Datentypen

Built-In Datentypen

- Groovy unterstützt **alle** eingebauten Datentypen von Java und hat zusätzlich eigene Datentypen und erweiterte Funktionen
- *primitive Datentypen*
  - Logisch – boolean
  - Ganzzahl – byte, short, int, long
  - Fliesskommazahlen – float, double
  - Zeichen – char
- *komplexe Datentypen u.a.*
  - BigInteger
  - BigDecimal

Datentyp	Bezeichnung	Größe (Bits)	Wertebereich/Details
Logisch	boolean	n/a	true und false
Ganzzahl	byte	8	-128 bis 127
Ganzzahl	short	16	-32.768 bis 32.767
Ganzzahl	int	32	-2.147.483.648 bis 2.147.483.647
Ganzzahl	long	64	-9.223.372.036.854.775.808 bis 9.223.372.036.854.775.807
Ganzzahl	BigInteger	unbegrenzt	Unbegrenzt, nur durch verfügbaren Speicher eingeschränkt
Fließkommazahlen	float	32	ca. 1.4E-45 bis 3.4E+38
Fließkommazahlen	double	64	ca. 4.9E-324 bis 1.7E+308
Dezimalzahlen	BigDecimal	unbegrenzt	Hohe Präzision, nur durch verfügbaren Speicher eingeschränkt
Zeichen	char	16	16-Bit Unicode Zeichen

```java
class Beispiel {
    static void main(String[] args) {
        // Beispiel für einen long Datentyp
        long y = 2_036_854_775_807L; // oder 50000L

        // Beispiel für einen Gleitkommazahlen-Datentyp
        float a = 10.56f;

        // Beispiel für einen double Datentyp
        double b = 10.54;

        // Beispiel für einen BigInteger Datentyp
        BigInteger bi = 35g;

        // Beispiel für einen BigDecimal Datentyp
        BigDecimal bd = 3.5g;


    }
}
```

```
class Example {
    static void main(String[] args) {

def a = 1
boolean isInteger = a instanceof Integer
println(isInteger) // -> true

println(a)
    }
}
```

Komplexe Datentypen - Strings

```groovy
class Beispiel {
    static void main(String[] args) {
     def einfach = 'einfache Anführungszeichen'
     def doppelt = "doppelte Anführungszeichen\n"
     def slashy = /ein "Slashy-String" ohne 'Escape'/
     def dollar = $/andere Möglichkeit und Einfügen von "/"/$
     def dreifach = '''
Ich bin
ein String der über
mehr Zeilen geht\n'''

     def tripple = """
erste Zeile
zweite Zeile
dritte Zeile"""

    println(einfach)
    println(doppelt)
    println(slashy)
    println(dollar)
    println(dreifach)
    println(tripple)
   }
}
```

```
Ausgabe:
einfache Anführungszeichen
doppelte Anführungszeichen

ein "Slashy-String" ohne 'Escape'
andere Möglichkeit und Einfügen von "/"

Ich bin
ein String der über
mehr Zeilen geht



erste Zeile
zweite Zeile
dritte Zeile
```

statische vs. dynamische Typen

- Java = statisch typisiert

```java
public class MyClass {
    public static void main(String args[]) {
      String s = "Hallo Welt";
      System.out.println(s); // -> Hallo Welt

      s = 123;
      System.out.println(s); //-> Compiler-Fehler
      }
}
```

- Groovy = dynamisch typisiert

```groovy
class Beispiel {
    static void main(String[] args) {
    def s = "Hallo Welt"
    println s.getClass() // -> class java.lang.String
    s = 123
    println s.getClass() // -> class java.lang.Integer

  }
}
```

GString

```
class Beispiel {
    static void main(String[] args) {
     def person = [name: 'Thomas Smits', lehrt: 'PR3']
     def hochschule = "Hochschule Mannheim"
     def ausdruck = "Hallo, mein Name ist ${person.name}. Ich unterrichte ${person.lehrt}. An der ${hochschule}."

     println(ausdruck)
   }
}
```

Ausgabe:
Hallo, mein Name ist Thomas Smits. Ich unterrichte PR3. An der Hochschule Mannheim.

String Konkatenation

- *jeder* String kann mit "+" konkateniert werden

```
class Beispiel {
    static void main(String[] args) {
     def eins = "Ein String"
     def zwei = ' wird konkateniert'
     println(eins + zwei) // -> Ein String wird konkateniert
    }
}
```

String Index

- mit positiven & negativen Indizes auf Zeichen eines Strings zugreifen

```
class Beispiel {
    static void main(String[] args) {
    def greeter = "Hallo Welt"

    println(greeter[1]) //-> a
    println(greeter[-4]) // -> W
    }
}
```