

Tilordning

Tilordning (dvs gi en variabel en verdi) er også likt:

Python

```
n = 2 + v*(t+1)
omkr = 2 * 3.1416 * r
```

Java

```
n = 2 + v*(t+1)
omkr = 2 * 3.1416 * r
```



Noe ser anderledes ut men er egentlig likt

if-tester

If-tester bruker *krøllparenteser* i stedet for innrykk, og alle setninger avsluttes med semikolon:

Python

```
if v < 0:
    v = -v
```

Java

```
if (v < 0) {
    v = -v;
}
```

Dette betyr at linjeskift og innrykk ikke betyr noe i et Java-program:

```
if (v < 0) { v = -v; }
```

```
if (
    v < 0) { v = -
v; }
```



If-tester kan ha flere alternativer:

Python

```
if x < 0:
    f = -1
elif x == 0:
    f = 0
else:
    f = 1
```

Java

```
if (x < 0) {
    f = -1;
} else if (x == 0) {
    f = 0;
} else {
    f = 1;
}
```

Krøllparenteser brukes også i while- og for-løkker:

Python

```
while x <= 255:  
    total = total + x  
    x = 2 * x  
  
for i in range(1,10):  
    sum = sum + i
```

Java

```
while (x <= 255) {  
    total = total + x;  
    x = 2 * x;  
}  
  
for (int i = 1; i <= 10; i++) {  
    sum = sum + i;  
}
```

Utskrift

I java finnes to metoder for utskrift:

System.out.print skriver ut parameteren; hvis det er flere elementer, skjøtes de sammen med +.

System.out.println gjør det samme, men setter på et linjeskift.

Python

```
r = 2.0
v = 3.1416 * r * r
print("r =", r, end=" ")
print("gir areal", v)
```

Java

```
double r = 2.0;
double v = 3.1416 * r * r;
System.out.print("r = " + r);
System.out.println(" gir areal " + v);
```



Innpakning

I Java må alt ligge i en klasse

```
class MinKlasse {  
    :  
}
```

Funksjonen **main** starter det hele, og den må alltid deklarereres som

```
public static void main(String[] arg)
```

```
class Hei {  
    public static void main(String[] arg) {  
        System.out.println("Hei!");  
    }  
}
```



Hvordan får vi det hele i gang?

Kjøring

I Java brukes to programmer:

javac oversetter **.java**-filen til én eller flere **.class**-filer.

java utfører **.class**-filen(e)

```
$ javac Hei.java
$ java Hei
Hei!
$
```

```
class Hei {
    public static void main(String[] arg) {
        System.out.println("Hei!");
    }
}
```



Angivelse av variabeltype

- Python har **dynamisk typing** der variablene kan tilordnes verdier av ulike typer.
- Java har **statisk typing** der alle variabler har type angitt av deklarasjonen. De kan bare få verdier av den typen.

Python

```
i = 5
v = 2.5
v = v + i
s = "Svaret er"
print(s, v)
```

Java

```
int i = 5;
double v = 2.5;
v = v + i;
String s = "Svaret er";
System.out.println(s + " " + v);
```



Java har mange typer å velge blant

Javas typer

De viktigste typene i Java er:

int heltall (dvs 0, 1, 2, -3, ...)

double flyt-tall (dvs 0.0, 1.5, -22.7, 3.14, ...)

boolean logisk verdi (dvs **true** og **false**)

char enkelttegn (dvs 'a', 'b', '?', '5', ...)

String tekst (dvs "x", "abcd", "Hei!", ...)

I tillegg kommer klasser (som omtales i neste time).

Finn medianen (dvs det midterste) av tre tall

```
def finnMedian(a):
    return sorted(a)[1]

data = [0] * 3
leser = open("tall.data")
for i in range(3):
    data[i] = int(leser.readline())
print("Medianen er", finnMedian(data))
```

```
import java.io.File;
import java.util.Arrays;
import java.util.Scanner;

class Median {
    private static int finnMedian(int[] a) {
        Arrays.sort(a);
        return a[1];
    }

    public static void main(String[] arg) throws Exception {
        int[] data = new int[3];
        Scanner leser = new Scanner(new File("tall.data"));

        for (int i = 0; i < 3; i++) {
            data[i] = Integer.parseInt(leser.nextLine());
        }
        System.out.println("Medianen er " + finnMedian(data));
    }
}
```



Det er litt jobb å lære seg et nytt programmeringsspråk

Konklusjon

- Det finnes ikke noe beste språk for alle formål
- Det er en styrke å kunne flere språk
- Alle informatikere må regne med å lære flere språk
- Det er egentlig bare én måte å lære et nytt programmeringsspråk:
 - 1 Les dokumentasjonen
 - 2 Prøv selv med mange eksempler
 - 3 Om ikke alt fungerer som forventet, gjenta.