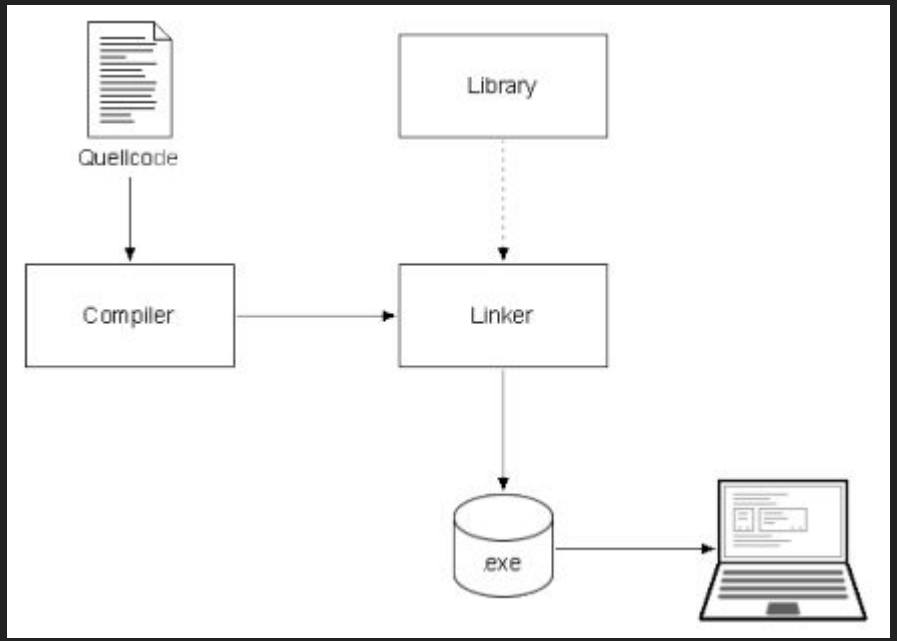
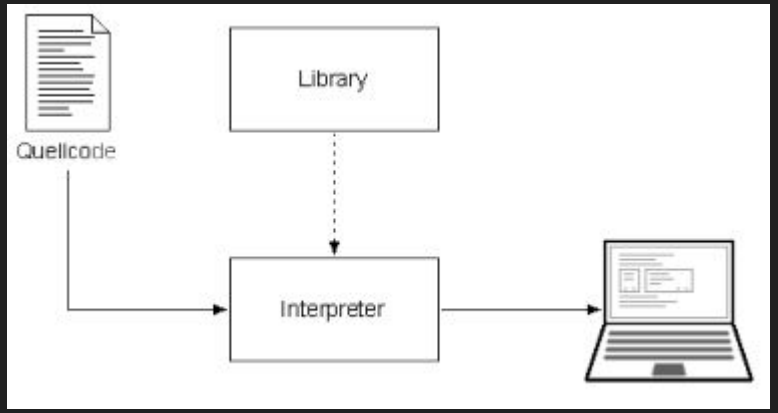
python™

Python Programmierung 01

Torben Friedrich Görner

Ausführung von Python Code



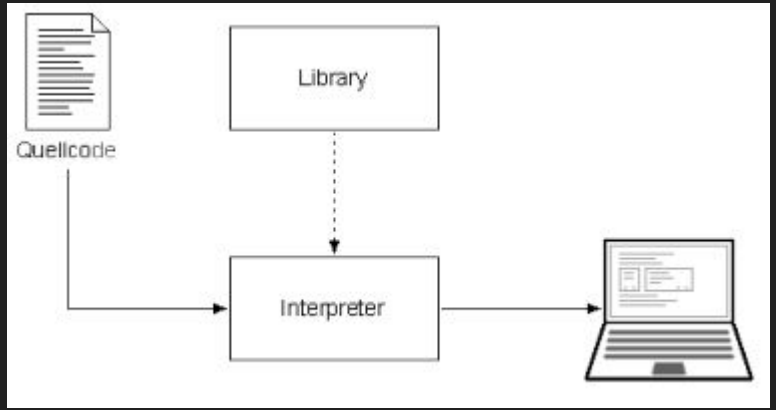


Compiler Modell (Links), Interpreter Modell (Rechts)

Ausführung von Python Code

Python Code wird von einem Interpreter ausgeführt.

