



Aufgabenblatt 4

Abgabe: Die Lösungen sollten zeitnah als IPYNB-Datei fertiggestellt werden. Der Code muss ausreichend kommentiert sein und die Variablen müssen sinnvoll benannt werden. Sie müssen die Aufgabe selbst programmiert haben. Sie können Fragen in Form von Kommentaren im Code stellen, falls etwas nicht funktioniert hat. Die Antwort erfolgt dann im Praktikum mündlich. Sie dürfen nie mehr als drei Aufgabenblätter im Rückstand sein.

Hilfsmittel: Kein Copy-Paste aus dem Internet, alles muss selbstständig programmiert sein. Sie dürfen die IPython-Notebook-Skripte aus der Vorlesung (liegen nach der jeweiligen Vorlesung auf Ilias) und die Python-Einführung auf Ilias verwenden. Außerdem ist die Hilfe-Funktion `help(...)` und die Methode `dir(...)` zur Auflistung der verfügbaren Funktionen zu empfehlen.

Anwesenheit: Grundsätzlich herrscht Anwesenheitspflicht. Ein Attest ist notwendig, wenn jemand nicht kommen kann. Bei unentschuldigtem Fehlen ist das Praktikum nicht bestanden. Von der Teilnahme an der Klausur wird in diesem Fall dringend abgeraten.

Aufgabe 4.0

Schreiben Sie die Funktion `temperature(s)`, welche einen String liefert bekommt. Der String soll aus einer Zahl und einer Einheit (C, F, K) bestehen und soll überprüft werden. Wenn der Typ des Eingangsparameters kein String ist, der String keine Zahl enthält und wenn die Einheit nicht aus (C, F, K) besteht, dann soll eine *Exception* geworfen werden.

Die Funktion soll eine Dictionary zurückliefern, welches die errechneten Temperaturwerte für Celsius, Fahrenheit und Kelvin enthält.

Umrechenformeln:

$$C = K - 273.15$$

$$F = C \cdot \frac{9}{5} + 32$$

```
>>> temperatur ( '0C' )           # 0 Celsius
{'K': 273.15, 'C': 0.0, 'F': 32.0
>>> temperatur ( '0F' )           # 0 Fahrenheit
{'K': 255.37, 'C': -17.78, 'F': 0.0
>>> temperatur ( '0K' )           # 0 Kelvin
{'K': 0.0, 'C': -273.15, 'F': -459.67}
```

Aufgabe 4.1

Schreiben Sie eine Funktion `prim(n)`, die einen Wert vom Typ `Bool` zurückliefert: Sie soll also `True` zurückliefern, falls die übergebene Zahl eine Primzahl darstellt und andernfalls `False` zurückliefern. Die übergebene Zahl `n` soll nach ihrem Typ überprüft werden. Falls der Typ kein Integer ist, dann soll eine *Exception* geworfen werden.

Aufgabe 4.2

Verwenden Sie die Funktion `prim(n)` aus der vorhergehenden Aufgabe, um eine Liste aller Zwillingsprimzahlen zwischen 2 und 10 000 zu erzeugen. Die übergebene Zahl `n` soll nach ihrem Typ überprüft werden. Falls der Typ kein Integer ist, dann soll eine *Exception* geworfen werden.

Anmerkung: Eine Zahl p heißt Zwillingsprimzahl, wenn $p + 2$ ebenfalls eine Primzahl ist.

Aufgabe 4.3

Spielereien mit Streudiagrammen unter Verwendung von `plt.scatter()`.

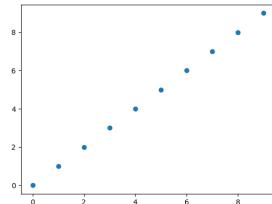
Dafür sollen Sie wie folgt `matplotlib` importieren:

```
import matplotlib.pyplot as plt
```

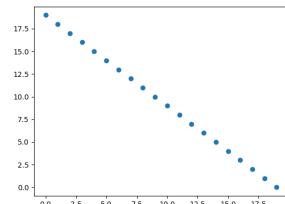
Außerdem sollen sie, dass die Graphiken innerhalb des Notebooks erzeugt werden das folgende Magic-Command für Anaconda verwenden.

```
%matplotlib inline
```

(a) Geben Sie Python-Code an, um folgendes Streudiagramm zu zeichnen:



(b) Geben Sie Python-Code an, um folgendes Streudiagramm zu zeichnen:



(c) Geben Sie Python-Code an, um folgendes Streudiagramm zu zeichnen; Sie dürfen dazu nicht mehr als 15 Zeilen Code verwenden:

