

Python: Woche 1

Grundlagen, Variablen, Ausgabe
Michael Stambach
Stellvertretung Ivo Blöchliger

Ich

- Michael Stambach
- KSBG 2018 – 2022
- Jetzt Studium Elektrotechnik, ETH
- Stellvertretung für drei Wochen

Vorstellungsrunde

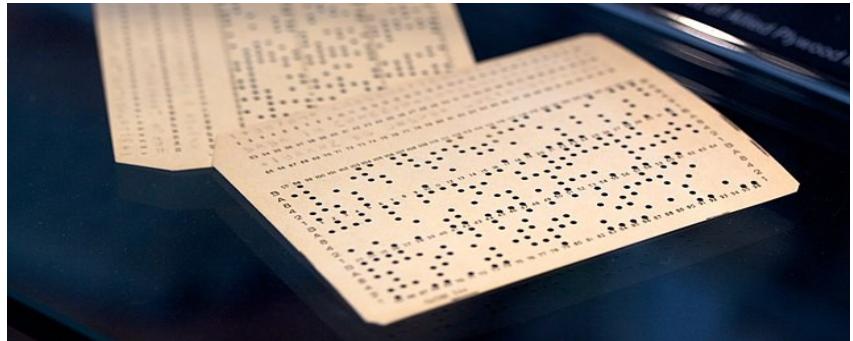
Name & Programmiererfahrung?

Themen Heute

- Programmieren allgemein
- Weshalb Python
- Programmstruktur
- Variablen
- Datentypen
- Ausgabe

Programmieren allgemein

- Computer sind universelle Rechenmaschinen
- Anweisungen werden abgearbeitet
- Programm: Sammlung von Befehlen / Anweisungen
- Meist eine Textdatei
- Bearbeitbar mit Texteditor



Marcin Wichary from San Francisco, Calif., CC BY 2.0
<<https://creativecommons.org/licenses/by/2.0/>>, via Wikimedia Commons

```
useEffect(() => {
  if (joboffersFrontpage.totalPages === 0 && !joboffersFrontpage.isPending) {
    dispatch(listLoadNextPage(JOBOFFERS, { listName: 'frontpage' }))
  }
}, [])

let pastEventCount = 0

if (eventsFrontpageUpcoming.lastPageLoaded > 0) {
  pastEventCount = 3 - (eventsFrontpageUpcoming.items.length % 3)
```

Programmiersprachen

- Programme können in verschiedenen Sprachen geschrieben werden
- Diese haben verschiedene Vor- und Nachteile
- Geschwindigkeit, Komplexität, Stabilität
- Wir benutzen Python

Wieso Python?



- **Anfängerfreundlich**
 - Dynamische Datentypen
 - Unkomplizierter Syntax
 - Viele Libraries, Ressourcen
- **Läuft überall**
- **Nachteile**
 - Eher langsam
 - Python muss installiert sein

Programmstruktur

- **Textdatei**
- **Dateiendung ".py"**
- **Sammlung von Befehlen**
- **Ausführung Zeile für Zeile von oben nach unten**

Minimales Python-Programm

```
print("Hello World!")
```

- Eine einzige Anweisung
- Gibt **Hello World!** auf der Konsole aus
- Häufig einfachstes Beispiel für Programmiersprachen
- Funktionsaufruf der "print"-Funktion (dazu später mehr)

Etwas mehr...

Programm

```
print("Hallo!")  
print("Dieses Programm...")  
print("...hat drei Zeilen")
```



Ausgabe

```
Hallo!  
Dieses Programm...  
...hat drei Zeilen
```

Schauen wir genauer hin...

```
print("Hello World!")
```

Name der Funktion:
Gewünschte
Funktionalität

Argument:
Spezifischer Wert zur
Ausführung

*Weshalb ist unser Text von
Anführungszeichen umgeben?
Damit es ein Wert ist!*

Zahlen

- Bisher haben wir nur Zeichenfolgen (Strings) verwendet
- Strings sind immer von Anführungszeichen umschlossen
- Zahlen kann man in Python «direkt» verwenden
- Folgende Befehle geben die gleiche Ausgabe

```
print("12")
```

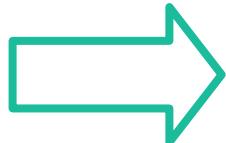
```
print(12)
```

Was ist der Unterschied?

