

# Variablen

Einführung

# Was sind Variablen?

Variablen sind Referenzen auf Speicherorte (Labels).

Sie zeigen auf einen Ort im Speicher. In Python können wir zum Beispiel eine Zahl einer Variable zuweisen.

```
nummer = 3
```

Hier haben wir der Variable nummer die Zahl 3 zugewiesen.

Das = nennt man deshalb auch den Zuweisungsoperator.

# Schreibweise von Variablen

Variablen-Namen dürfen **keine Leerzeichen** enthalten.

```
mein name = "John Doe"
```

⇒ Falsch

Variablen-Namen werden in Python in Snake-Case notiert

```
mein_name = "John Doe"
```

```
meinName = "John Doe" # falsch (CamelCase)
```

```
MeinName = "John Doe" # falsch (UpperCamelCase, Pascal-Case)
```

In Variablen-Namen werden idealerweise **keine Umlaute und Sonderzeichen** benutzt

# Singular & Plural

Man sollte darauf achten, Variablen möglichst im Plural zu schreiben, wenn es sich um eine sequenzartige Struktur handelt oder eine Mehrzahl ausgedrückt werden soll.

```
persons = 10
```

```
cities = ["Berlin", "Hamburg"]
```

Das einzelne Element einer Sequenz kann im Singular adressiert werden:

```
for city in cities:  
    print(city)
```

# Datentypen

Variablen werden **ohne Angabe des Datentyps** verwendet. In Python haben Variablen keinen festen Datentyp (wie zum Beispiel in Java/C++), sondern der Datentyp wird dynamisch vergeben.

```
i = 3
```

⇒ Der **Datentyp** von i ist jetzt vom Typ **Integer**

```
i = "John Doe"
```

=> Der **Datentyp** von i ist jetzt vom Typ **String**

# Was ist ein Datentyp?

Jede Variable hat einen Datentypen. Ein **Datentyp** kann eine Zahl sein, z.B. 4 oder ein String, z.B. "Berlin". Der Datentyp beschreibt die Art und Weise, wie ein Wert im Speicher verarbeitet wird.

Mit Zahlen kann man rechnen, mit Strings kann man Texte verarbeiten. Eine Zeichenkette und eine Zahl kann man zum Beispiel nicht addieren.

Um den Datentypen einer Variablen herauszufinden, nutzen wir die Funktion `type()`.

```
x = 3
```

```
type(x)
```

```
<class 'int'>
```

# Einfache Datentypen

Es gibt 6 einfache und somit **unveränderliche** Datentypen in Python.

**Integer** (-23, 200000)

**Fließkommazahlen** (-0.03, 200.25)

**Boolsche Werte** (True oder False)

**Zeichenketten** (String)

**Complex** (Komplexe Zahlen, welche aus Real- und Imaginärteil bestehen)

der spezielle Datentyp **None**

# ID

Jedes Objekt in Python hat eine ID. Da in Python alles ein Objekt ist, hat jede Variable eine ID. Damit kann man zum Beispiel prüfen, ob zwei Variablen auf dasselbe Objekt referenzieren.

```
number_trees = 9  
id(number_trees)  
140709802149568
```

Diese ID ist für jede Pythonsitzung einzigartig. Wir verwenden sie in expliziter Form mehrheitlich für Debugging- und Testzwecke. Python nutzt die ID intern für den Identitäts-Operator `is`.



# Python 3

## The standard type hierarchy

