

ÜBUNG

ZUR EINFÜHRUNG IN DIE PROGRAMMIERUNG

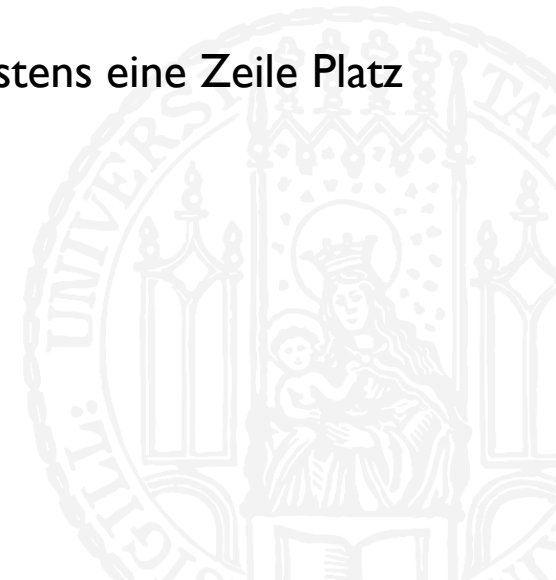
MEHR FRAGEN

- Stellt mir mehr Fragen!
- Mailadresse: Leonie.Weissweiler@campus.lmu.de



ÜBUNGSABGABE

- Bitte kopiert nicht den gesamten Inhalt eurer Konsole in das Abgabeformular.
- Überlegt euch für jede Aufgabe, auch für die Bash-Aufgaben, was der Befehl bzw. das Programm war, das den gewünschten Effekt hatte und gebt diesen ab!
- Zur Lesbarkeit bitte immer die Unteraufgabe (z.B. 1-2) darüber schreiben und danach mindestens eine Zeile Platz lassen.



WIEDERHOLUNG: COMMAND PROMPT

Benutzername Computername Aktueller Ordner


```
Leonie@Ubuntu:~/Python/Übungsblatt1 $
```



WIEDERHOLUNG: CD

Befehl Zielordner

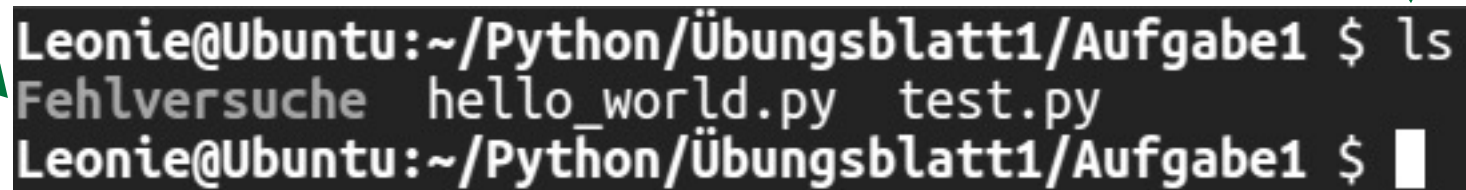
```
Leonie@Ubuntu:~/Python/Übungsblatt1 $ cd Aufgabe1  
Leonie@Ubuntu:~/Python/Übungsblatt1/Aufgabe1 $ █
```



WIEDERHOLUNG: LS

Dateien und Ordner im
aktuellen Ordner

Befehl



```
Leonie@Ubuntu:~/Python/Übungsblatt1/Aufgabe1 $ ls
Fehlversuche hello_world.py test.py
Leonie@Ubuntu:~/Python/Übungsblatt1/Aufgabe1 $
```



WIEDERHOLUNG: MKDIR

Befehl Name des neuen Ordners

```
Leonie@Ubuntu:~/Python/Übungsblatt1 $ mkdir Aufgabe2
Leonie@Ubuntu:~/Python/Übungsblatt1 $ ls
Aufgabe1  Aufgabe2
Leonie@Ubuntu:~/Python/Übungsblatt1 $
```