Dieser “interpretiert” den Python Code und führt entsprechend die Anweisungen aus.



Python

- Die Programmiersprache Python wurde Anfang der 1990er Jahre von Guido van Rossum am Zentrum für Wissenschaft und Informatik in Amsterdam entwickelt.
- Der Name Python hat nichts mit der Schlange zu tun. Er bezog sich auf die britische Komikertruppe Monty Python. (Die Assoziation zur Schlange kam über das Python-Logo.)
- Im Januar 1994 erschien Python in der Version 1.0.
- Python 2.0 erschien im Oktober 2000 und wird mittlerweile nicht mehr unterstützt.
- Die Python-Version 3.0 erschien im Dezember 2008.

Python

Vorteile

- kostenlos und Open-Source
- modern und wird weiterentwickelt
- vielseitig und bietet Lösungen für unterschiedlichste Anwendungsprobleme
- plattformunabhängig (Läuft auf Windows, Mac, Linux, etc.)
- Python Code ist meist kurz (viel kürzer als beispielsweise Java)
- dynamische Typisierung
- hohe Anwendung im Bereich Data Science, KI, und maschinellem Lernen

Python

Nachteile

- Durch Interpreter schlechtere Performance als Compiler Sprachen wie Java, C oder C++
- agiert nicht so hardwarenah
- benötigt durch dynamische Typisierung mehr Speicherplatz als andere Sprachen
- nicht so gut geeignet für mobile Anwendungen

Python - Kommentare

```
[1]  # syntaktisch korrekter einzeiliger Kommentar
     print("Dieser String wird ausgegeben.")    # Dieser String nicht.
     # Der folgende Befehl wurde auskommentiert:
     # print(42)

     Dieser String wird ausgegeben
```

Python - Kommentare

```
[1] """Dieser Kommentar
    läuft über insgesamt
    drei Zeilen."""
    print(42)

    42
```

Python - Datentypen

- **Ganze Zahlen** (integer, Python-Syntax: int), Bsp. 42, -42
- **Fließkommazahlen** (floation point number, Python-Syntax: float), Bsp. 3,141
- **Boolesche Zahlen** (boolean number, Python-Syntax: bool), True & False
- **Zeichenketten** (string, Python-Syntax: string), Bsp. “Hallo Welt”

Python - Datentypen

- **Listen** (list, Python-Syntax: list) sind in einer geordneten Reihenfolge zusammengefasste Daten beliebigen Typs, die in eckigen Klammern „[“ und „]“ notiert werden. Die Daten werden durch Kommata getrennt.
  - Bsp. zahlen = [0,1,2,3,4,5], namen = [“Torben”, “Franziska”]
- **Tupel / Paare** (tuple, Python-Syntax: tuple),
  - Bsp. Rechteck Höhe und Breite = Tupel (42,10)

Python - Datentypen

- **Dictionaries** (Python-Syntax: dict) sind mithilfe von Schlüssel-Wert-Paaren (key-value pairs) zusammengefasste Daten, die aber in geschweiften Klammern „{“ und „}“ notiert werden. Schlüssel-Wert-Paare werden durch Kommata getrennt, zwischen einem Schlüssel und dem zugehörigen Wert steht ein Doppelpunkt.
  - Bsp. Telefonbuch {“Torben”:1234231, “Franziska”:432131}

Python - Wertzuweisung

- <Name der Variablen> = <Wert>
- Wir müssen keinen Datentypen angeben
- Beispiele
  - x = 5
  - name = ”Müller”
  - planeten = [’Merkur’, ’Venus’, ’Erde’, ’Mars’]

