

# DA102A—F1: Java vs. Python

Malmö universitet

Institutionen för datavetenskap och medieteknik

2025-??-??

# Innehåll

Repetition

Java

Skillnader

Allmänt

Datatyper

Input/output

Villkorssatser

Loopar

Funktioner & Metoder

Sammanfattning

# Repetition

# Repetition

Saker vi har gjort i DA100A

- ▶ Datatyper
- ▶ Villkorssatser
- ▶ Loopar
- ▶ Funktioner
- ▶ Listor

# Repetition

## Datatyper

Vi har använt datatyperna

- ▶ `int` för heltal
- ▶ `float` för flyttal
- ▶ `bool` för sant/falskt
- ▶ `str` för text
- ▶ `list` för listor (arrayer)

# Input/output

```
1 name = input("Your name: ")
2 age = int(input("Your age: ")) # Vi castar till en int
3
4 print("Hello,", name, "you are", age, "years old.")
```

# Repetition

## Villkorssatser

Vi har skapat villkorsatser så att programmet kan agera utifrån olika situationer

```
1 name = input("May I have your name, please?" )
2 if len(name) > 6:
3     print("Your name is too long, I do not want it.")
4 elif len(name) < 3:
5     print("Your name is too short, not fit for me.")
6 else:
7     print("Thank you, your name is now mine.")
```

# Repetition

## Loopar

Vi har gjort upprepningar med `while` och med `for`

```
1 i = 0
2 while i < 10:
3     print(i)
```

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



# Repetition

## Funktioner

Vi har skapat och anropat funktioner

```
1 def my_function(par1,par2):  
2     # Do stuff  
3     res = par1+par2  
4     return res  
5 a = 1  
6 b = 2  
7 print(my_function(a,b))
```

# Repetition

## Listor

Vi har skapat listor med flera element

```
1 hobbits = ["Sam", "Frodo", "Merry", "Pippin", "Fatty"]
2 print(hobbits[0])
```

```
1 matrix = [[1,2,3],[4,5,6],[7,8,9]]
2 for row in range(len(matrix)):
3     for col in range(len(matrix[row])):
4         print(matrix[row][col])
```

# Java

## Vad är Java?

Innan vi går in på skillnader i syntax, så ska vi kolla lite på Java som språk.

Java är ett kompilerat språk (där Python till är ett interpreterat språk). Det innebär att Java behöver översätta källkoden från text till byte-kod innan vi kan köra programmet. (Python gör detta medans programmet körs)

Du kan inte köra ditt program för ens efter att det är kompilerat.

Under kompileringen så görs en felsökning av koden och hittar den fel så misslyckas kompileringen.

Java är ett strikt objekt-orienterat språk.

## Var används Java?

En gång i tiden var Java sexigt.

Telefon-appar skrevs i Java.

Många system kör Java i sin *back-end* (särskilt äldre system)

*Minecraft* är kodat i Java

# Skillnader

Allmänt  
Datatyper  
Input/output  
Villkorssatser  
Loopar  
Funktioner & Metoder

# Skillnader

## Semikolon

En viktig skillnad mellan Python och Java är att man i Java **måste** markera när en sats är slut.

I Python görs detta automagiskt när du byter rad.

I Java (och C) behöver du markera när en sats är över med ett semikolon “;”

# Skillnader

## Block

I Python markerar du block-tillhörighet med indragningar. (Exempelvis if-satser och loopar)

I Java markerar du block-tillhörighet med *måsvingar* `{}`

Symbolerna `{}` har flera olika namn:

- ▶ Måsvingar (detta är det vedertagna namnet)
- ▶ Krullparanteser
- ▶ På engelska: *curly brackets*
- ▶ På danska: *tuborg-klammer*



# Skillnader

## Kommentarer

```
1 # Detta är en kommentar
2 """
3 Detta är en lång kommentar
4 """
```

```
1 // Detta är en kommentar
2 /*
3 Detta är en lång kommentar
4 */
```

