



THU

**Technische
Hochschule
Ulm**

Vorkurse der Fakultät Informatik
Programmier-Crashkurs – Aufgaben

M.Reiter & Dr. P.Hönig
September 2025



1. Variablen, Ein- und Ausgabe

Tipp: legen Sie für jede Aufgabe eine neue Datei an! Verwenden Sie die Aufgabennummer als Name z.B. `aufgabe_1_1.py`

Aufgabe 1.1

Erstellen Sie ein Programm, das nach dem Namen des Benutzers fragt und diesen einliest. Anschließend soll der Name für eine Begrüßung verwendet werden. Die Ausgabe könnte dabei wie folgt aussehen:

```
Hallo Julia!
```

Aufgabe 1.2

Finden Sie für folgende Verwendungszwecke geeignete Datentypen. Es kann für jeden Verwendungszweck mehrere Datentypen geben.

Art der Variable	Geeigneter Datentyp
Alter eines Menschen	
Jahresgehalt in ganzen Euro-Beträgen	
Die eulersche Zahl e mit 7 Nachkommastellen	
Das Gewicht in kg	
Initialen einer Person	
Lieblingsessen einer Person	
Anwesenheit einer Person in einem Raum	
Anzahl Bücher, die eine Person gelesen hat	
Preis für ein Produkt	
Zustand, ob Lampe eingeschaltet ist, oder nicht	

Sie können diese Aufgabe auf Papier oder am Rechner mit einem beliebigen Editor lösen.

Aufgabe 1.3

Programmieren Sie eine Anwendung, die zwei Zahlen einliest und anschließend die Ergebnisse der Addition, Subtraktion, Multiplikation, Division und Modulo der beiden Zahlen ausgibt!

Bei der Eingabe von 5 und 2 könnte die Ausgabe wie folgt aussehen:

```
5 + 2 = 7
5 - 2 = 3
...
```

Aufgabe 1.4

Erstellen Sie eine Code-Lösung zur Berechnung des Umfangs und der Fläche eines Kreises. Die berechneten Werte sollen anschließend ausgegeben werden. Für die Berechnung selbst soll der Radius des Kreises vom Benutzer eingelesen werden.

- › Formel für den Kreisumfang: $umfang = 2 * PI * radius$
- › Formel für die Kreisfläche: $flaeche = PI * radius * radius$

Tipp: Weisen Sie der Konstante PI den Wert 3.14159 zu

Aufgabe 1.5

Sie haben ein Curry-Rezept für 6 Personen mit den folgenden Zutaten:

- › 200g Kartoffeln
- › 1 Zwiebel
- › 250g Tomaten
- › 400g Kichererbsen
- › 150 ml Kokosmilch
- › 2 EL Olivenöl

Entwickeln Sie eine Code-Lösung, die die Anzahl an Personen, für die das Gericht gekocht werden soll, einliest, die Zutaten entsprechend umrechnet und in einer Ausgabe darstellt.

Tipp: Einen Zeilenumbruch erhalten Sie mit Hilfe von "`\n`" in der `print()`-Funktion, z.B.:
`print("Hello, \nworld!")`

Die dazugehörige Ausgabe lautet dann:

```
Hello,  
world!
```

Aufgabe 1.6

Schreiben Sie ein Programm, das eine Zahl vom Benutzer einliest und anschließend eine Multiplikationstabelle von 1 bis 10 erzeugt.

Das Ergebnis soll zudem noch als Produkt der „gegenüberliegenden“ Grenzzahl dargestellt werden, also Zahl 1 mit Grenzzahl 10, Zahl 2 mit Grenzzahl 9...

Die Ausgabe könnte für die Eingabe von 4 wie folgt aussehen:

```
1 * 4 = 0.4 * 10  
2 * 4 = 0.8888888888888888 * 9  
3 * 4 = 1,5 * 8  
...  
8 * 4 = 10.666666666666666 * 3  
9 * 4 = 18 * 2  
10 * 4 = 40 * 1
```

Aufgabe 1.7

Werten Sie die folgenden Wahrheitswerte **ohne Zuhilfenahme** eines Programms aus:

Ausdruck	Wahrheitswert
$(3 < 4)$	
"Hallo" == "hallo"	
$zahl_a = 5$ $zahl_b = 4$ $zahl_b = zahl_b + 2$ $zahl_a < zahl_b$	
$5 \leq 10$	
$zahl_a = 3$ $zahl_b = 7$ $zahl_a = zahl_a + 4$ $zahl_a \neq zahl_b$	

Geben Sie die vorliegenden Ausdrücke anschließend in Ihrer Entwicklungsumgebung ein und überprüfen Sie Ihre Ergebnisse!

