# Programmzeilen mit der IDLE Shell testen

Gib die Programmzeilen nach dem Prompt (>>>) ein. Ist das Ergebnis OK oder bekommst du eine Fehlermeldung?

Tipp: Mit ALT + p kannst du die letzte Zeile wiederholen.

### Zeichenketten (strings) testen

|  |  |  |
| --- | --- | --- |
| **Programmzeile** | **OK** | **Fehler** |
| Hallo Welt |  |  |
| "Hallo Welt" |  |  |
| "Hallo Welt' |  |  |
| 'Hallo Welt' |  |  |
| print("Hallo Welt") |  |  |
| pirnt("Hallo Welt") |  |  |

### Zeichenketten in Variablen schreiben

|  |  |  |
| --- | --- | --- |
| **Programmzeile** | **OK** | **Fehler** |
| nachricht = Hallo Welt |  |  |
| nachricht = "Hallo Welt" |  |  |
| nachricht |  |  |
| print(nachricht) |  |  |
| len(nachricht) |  |  |
| type(nachricht) |  |  |
| nachricht.upper() |  |  |

### Zahlen (integer und float) testen

|  |  |  |
| --- | --- | --- |
| **Programmzeile** | **OK** | **Fehler** |
| 3 |  |  |
| 3.14 |  |  |
| 3,14 |  |  |
| print(3) |  |  |
| print(3.14) |  |  |
| print(3,14) |  |  |

### Zahlen in Variablen schreiben

|  |  |  |
| --- | --- | --- |
| **Programmzeile** | **OK** | **Fehler** |
| zahl = 3 |  |  |
| zahl = 3.14 |  |  |
| zahl |  |  |
| print(zahl) |  |  |
| len(zahl) |  |  |
| type(zahl) |  |  |
| zahl.upper() |  |  |

### Die Farbe kennzeichnet einen Text mit besonderer Bedeutung

|  |  |
| --- | --- |
| **Farbe** | **Bedeutung** |
| rot |  |
| grün |  |
| violett |  |

# Mit der IDLE Shell rechnen

Gib die Programmzeilen nach dem Prompt (>>>) ein. Ist das Ergebnis OK oder bekommst du eine Fehlermeldung?

### Rechnen mit Zeichenketten

|  |  |  |
| --- | --- | --- |
| **Programmzeile** | **OK** | **Fehler** |
| "Hallo" + "Welt" |  |  |
| "Hallo" + " " + "Welt" |  |  |
| "Hallo " + "Welt" |  |  |
| "Hallo " - "Welt" |  |  |
| print("Hallo " + "Welt") |  |  |
| 3 \* "Hallo Welt" |  |  |
| 3 \* "Hallo Welt " |  |  |
| 3 / "Hallo Welt " |  |  |
| print(3 \* "Hallo Welt ") |  |  |
| nachricht = "Hallo Welt " |  |  |
| print(3 \* nachricht) |  |  |

### Rechnen mit Zahlen

|  |  |  |
| --- | --- | --- |
| **Programmzeile** | **OK** | **Fehler** |
| 3 + 4 |  |  |
| 3 - 4 |  |  |
| 3 \* 4 |  |  |
| 3/4 |  |  |
| 30//4 |  |  |
| 30%4 |  |  |
| ergebnis = 3 \* 4 |  |  |
| ergebnis/5 |  |  |
| ergebnis//5 |  |  |
| print(ergebnis/5) |  |  |

### Punktrechnung vor Strichrechnung

|  |  |  |
| --- | --- | --- |
| **Programmzeile** | **OK** | **Fehler** |
| 3 + 4 \* 2 |  |  |
| (3 + 4) \* 2 |  |  |
| 12 – 3 / 2 |  |  |
| (12 – 3) / 2 |  |  |
| 12 – 3 // 2 |  |  |
| (12 – 3) // 2 |  |  |

### Zahl in Zeichenkette umwandeln – Zeichenkette in Zahl umwandeln

|  |  |  |
| --- | --- | --- |
| **Programmzeile** | **OK** | **Fehler** |
| zahl = 3 |  |  |
| str(zahl) |  |  |
| zahl = 3.14 |  |  |
| str(zahl) |  |  |
| zeichenkette = "3.14" |  |  |
| int(zeichenkette) |  |  |
| float(zeichenkette) |  |  |

# Mit dem IDLE Editor ein Programm schreiben

Mit dem Menüpunkt File/ New File erzeugst du ein leeres Editor-Fenster mit dem Titel "untitled".

