

# WOCHE 3

## PYTHON: BUILT-INS, OPERATOREN, DATENTYPEN

# LINK ZU DEN FOLIEN



# LERNZIELE

- Gemeinsamkeiten Python & Scratch erkennen
- Python als *professionelle Programmiersprache* kennenlernen
- Wichtige Begriffe:



**print()/input()**



Operatoren



Datentypen



Variablen

# WAS KENNEN WIR BEREITS?

## Anweisungen

gehe 10 er-Schritt

drehe dich  um 15 Grad

## Bedingungen

falls  dann

falls  dann  
sonst

## Funktionen

Definiere quadriere x

quadriere 5

## Schleifen

wiederhole 10 mal

wiederhole fortlaufend

## Variablen

punkte setze punkte auf 0

ändere punkte um 1

# WAS IST PYTHON?

- Interpretierte Programmiersprache
- Open-Source
- Aktuelle Version: 3.7.1
- Professionelle, "general purpose"-Sprache

Laut [TIOBE-Index](#) aktuell 4. beliebteste Programmiersprache

# INSTALLATION: PYTHON





# VORTEILE VON PYTHON

- ✓ Open-Source
- ✓ Große Community -> viele Bibliotheken
- ✓ Python legt den Fokus auf ..
  - ..Schneller Erlernbarkeit
  - ..Guter Lesbarkeit



# JAVA VS. PYTHON

## HELLO WORLD IN JAVA:

```
public static void main(String[] args){  
    System.out.println("Hello World!");  
}
```


## HELLO WORLD IN PYTHON:

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

# TEXTE AUSGEBEN

## DIE `print()`-FUNKTION

# BEISPIELE: `print()`

- Mithilfe von `print()` lassen sich "Objekte" ausgeben
- Nützlich um:
  - Informationen anzuzeigen
  - Programmfehler zu suchen  
( *Debugging*)

Ein Text:

```
print("Hallo Welt")
```

Ein Ausdruck (zB. ein math. Term):

```
print(9+4)
```


Das Ergebnis einer Funktion:

```
print(max(5, 6, 7, 1, 2, -4))
```

# BENUTZEREINGABEN

## DIE `input()`-FUNKTION

# BEISPIELE: `input()`

- Ermöglicht Speichern von Benutzereingaben
- Parameter = Text, der angezeigt wird
- Eingabe wird als `str` gespeichert 

```
text = input("Hier Text eingeben:")  
print(text)
```

```
a = int(input("Erste Zahl:"))  
b = int(input("Erste Zahl:"))  
print(a < b)
```



# LIVE-CODING

# AUFGABENBLATT 1







# DATENTYPEN

**STRING,FLOAT,INTEGER, BOOLEAN**



# DATENTYPEN

- Python unterscheidet zwischen Daten verschiedenen *Typs*
- 4 Grunddatentypen:
  -  Strings(**str**)
  -  Integer(**int**)
  -  Float(**float**)
  -  Boolean(**bool**)

# INTEGER

- Ganze Zahlen
- Sowohl positiv als auch negativ möglich
- Beispiele:

**128**

**-4**

**182932304**

# FLOAT

- Fließkommazahlen
- Zahl enthält immer Nachkommawert
- Sowohl positiv als auch negativ möglich
- Beispiele:

**-4.2**

**0.03000049**

**18.0**


# STRINGS

- Zeichenketten = Liste von Zeichen
- Ein String kann bestehen aus:
  - Zahlen
  - Buchstaben
  - Sonderzeichen
- Beispiele

`"Hallo Welt"`    `"x3af!"`    `"1234999000"`

`""Ich bin ein langer Satz!!1""`

# BOOLEAN

- Wahrheitswerte: **True** oder **False**
- *Schleifen* und *Bedingungen* arbeiten mit sog. *booleschen Ausdrücken*
-  Boolescher Ausdruck :
  - Ausdruck, der entweder wahr oder falsch ist

[Video zum Thema Aussagenlogik/boolesche Algebra](#)

# ALLTAGSBEISPIELE

- "Ist der Benutzer eingeloggt?"
- "Erkennt die Kamera eine Person?"
- "Hat der Benutzer heute Geburtstag?"
- "Wurde dieser Knopf gedrückt?"