Aufgabe 1.8

Sie haben folgenden Code vorliegen, erklären Sie, was hier passiert:

```
a = float( input("Geben Sie die Beschleunigung ein: ") )
t = float( input("Geben Sie die Zeit ein, die seit dem Start der Bewegung vergangen ist: ") )

# Zeit-Geschwindigkeit-Gesetz
v = a * t

# Zeit-Ort-Gesetz
s = 0.5 * a * t * t

print(f"Die Geschwindigkeit nach Zeit {t} beträgt {v} und die Strecke entspricht {s}!")
```

Aufgabe 1.9

Ihr Freund ist ein Programmier-Neuling und hat Ihnen den folgenden Code geschickt, damit Sie ihm helfen, das Programm zum Laufen zu bekommen. Korrigieren Sie die Fehler im vorliegenden Code:

```
m = input("Geben Sie die Masse des Objekts (in kg) ein: ")
v = input("Geben Sie die Geschwindigkeit des Objekts (in m/s) ein: ")

Berechnung der kinetischen Energie
E_k = 0.5 * m * v * v

print("Die kinetische Energie des Objekts beträgt {E_k} Joule.")
```

Tipp: Nutzen Sie Ihre Entwicklungsumgebung um Ihre Änderungen zu überprüfen!



2. Bedingte Verzweigungen

Aufgabe 2.1

Folgende Variablen sind definiert:

$a = 4$
 $b = 3$
 $c = 3$
 $d = 4.5$
 $e = 1$

Werten Sie die Wahrheitswerte der folgenden Ausdrücke ohne Zuhilfenahme eines Programms aus:

Ausdruck	Wahrheitswert
$a \neq 4$	
$a > b$	
$b == c$	
$(b + 1) == 4$	
$(a / b) == 1$	
$(b / a) > 0$	
$(a \% e) \neq 0$	
$(a > 0) \text{ and } (a \leq 4)$	
$\text{not } (e < c)$	

Überprüfen Sie Ihre Ergebnisse anschließend mit der Eingabe in Ihre Entwicklungsumgebung!

Aufgabe 2.2

Lesen Sie zwei Zahlen vom Benutzer ein und werten Sie mit Hilfe von Programmcode aus, welche der beiden Zahlen größer und welche kleiner ist, oder ob die Zahlen gleich sind. Anschließend soll eine entsprechende Ausgabe erfolgen.

Aufgabe 2.3

Entwickeln Sie eine Code-Lösung, die unterscheidet, ob eine eingelesene Zahl gerade oder ungerade ist und dieses Ergebnis dem Benutzer zurückmeldet.

Aufgabe 2.4

Erstellen Sie ein Taschenrechner-Programm, das zwei Zahlen und einen Operator ("+", "-", "*" oder "/") einliest. Berechnen Sie das Ergebnis der gewählten Operation und geben Sie das Ergebnis aus.

Aufgabe 2.5

Entwickeln Sie ein Programm, das den Benutzer nach dem Monat fragt und basierend auf dem Monat die entsprechende Jahreszeit (Frühling, Sommer, Herbst oder Winter) ausgibt.

Aufgabe 2.6

Im Beipackzettel eines Medikaments für Kinder wird folgende Dosierung festgelegt:

Gewicht in kg	Einzel-dosis in mg	Tagesdosis in mg
7 - 9	100	300
10 - 12	150	600
13 - 18	200	1000
19 - 25	300	1200

Schreiben Sie ein Programm, welches das Körpergewicht einliest und anschließend die Einzeldosis und das Zeitintervall in Stunden ausgibt, die zwischen zwei Medikamenteneinnahmen liegen sollten, um die Tagesdosis gleichmäßig über den gesamten Tag mit 24 Stunden zu verteilen.

Aufgabe 2.7

Schreiben Sie ein Programm, das eine Zahl einliest und anschließend folgende Unterscheidungen und entsprechende Ausgaben macht:

- › Ist die Zahl ohne Rest durch 3 teilbar, soll „Fizz“ ausgegeben werden
- › Ist die Zahl ohne Rest durch 5 teilbar, soll „Buzz“ ausgegeben werden
- › Ist die Zahl ohne Rest durch 3 und durch 5 teilbar, soll „FizzBuzz“ ausgegeben werden
- › Ist die Zahl weder durch 3 noch durch 5 ohne Rest teilbar, soll nichts ausgegeben werden



3. Schleifen

Tipp: Erstellen Sie eine durchlaufbare Zahlenliste mit Hilfe der `range()` –Funktion:

```
liste = range(0, 10)
```

`liste` enthält nun ein *iterierbares* (=durchlaufbares) Objekt an Werten angefangen von 0 bis einschließlich 9 und eignet sich hervorragend zum Durchlaufen in einer `for`-Schleife.