Zahlen

- Mit Zahlen kann man Rechnen!

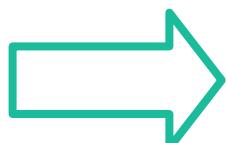
```
print(9 + 10)
```



```
19
```

- Was passiert mit Strings?

```
print("9" + "10")
```



```
910
```

Eine Auswahl an Datentypen

"Hallo"
21
3.1415
True

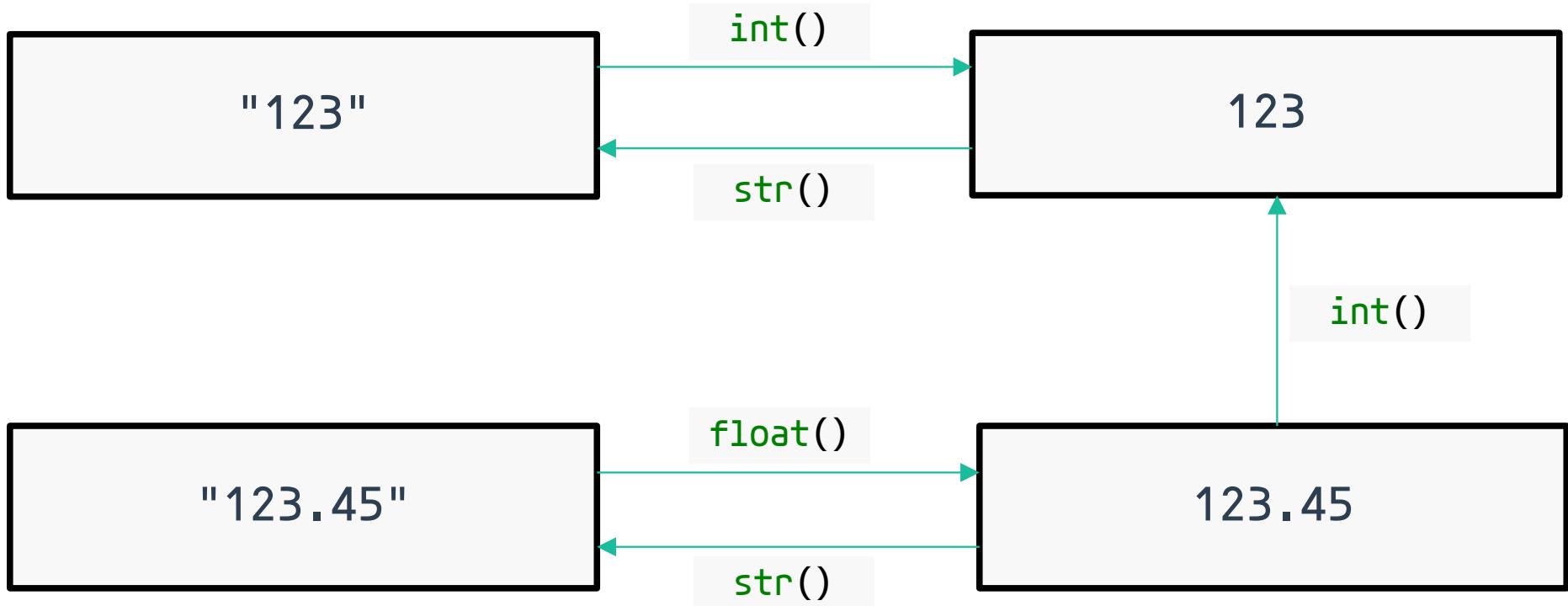
Zeichenfolge "str"