Gib die Programmzeilen im Editor-Fenster ein. Die Farbe dort kennzeichnet einen Text mit besonderer Bedeutung.

Speichere den Inhalt mit dem Menüpunkt File/ Save unter dem Dateinamen "mein\_erstes\_programm" im Ordner Python/03\_Rechnen\_mit\_Zeichenketten\_und\_Zahlen

Starte das Programm mit dem Menüpunkt Run/ Run Module.

Notiere die print-Ausgaben in der Tabelle.

### Rechnen

|  |  |
| --- | --- |
| **Programmzeile** | **print-Ausgabe** |
| # Das ist ein Kommentar |  |
| # Rechnen |  |
| nachricht = "Hallo Welt " |  |
| print(3 \* nachricht) |  |
| ergebnis = 3 \* 4 |  |
| print(ergebnis/5) |  |

### Umwandeln

|  |  |
| --- | --- |
| **Programmzeile** | **print-Ausgabe** |
| # Umwandeln |  |
| zahl = 3.14 |  |
| print(zahl) |  |
| print(str(zahl)) |  |
| zeichenkette = "3.14 " |  |
| print(zeichenkette) |  |
| print(3 \* zeichenkette) |  |
| print(3 \* float(zeichenkette)) |  |

### Variablen ausgeben

|  |  |
| --- | --- |
| **Programmzeile** | **print-Ausgabe** |
| # Variablen ausgeben |  |
| print("nachricht =", nachricht) |  |
| print("ergebnis =", ergebnis) |  |
| print("zahl =", zahl) |  |
| print("zeichenkette =", zeichenkette) |  |

# Aufgabe: Ziffern mit Punkten darstellen

Die Ziffern auf einem Kassenzettel sind aus Punkten zusammengesetzt. Mit 5x7 Punkten können die Ziffern 0 bis 9 gut lesbar dargestellt werden.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ● | ● | ● |  |
| ● |  |  |  | ● |
| ● |  |  |  | ● |
| ● |  |  |  | ● |
| ● |  |  |  | ● |
| ● |  |  |  | ● |
|  | ● | ● | ● |  |

Schreibe ein Programm, das mit print-Befehlen und dem Zeichen \* oder ● eine Ziffer in ein 5x7 Raster druckt:

1. Drucke eine Null
2. Drucke eine Acht
3. Drucke eine Eins
4. Drucke eine Ziffer deiner Wahl

Tipp: Das Zeichen ● hat den Unicode 25CF. Es kann über die Windows-Zeichentabelle eingegeben werden.



# Eine Liste speichert viele änderbare Elemente

Gib die Programmzeilen nach dem Prompt (>>>) ein.

Merke: Der Index steht in eckigen Klammern. Der Index beginnt mit Null!

Notiere die print-Ausgaben in der Tabelle.

### Eine Liste anlegen – mit eckigen Klammern!

|  |  |
| --- | --- |
| **Programmzeile** | **print-Ausgabe** |
| vornamen = ["Axel", "Elke", "Martin] |  |
| print(vornamen) |  |
| print(vornamen[0]) |  |
| print(vornamen[0:2]) |  |
| print(vornamen[-1]) |  |
| vornamen[2] = "Fritz" |  |
| print(vornamen) |  |

### Eine Liste erweitern

|  |  |
| --- | --- |
| **Programmzeile** | **print-Ausgabe** |
| vornamen = vornamen + ["Heike", "Sabine"] |  |
| print(vornamen) |  |
| vornamen += ["Markus"] |  |
| print(vornamen) |  |

### Eine leere Liste anlegen und füllen

|  |  |
| --- | --- |
| **Programmzeile** | **print-Ausgabe** |
| buchstaben = [] |  |
| buchstaben.append("a") |  |
| print(buchstaben) |  |
| buchstaben.append("b") |  |
| print(buchstaben) |  |

### Elemente einer Liste löschen

|  |  |
| --- | --- |
| **Programmzeile** | **print-Ausgabe** |
| print(vornamen) |  |
| vornamen.remove("Heike") |  |
| print(vornamen) |  |
| del vornamen[0] |  |
| print(vornamen) |  |
| del vornamen |  |
| print(vornamen) |  |

### Ein zufälliges Element aus einer Liste auswählen

|  |  |
| --- | --- |
| **Programmzeile** | **print-Ausgabe** |
| import random |  |
| handzeichen = ["Schere", "Stein", "Papier"] |  |
| print(random.choice(handzeichen)) |  |

# Ein Tupel speichert viele nicht änderbare Elemente

Gib die Programmzeilen nach dem Prompt (>>>) ein.

