



# Übungen – Auffrischung der Grundlagen

1. Erstellen Sie eine **Liste** namens `fruits` mit den Werten "apple", "banana" und "cherry".
  - Fügen Sie dann die Frucht "`orange`" am Ende der Liste hinzu und
  - "`grape`" am Anfang der Liste.

Geben Sie die endgültige Liste aus (print).



# Übungen – Auffrischung der Grundlagen

2. Erstellen Sie eine **Liste** namens **numbers** mit den Werten 1, 2, 3, 4, 5.

- Entfernen Sie die Zahl 3 aus der Liste und geben Sie die Liste aus.
- Entfernen Sie dann das letzte Element - **nutze pop() funktion** -und geben Sie die Liste erneut aus.



# Übungen – Auffrischung der Grundlagen

3. Erstellen Sie ein **Tuple** namens `coordinates` mit den Werten `10, 20, 30`.

Greifen Sie auf das erste und das letzte Element des Tuples zu und geben Sie diese aus (print).



# Übungen – Auffrischung der Grundlagen

4. Erstellen Sie ein **Tuple** **person** mit den Werten "Alice", 25, "Engineer".

Entpacken Sie die Werte in die Variablen **name**, **age** und **profession** und geben Sie diese aus.



# Übungen – Auffrischung der Grundlagen

5. Erstellen Sie zwei **Tuples** `tuple1` mit den Werten `1, 2, 3` und `tuple2` mit den Werten `4, 5, 6`.

Kombinieren Sie die beiden Tuples zu einem neuen Tuple `merged_tuple` und geben Sie dieses aus.



# Übungen – Auffrischung der Grundlagen

6. Erstellen Sie ein Dictionary `car` mit den Schlüsseln `"brand"`, `"model"` und `"year"`, und den Werten `"Toyota"`, `"Corolla"`, `2010`.

Fügen Sie einen neuen Schlüssel `"color"` mit dem Wert `"blue"` hinzu und aktualisieren Sie den Wert von `"year"` auf `2022`. Geben Sie das endgültige Dictionary aus.

**Iterieren** Sie durch das Dictionary und geben Sie für jeden Schlüssel den entsprechenden Wert aus.

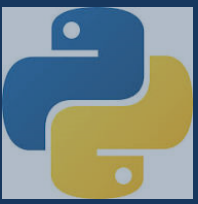


# Übungen – Auffrischung der Grundlagen

7. Erstellen Sie eine **Liste** mit den Werten: 1, 2, 2, 3, 4, 4, 5

Wandeln Sie die Liste in einen **geeigneten Datentyp** um, sodass alle doppelten Werte entfernt werden.

Speichern Sie das Ergebnis in der Variable `unique_numbers` und geben Sie es aus.



# Übungen – Auffrischung der Grundlagen

8. Erstellen Sie eine Variable `temperature` mit einem Startwert, z. B. `22`.

Schreiben Sie ein Programm, das die Temperatur überprüft und je nach Wert folgendes ausgibt:

- Wenn `temperature` größer als 30 ist, soll ausgegeben werden: `"Zu heiß"`
- Wenn `temperature` kleiner als 10 ist, soll ausgegeben werden: `"Zu kalt"`
- In allen anderen Fällen: `"Angenehm"`

Ändern Sie den Wert von `temperature` und testen Sie, ob alle drei Fälle korrekt funktionieren.





# Übungen – Auffrischung der Grundlagen

9. Erstelle ein Programm, das die **Zahlen von 1 bis 20** durchgeht.

Geben Sie dabei nur die Zahlen aus (print), die ohne Rest durch **3** teilbar sind.

z.B.  $7 \div 3 = 2$ , Rest 1 -> wird nicht ausgegeben

Hinweis: Löse die Aufgabe ohne den  $/$ -Operator zu verwenden.

Tipp: Es gibt einen anderen Operator, mit dem man den Rest einer Division berechnen kann.



# Übungen – Auffrischung der Grundlagen

10. Erstellen Sie eine Variable `text` mit dem Wert "Hallo Python Welt"

- Geben Sie die **Länge** des Textes aus.
- **Wandeln** Sie den gesamten Text in **Großbuchstaben** um.
- Ersetzen Sie das Wort "Welt" durch "Community"
- Geben Sie das Ergebnis jeweils mit `print()` aus.

Hinweis: Verwenden Sie eingebaute String-Methoden, wann immer möglich – vermeiden Sie eigene Schleifen oder manuelle Ersetzungen.



