



CORRELAID

GOOD CAUSES. BETTER EFFECTS.

Python Workshop

Termin 1: Python Basics

Torben Abts & Jonas Hummel

Datum

Wer wir sind



Wir sind ein deutschlandweites Netzwerk von über 2,000 Data Scientists, die die Welt durch Data Science verbessern wollen.

#MetaWeltretter



Data4Good Projekte



Der Kern unserer Arbeit ist die Durchführung von pro-bono Projekten mit Non-Profit-Organisationen.

Mehr als 55 Data4Good Projekte wurden seit 2015 umgesetzt

Logos auf Konstanzer Projekte upo

Einige unserer Kooperationspartner:innen:

project
together



CORRELAID
GOOD CAUSES. BETTER EFFECTS.

Mitmachen



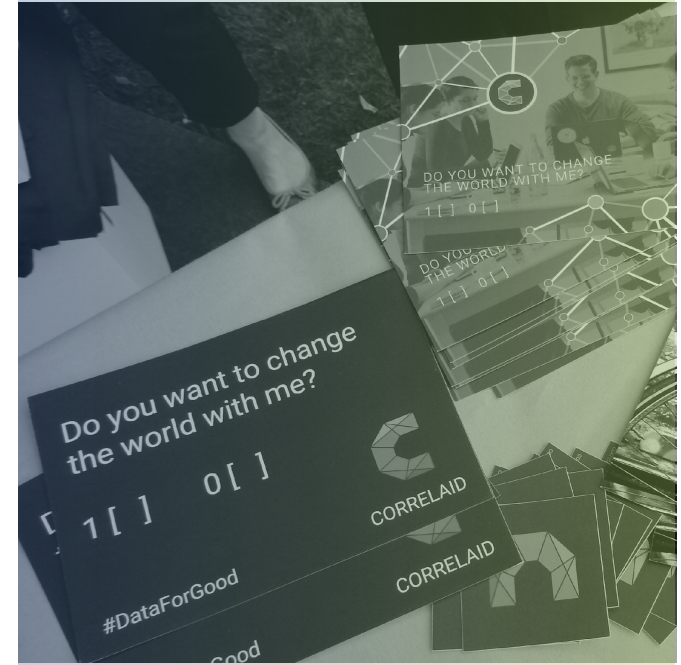
CorrelAid ist ein offenes Netzwerk für alle Menschen



Jede:r der oder die unseren Code of Conduct respektiert, ist willkommen



Neben der Projektarbeit kannst du auf unterschiedliche Art und Weise bei uns aktiv werden



Agenda

1. Intro
2. Absolute Basics
3. Datentypen
4. Datencontainer
5. Loops und Conditional Statements
6. Funktionen



1. Intro - Was ist Python?

- Allzweck-Programmiersprache
- Entwickelt von Guido van Rossum in den 1980er Jahren
- Name leitet sich von "Monty Python's Flying Circus" ab
- Aufstieg unter anderem dank Google (Machine Learning Libraries)



1. Intro - Warum Python?

Flexibel
&
Einfach



Große Community



Windows
MacOS
Linux



Viele
Libraries
(Erweiterungen)



Flexible in
Programmier-
paradigmen



Kostenlos!

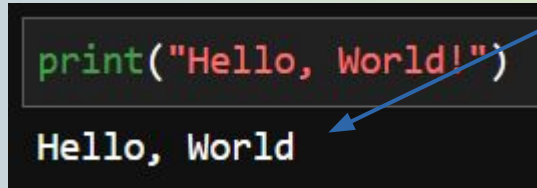


2. Absolute Basics

- Hello, World!
- Variablen
- Mathematische Operatoren
- Logische Operatoren
- Kommentare



2. Absolute Basics - Hello, World!



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

- Input
- Output
- () → Funktion wird auf Inhalt der Klammer ausgeführt
- print() → Output in die Konsole
- "" oder " → zum Schreiben von Ausdrücken (später mehr)



2. Absolute Basics - Variablen

```
tutor1 = "Torben"
tutor2 = "Jonas"
age_tutor1 = 22
age_tutor2 = 23

print(tutor1, age_tutor1)
print(f"{tutor2} ist {age_tutor2} alt.")

Torben 22
Jonas ist 23 alt.
```

- Man kann Code in Variablen speichern
- Jegliche Datentypen können zugewiesen werden
- Assignment Operator =
- Konvention: klein schreiben und _ als Leerzeichen



2. Absolute Basics - Mathematische Operatoren

```
# Addition
sum_numbers = 7 + 8 + 15
print("sum_numbers:", sum_numbers)

# Subtraktion
number1, number2, = 25, 17
diff_numbers = number1 - number2
print("diff_numbers:", diff_numbers)

sum_numbers: 30
diff_numbers: 8
```

- Alle mathematischen Operatoren können auch in Python genutzt werden
- Bei numerischen Ausdrücken funktionieren diese immer
- Bei anderen Datentypen kommt es darauf an, ob diese "sinnvoll" sind (Achtung, kann zu anderem Ergebnis führen!)



