Benutzereingabe: input()-Funktion

1. Benutzereingaben mit Text

Bisher haben unsere Programme nur Daten an die Benutzer ausgegeben, aber haben noch keine Daten durch die Benutzer eingeben lassen. Das ist aber eine sehr häufig benötigte Funktion.

Für die Benutzereingabe sind in der Regel zwei Elemente nötig:

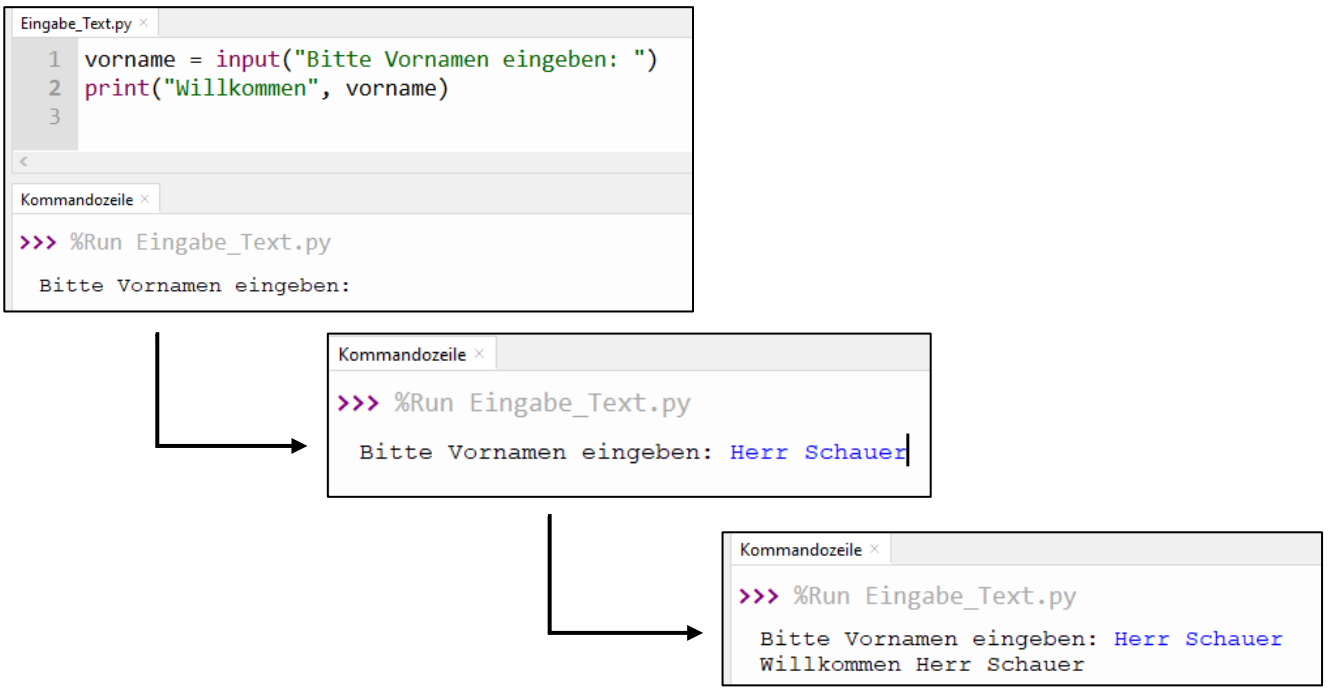
1. Eine Nachricht an den Benutzer, dass etwas eingegeben werden sollen.
2. Die Benutzereingabe muss in einer Variablen abgespeichert werden, damit die eingegebenen Daten im Programm später weiterberarbeitet werden können.

In Python wird dafür die Funktion input() verwendet. Diese Funktion gibt eine Nachricht auf dem Bildschirm aus und wartet dann auf Tastatureingaben, die mit der Taste „Enter“ abgeschlossen werden. Die eingegebenen Zeichen werden dann als Zeichenkette / String in einer Variablen abgespeichert. Das folgende Beispiel macht die Funktionsweise deutlich:

```
1 vorname = input("Bitte Vornamen eingeben: ")
2 print("Willkommen", vorname)
```

Erklärung

Mit der Anweisung input wird der Text „Bitte Vornamen eingeben: “ auf dem Bildschirm ausgegeben. Die Benutzer des Programms haben dann Gelegenheit, Daten mittels Tastatur einzugeben und die Eingabe mit Taste „Enter“ abzuschließen. Sobald das geschieht, werden die eingegebenen Daten in der Variablen *vorname* abgespeichert, so dass sie in der anschließenden Ausgabe verwendet werden können.



2. Benutzereingaben mit Zahlen

Die Funktion input() hat eine auf den ersten Blick überraschende Eigenschaft: Sie interpretiert alle eingegebenen Zahlen als Zeichen, d.h. auch Ziffern werden als Zeichen und nicht als Zahlen betrachtet. Das wird im folgenden Programm deutlich:

Eingabe_Zahl.py

```
1  zahl = input("Bitte geben Sie eine Zahl ein: ")
2  haelfte = zahl / 2
3  print("Die Hälfte der eingegebenen Zahl ist: ", haelfte)
```

Ausgabe (mit Fehlermeldung):

Kommandozeile

```
>>> %Run -c $EDITOR_CONTENT

  Bitte geben Sie eine Zahl ein: 5
  Traceback (most recent call last):
    File "<string>", line 2, in <module>
  TypeError: unsupported operand type(s) for /: 'str' and 'int'
```

Hier wird ausgesagt, dass die Operation „/“, also die Division, mit den Datentypen „str“ (= String, Zeichenkette) und „int“ (= Ganzzahl) nicht durchgeführt werden kann, was ja auch stimmt – Es macht ja auch keinen Sinn, eine Zeichenkette durch die Zahl 2 zu teilen!

Lösung: Typumwandlung

Eine Typumwandlung (cast) wird benötigt, um einen Datentyp in einen anderen Datentyp umzuwandeln. Hierfür stellt Python die Funktionen **int(), float(), str()** und **bool()** zur Verfügung.

Da Python Werte der Funktion input() immer als Text („String“) interpretiert, muss man die eingegebenen Zeichen beim Abspeichern in der Variablen mittels der Funktion float() in eine Dezimalzahl umwandeln:

Eingabe_Zahl.py *

```
1  zahl_zeichenkette = input("Bitte geben Sie eine Zahl ein: ")
2  zahl_dezimalzahl = float(zahl_zeichenkette)
```

(Bessere) Alternative

Eingabe_Zahl.py *

```
1  zahl = float(input("Bitte geben Sie eine Zahl ein: "))
```

Das korrigierte und funktionierende Programm sieht dann so aus:

Eingabe_Zahl.py *

```
1  zahl = float(input("Bitte geben Sie eine Zahl ein: "))
2  haelfte = zahl / 2
3  print("Die Hälfte der eingegebenen Zahl ist:", haelfte)
```

Kommandozeile

```
Bitte geben Sie eine Zahl ein: 5
Die Hälfte der eingegebenen Zahl ist: 2.5
```