# Skillnader

## Namnkonventioner

I Python använder man ofta understreck i variabelnamn om det är ett namn som innehåller flera ord. Exempelvis: `my_long_name`

I Java använder man istället *mixed case*. Exempelvis: `myLongName`

# Datatyper

I Java deklarerar man datatypen när man initialiserar en variabel.

```
1 int a = 5; // heltal
2 float b = 5.5f // flyttal (32 bitar (7 decimaler)) notera f
3 double c = 5.5 // decimaltal (64 bitar (15/16 decimaler))
4 boolean d = true; // notera liten bokstav
5 String e = "Hej"; // notera citattecken och stor bokstav
6 char f = 'a'; // notera apostrof
```

# Datatyper

Java är ett strikt typat språk. Det innebär att en variabel har en bestämd datatyp redan när den skapas, och den variabeln kan inte innehålla några andra datatyper.

```
1 # Detta är okej Python-kod
2 a = 5
3 a = 5.5
4 a = "Najs"
```

```
1 // Detta är inte okej Java-kod
2 int a = 5;
3 a = 5.5; // Error
4 double a = 5.5; // Error
```

# Datatyper

## Float och double

I Python använder man `float` för decimaltal. Medan Java använder både `float` och `double`

En `float` i Java är sparad med 32 bitar, medan en `double` är sparad i 64 bitar. Detta betyder att du får en högre precision med `double`

Som regel kommer vi att använda `double` för att lagra decimaltal.

# Input/output

```
1 # Python
2 name = input("Name: ")
3 age = int(input("Age: "))
4 print(name, age)
```

```
1 // Java
2 Scanner input = new Scanner(); // Nödvändig rad
3 System.out.println("Name: ");
4 String name = input.nextLine();
5 System.out.println("Age: ")
6 int age = input.nextInt();
7 System.out.println(name + " " + age);
```

# Villkorssatser

## Python

```
1 if a == True and b == False:
2     pass
3 elif c == True or d == True:
4     pass
5 else:
6     pass
```

# Villkorssatser

## Java

```
1  if (a == true && b == false){ // Notera parenteserna
2      // Notera att && betyder och
3  }
4  else if (c == true || d == true){ // Vi skriver ut elif
5      // Notera att || betyder eller
6  }
7  else{
8      // Notera alla { } som markerar block
9  }
```



# Loopar

## While

```
1 # En while-loop i Python
2 i = 0
3 while i < 10:
4     print(i)
5     i += 1
```

```
1 // En while-loop i Java
2 int i = 0;
3 while (i < 10){ // Notera återigen parenteserna runt villkoret
4     System.out.println(i);
5     i++; // Detta motsvarar i += 1
6 }
```

# Loopar

## For

```
1 # For-loop i Python
2 for i in range(0,10,1): # Start, Stop, Steg
3     print(i)
```

```
1 // For-loop i Java
2 for (int i=0; i<10; i++){ // Start, Stop, Steg
3     System.out.println(i); // Notera återigen alla parenteser
4 }
```

# Funktioner

## Funktioner i Python

```
1  # Funktion i Python
2  def my_function(par1, par2):
3      result = par1+par2
4      return result
```

# Funktioner

## Metoder i Java

Java är helt objektorienterat och har istället för funktioner *metoder* (dessa finns i Python också—har du läst kursen Programmering 2 känner du till dem)

Metoder är i princip samma sak som funktioner, men kopplade till objekt.

```
1 // Metod i Java
2 public double myFunction(double par1, double par2){
3     double result = par1+par2;
4     return result;
5 }
```

# Metoder

## Metodhuvudet

```
1 public double myFunction(double par1, double par2){}
```

`public` anger tillgänglighet

`double` (det första) anger returtyp

Parametrarna behöver också ha bestämda datatyper

En metod som inte ska returnera något värde har retur-typen `void`

# Metoder

## Överskuggning

I Python kunde man ange default-parametrar

I Java behöver vi *överskugga* funktionerna

```
1 public double myFunction()
```

# Sammanfattning