# ÜBERSICHT: DATENTYPEN

Strings      Zeichenketten

`"Hallo Welt"`

`'Gustav Gans'`

`"""Text`

`mit mehreren  
Zeilen"""`

---

Float      Fließkommazahlen

`5.14`

---

Integer      Ganzzahlen

`5, 99, -23`

---

Boolean      Wahrheitswerte

`True, False`

# OPERATOREN

- Verknüpfen *Werte* (Variablen, Konstanten, Funktionswerte..) zu *Ausdrücken*
- Arten von Operatoren:
  - Rechenoperatoren
  - Vergleichsoperatoren
  - Bitoperatoren\*

\*.. schauen wir uns nicht im Detail an: [Dokumentation - Bitoperatoren](#)



# RECHENOPERATOREN

**+** **-** **\***  
**/**

Grundrechenarten

5+7, 6\*5,  
-3.0+1.4

**\*\***

Potenz (zB. 5<sup>3</sup>)

5\*\*3

**//**

Ganzzahldivision  
("Wie oft passt X in Y?")

```
7 // 3 # ergibt 1
```

**%**

Modulo-Operator  
("Wieviel Rest ergibt  
Division?")

```
14 % 4 # ergibt 2
```

# VERGLEICHSOPERATOREN

- Vergleich 2 Werte miteinander
- Liefert als Ergebnis **True** bzw. **False** (**bool**) zurück
- Können mit **and** und **or** verknüpft werden

< "größer"

<= "kleiner-gleich"

== "gleich"

>= "größer gleich"

> "größer"

!= "ungleich"

== "gleich"

# BUILT-IN-FUNKTIONEN

- Python bringt Reihe an direkt benutzbaren Funktionen mit
  - sogenannte Built-in-Funktionen
- Schon kennengelernt: `print()` und `input()`
- Viele weitere Funktionen zB. um:
  - Elemente zu sortieren
  - Daten in andere Datentypen umwandeln
  - Minima/Maxima finden

# BUILT-IN-FUNKTIONEN

<code>abs()</code>	<code>delattr()</code>	<code>hash()</code>	<code>memoryview()</code>	<code>set()</code>
<code>all()</code>	<code>dict()</code>	<code>help()</code>	<code>min()</code>	<code>setattr()</code>
<code>any()</code>	<code>dir()</code>	<code>hex()</code>	<code>next()</code>	<code>slice()</code>
<code>ascii()</code>	<code>divmod()</code>	<code>id()</code>	<code>object()</code>	<code>sorted()</code>
<code>bin()</code>	<code>enumerate()</code>	<code>input()</code>	<code>oct()</code>	<code>staticmethod()</code>
<code>bool()</code>	<code>eval()</code>	<code>int()</code>	<code>open()</code>	<code>str()</code>
<code>breakpoint()</code>	<code>exec()</code>	<code>isinstance()</code>	<code>ord()</code>	<code>sum()</code>
<code>bytearray()</code>	<code>filter()</code>	<code>issubclass()</code>	<code>pow()</code>	<code>super()</code>
<code>bytes()</code>	<code>float()</code>	<code>iter()</code>	<code>print()</code>	<code>tuple()</code>
<code>callable()</code>	<code>format()</code>	<code>len()</code>	<code>property()</code>	<code>type()</code>
<code>chr()</code>	<code>frozenset()</code>	<code>list()</code>	<code>range()</code>	<code>vars()</code>
<code>classmethod()</code>	<code>getattr()</code>	<code>locals()</code>	<code>repr()</code>	<code>zip()</code>
<code>compile()</code>	<code>globals()</code>	<code>map()</code>	<code>reversed()</code>	<code>__import__()</code>
<code>complex()</code>	<code>hasattr()</code>	<code>max()</code>	<code>round()</code>	

# AUFGABENBLATT 2



A blurry, low-angle shot of a man in a Superman costume running through a hallway. The man is in the lower right foreground, moving towards the left. The hallway has a tiled floor and walls with posters. One poster is clearly visible with the word "LAPTOP" at the top. The image is out of focus, giving a sense of motion.

**VIELEN DANK!**