**Globale Verständnisfragen**

Zum Abschluss dieses Lernmoduls wollen wir noch einmal die wichtigsten Erkenntnisse und Informationen, die Sie aus diesem Lernmodul mitnehmen sollen, mit einigen zusammenfassenden Fragen verschiedenen Typs abfragen und wiederholen. Sie können diese Fragen als "inverted classroom" verstehen, und damit als komplementär zu den vorgeschlagenen größeren Übungen im nächsten Abschnitt. Die Idee ist, dass Sie sich hier überlegen können, ob Sie die gestellten Fragen selbst beantworten können. Falls es noch Unklarheiten gibt, können Sie die entsprechenden Antworten nochmals direkt in den passenden Abschnitten des Lernmoduls rekapitulieren. Dementsprechend sind die Antworten auch absichtlich skizzenhaft. Bitte zögern Sie nicht, bei konkreten Fragen einen der Kanäle zur Hilfe zur Selbsthilfe zu nutzen.

**Vokabeltest**

**Was ist eine CSV-Datei?**

CSV ist ein einfaches textbasiertes Format, um Tabellen in Dateien abzuspeichern, in einer standardisierten und plattformunabhängigen Art und Weise. Im Wesentlichen wird jede Zeile einer Tabelle abgespeichert als string, mit einem standardisierten Trennsymbol zwischen den Spalten der Tabelle. Wie der Name impliziert, ist das Standard-Trennsymbol das Komma.

**Was ist eine Exception?**

Eine Exception ist eine programmatische Möglichkeit, bei einem antizipierten Fehler vom normalen Programmablauf abzuweichen, anstatt das Programm "crashen zu lassen". Exceptions sind somit eine gute Programmierpraxis, um bspw. Fehleingaben durch Benutzer\*innen zu antizipieren. Ihr Syntax wird hier erläutert. Exceptions sind zudem eng verbunden mit Debuggern, und machen so unser Leben als Programmierer\*innen einfach(er).

**Was ist eine Assertion?**

Eine Assertion ist gewissermaßen eine "Exception light". Sie generiert eine Exception, kann aber zu effizienterer Fehlersuche beitragen. Sie eignet sich insbesondere für antizipierte Fehler seitens der Benutzer\*innen, wie z.b. eine Vorzeichenbedingung.

**Syntaxfragen**

**Wie öffnet man eine Textdatei zum Lesen? Wie zum Schreiben? Wie zum Weiterschreiben am Dateiende?**

Dateien werden in Python mittels file = open(filename, mode) geöffnet. Hierbei ist filename ein String, der Pfad und Dateiname beinhaltet, der Standardpfad ist das aktuelle Verzeichnis. mode bestimmt die Art des Dateizugriffs: mode=r bedeutet Lesezugriff, mode=w Schreibzugriff mit Überschreiben des Inhalts, und mode=a Schreibzugriff mit Anhängen an das Dateiende. Jeder open() befehl sollte mittels file.close() beendet werden. All dies gilt (bei uns) nur für Textdateien.

**Wie fängt man eine bekannte Exception?**

Generell wird Code, in dem man eine Exception fangen möchte, in einem Block platziert, der von try: und except: eingeschlossen wird. Nach dem except Befehl folgt ein weiterer Block mit der Behandlung des aufgetretenen Fehlers. Eine benannte Exception, bspw. ein ValueError, wird einfach nach except aufgeführt: except ValueError: block.

**Wie fängt man eine unbekannte Exception?**

Hier wird einfach kein Exception-Name nach except aufgeführt.

**Einfache Verständnisfragen**

**Erklären Sie, nachdem Sie die gesamte Lerneinheit bearbeitet haben, die Funktionsweise der Funktion is\_float() aus der Miniübung zu CSV-Dateien.**

Der Quellcode der Funktion lautet:

def is\_float(n):

try:

float(n)

return True

except ValueError:

return False

Bekanntlich kann mit dem Befehl float() irgendeine Zeichenkette (oder allgemein irgendeine Variable beliebigen Typs) in den Datentyp float konvertiert werden. In der Funktion wird dieser Befehl nun in eine Exception gewickelt. Schlägt die Konvertierung fehl, so wird ein ValueError geworfen, und wir folgern, dass dann die Eingabe keine in einen float konvertierbare Variable war.

**Was ist der Unterschied zwischen einer Assertion und einer Exception?**

Beides sind Möglichkeiten, um im Programmablauf auf Fehler zu reagieren. try/except Blöcke sind hilfreicher bei Fehlern der Benutzer\*innen, während assert eher hilfreich ist, um eigene Fehler zu vermeiden, und wenn sie auftreten, sie schneller zu finden.

**Wann sollten Sie Assertions verwenden, wann Exceptions?**

Assertions sind prinzipiell effizienter im Sinne der Ausführungsgeschwindigkeit von Programmen. Warum dies so ist, führt allerdings im Rahmen dieses Einsteigerkurses zu weit.

**Was ist der Unterschied zwischen einem Modul und einem Paket?**

Ein Modul ist eine Sammlung von Dateien, die gemeinsame Funktionalität bereitstellen. Die Dateien müssen dazu in einem Verzeichnis liegen. Ein Paket ist eine Sammlung von Modulen.

**Transferfragen**

**Welche Exceptions können beim Lesezugriff auf eine Textdatei auftreten?**

Die wichtigsten Exception bei dieser Operation ist FileNotFoundError, die genau so funktioniert, wie der Name impliziert. Viele Fehler bei Dateioperationen besitzen keine "eigene" Exception, vielmehr wird alles auf OSError abgebildet. Details finden Sie hier.

**Welche Exceptions können beim Schreibzugriff auf eine Textdatei auftreten?**

Siehe vorherige Frage.

**Finden Sie heraus, wie Sie abfragen können, ob eine gegebene Datei mit absolutem Pfad existiert.**

Hierzu binden Sie das Modul os.path ein und verwenden den Befehl os.path.exists().

**Finden Sie heraus, wie Sie eine Datei in Python kopieren können, ohne sie explizit zu lesen und zu schreiben.**

Diese Frage ist ein Klassiker für etwas, was auf StackOverflow hervorragend beantwortet wird: https://stackoverflow.com/questions/123198/how-to-copy-files

Spielen Sie BugBounty mit Kommiliton\*innen: Ein\*e von Ihnen schreibt einen Codeschnipsel mit einem absichtlichen, idealerweise gut versteckten Fehler. Eine Idee dazu ist, in der Dokumentation eine gewisse Funktionalität zu behaupten, die in der Implementierung für Sonderfälle nicht erreicht wird, oder allgemein fast nicht erreicht wird. Ein\*e andere von Ihnen versucht, diesen Fehler im Debugger zu lokalisieren.