Aufgabe 3.1

Erstellen Sie ein Programm, welches einen Text und eine Anzahl an Wiederholungen vom Benutzer einliest, um den eingetragenen Text anschließend in der Ausgabe entsprechend oft zu wiederholen. Schreiben Sie die Wiederholung mit allen Ihnen bekannten Schleifen.

Aufgabe 3.2

Schreiben Sie Programmcode, der eine Zahl vom Benutzer einliest und anschließend die Summe aller Zahlen von 1 bis zu einschließlich der eingegebenen Zahl bildet. Am Ende wird das Ergebnis ausgegeben.

Aufgabe 3.3

Programmieren Sie eine Anwendung, die vom Benutzer so lange abfragt, ob es noch regnet, bis der Benutzer mit der Eingabe von „nein“ antwortet. An dieser Stelle wird das Programm nach einer letzten Meldung, dass man nun keinen Schirm mehr benötigt, beendet.

Aufgabe 3.4

Lesen Sie eine Zahl als Benutzereingabe ein. Schreiben Sie nun einen Programmcode, der ermittelt, wie viele Zahlen von 1 bis einschließlich der eingegebenen Zahl ohne Rest durch 3 teilbar sind. Geben Sie das entsprechende Ergebnis anschließend aus.

Aufgabe 3.5

Entwickeln Sie ein Programm, das alle Primzahlen von 1 bis zu einer vom Benutzer eingegebenen Zahl berechnet und ausgibt. Eine Primzahl ist eine Zahl, die nur durch 1 und sich selbst ohne Rest teilbar ist.

Aufgabe 3.6

Erstellen Sie Programmcode, der alle geraden Zahlen von 1 bis zu einer vom Benutzer eingegebenen Zahl addiert und anschließend das Ergebnis ausgibt.

Aufgabe 3.7

Schreiben Sie ein Programm, welches eine Zahl einliest, um dann ein Quadrat in der Konsole zu malen. Dabei soll die Zeile entsprechend viele Zeichen enthalten, wie Reihen ausgegeben werden, nämlich entsprechend der Benutzereingabe.

Mit der Eingabe von 4 könnte die Ausgabe wie folgt aussehen:

```
####  
####  
####  
####
```

Aufgabe 3.8

Schreiben Sie ein kleines Spiel, bei welchem der Spieler eine vom Computer zufällige Zahl erraten muss. Der Spieler hat dabei drei Versuche. Die Zufallszahl soll zwischen 0 und 10 sein. Falls der Spieler falsch liegt, wird ausgegeben, ob die Zufallszahl größer oder kleiner ist als die eingegebene Zahl.

Falls der Spieler seine drei Versuche aufgebraucht hat, wird das Spiel beendet. Ebenso ist das Spiel beendet, wenn der Spieler die Zahl errät.

Tip: Mit Hilfe der beiden nachfolgenden Code-Zeilen lässt sich eine Zufallszahl angefangen von `zahl_a` bis `zahl_b` erzeugen. Achtung: `zahl_b` ist dabei nicht mehr eingeschlossen.

In der ersten Zeile wird das Modul `random` importiert. Diese Zeile sollte tatsächlich ganz oben in Ihrem Code stehen. Die zweite Zeile ruft eine Funktion von `random` auf, um eine zufällige Zahl innerhalb der angegebenen Grenzen auszuwählen.

```
import random  
zufallszahl = random.randrange(zahl_a, zahl_b)
```

Aufgabe 3.9

Entwickeln Sie Programmcode, der einen Würfel simuliert. Hierfür wird eine zufällige Zahl zwischen 1 und 6 (wie ein Würfel) generiert und dem Benutzer mitteilt, welche Zahl geworfen wurde. Der Benutzer wird nach jedem Wurf gefragt, ob er noch einmal würfeln möchte. Das Programm sollte fortlaufend würfeln, solange der Benutzer dies möchte.

Aufgabe 3.10

Entwickeln Sie ein Programm, welches den Durchschnitt von einer Reihe an Zahlen berechnen soll. Zunächst wird vom Benutzer abfragt, wie viele Zahlen erfasst werden sollen. Anschließend wird die Eingabeaufforderung entsprechend oft ausgeführt. Eingegebene Zahlen werden fortlaufend zusammenaddiert und zum Schluss ihr Durchschnitt ermittelt und dem Benutzer mitgeteilt.

Aufgabe 3.11

Schreiben Sie Programmcode mit einer Endlosschleife! ☺
Versuchen Sie, mehr als eine Lösung zu finden.