2. Absolute Basics - Mathematische Operatoren

Übersicht mathematischer Operatoren

Syntax	Math	Operation Name
<code>a+b</code>	$a + b$	addition
<code>a-b</code>	$a - b$	subtraction
<code>a*b</code>	$a \times b$	multiplication
<code>a/b</code>	$a \div b$	division (see note below)
<code>a//b</code>	$\lfloor a \div b \rfloor$	floor division (e.g. $5//2=2$) - Available in Python 2.2 and later
<code>a%b</code>	$a \bmod b$	modulo
<code>-a</code>	$-a$	negation
<code>abs(a)</code>	$ a $	absolute value
<code>a**b</code>	a^b	exponent
<code>math.sqrt(a)</code>	\sqrt{a}	square root



2. Absolute Basics - Mathematische Operatoren

Diese mathematischen Operatoren können direkt in die Assignment Operators eingebaut werden.

`+=` (increment assignment)

Adds a value and the variable and assigns the result to that variable.

`-=` (decrement assignment)

Subtracts a value from the variable and assigns the result to that variable.

`*=` (multiplication assignment)

Multiplies the variable by a value and assigns the result to that variable.

`/=` (division assignment)

Divides the variable by a value and assigns the result to that variable.

`=` (power assignment)**

Raises the variable to a specified power and assigns the result to the variable.

`%=` (modulus assignment)

Computes the modulus of the variable and a value and assigns the result to that variable.

`//=` (floor division assignment)

Floor divides the variable by a value and assigns the result to that variable.



2. Absolute Basics - Logische Operatoren

```
print(True and False)
print(True or False)
print(not False and True)
```

```
num0 = 0
print(num0 or True)
print(5 >= 7)
```

```
False
True
True
True
False
```

- Ergeben stets Wahrheitsausdrücke (True/False)
- Wichtig für Vergleiche und Bedingungen



2. Absolute Basics - Logische Operatoren

Übersicht Logische Operatoren

Operator	Description	Example
and	Returns True if both statements are true	<code>x < 5 and x < 10</code>
or	Returns True if one of the statements is true	<code>x < 5 or x < 4</code>
not	Reverse the result, returns False if the result is true	<code>not(x < 5 and x < 10)</code>

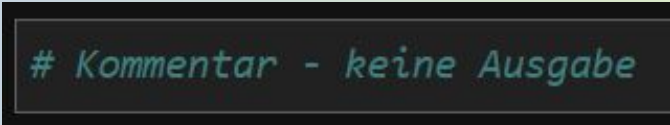


2. Absolute Basics - Logische Operatoren

Übersicht Vergleichsoperatoren

Operator	Name	Example
==	Equal	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

2. Absolute Basics - Kommentare



```
# Kommentar - keine Ausgabe
```

- # Kommentar
- Sollte unbedingt genutzt werden, um den Code verständlicher zu machen



3. Datentypen

- Strings
- Ints
- Floats
- Boolean



3. Datentypen - Strings



3. Datentypen - Ints



3. Datentypen - Floats



3. Datentypen - Booleans



4. Datencontainer

- Listen
- Tuples
- Dictionaries
- Sets

4. Datencontainer - Listen



4. Datencontainer - Tuples



4. Datencontainer - Dictionaries



4. Datencontainer - Sets



5. Loops und Conditional Statements

- for-Loops
- while-Loops
- if-Statement
- else/elif

5. Loops und Conditional Statements - for-Loops



5. Loops und Conditional Statements - while-Loops



5. Loops und Conditional Statements - if-Statement



5. Loops und Conditional Statements - else/elif



6. Funktionen

Weiterführende Quellen

Lehrbücher

- [A Beginners Guide to Python 3 Programming](#) (John Hunt)





CORRELAID
GOOD CAUSES. BETTER EFFECTS.

Python Workshop

Termin 2: Datenanalyse mit Python

Torben Abts & Jonas Hummel

Datum



CORRELAID

GOOD CAUSES. BETTER EFFECTS.

Python Workshop

Termin 3: Webscraping

Torben Abts & Jonas Hummel

Datum