

Kotlin - Kapitel 1

Datentypen:

Was sind Datentypen ?

Datentypen sind in unserem Fall erstmal nur Typen von Werten mit denen wir arbeiten.

Wir verwenden folgende Datentypen:

Integer: Mit Integers können wir ganze Zahlen (also -1,0,1,2,...565754,...) benutzen.

Wie erstellen wir einen **Integer** in Kotlin ?

```
var unserInt: Int  
unserInt = 2
```

Double: Mit einem Double können wir Zahlen mit Komma darstellen (also 1.2, 6.7).

Wie erstellen wir einen **Double** in Kotlin ?

```
var unserDouble: Double  
unserDouble = 2.2
```

String: Mit Strings können wir jeden möglichen Text speichern. Wichtig ist dabei, dass dieser Text immer in den " steht.

Wie erstellen wir einen **String** in Kotlin ?

```
var unserString: String  
unserString = "unser String"
```

Character: Ein Character ist nur ein Zeichen auf eurer Tastatur und wird innerhalb von ' geschrieben.

Wie erstellen wir einen **Character** in Kotlin ?

```
var unserChar: Char  
unserChar = 'a'
```

Wie funktioniert das Erstellen ?

```
1 2 3 4  
var unserInt: Int  
5 6  
unserInt = 2
```

1: Wir schreiben am Anfang **var**. Damit sagen wir dem Programm das wir eine Variable schreiben wollen.

2: Hier wird der Name geschrieben, den wir unserer Variable geben wollen. Dort kann alles stehen.

3: Die Doppelpunkte muss man schreiben. Sie heißen in unserem Fall, dass wir etwas zu unserer Variable hinzufügen wollen - nämlich einen Typen.

4: Wir müssen nun unseren Typen auch angeben, also Int, Double, String oder Char. In unserem Fall dann eben ein Int

5: Hier greifen wir auf unsere Variable zu.

6: Das „=" ist immer eine Zuweisung.
Was passiert ?

Kommentare:

In jeder Programmiersprache gibt es Kommentare. Kommentare sind dafür da, dass ihr Text schreiben könnt, der aber bei der Ausführung einfach ignoriert wird. Das hilft dabei den Code übersichtlicher zu machen und um Merkhilfen zu notieren.

```
// Es gibt auch Kommentare
```

```
/*  
  und  
  auch  
  welche  
  über  
  mehrere  
  Zeilen  
*/
```

1. Für Kommentare über eine Zeile muss einfach nur ein `//` davor gesetzt werden.
2. Bei welchen über mehrere Zeilen wird der Kommentar mit `/*` begonnen und mit `*/` beendet.

Bedingungen:

Für Programme sind Bedingungen immens wichtig. Mit Bedingungen wird gesteuert und auf Eingaben reagiert. (Diese Erklärung ist nicht so wichtig):

```
if (Bedingung) {  
    // TODO  
}
```

Hier haben wir eine einfache if - Verzweigung. Nur wenn unsere Bedingung gilt soll etwas gemacht werden. (natürlich funktioniert dieser Code nicht. Statt Bedingung muss wirklich eine Bedingung gelten.)

Boolean: Wir haben bei unseren Typen noch einen vergessen, nämlich den Boolean. Der Boolean kann nur die Werte **False** (für nein) und **True** (für ja) annehmen.

-> Jede Bedingung wertet entweder zu False oder zu True aus.

Was heißt das für uns ?

```
if (true) {  
    // TODO  
}
```

Wir wissen damit wie unsere if - Verzweigung funktioniert und das der Code links auch klappt. Wir können also alles als if Bedingung verwenden was als Ergebnis einen dieser Booleans zurück gibt.

Wie schreiben wir denn unsere Bedingungen:
Mit Vergleichen ?

```
if (x < 5){  
    //TODO  
}
```

Jetzt überprüft unser Programm ob unsere Variable x kleiner als fünf ist.

Bei Ja -> führt das Programm alles aus was in den geschweiften Klammern steht.

Bei Nein -> springt das Programm ans Ende der Klammern.

Kotlin - Kapitel 1

Wir kennen jetzt also den Satz „Wenn das passiert mache“

Den Satz können wir erweitern mit „Wenn das passiert mache ..., sonst mache“

```
1  if (x < 5){  
    //TODO  
  }  
2  else {  
    }  
3
```

Was macht das Programm diesmal ?

Bei Ja -> springt das Programm zu 1 und wenn alles von 1 abgearbeitet wurde zu 3.

Bei Nein -> springt das Programm zu 2 und wenn es da alles abgearbeitet hat zu 3.

Eine Möglichkeit für Bedingungen gibt es noch „Wenn das passiert mache, Sonst mache wenn das passiert, sonst mache ...“

```
if (x < 5){  
    //TODO  
}  
else if (x > 5){  
    //TODO  
}  
else {  
    //TODO  
}
```

Wir können also beliebig viele Bedingungen abfragen.

Aufgabe 1

a) Trage ein welche Werte v,w,x,y,z am Ende haben werden...

```
var v = 20  
var w = 3  
var x = 3  
var y = 4  
var z = 10  
  
if (x < v){  
    x = x + v  
    z = 4  
}  
else if (x > 5){  
    y = y + x  
}  
else {  
    w = 3  
    v = 3  
}
```

v = _____

w = _____

x = _____

y = _____

z = _____

b)

```

var v = 1
var w = 2
var x = 3
var y = 4
var z = 5

if (x > 5){
    x = x + v
    z = 4
}
else if (x > 3){
    y = y + x
    v = v * 2
    z = z * 2
}
else {
    w = 3
    v = 3

    if (x == y && x == v){
        x = v * w
    }
}

y = 3
z = 3

```

V = _____

W = _____

X = _____

Y = _____

Z = _____

c)

```

var v = 1
var w = 2
var x = 3
var y = 4
var z = 5

if (x < 5){
    if (x < 4){
        if (x < 3){
            if (x < 2){
                if (x < 1){
                    x = x * 1
                }
                else{
                    x = x + 1
                }
            }
            else{
                x = x + 2
            }
        }
        else{
            x = x + 3
        }
    }
    else{
        x = x + 4
    }
}

v = x + 1
w = x + 2
y = x + 3
z = x + 4

```

V = _____

W = _____

X = _____

Y = _____

Z = _____