Merke: Der Index steht in eckigen Klammern. Der Index beginnt mit Null!

Notiere die print-Ausgaben in der Tabelle.

### Ein Tupel anlegen – mit runden Klammern!

|  |  |
| --- | --- |
| **Programmzeile** | **print-Ausgabe** |
| punkt = (-10, 5, 7) |  |
| print(punkt) |  |
| print(punkt[0]) |  |
| print(punkt[0:2]) |  |
| print(punkt[-1]) |  |
| punkt[2] = 15 |  |
| punkt = (-10, 5, 15) |  |
| print(punkt) |  |

### Ein Element im Tupel suchen

|  |  |
| --- | --- |
| **Programmzeile** | **print-Ausgabe** |
| print(punkt) |  |
| print(punkt.count(7)) |  |
| print(punkt.index(7)) |  |

# Ein Dictionary speichert Paare von Schlüssel und Wert

Mit dem Menüpunkt File/ New File erzeugst du ein leeres Editor-Fenster mit dem Titel "untitled".

Gib die Programmzeilen im Editor-Fenster ein.

Speichere den Inhalt mit dem Menüpunkt File/ Save unter dem Dateinamen "woerterbuch" im Ordner Python/04\_Listen\_und\_Woerterbuecher

Starte das Programm mit dem Menüpunkt Run/ Run Module.

Notiere die print-Ausgaben in der Tabelle.

### Ein leeres dictionary anlegen – mit geschweiften Klammern – und füllen

|  |  |
| --- | --- |
| **Programmzeile** | **print-Ausgabe** |
| # Wörterbuch Englisch – Deutsch |  |
| # Leeres dictionary anlegen |  |
| englisch\_deutsch = {} |  |
| # dictionary füllen |  |
| englisch\_deutsch["cat"] = "Katze" |  |
| englisch\_deutsch["dog"] = "Hund" |  |
| englisch\_deutsch["cow"] = "Kuh" |  |
| print(englisch\_deutsch) |  |
| print(englisch\_deutsch["dog"]) |  |
| englisch\_deutsch["sheep"] = "Schaf" |  |
| print(englisch\_deutsch) |  |

### Schlüssel und Werte des dictionary ausgeben

|  |  |
| --- | --- |
| **Programmzeile** | **print-Ausgabe** |
| # Schlüssel ausgeben |  |
| print(englisch\_deutsch.keys()) |  |
| print(englisch\_deutsch.values()) |  |

### Neues dictionary mit Vertauschung von Schlüssel und Wert erstellen

Wenn jeder Wert nur einmal im dictionary vorkommt, geht das mit folgender Programmzeile (Erläuterung später).

|  |  |
| --- | --- |
| **Programmzeile** | **print-Ausgabe** |
| # neues dictionary erstellen |  |
| deutsch\_englisch = dict((v,k) for k,v in englisch\_deutsch.items()) | |
| print(deutsch\_englisch) |  |
| # Schlüssel ausgeben |  |
| print(deutsch\_englisch.keys()) |  |
| print(deutsch\_englisch.values()) |  |

# Der Computer fragt …

Mit dem Menüpunkt File/ New File erzeugst du ein leeres Editor-Fenster mit dem Titel "untitled".

Gib die Programmzeilen im Editor-Fenster ein.

Speichere den Inhalt mit dem Menüpunkt File/ Save unter dem Dateinamen "summe\_ausgeben" im Ordner Python/05\_Benutzereingaben

Starte das Programm mit dem Menüpunkt Run/ Run Module.

Notiere die print-Ausgaben in der Tabelle.

