Python 2.x und Python 3.x

- Python 3.0 wurde Ende 2008 veröffentlicht
  - Zu dem Zeitpunkt war Python 2.6 aktuell
- Python 2.x bekam weiterhin Updates
  - 2 Python-Versionen parallel…
- Support Ende von 2.x Anfang 2020

- Python 2.x Installationen noch oft vorzufinden
  - Viele (kleinere) Module nur für Python 2

Python 2.x und Python 3.x - Unterschiede

- Print-Funktion

Python 2.x

Eingabe:

```
print “Hello, World!”
print(“Hello, World!”)
```

Ausgabe:

```
Hello, World!
Hello, World!
```

Python 3.x

Eingabe:

```
print “Hello, World!”
print(“Hello, World!”)
```

Ausgabe:

```
SyntaxError: invalid syntax
Hello, World!
```

Python 2.x und Python 3.x - Unterschiede

- Integer-Division

Python 2.x

Eingabe:

```
print "3 / 2 =", 3 / 2
print "3 // 2 =", 3 // 2
print "3 / 2.0 =", 3 / 2.0
print "3 // 2.0 =", 3 // 2.0
```

Ausgabe:

```
3 / 2 = 1
3 // 2 = 1
3 / 2.0 = 1.5
3 // 2.0 = 1.0
```

Python 3.x

Eingabe:

```
print("3 / 2 =", 3 / 2)
print("3 // 2 =", 3 // 2)
print("3 / 2.0 =", 3 / 2.0)
print("3 // 2.0 =", 3 // 2.0)
```

Ausgabe:

```
3 / 2 = 1.5
3 // 2 = 1
3 / 2.0 = 1.5
3 // 2.0 = 1.0
```

Python 2.x und Python 3.x - Unterschiede

- Weitere Unterschiede:

https://sebastianraschka.com/Articles/2014_python_2_3_key_diff.html

Syntax von Python

- Unterscheidet sich von den gängigen Sprachen
- Fokus auf Lesbarkeit

JAVA

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

Python

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

Syntax von Python - Einrückung

- Leerzeichen haben eine Bedeutung
- Es darf kein Mix aus Tab und Leerzeichen benutzt werden
  - Manche Editoren ersetzen automatisch Tabs mit Leerzeichen (und andersrum)

```
a = 5
b = 3
if a > b:
  print(a)
else:
  print(b)
```

```
a = 5
b = 3
if a > b:
print(a)
else:
print(b)
```

-> IndentationError

Syntax von Python – Groß- und Kleinschreibung; Semikolons

- Python ist case sensitive
- Print() =/= print()

- Semikolons werden in Python generell nicht benötigt
- Können benutzt werden um mehrere Befehle in eine Zeile zu packen

Syntax von Python – Variablen- und Funktionsnamen

- Ein Name darf…
  - Alphanumerische Zeichen und Unterstriche enthalten
  - Keine Leerzeichen oder Sonderzeichen enthalten
  - Nicht mit einer Zahl beginnen (Buchstaben/Unterstrich erlaubt)

Syntax von Python – Variablen- und Funktionsnamen

- Namen dürfen keine der folgenden Schlüsselwörter sein:

Keywords in Python programming language				
False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield

Syntax von Python – Kommentare

- Beschreibungen des Codes
- Hilfreich für andere Leute

```
# Einzeiliger Kommentar
"""
Ein
Mehrzeiliger
Kommentar
"""


x = 5    # setze die Variable x = 5
```

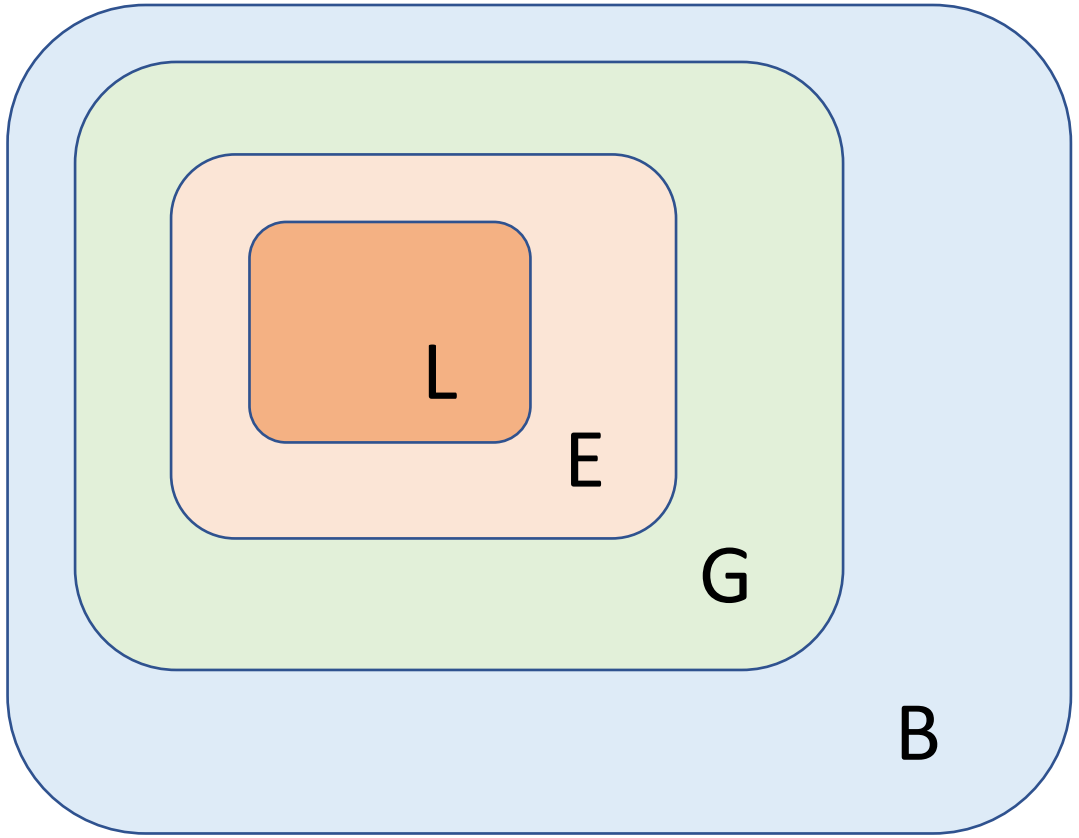
Syntax von Python – Module & Namespaces

- Module erweitern Python um neue Funktionen (=Library)
- Namespaces sind *Räume* innerhalb eines Programms, in denen ein Name gültig ist

Syntax von Python – Module & Namespaces

```
x = "global"

def outerFunction():

    def innerFunction():
        print(x)

    innerFunction()

outerFunction()
```

Syntax von Python – Module & Namespaces



1. **L**ocal: Innerster Scope einer Funktion
2. **E**nclosing: Umschließende Funktion
3. **G**lobal: Namensraum des Moduls
4. **B**uilt-in: Alle in Python schon existierenden Namen

Syntax von Python – Module & Namespaces

- Einbinden eines Moduls mit dem import-Befehl
- Zum Beispiel: random-Modul

```
import random

print(random.randint(3, 9))

# print(dir(random))
```

Syntax von Python – Module & Namespaces

- Funktionen des Built-In Namespaces

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