

# Java uden brug af IDE

Vi har indtil nu arbejdet med processing i processings egen "ide" (integrated development environment). Det er smart da alt hvad man skal bruge er pakket fint ind og man kan starte sine programmer med et enkelt tryk!

Men nu skal vi prøve at bevæge os lidt om bag "scenen" og se hvad der i virkeligheden sker!

## Varighed:

Cirka 3 moduler. Et modul til gennemgang, et modul til installation,- og et modul til præsentation for klassen.

## Grupper:

I må gerne arbejde sammen. Men installere og skrive kode på deres egen computer!!

## Læringsmål:

- Kunne bevæge sig imellem mapper fra en konsol
- Overordnet forståelse for Javas virkemåde: <https://www.javatpoint.com/java-bytecode>
- Kunne skrive et simpelt java program uden "ide" og selv kompilere og eksekvere programmet fra en konsol

## Installation og opsætning:

### Konsol kommandoer:

For at gå ind i en mappe skriv:

Bevæge sig ind i en mappe i en mappe

Eller skriv hele "stien" f.eks.:

Gå ud af mappe, skriv:

I **mac/linux** se indholdet i en mappe skriv:

I **windows** se indhold af mappe skriv:

Start program skriv `./programnavn`

**`cd mappenavn`**

**`cd mappenavn/andet_mappenavn`**

**`cd c:\user\ajrp\Desktop`**

**`cd ..`**

**`ls`**

**`dir`**

**`./programnavn`**

### Installation af Atom

Vi installere "Atom". Det er en meget avanceret editor, næsten en IDE. Men indtil nu bruger vi den bare som "editor":

<https://atom.io/>

### Installation af Java

Når vi senere skal bruge processing som et bibliotek i java er det nødvendigt at have Java 8, da senere versioner af Java ikke er supporteret af processing, download java 8 og installer fra nedenstående link:

<https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>

eller hent fra teams **"3i / generelt / filer / libs"**

### Java i konsol:

Når java skal anvendes direkte i en konsol, bruger man to kommandoer

- **javac:** der betyder java kompilering. Den bygger såkaldt bytecode
- **java:** der eksekverer/starter en fil i "bytecode"
- **java -version:** Skriv dette i din konsol, så du er sikker på du anvender java 8

*Hvis ikke du har java 8, se nedenfor*

### Java opsætning specielt Windows:

Åben "**Advanced System Settings**" / "**Environment Variables**" / "**System variables**".

Vælg variablen "**Path**" under system variables og tryk "**edit**".

Vælg "**new**" eller "**edit**" og opret en sti, der peger på din nye java.

Husk tryk "**ok**"!!! (vigtigt for at gemme).

Hos mig er stien f.eks. "**C:\ProgramFiles\Java\jdk1.8.0\_301\bin**".

### Java opsætning specielt Mac:

Åben en konsol!

For at se java versioner tast: **/usr/libexec/java\_home -V**

Skift til installeret java 8: **export JAVA\_HOME=\$(/usr/libexec/java\_home -v1.8.0\_301)**

(Selvom versionen er v1.8.0\_301 behøver man kun skrive v1.8 resten "gætter" konsollen)

**Test:** Start en ny konsol og tjek om kommandoerne "java" og "javac" virker.

Skriv også "java -version" og se om du nu har java 8!!

### Opgaven:

1. I skal lave en en fil "MitProgram.java" i Atom
2. Skriv et java program der kan:
  - a. Almindelig opgave: Udskrive 10 tal fra et array i jeres konsol
  - b. Svær opgave: Udskrive de 10 tal i tilfældig rækkefølge  
materiale: <https://www.w3schools.com/java/default.asp>
3. Gå ind en konsol og find mappen hvor programmet ligger. Kør kommandoen.  
**javac MitProgram.java**
4. Hvis kompilering virkede burde i nu have en fil kaldet "MitProgram.class" i jeres mappe. I skal nu køre programmet:  
**java MitProgram**

### Evaluering af opgaven:

Løbende snak med de enkelte grupper.

Til sidst vælges en tilfældig gruppe, der skal præsentere FOR KLASSEN ;-):

1. Forklar opsætningen på jeres maskine
2. Gennemgå koden, skriv kommentarer hvis i er i tvivl om hvordan i forklarer det
3. Kompiler jeres program
4. Kør jeres program