

**Das EVA-Prinzip mit TigerJython - vom Integer zum String****#Benötigtes Modul**

```
from random import randint
```

**#Eingabe**

```
ende = inputInt("Ich berechne dir eine Zufallszahl zwischen 1 und der ganzen Zahl, die du jetzt eingeben kannst:")
```

**#Verarbeitung**

```
zufallszahl = randint(1,ende) #
```

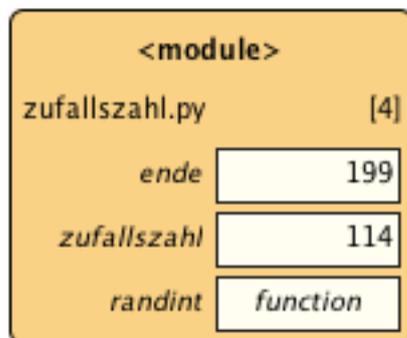
**#Ausgabe**

```
print zufallszahl
```

**Aufgaben**

**Hinweis** zur Aufgabenstellung: „Ergänze“ (ab 3. ) bedeutet, dass zum vorhandenem Programmtext die notwendigen Zeilen hinzugefügt werden, so dass das Programm immer länger wird.

1. Implementiere das o.a. Programm. Nach einem # steht ein Kommentar, der vom Python-Interpreter nicht abgearbeitet wird, sondern beschreibende Funktion hat.
2. **Speichere** das Programm in deinem Ordner. Die Benennung wurde im Unterricht erläutert.
3. Schalte den Debugger von TigerJython ein. Beobachte den Programmablauf. Beschreibe die Einträge und deren Inhalte.



4. Ergänze die Ausgabe mit Hilfe der Dialog-Übersicht von TigerJython, so dass ein modaler Dialog<sup>1</sup> die Ausgabe der Zahl übernimmt.
5. Ergänze die Ausgabe um einen Text. Hinweis: um eine Variable zusammen mit einem Text auszugeben, wird die Variable mit einem Komma vom Textteil getrennt.

Beispiel

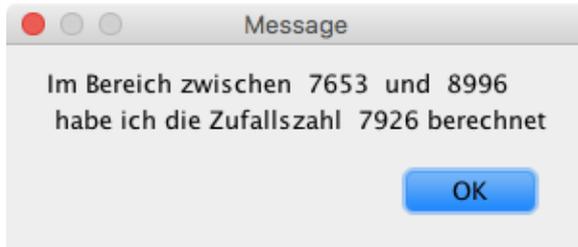
```
msgDlg("Ich habe die Zufallszahl ", zufallszahl, "berechnet")
```

Textteil
Variable
Textteil

6. Ergänze die Eingabe um einen modalen Dialog mit einer sinnvollen Titelzeile.
7. Ergänze die Ausgabe in der Konsole (print-Anweisung) um einen Text.

<sup>1</sup> Ein modaler Dialog öffnet sich über einer Seite meist nach einer Aktion (etwa: eine Taste wird gedrückt). Der Dialog wartet auf Nutzer-Eingaben. Die Seite darunter ist nun nicht erreichbar, oft ist sie abgedunkelt.

8. Öffne ein neues Tab im Editor. Kopiere alles aus dem ersten Tab (CTRL a , dann CTRL v drücken oder über den Menüpunkt *Bearbeiten*) in das neue Tab.
9. Ändere das bestehende Programm jetzt so ab, dass nicht nur das Ende des Bereiches aus dem die Zufallszahl berechnet wird, sondern auch die Startzahl des Zufallsbereiches eingegeben werden kann. Überflüssige Mehrfacheingaben und Ausgaben des vorherigen Programms löscht du einfach. Beispielsweise könnte die Ausgabe deines Programmes so aussehen:



Um einen Zeilenumbruch an einer bestimmten Stelle durchzuführen, damit die Ausgabe in der Breite nicht zu lang ist, brauchst du dort nur die Zeichen

`\n` (Backslash=umgedrehter Schrägstrich und den Buchstaben n) in den Ausgabebetext schreiben.

### Hinweise

Du hast Variablen, Funktionen und Module kennengelernt. Die Variablen *zufallszahl* und *ende* haben den Datentyp Integer (Ganzzahl). Die Funktion *randint* bekommt zwei Parameter: die Zahl 1 als Anfang des Bereichs aus dem die Zufallszahl berechnet wird und die Variable *ende*. Die Funktion *randint* gibt eine ganze Zahl zurück.

### Weitere Aufgaben

1. Lese im Skript "Python - Eine Einführung in die Computer-Programmierung von Tobias Kohn" im Kapitel 3 die Einführung und weiter im Kapitel 3.1 *Grundoperationen und das Zahlenformat*. Danach solltest Du mit den Grundrechenarten und deren Anwendung in Python vertraut sein. Somit lässt sich die BMI-Berechnung einfach durchführen.
2. Schreibe ein Programm, das nach Eingabe der Körpergröße in Meter (z.B. 1,56 m) und des Gewichts in kg (z.B. 64,3 kg) den BMI (Body Mass Index) berechnet. Dieser wird ausgegeben. Die Wahl der Eingabedialoge und Ausgabedialoge ist dir freigestellt. Informiere dich durch eine sachgerechte Recherche über den BMI und verwende diese Ergebnisse für den Verarbeitungsteil deines Programms.

**Hinweise:** Du brauchst eine andere Eingabefunktion. Probiere aus und bereite dich darauf vor, deine Auswahl begründen zu können. Benenne die Variablen sinnvoll!

Beachte, dass bei einer Eingabe von Kommazahlen, z.B. 1,56 in Python statt *Komma* ein *Punkt* gesetzt werden muss: 1.56 . Sonst gibt es einen Fehler.

3. Ergänze dein BMI-Programm so, dass auch der Name des Benutzers erfragt und bei der Ausgabe berücksichtigt wird, z.B.

