

Ein- und Ausgabe

Philipp Hanisch, Valentin Roland

6. Oktober 2022

Python-Grundlagen

1. Input

2. Typen

3. Ausgabe

4. Aufgaben

Input

Einlesen von der Konsole

- Motivation: Nutzerinteraktion
- `input()` oder `input(<prompt>)`
- liest Zeile von der Konsole
- liefert einen String

```
1 ^^Itext = input() # wartet auf Nutzereingabe
2 ^^Iprint(text)
3
4 ^^Iname = input('Wie heißt du?') # Nutzer erhält Eingabehinweis
5 ^^Iprint('Hallo ' + name '!')
6     ^^I
```

Problem

```
1 ^^Inumber = input('Gib eine Zahl ein: ')
2 ^^Iresult = 5 + number
3 ^^Iprint('Deine Zahl plus 5 ergibt: ' + result)
4 ^^I
```

Problem: Fehlerhafte Typen

```
1 ^^Inumber = input('Gib eine Zahl ein: ')\n2 ^^Iresult = 5 + number # Fehler: number ist ein String!\n3 ^^Iprint('Deine Zahl plus 5 ergibt: ' + result)\n4 ^^I
```

Typen

- dynamische Typisierung
- erlaubte Operationen
- Typumwandlung (Casting)

Rückblick: Typübersicht

Name	Funktion
<code>int</code>	Ganzzahl "beliebiger" Größe
<code>float</code>	Kommazahl "beliebiger" Größe
<code>str</code>	Zeichenkette
<code>bool</code>	Wahrheitswert (<code>True</code> , <code>False</code>)
<code>list</code>	gewöhnliche Liste
<code>tuple</code>	unveränderbares n-Tupel
<code>set</code>	(mathematische) Menge von Objekten
<code>dict</code>	Hash-Map

Grundlagen: Typen

- Typbestimmung: `type(...)`
- Casting: `int('42')`, `float('10.3')`, etc.
- potentielle Fehlerquelle

```
1  ^^Ia = int('42')
2  ^^Itext = input('Deine Eingabe: ')
3
4  ^^Ib = int(text) # kann einen Fehler werfen
5
6  ^^I3 + 6.3 # => 9.3
7  ^^Iint(3 + 6.3) # => 9
8
9  ^^Idef harm_folge(n):
10 ^^I    ^^Iif type(n) == int:
11 ^^I    ^^I    ^^Ireturn 1/n
12 ^^I
```

Ausgabe

Ausgabe über print(...)

```
1 ^^Iprint('Wundervoller Text')
2 ^^Iprint('Noch', 'mehr', 'wundervoller', 'Text')
3
4 ^^Iday = input('Welcher Tag ist heute?')
5 ^^Itime = input('Wie spät ist es?')
6 ^^Iprint('Es ist ' + day + ', um ' + time + '.')
7 ^^I
```

Ausgabe über print(...)

```
1 ^^Iprint('Wundervoller Text')
2 ^^Iprint('Noch', 'mehr', 'wundervoller', 'Text')
3
4 ^^Iday = input('Welcher Tag ist heute?')
5 ^^Itime = input('Wie spät ist es?')
6 ^^Iprint('Es ist ' + day + ', um ' + time + '.')
7
8 ^^I# oder schöner:
9 ^^Iprint('Es ist {}, um {}'.format(day, time))
10 ^^I
```

Stringformatierung

String können mit `'...'.format(...)` formatiert werden:

```
1 ^^Inumber = 3
2 ^^Iitem = 'Katzen'
3 ^^Iplace = 'APB'
4
5 ^^I# per Reihenfolge
6 ^^I'Es gibt {} {} im {}!'.format(number, item, place)
7
8 ^^I# per Name
9 ^^I'Es gibt {amount} {things} im {where}!'.format(where=place,
10         amount=10, things=item)
11 ^^I# per Reihenfolge und Name
12 ^^I'Es gibt {} {things} im {}!'.format(number, place, things='
13     Koalas')
```

Aufgaben

1. Fragt den Nutzer nach Vor- und Nachnamen und begrüßt ihn!
2. Lest zwei Zahlen ein und berechnet die Summe!
3. Lest eine (vom Nutzer vorher festzulegende) Anzahl an Zahlen ein. Berechnet die Summe (Maximum, Mittelwert, ...) und gebt sie zusammen mit der Liste der Zahlen aus!
4. Modifiziert euer Programm so, dass der Nutzer nach jeder Zahl gefragt wird, ob er weitere Zahlen eingeben möchte!
5. Wählt eine beliebige natürliche Zahl. Lasst den Nutzer eure Zahl raten. Wenn er falsch geraten hat, gebt einen Hinweis ($>$, $<$)!
6. Bearbeitet die Aufgaben auf der Kurswebsite!