


AUFGABEN SESSION 8:

AUFGABE 1: AUSGABE VON PI, SIN(3) UND

1. Importiere die Standardbibliothek math
2. Gebe Pi auf der Konsole aus
3. Gebe Sinus(3) auf der Konsole aus
4. Berechne 151515! faculty(151515) – auf eigene Gefahr! 
hint: faculty ist im Modul math enthalten.

AUFGABE 2: WAS BIN ICH?

Du erhältst eine nicht leere Liste mit einer Länge $n > 0$. In der Liste befinden sich Strings und Integer. Definiere ein Modul (eigene .py Datei) mit der Funktion `was_bin_ich(param)`. Die Funktion erhält einen Parameter `param` und prüft, ob es sich dabei um einen Integer oder einen String handelt. Nach der Prüfung gibt die Funktion die Zeichenkette zurück: 'Ich bin ein {Datentyp}: {Wert}'

Implementiere/importiere dieses Modul in einer neuen Datei (z.B. `main.py`) und prüfe die gegebene Liste.

Hint: Du benötigst in der `main.py` eine Schleife um die einzelnen Elemente der Liste zu 'holen'.

Input:

```
liste = ['Hi', 13, 'Jo', 'Ichbins', 35, 'Banane', 'Kevin', 35, '17', 'Kamera']
```

Expected output:

```
Ich bin ein String: Hi
Ich bin ein Integer: 13
Ich bin ein String: Jo
Ich bin ein String: Ichbins
Ich bin ein Integer: 35
Ich bin ein String: Banane
Ich bin ein String: Kevin
Ich bin ein Integer: 35
Ich bin ein String: 17
Ich bin ein String: Kamera
```

AUFGABE 3: WER KANN MISCHELN?

Erweitere Aufgabe 2 nun um die Funktion `shuffle`. Diese findet sich im Modul `random` der Standardbibliothek. Bevor die Liste geprüft wird soll diese zufällig gemischt werden.

Input:

```
liste = ['Hi', 13, 'Jo', 'Ichbins', 35, 'Banane', 'Kevin', 35, '17', 'Kamera']
```

Expected output (reihenfolge kann abweichen da shuffle):

```
Ich bin ein String: 17
Ich bin ein Integer: 13
Ich bin ein String: Ichbins
Ich bin ein String: Kamera
Ich bin ein String: Jo
Ich bin ein String: Hi
Ich bin ein Integer: 35
Ich bin ein String: Banane
Ich bin ein String: Kevin
Ich bin ein Integer: 35
```

AUFGABE 4: FAMILIENSTAMMBAUM



Lade das File „Klassen_Session_8.py“ in dein PyCharm Projekt. Erstelle dann ein neues File in dem du die zur Verfügung stehende Klasse aus dem File importierst. Initialisiere dann 7 Personen mit ihrem Namen. Verknüpfe die 7 Personen über die Attribute „elternteil_1“ und „elternteil_2“ so, dass am Ende ein Stammbaum über drei Ebene (Kind, Eltern, Großeltern) entsteht. Ruf dann die Methode „sag_hallo()“ für das Kind und beide Eltern auf. Dabei sollen die erstellten Objekte der Eltern nicht direkt benutzt werden.

AUFGABE 5: FAMILIENSTAMMBAUM HARDCORE



Erweitere Aufgabe 4, indem du alle den Stammbaum direkt bei der Initialisierung der Objekte erstellst. Nutze dazu die Attribute „elternteil_1“ und „elternteil_2“ aus der `__init__` Methode.