### Der Computer erwartet Zahlen

|  |  |
| --- | --- |
| **Programmzeile** | **print-Ausgabe** |
| # Summe von zwei Zahlen ausgeben |  |
| # Benutzereingaben anfordern |  |
| zahl1 = input("Gib die erste Zahl ein ") |  |
| zahl2 = input("Gib die zweite Zahl ein ") |  |
| # Strings in Dezimalzahlen umwandeln |  |
| zahl1 = float(zahl1) |  |
| zahl2 = float(zahl2) |  |
| # Summe ausgeben |  |
| print("Die Summe der Zahlen ist", zahl1 + zahl2) |  |

### Der Computer erwartet Strings

|  |  |
| --- | --- |
| **Programmzeile** | **print-Ausgabe** |
| # Summe von zwei Strings ausgeben |  |
| # Benutzereingaben anfordern |  |
| str1 = input("Gib den ersten String ein ") |  |
| str2 = input("Gib den zweiten String ein ") |  |
| # Summe ausgeben |  |
| print("Die Summe der Strings ist", str1 + str2) |  |

### Aufgabe: Wörterbuch erweitern

Das Dictionary englisch\_deutsch soll erweitert werden. Schreibe das Programm dazu.

1. Lege das Dictionary englisch\_deutsch an und fülle es mit 3 Paaren.
2. Fordere ein neues englisches Wort an – den Schlüssel.
3. Fordere das passende deutsche Wort an – den Wert.
4. Erweitere das Dictionary mit Schlüssel und Wert.
5. Drucke das erweiterte Dictionary

Was passiert, wenn der Schlüssel bereits vorhanden ist?

# Der Computer unterscheidet zwischen "wahr" und "falsch"

Gib die Programmzeilen nach dem Prompt (>>>) ein.

Notiere die print-Ausgaben in der Tabelle.

### Bedingung mit Zahlen

|  |  |
| --- | --- |
| **Programmzeile** | **print-Ausgabe** |
| print(1 == 2) |  |
| print(1 != 2) |  |
| print(1 < 2) |  |
| print(1 > 2) |  |
| print(3 == 3) |  |
| print(3 != 3) |  |
| print(3 <= 3) |  |
| print(3 >= 3) |  |

### Bedingung mit Strings

|  |  |
| --- | --- |
| **Programmzeile** | **print-Ausgabe** |
| print("Hallo" == "Welt") |  |
| print("Hallo" != "Welt") |  |
| print("Montag" == "Montag") |  |
| print("Montag" != "Montag") |  |

### Bedingung mit range(stop)

|  |  |
| --- | --- |
| **Programmzeile** | **print-Ausgabe** |
| bereich = range(10) |  |
| print(list(bereich)) |  |
| print(1 in bereich) |  |
| print(10 in bereich) |  |

### Bedingung mit range(start, stop)

|  |  |
| --- | --- |
| **Programmzeile** | **print-Ausgabe** |
| bereich = range(2, 10) |  |
| print(list(bereich)) |  |
| print(1 in bereich) |  |
| print(9 in bereich) |  |

### Bedingung mit range(start, stop, step)

|  |  |
| --- | --- |
| **Programmzeile** | **print-Ausgabe** |
| bereich = range(0, 10, 2) |  |
| print(list(bereich)) |  |
| print(3 in bereich) |  |
| print(8 in bereich) |  |

# Der Computer unterscheidet Fälle

### Ist die Bedingung "wahr", werden die Programmzeilen darunter ausgeführt.

Gib die Programmzeilen nach dem Prompt (>>>) ein.

Notiere die print-Ausgaben in der Tabelle.

### Eine Bedingung – zwei Fälle

|  |  |
| --- | --- |
| **Programmzeile** | **print-Ausgabe** |
| wert = 5 |  |
| if wert < 10: |  |
| print("wert ist kleiner als 10") |  |
| print("Ich gehöre auch zu der Bedingung") |  |

### Eine Bedingung und die Alternative – zwei Fälle

|  |  |
| --- | --- |
| **Programmzeile** | **print-Ausgabe** |
| if wert < 10: |  |
| print("wert ist kleiner als 10") |  |
| else: |  |
| print("wert ist größer oder gleich 10") |  |

### Mehrere Bedingungen und die Alternative – vier Fälle

|  |  |
| --- | --- |
| **Programmzeile** | **print-Ausgabe** |
| if wert == 10: |  |
| print("wert ist gleich 10") |  |
| elif wert == 4: |  |
| print("wert ist gleich 4") |  |
| elif wert == 5: |  |
| print("wert ist gleich 5") |  |
| else: |  |
| print("keine Bedingung ist erfüllt") |  |