

Java for deg som kan Python

IN1010

23.1.2023, 2. time

Siri Moe Jensen

Oversikt

- Hvorfor Java?
 - Python \Rightarrow Java
 - Kom i gang
-
- I morgen: Klasser ved Stein Gjessing
 - Neste uke: 2 * 2 timer forelesning om Java
 - Deretter (uke 3): Arv

Innhold

Introduksjon til Java

- Hvorfor flere språk
- Hva er et objektorientert språk
- Java: Pluss og minus
- Hva trenger jeg for å programmere i Java i IN1010?

Java versus Python

- Et komplett program
- Variabler, verdier og typer
- Skrive til terminal

Språk lages med ulike utgangspunkt

Java (James Gosling/ Sun Microsystems; 1996)

Mål: It must

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

Python (Guido van Rossum; 1989)

Filosofi

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

Sentrale anvendelsesområder

Python

- Data Science & Data Analytics
- Kunstig intelligens (AI), dyp læring
- .. og mye mer

- Stadig vanligere:
Introduksjonskurs i programmering

Java

- Enterprise Application utvikling
- Android app-utvikling
- ...og mye mer

Det finnes ikke ett "beste språk"

Mange faktorer avgjør

- Anvendelse (domene og oppgaver)
- Eksisterende arkitektur, kode og kompetanse
- Personlige preferanser
- Trender i business (vid forstand), samfunn og teknologi
- Teknologeutvikling, utvikling av språkene

Kontinuerlig utvikling

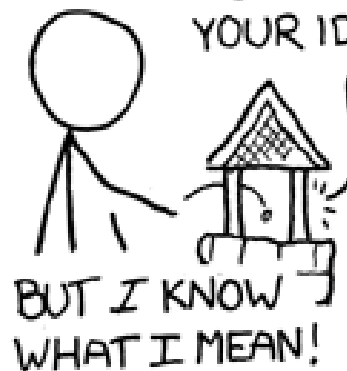
Men:

- Java og Python er begge blant topp 3 på de fleste rangeringer av programmeringsspråk: Antall søk, antall arbeidsannonser, popularitet blant utviklere, etc.

Hvorfor lære flere programmeringsspråk?

- Python er et enkelt språk å komme i gang med, og relativt lett å lese
- Men det betyr også at endel sentrale aspekter – og noen utfordringer - kamufleres eller unngås
- I Java ser du mer direkte "hva som skjer" – og blir kjent med flere sentrale programmeringsmekanismer
- Du vil høyst sannsynlig ha nytte av disse – og andre språk
 - For bruk i videre studier eller jobb
 - For å vite hva som er generelt – og spesielt for ett språk
 - For å bli kjent med flere måter å "se" og løse oppgaver på
 - For å lære språk nummer 3 og 4 enda raskere 😊

YOU'LL NEVER FIND A
PROGRAMMING LANGUAGE
THAT FREES YOU FROM
THE BURDEN OF
CLARIFYING
YOUR IDEAS.



"Java og Python er objektorienterte språk"

- Hva betyr det?
 - ? Vi kan definere klasser og opprette objekter av disse *begge*
 - ? Vi kan bruke en objektorientert tankegang i modellering og design *begge*
 - ? Alle data er (egentlig) representert som objekter *Python*
 - ? All kode ligger i klasser, dvs vi må skrive minst én klasse *Java*
- Et Javaprogram har *all* kode inne i en eller flere klasser
 - Vi skal bruke *statiske* metoder i Java (finnes også i Python, men er ikke pensum i IN1000)
 - En statisk metode kan brukes selv om vi ikke har noen objekter av klassen – den hører ikke til ett objekt, men til klassen
 - Mer om dette i morgen

Kjøring av Java-programmer

- Du må installere Java Development Kit (JDK)
- Du trenger to programmer for å kjøre et Java-program
 - **javac** oversetter (kompilerer) .java-filen(e) til en eller flere .class-filer
 - **java** utfører .class-filen(e)

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

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

Installasjon av Java: Se Tekniske ressurser og hjelp på semestersiden

Fra Python til Java

Gitt følgende Python program – hvordan skriver og kjører vi (nesten) samme program i Java?

```
ant_programmer = 100  
print("Dette er mitt " + str(ant_programmer) + ". Python-program")
```

Eksempel i Java (kommentarer i senere slides)

```
class Program {  
    public static void main(String[] args) {  
        int antallProgrammer;  
        antallProgrammer = 1;  
        System.out.println("Dette er mitt" + antallProgrammer + ". Java-program!");  
        System.out.print("Denne linjen venter på .println() kall eller programslett");  
    }  
}
```

Innpakning

- I Java må alt ligge i klasser
- Utføring av programmet starter alltid i en metode som heter main – deklarerer slik:

```
class Program {  
  
    public static void main(String[] args) {  
  
    }  
  
}
```