Python - Arithmetische Operatoren

Operation	Operator	Code-Beispiel	Darstellung in der Mathematik
Addition	+	`a + b`	$a + b$
Subtraktion	-	`a - b`	$a - b$
Multiplikation	*	`a * b`	$a \cdot b$
Division	/	`a / b`	$a : b,\ a \div b,\ {}^{a}/_{b}$
Ganzzahlige Division	//	`a // b`	$\lfloor a : b \rfloor,\ \lfloor a \div b \rfloor,\ \lfloor {}^{a}/_{b} \rfloor$
Modulo (Restwert)	%	`a % b`	$a \bmod b$
Potenz	**	`a ** b`	a^b

Python - Vergleichsoperatoren

Operation	Operator	Code-Beispiel	Darstellung in der Mathematik
größer als	>	a > b	$a > b$
größer oder gleich	>=	a >= b	$a \geq b$
kleiner als	<	a < b	$a < b$
kleiner oder gleich	<=	a <= b	$a \leq b$
gleich	==	a == b	$a = b$
ungleich	!=	a != b	$a \neq b$

Python - Boolesche Operatoren

Operation	Operator	Code-Beispiel	Darstellung in der Aussagenlogik
UND	`and`	`a and b`	$a \wedge b$
ODER	`or`	`a or b`	$a \vee b$
NICHT	`not`	`not a`	$\neg a$

Python - Boolesche Operatoren

Übung

Gegeben a = True, b = True, c = False

- a and b
- a and c
- a and b and c
- a and not c
- a or c
- not a or c
- a or (b and c)

Python - Boolesche Operatoren

Übung (Lösung)

Gegeben a = True, b = True, c = False

- a and b = True
- a and c = False
- a and b and c = False
- a and not c = True
- a or c = True
- not a or c = False
- a or (b and c) = True

Python - Kontrollanweisungen

Die bedingte Anweisung: if

```
if bedingung:
    anweisungen
```

```
[1]  alter = 15
     if alter < 18:
         print("Du bist noch nicht volljährig.")
     Du bist noch nicht volljährig.
```

Python - Kontrollanweisungen

Die bedingte Anweisung: if

```
if bedingung:
    bedingte_anweisungen
else:
    alternative_anweisungen
```

```
[1]  die_sonne_scheint = False
     if die_sonne_scheint:
         print("Es ist schönes Wetter.")
     else:
         print("Es ist kein schönes Wetter.")

     Es ist kein schönes Wetter.
```

Python - Kontrollanweisungen

Die bedingte Anweisung: if

```
if bedingung_1:
    anweisungen_1
elif bedingung_2:
    anweisungen_2
elif bedingung_3:
    anweisungen_3
else:
    alternative_anweisungen
```

Python - Übung

Schreibe ein Programm, das überprüft, ob eine Person einen Film schauen darf.

- Gegeben sei ein Alter und ein Boolean, ob eine Person mit ihren Eltern da ist.
- Es soll ausgegeben werden (print), welche Filme eine Person schauen darf. Filme ab 0,6,12,16,18
- Teste dein Programm mit verschiedenen Werten für Alter und Boolean

Python - Kontrollanweisungen

Schleifen - While

Idee : Tue etwas, solange eine Bedingung gilt.

```
while bedingung:
    anweisungen
```

Python - Kontrollanweisungen

Schleifen - While

Idee : Tue etwas, solange eine Bedingung gilt.

```
while bedingung:
    anweisungen
```

Python - Übung

Schreibe ein Programm, das die Fakultät einer Zahl bestimmt. Bsp. 5! = 120 (1*2*3*4*5)

- Eingabe ist eine Zahl, Ausgabe die Fakultät
- Nutze hierfür eine Schleife, sodass du die Zahl beliebig austauschen kannst

Tipp:

Nutze eine Variable (integer) als Schleifenzähler.