

## Groovy kennismaking

1. Download en installeer Java JDK 11
2. Download Groovy 3.0 en pak uit naar een later vindbare directory
3. Stel de JAVA\_HOME en GROOVY\_HOME variabelen in
4. Test je omgeving door vanuit een MS-DOS (cmd) venster de GroovyConsole te openen
5. Maak in de GroovyConsole een script waarin:
  - a. Je een variabele maakt en vult met je naam
  - b. Een variabele maakt en vult met je leeftijd
  - c. Een andere variabele maakt met een getal onder de 10
  - d. Een resultaat variabele maakt met daarin de optelsom van de vorige twee variabelen
  - e. De waarde van de resultaat variabele toont in de console

## Datatypes, methoden en operatoren

1. Maak een aantal variabelen in een Groovy script van verschillend datatype. Let erop dat je die in Groovy niet handmatig hoeft op te geven (niet typed).
2. Zet een getal in een String variabele (dus plaats er quotes omheen)
  - o Zorg ervoor dat je een cijfer bij dit getal optelt en het resultaat in de console toont. Er zal een conversiestap nodig zijn. Gebruik hiervoor:
    - (Groovy style:) de notatie: `def x = input as Integer` om van x een Integer te maken
    - (Java style:) de `Integer.parseInt` methode of `Double.parseDouble` bij een kommagetal.

Gebruik voor de volgende opdrachten de documentatie van Java m.b.t. Strings:

<https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/lang/String.html>

3. Maak een methode die een meegegeven tekst omzet naar hoofdletters
4. Maak een methode die twee opgegeven getallen bij elkaar optelt
5. Maak een methode die van een meegegeven getal opgeeft of het een getal kleiner is dan 10, groter is dan 10 of precies 10 is.
6. Maak een methode die een deel van een opgegeven tekst laat zien, waarbij aan de methode ook de start- en eindpositie worden opgegeven
7. Maak een methode die van een opgegeven getal aangeeft of het een even of oneven getal is. Tip: maak gebruik van de modulus operator % (en bij voorkeur *niet* van Google dit keer).

## Mini Casus: Producten beheer (1)

Wij gaan een applicatie maken die een verzameling van producten kan bijhouden.

Eerst dienen wij hiervoor grofweg een aantal stappen te doorlopen alvorens wij in een volgend hoofdstuk daadwerkelijk de verzameling gaan wijzigen.

Voor nu:

Maak een variabele waarin wij een verzameling gaan opslaan en stop daarin een set blokhaken [].

Maak een methode voegToe waarmee wij later elementen gaan toevoegen aan onze verzameling onder bepaalde voorwaarden.

Maak een methode wijzig waarmee wij later een opgegeven element gaan wijzigen naar een opgegeven waarde

1) Vraag de eindgebruiker om telkens een keuze te maken wat hij of zij wilt. Dit kan door gebruik te maken van de `System.in.newReader().readLine()` functie. De onderstaande stukken code mag je voor nu één-op-één kopiëren:

```
println "Geef je keuze op:"  
...  
def invoer = System.in.newReader().readLine();
```

Je kunt de gebruiker de keuze geven uit vier opties:

- 1) Toon producten
- 2) Voeg product toe
- 3) Wijzig product
- 4) Verwijder product

Doe dit in een loop, omdat anders de applicatie al klaar is na de eerste keus.

Zorg ervoor dat er in de loop wordt afgevangen wanneer er bijvoorbeeld op 'q' wordt gedrukt om de applicatie mee te sluiten:

```
def invoer = System.in.newReader().readLine(); // evt as Integer erachter...  
while (invoer != 'q') {  
    //println "Je keuze was: ${invoer}";  
    if ...  
}
```

**Succes!**