Einführung in das Programmieren

Übungsblatt 4

Christoph Draxler, Vanessa Reichel

12.06.2023

1 Zugriff auf Dateien: Urlaubsfieber

Mithilfe der Library `os` kannst du auf Dateien auf deinem Laptop zugreifen. Vergiss nicht die Library mit `import os` zu importieren, um alle Funktionen dieser Library nutzen zu können.
Achtung: Bei Befehlen, die Dateien oder Ordner löschen, ist Vorsicht geboten! Es gibt **keinen Papierkorb!** Die Wiederherstellung ist **wenn überhaupt** nur sehr umständlich möglich.

- Lade den Ordner **Urlaub** aus dem LSF herunter und speichere diesen Ordner auf Deinem Rechner ab.
- Welche Daten liegen in diesem Ordner? Bestimme, ob es sich dabei um Files oder Ordner handelt.
- Welche Dateien liegen im Unterordner `Reservierungsbestätigungen`?
- Lies die Datei `Sehenswürdigkeiten.txt` aus und speichere alle Sehenswürdigkeiten in einer Liste ab. **Tipp:** Mit dem Befehl .strip() lassen sich die Zeilenumbrüche, die ebenfalls aus dem File ausgelesen werden, entfernen.
- Privatsphäre muss sein: Lösche die Datei `Tagebuch.txt`

2 Zugriff auf Audiodateien

Mit der Library `wave` kann Python auch auf Audiodateien im WAV-Format zugreifen. Mit den folgenden Befehlen können Sie wichtige Angaben über WAV-Audiodateien erfahren:

```
import wave

filename = 'XYZ' # hier einen Dateinamen eingeben!
with wave.open(filename, 'rb') as audiofile:
  params = audiofile.getparams()
```

`params` enthält nun Angaben zur Sampling- oder Framerate, Anzahl Kanäle, usw. in Form eines sog. *named tuple*, d. h. einer Liste fester Länge mit benannten Elementen. Diese Elemente kann man entweder mittels eines numerischen Index extrahieren, oder mittels Feldnamen:

```
params[2]
params.framerate
```

In beiden Fällen wird eine Framerate von 44100 zurückgegeben.

- Schreiben Sie eine Funktion `playtime(audiofile)`, die für eine Audiodatei im WAV-Format die Abspieldauer in Sekunden angibt. Wie berechnet Sie diese Dauer?
- Schreiben Sie eine Funktion `totalPlaytime(directory)`, die in einem Verzeichnis nach Audiodateien im WAV-Format sucht und deren Abspieldauer summiert. Verwenden Sie dazu den Befehl `os.walk()`

3 Reguläre Ausdrücke

Mithilfe von regulären Ausdrücken kann ein **Muster** definiert werden. Ein Muster kann zum Beispiel eine Folge aus bestimmten Buchstaben (Groß- und Kleinbuchstaben), Zahlen oder Sonderzeichen sein. Das Muster wird verwendet, um einen String nach diesem Muster zu durchsuchen. Dieses Verfahren ist sehr effizient. Oftmals können dadurch komplexe `if` und `else`- Bedingungen umgangen werden. Reguläre Ausdrücke gibt es in vielen Programmiersprachen (sie sind nicht python-spezifisch). Es lohnt sich daher, sich mit regulären Ausdrücken vertraut zu machen.
Tipp für Interessierte: Im Internet existiert eine Vielzahl an RegEx-Cheatsheets (RegEx = **Reg**ular **Ex**pression), die die wichtigsten Informationen anschaulich zusammenfassen.

Löse die folgende Aufgabe mithilfe von regulären Ausdrücken:

Die Funktion `passwort_ueberpruefen(passwort)` überprüft ob ein eingegebenes Passwort aus mindesten 8 Zeichen (egal ob chars, Ziffern oder Sonderzeichen) besteht, mindestens einen Großbuchstaben und mindestens eine Ziffer enthält. Sind diese Bedingungen erfüllt, wird `"Das Passwort wird akzeptiert"` zurückgegeben, ansonsten wird der Nutzer informiert, welche Bedingung noch erfüllt werden muss, damit das Passwort akzeptiert wird.

4 Wiederholung

4.1 Lückentext: `addition()`

Versuche zu verstehen, was die Funktion berechnen soll. Fülle dann die Lücken aus und überprüfe, ob deine Version der Funktion die gewünschten Ergebnisse liefert.

```
def addition(von, bis, inklusion):
  if von <= bis:
    if von == bis and inklusion:
      return(_____)
    elif von == bis:
      return(_____)
    else:
      return(von + _____(von +1, bis, inklusion))
  else:
    return(addition(_____, _____, inklusion))
```

Stimmen deine Ergebnisse mit den folgenden überein?
`addition (1, 6, False) = 15`
`addition (1, 6, True) = 21`
`addition (100, 1, True) = 5050`

- Markiere am linken Rand den oder die Terminalfälle der Rekursion.
- Wozu wird das Argument `inklusion` verwendet?
- Welche Auswirkung hat es, wenn man die Bedingungen in der zweiten und dritten `if`-Klausel vertauscht?

4.2 Das große Würfeln

Ein Würfel wird von einem Nutzer `x`-mal geworfen. Schreibe eine Funktion, die `x`-mal würfelt und festhält, welche Zahl wie oft gewürfelt wurde.

4.3 Wer wird Millionär

Gegeben ist ein dictionary: die Keys sind Fragen; die Values die entsprechenden Antworten. Schreibe eine Funktion, die eine zufällige Frage auswählt und den Nutzer über die Konsole eine Antwort eingeben lässt. Informiere den Nutzer, ob die eingegebene Antwort stimmt und was die richtige Antwort ist, falls eine falsche Antwort gegeben wurde!

```
qanda = {
"Womit beginnt eine Funktionsdefinition in Python?" :  "def",
```

```
"Mit welchem Befehl kann man ein dictionary erstellen?" :  "dict()",
"Mit welchem Befehl wird eine Liste sortiert?" :  "sort()",
"Ein dictionary besteht aus Keys und ___?" :  "Values",
"Wie heißt der erste Bestandteil einer Schleife?" :  "Initialisierung",
"Eine rekursive Funktion besteht aus einem Rekursionsfall und einem ___?" :  "Terminalfall",
"Mit welchem Befehl kann eine Zahl in einen String konvertiert werden?" :  "str()"
}
```