4. Listen

Aufgabe 4.1

Folgende Variablen sind definiert:

```
liste_a = ["Banane", "Kirsche", "Apfel", "Birne", "Pflaume"]
liste_b = range(0, 4)
```

Ermitteln Sie die Werte der folgenden Ausdrücke ohne Zuhilfenahme eines Programms:

Ausdruck	Wert
<code>len(liste_a)</code>	
<code>liste_a[2]</code>	
<code>len(liste_b)</code>	
<code>liste_b[3]</code>	
<code>liste_b[1] + liste_b[2]</code>	
<code>liste_a[(len(liste_a)-2)]</code>	
<code>liste_b[(1+1)]</code>	
<code>liste_a[4]</code>	
<code>len(liste_a) > len(liste_b)</code>	

Überprüfen Sie Ihre Ergebnisse anschließend mit der Eingabe in Ihre Entwicklungsumgebung!

Aufgabe 4.2

Erzeugen Sie eine Liste von 1 bis 15. Gehen Sie die Liste anschließend iterativ durch, und geben Sie jede dritte Zahl aus.

Hinweis: Ihre Lösung sollte die Zahlen 3, 6, 9, 12 und 15 ausgeben.

Aufgabe 4.3

Erstellen Sie ein Programm, welches vom Benutzer 10 Zahlen einliest und in einer Liste zwischenspeichert. Am Ende soll die Liste wieder für den Benutzer ausgedruckt werden.

Lösen Sie diese Aufgabe einmal mit der Funktion *append* und einmal mit Hilfe vom *Index*.

Aufgabe 4.4

Schreiben Sie ein Programm, das so lange Texteingaben vom Benutzer entgegennimmt, bis der Benutzer die Eingabe „stopp“ tätigt. Geben Sie dem Benutzer aus, wie viele Texteingaben betätigt wurden und welche das waren.

Aufgabe 4.5

Drucken Sie die folgende verschachtelte Liste aus, sodass jede Zahl in einer eigenen Zeile steht!
`schachtel_liste = [[1, 2], [3, 4], [5, 6]]`

Aufgabe 4.6

Die folgende verschachtelte Liste ist definiert:
`schachtel_liste = [[1, 1, 3], [4, 5, 6, 7, 8], [9, 0, 0, 10]]`

Ermitteln Sie den jeweiligen Zahlenindex ohne Zuhilfenahme eines Programms.

Gesuchte Zahl	Index
3	
4	
6	
8	
9	

Überprüfen Sie Ihre Ergebnisse anschließend mit der Eingabe in Ihre Entwicklungsumgebung!

Aufgabe 4.7

Die folgende verschachtelte Liste ist definiert:
`schachtel_liste = [[10, 9, 8, 7], [3, 4, 5, 6, 7, 8], [0, 0, 0, 0]]`

Werten Sie die folgenden Ausdrücke ohne Zuhilfenahme eines Programms aus.

Ausdruck	Wert
<code>len(schachtel_liste)</code>	
<code>len(schachtel_liste[1])</code>	
<code>schachtel_liste[1][3]</code>	
<code>schachtel_liste[0][0]</code>	
<code>schachtel_liste[0][4]</code>	

Überprüfen Sie Ihre Ergebnisse anschließend mit der Eingabe in Ihre Entwicklungsumgebung!

Aufgabe 4.8

Erstellen Sie eine Liste von 1 bis 20. Ersetzen Sie alle durch 3 teilbaren Zahlen mit einem „*“.
 Drucken Sie die veränderte Liste zum Schluss aus.

Tipp: `liste = range(zahl_a, zahl_b)` erzeugt ein iterierbares Objekt, bei dem einzelne Bestandteile nicht überschrieben werden können. Um eine Liste mit Zahlen zu erhalten, muss man explizit eine Zahlen-Liste daraus erzeugen: `liste = list(range(zahl_a, zahl_b))`



5. Funktionen

Aufgabe 5.1

Schreiben Sie ein Programm, das drei Zahlen vom Benutzer einliest und anschließend die kleinste und die größte Zahl ermittelt und dem Benutzer entsprechend ausgibt.

Nutzen Sie für die Benutzereingabe und die Berechnung der Kennzahlen jeweils eine eigene Funktion.

Aufgabe 5.2

Entwickeln Sie Programmcode für die Berechnung des Flächeninhalts eines Rechtecks.

Erstellen Sie dafür eine Funktion, die die Länge und Breite übergeben bekommt und den berechneten Flächeninhalt zurückliefert.

Tipp: Die Formel zur Berechnung des Flächeninhalts lautet: $A = \text{Länge} \times \text{Breite}$

Aufgabe 5.3

Lesen Sie eine Zahl als Benutzereingabe ein. Schreiben Sie eine Funktion, die alle Quadratzahlen von 1 aufwärts bis einschließlich zur eingegebenen Zahl ausgibt.

Tipp: Für die Eingabe von „5“ soll die Ausgabe wie folgt aussehen:

```
1^2 = 1
2^2 = 4
3^2 = 9
4^2 = 16
5^2 = 25
```