NEUER BEFEHL: PWD

Kompletter Pfad des aktuellen Ordners

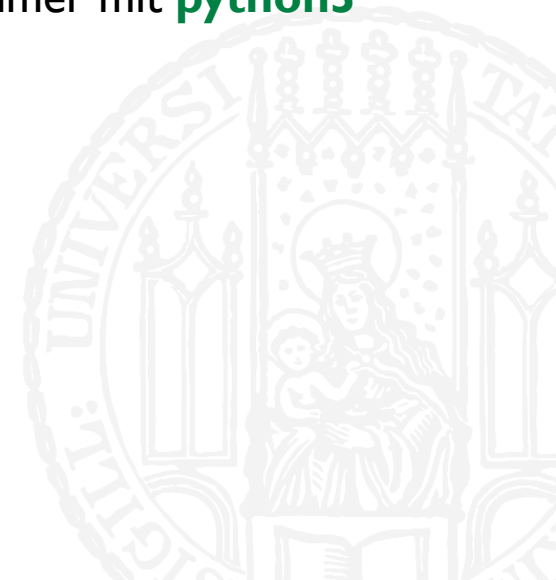
Befehl

```
Leonie@Ubuntu:~/Python/Übungsblatt1 $ pwd
/home/leonie/Python/Übungsblatt1
Leonie@Ubuntu:~/Python/Übungsblatt1 $
```



PYTHON VERSIONEN

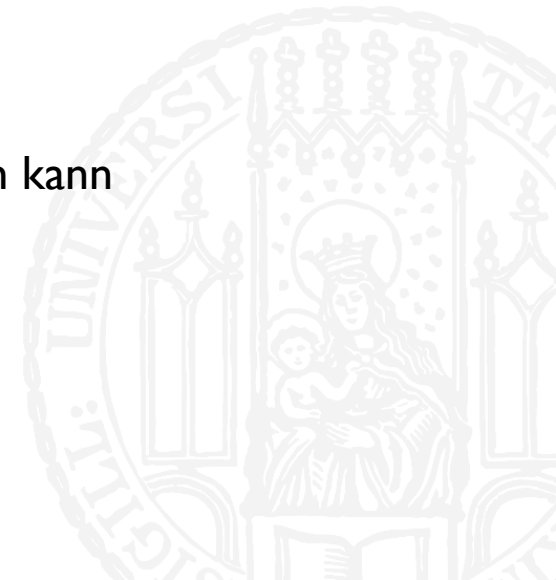
- Es gibt zwei Python Versionen, Python 2 und Python 3
- Python 2 und Python 3 sind nicht kompatibel!
- In der Konsole wird `python` zu `python2` vervollständigt, deswegen Python3-Programme immer mit `python3` aufrufen
- Bei Internetrecherchen aufpassen, ob sich die Quelle auf Python 2 oder 3 bezieht



SHEBANG

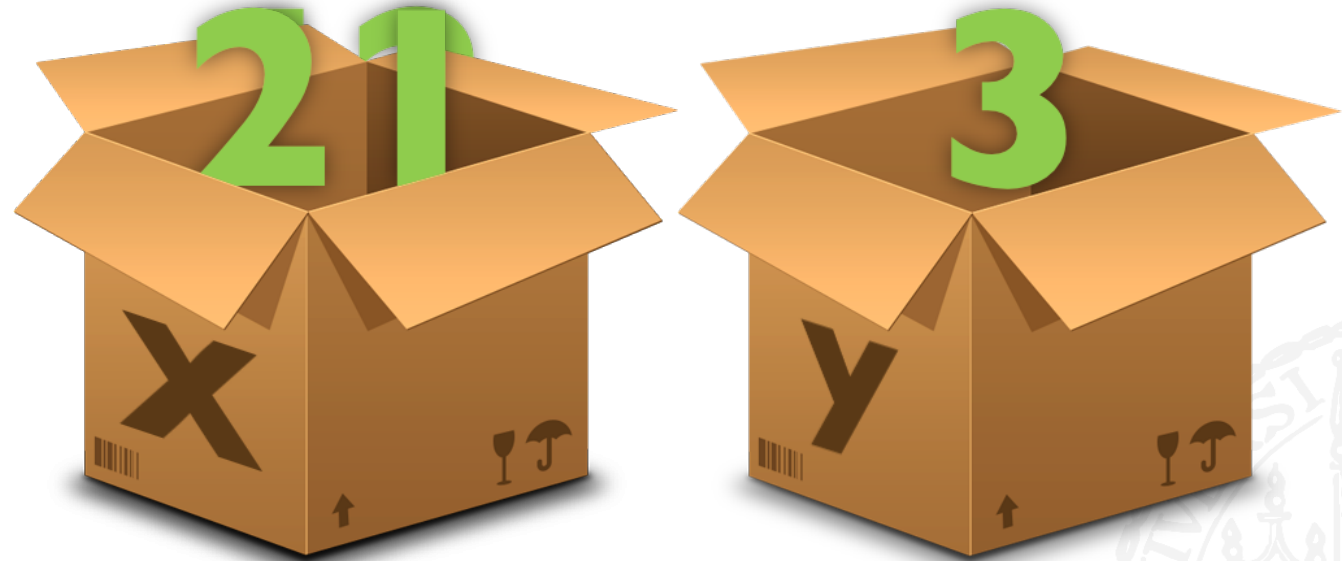
```
#!/usr/bin/python3
#Aufgabe 1-2
#WS 2016/17
#Autorin: Leonie Weißweiler
print ('Hello World')
```