- Filen må ha samme navn som klassen, her **Program.java**

Blokker og programsetninger

- I Java brukes krøllparenteser (`{ }`) i stedet for innrykk.
- Alle Java-setninger avsluttes med semikolon (`;`).

```
if (antallProgrammer > 0) {  
    System.out.println("Dette er mitt " + antallProgrammer + ". Java-program!");  
}
```

- Linjeskift og innrykk betyr altså ikke noe i et Java-program:

```
    if (antallProgrammer  
> 0) {  
        System.out.println  
            ("Dette er mitt " + antallProgrammer + ". Java-program!");  
    }
```

Angivelse av variabeltype

- Python har *dynamisk typing* der typen er knyttet til verdiene ; derfor kan en variabel tilordnes verdier av en hvilken som helst type.
- Java har *statisk typing* der både variablene og verdiene har type; variabler kan bare få verdier av den riktige typen.
- Derfor: *I Java må alle variabler deklarereres.*

```
int antallProgrammer = 1;  
double pris;  
pris = 3.25;  
boolean ferdig = true, ok;
```

```
System.out.println("Pris = " + pris + " og ferdig = " + ferdig);
```

```
>java Program  
Pris = 3.25 og ferdig = true
```

Javas viktigste typer

int	heltall (dvs 0, 255, -3, ...)	
double	flyttall (dvs 0.0, 1.5, -22.873, ..)	
boolean	logisk verdi (dvs true , false)	
char	enkeltegn (dvs ' 1 ', ' a ', ' * ', ...)	NB: enkel apostrof (fnutt)


Dessuten finnes klassen

String	tekst (dvs " abcd ", "", " \$50 ", " 146 ", ..)	NB: dobbel apostrof
---------------	--	---------------------

Utskrift

I java finnes to metoder for utskrift:

- `System.out.print` skriver ut parameteren (kun en parameter!)
 - Hvis det er flere elementer skjøter du dem sammen med +
 - Eller man kan kalle `System.out.print` flere ganger
- `System.out.println` gjør det samme, men legger til linjeskift.



samme resultat!

```
System.out.print("Navn: " + "Donald Duck");  
System.out.println();
```

```
System.out.println("Navn: " + "Donald Duck");
```


if-tester kan ha flere alternativer

```
if (antallProgrammer== 0) {  
    System.out.println("Dette er mitt null'te Java-program!");  
} else if (antallProgrammer == 1) {  
    System.out.println("Dette er mitt forste Java-program!");  
} else {  
    System.out.println("Dette er mitt " + antallProgrammer + "Java-program!");  
}
```

Legg merke til:

- Java har parenteser rundt if-testen (i stedet for kolon etter)
- Java har ingen **elif** – i stedet skriver vi **else if**

Løkker

While-løkker

```
while (antallProgrammer < 10000) {  
    // Fortsett å programmere i Java  
    antallProgrammer += 1;  
}
```

For-løkker med teller (erstatte `... in range(10000)` i Python)

```
for (int i = 0; i < 10000; i++) {  
    // Skriv ett program til i Java  
}
```

Hvor mange feil finner du?

Feil.java

```
class Feil {  
    public static void main (String[] args) {  
        pris = 25;  
        pris = 2*pris  
        if pris > 200 {  
            pris = 200;  
        }  
        else:  
            pris = pris + 50;  
        print("Pris: " + pris);  
    }  
}
```

Et komplett lite eksempel

```
FeilUtenFeil.java
1 class FeilUtenFeil {
2     public static void main (String[] args) {
3         int pris = 25;
4         pris = 2*pris;
5         if (pris > 200) {
6             pris = 200;
7         }
8         else {
9             pris = pris + 50;
10        }
11        System.out.println("Pris: " + pris);
12    }
13 }
```