# Übungen – Auffrischung der Grundlagen

11. Erstellen Sie die Variable `name` = "Anna Müller"

- Trennen Sie den Namen in Vorname und Nachname (Nutzen Sie dafür eine passende Zeichenkettenmethode, um die beiden Namensteile voneinander zu trennen und separat auszugeben.)
- Speichern Sie beides in zwei Variablen `vorname` und `nachname`
- Kombinieren Sie beides mit einem f-string zu folgendem Satz:  
"Hallo, Anna Müller! Schön, dass du da bist."
- Geben Sie den Satz aus.



# Übungen – Auffrischung der Grundlagen

12. Erstellen Sie die Variable `wort = "Programmieren"`

Nutzen Sie dafür slice Syntax `text[start:ende:schritt]` , um gezielt auf bestimmte Teile der Zeichenkette zuzugreifen.

- Geben Sie nur die ersten 5 Buchstaben aus
- Geben Sie die letzten 4 Buchstaben aus
- Geben Sie jedes zweite Zeichen aus (P, o, r, ...)
- Drehen Sie das Wort um (rückwärts)



# Übungen – Auffrischung der Grundlagen

13. Gegeben:

$a = 17$

$b = 4$

- Berechnen Sie, **wie oft** die Zahl  $b$  **ganz** in  $a$  hineinpasst. Speichern Sie das Ergebnis in **ganze\_anzahl**.
- Berechnen Sie den Rest, der übrig bleibt, wenn man  $a$  durch  $b$  teilt. Speichern Sie das Ergebnis in **rest**.
- Berechnen Sie, wie viel jeder Person bekommt, wenn  $a$  Euro gleichmäßig an  $b$  Personen verteilt werden. Speichern Sie das Ergebnis in **anteil**.
- Speichern Sie in **potenz**, wie viel  $a$  hoch  $b$  ergibt.
- Geben Sie alle Ergebnisse mit `print()` aus – jeweils mit einer kurzen Beschreibung.



# Übungen – Auffrischung der Grundlagen

14. Gegeben:

`x = 5`

`y = 10`

`z = x`

- Prüfen Sie, ob `x` und `y` gleich groß sind. Speichern Sie das Ergebnis in `ist_gleich`.
- Prüfen Sie, ob `x` kleiner ist als `y` und gleichzeitig größer als `2`. Speichern Sie das Ergebnis in `zwischenwert`.
- Prüfen Sie, ob `x` nicht gleich `y` oder `z` größer als `3` ist. Speichern Sie das Ergebnis in `bedingung1`.
- Prüfen Sie, ob `x` und `z` wirklich identisch sind (nicht nur inhaltlich gleich, sondern identisch im Speicher). Speichern Sie das Ergebnis in `identisch`.
- Geben Sie alle Ergebnisse mit `print()` aus – z. B.:

`x und y sind gleich: False`

`x liegt zwischen 2 und y: True`

`Bedingung erfüllt: True`

`x und z sind identisch: True`



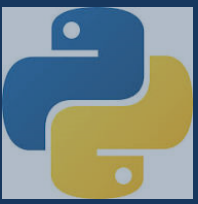
# Übungen – Auffrischung der Grundlagen

15. Gegeben:

```
farben = ["rot", "grün", "blau"]
```

```
text = "Programmieren macht Spaß"
```

- Prüfen Sie, ob "grün" in der Liste farben enthalten ist. Speichern Sie das Ergebnis in `gruen_drin`.
- Prüfen Sie, ob "gelb" nicht in der Liste farben enthalten ist. Speichern Sie das Ergebnis in `gelb_nicht_drin`.
- Prüfen Sie, ob der Buchstabe "S" im Text vorkommt. Speichern Sie das Ergebnis in `buchstabe_drin`.
- Prüfen Sie, ob das Wort "lernen" nicht im Text vorkommt. Speichern Sie das Ergebnis in `wort_nicht_drin`.
- Geben Sie alle Ergebnisse mit `print()` aus.



# Übungen – Auffrischung der Grundlagen

16. Schreiben Sie ein Programm, das den Benutzer auffordert, ein Passwort einzugeben.

Solange das Passwort **nicht korrekt** ist, soll die Eingabe **wiederholt** werden.

Erst wenn das richtige Passwort eingegeben wurde, soll "**Zugriff erlaubt**" ausgegeben werden.

Gegeben:

```
korrektes_passwort = "python123,,
```

Hinweis: Verwenden Sie `input()` , um die Eingabe vom Benutzer zu erhalten.





# Übungen – Auffrischung der Grundlagen

## Hilfreiche Ressourcen:

<https://docs.python.org/3/>

<https://www.w3schools.com/python/>

<https://www.programiz.com/python-programming>