- Beispielprogramm 1-4.py wie beim letzten Mal einfügen
- `#!/usr/bin/python`
- Falls wir das Programm ausführbar machen wollen damit es ohne `python3` aufgerufen werden kann
- Die Shebang line sagt dem Betriebssystem, wo es den Interpreter für `python` finden kann



VARIABLEN

```
>>> x = 42
>>> print(x)
42
>>> x = 21
>>> print(x)
21
>>> y = 3
>>> print(y)
3
>>> print(x)
21
```



VARIABLEN

```
>>> straÙe = "OettingenstraÙe"
```

```
>>> hausnummer = 67
```

```
>>> print(straÙe + hausnummer)
```

```
Traceback (most recent call last):
```

```
  File "<stdin>", line 1, in <module>
```

```
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```



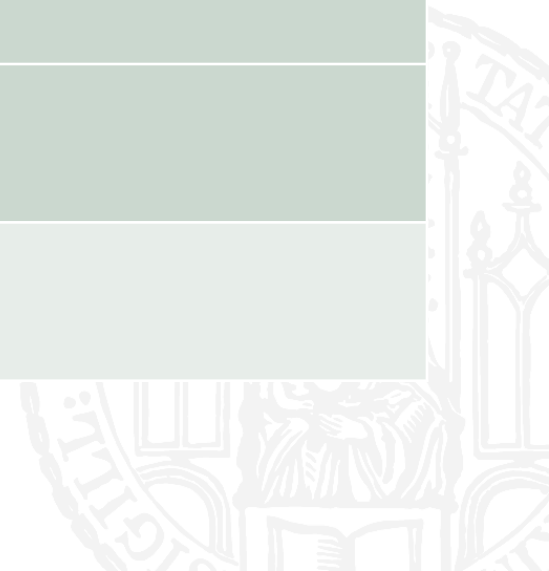
DATENTYPEN

- Jeder Wert hat einen Typ, zum Beispiel:
 - Integer (3, 42, -100)
 - String (“Hallo Welt”, “CIS”)
- Typen verhalten sich unterschiedlich:
 - $3+3=6$,
 - “hallo” + “welt” = “hallo welt”
 - “3” + “3” = “33”



DATENTYPEN

Datentyp	Inhalt	Operatoren
integer	Ganze Zahl	+ - * / > < <= >=
float	Kommazahl	
string	Text	+ *
boolean	Wahrheitswert (True oder False)	&& !



VERGLEICHE

- Man kann Werte mit passenden Typen vergleichen und erhält boolean Werte
- ```
>>> 3 < 5
True
```
- ```
>>> 3 < 5.4  
True
```
- ```
>>> "a" < "b"
True
```
- ```
>>> 3 < "3"  
Traceback (most recent call last):  
File "<stdin>", line 1, in <module>  
TypeError: unorderable types: int() < str()
```



TYPE CASTING

- Man kann manche Werte zwischen Typen konvertieren (“Casten”)
- ```
>>> int(5.6)
5
```
- ```
>>> str(4)
'4'
```
- ```
>>> int("54")
54
```
- ```
>>> int("Max")
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
ValueError: invalid literal for int() with base 10: 'Max'
```



TYPE CASTING

- Man kann manche Werte zwischen Typen konvertieren (“Casten”)
- ```
>>> bool(0)
False
```
- ```
>>> bool(1)  
True
```
- ```
>>> bool(-42.5)
True
```
- ```
>>> bool(“”)  
False
```
- ```
>>> bool(“abc”)
True
```



# QUIZ



# SPRECHSTUNDE