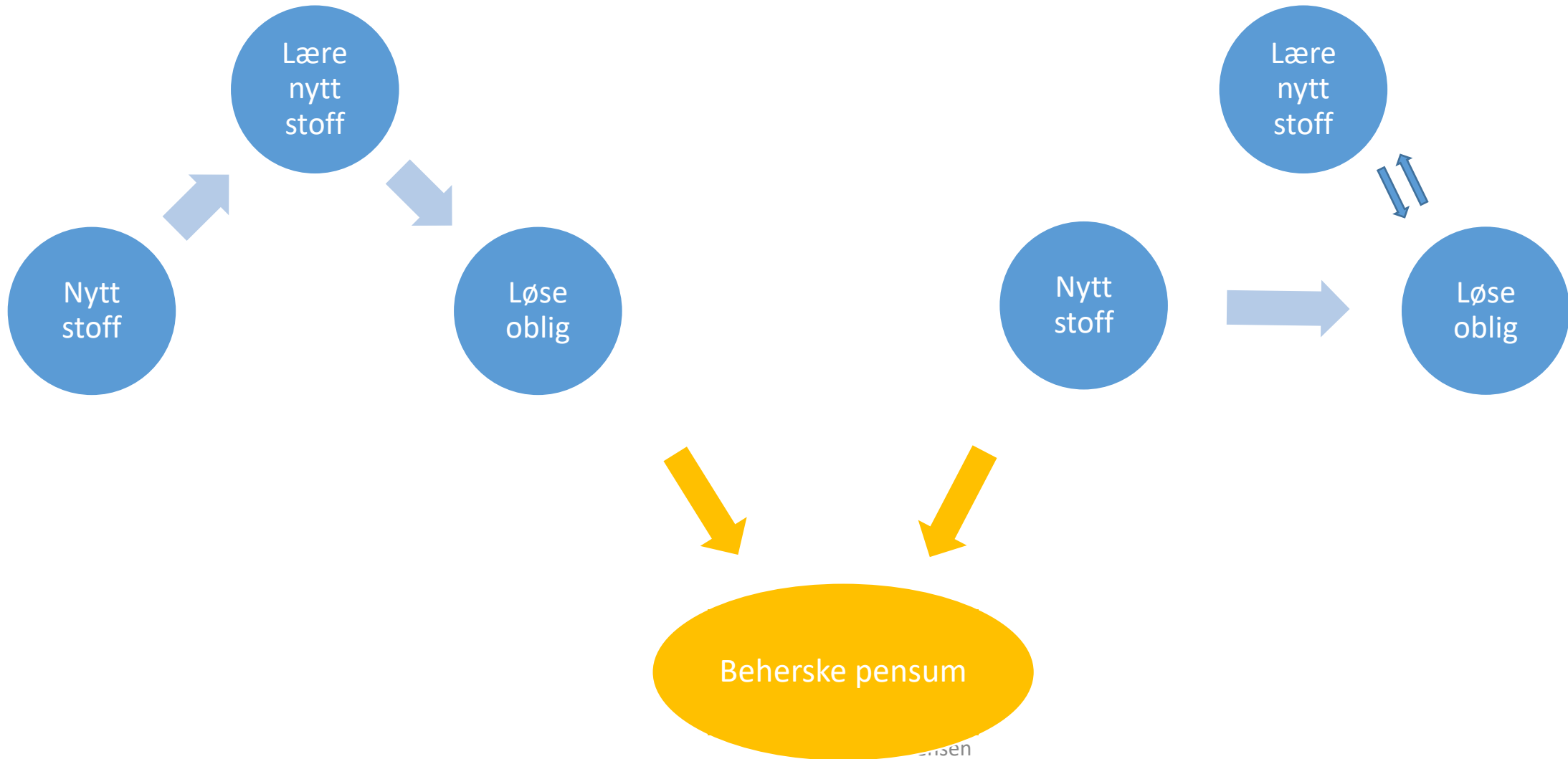
```
$ javac FeilUtenFeil.java
```

```
$ java FeilUtenFeil
```

Komme i gang

- Viktige lenker fra semestersiden (under [Tekniske ressurser og hjelp](#) og [Uke1](#)):
 - Installasjon og oppsett av Java
 - Java for menneskelige lesere (inkl deg selv) – Kodestil i Java
 - Programeksemples, Trix-oppgaver
- Under [Notater](#): Engelsk – norsk ordliste for IN1000 og IN1010
- Mer om programmering i Java denne uken og neste
 - Klasser og objektorientert programmering i morgen
 - Arrayer, metoder, input fra terminal og filhåndtering neste mandag
 - Mer om datastrukturer og representasjon neste tirsdag
 - Trening og flere detaljer i seminar og oppgaver inkl *Innlevering 0* (frist 30.1)!
- Nyttig: [Java Tutorials](#) og [Java API](#) (google [java 8 api detteskaljegbruke](#))

Hvilken vei tror du er mest effektiv?



Effektivitet

- Hva er målet?
 - Lære å programmere i Java, forstå konseptene og mekanismene som brukes
 - Solid grunnlag for videreutvikling i flere retninger
 - Ikke å utvikle spesialiserte programmer i et konkret produksjonsmiljø
- Bruk din egen tid best mulig
 - Skriv mange programmer
 - Men sørg for at du lærer (evt trener på) noe for hvert program
 - ⇒ jo mindre og senere hjelp, desto mer læring – som sitter
finn feil og svar på spørsmål selv før du bruker gruppelærer, løsningsforslag, ..
- Bruk gruppelærers tid best mulig
 - Følg undervisningen, bruk ressursene inkl semestersiden
 - Bruk repetisjonsgruppen tirsdag 12:15-14 ved behov
 - Spør i felles fora: Gruppetimer før lab, Discourse-kanaler ikke epost

Oppsummering

- 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
- Flere språk gir dypere forståelse

- Det er egentlig bare én måte å lære et nytt programmeringsspråk på:
 - les dokumentasjonen (lærebok, notater, Java Tutorials og Java 8 API)
 - prøv selv med mange eksempler (evt i pythontutor – velg Java som språk)
 - om ikke alt fungerer som forventet, gjenta

- Etter uke 2 regner vi med at du er "flytende" i Java