Ganzzahl "int"

Fliesskommazahl "float"

Logischer Wert "bool"

Explizite Typenkonversion



Variablen: Ein Beispiel

- Wir haben ein Quadrat mit Seitenlänge 3
- Wir schreiben ein Programm um Fläche und Umfang zu berechnen
- Funktioniert, aber was ist wenn wir ein anderes Quadrat wollen?

```
print("Umfang: ")  
print(4 * 3)
```

```
print("Fläche: ")  
print(3 * 3)
```

```
Umfang:  
12  
Fläche:  
9
```

Variablen: Wiederverwendung

- Wir führen eine Variable ein
- Jetzt muss man nur noch eine statt drei Stellen anpassen
- Der Rest des Programmes verhält sich identisch zu vorher

```
s = 3  
  
print("Umfang: ")  
print(4 * s)  
  
print("Fläche: ")  
print(s * s)
```

Variablen

- **Variablen stehen stellvertretend für einen Wert**
- **Gleiches Konzept wie in der Mathematik**
- **Aber: Variablen können sich im Laufe des Programmes verändern.**
- **Ein Wert einer Variable hat immer einen spezifischen Datentyp**

Setzen einer Variable

- Setzen einer Variable erfolgt mit "=".
- Name der Variable links, Wert rechts.
- Eine Variable und ihr Datentyp müssen in Python nicht explizit deklariert werden.
- In vielen anderen Programmiersprachen ist das nicht so!

```
text = "Python"
```

Variablen aktualisieren

- Beim Setzen einer Variable kann die Variable selbst benutzt werden
- Es wird dabei immer der alte Wert benutzt
- Es gibt eine noch kompaktere Schreibweise

Beispiel: "a" um 2 erhöhen

```
a = a + 2
```

Noch kompakter...

```
a += 2
```

Was gibt folgendes Programm aus?

```
a = 27  
  
b = a - 15  
b = b * 2  
a -= b  
a *=3  
  
print(a)
```

Zeile	a	b
1		
2		
3		
4		
5		

Funktioniert dieses Programm?

```
a = b + 2  
b = 4  
  
print(a)
```

Nein!

Ausführung von oben nach unten:
Variable existiert nicht, bevor sie
gesetzt wird

Operationen: Arithmetik

Symbol	Operation
+	Addition
-	Subtraktion
*	Multiplikation
/	Division
%	Divisionsrest / Modulo
//	Ganzzahldivision
**	Potenz ("hoch")

Der Datentyp kann sich dabei ändern!

Ausgabe von Variablen

Drei print Aufrufe und
Ausgabe über drei Zeilen
Geht das schöner?

```
Preis = 12
Print("Das kostet")
Print(preis)
Print("Franken")
```

Das kostet
12
Franken

Wir versuchen, die String in einem
print zu "zusammensetzen"

```
Preis = 12
Print("Das kostet " + Preis + " Franken")
```

Es ist nicht erlaubt eine Zahl zu
einer String zu addieren!

Error!

Ausgabe von Variablen

Wir können etwas explizit zu einer String umwandeln!

Dieser Code ist immer noch etwas umständlich...

Mit einem "f" vor einer String können wir eine "f-String" erstellen. Darin lassen sich Variablen direkt mit geschweiften Klammern verwenden

```
preis = 12  
print("Das kostet " + str(preis) + " Franken")
```

Das kostet 12 Franken

```
preis = 12  
print(f"Das kostet {preis} Franken")
```

Kommentare

- Anmerkungen in einem Programm
- In Python: Zeilen die mit "#" beginnen
- Werden beim Ausführen ignoriert



```
a = 2  
  
# a wird verdoppelt und in b gespeichert  
b = 2 * a
```

Aufgaben

- Heute noch keinen Code ausführen
- "Auf Papier" Programme analysieren
- Markdown-File
- Abgabe mit OneDrive-Link