

Basic-IO Grundlagen

Übung 3

Einführung in die Programmierung

Über dieses Übungsblatt

In diesem Übungsblatt lernen wir die Konzepte des letzten Übungsblatts in Python zu übertagen. Wir werden ebenfalls die ersten Schritte in unserem Hausgemachten minimalistischen Grafik-Frameworks Basic-IO gehen. **Sie dürfen auf allen Übungsblätter nur explizit erlaubte Bibliotheken (außer Basic-IO) importieren und benutzen.**

Python Listen

In den folgenden Aufgaben werden Sie mit der Python eigenen Datenstruktur **List** (Liste) arbeiten. Es handelt sich dabei um die Python-Version eines [Arrays](#). Listen wurden bereits im letzten Tutorium kurz angeschnitten, da aber das dieswöchige Tutorium ausfällt möchten wir Ihnen an dieser Stelle ein kurzes Beispiel geben wie man Listen benutzt.

Man kann Listen von verschiedenen Datentypen wie z.B. ganzen Zahlen (int), Fließkommazahlen (float) oder Charakterabfolgen (str) erstellen und mit dem `[]`-Operator auf einen bestimmten Eintrag zugreifen (und bei Bedarf verändern).

Anbei finden Sie drei mögliche Arten über eine Liste zu iterieren: (1) Mit der in der Vorlesung vorgestellten *while*-Schleife (2) Mit einem Iterator über die Anzahl an Einträgen in der Liste (gegeben durch die [len](#)-Funktion) (3) Mit einem Iterator über alle Einträge. **Alle drei Optionen sind valide Möglichkeiten über ein Array zu iterieren und geben volle Punktzahl. Option (1) ist maschinennäher und hilft uns zu verstehen, dass eine *for*-Schleife auch nichts anderes als eine *while*-Schleife ist. Option (2) ist ähnlich zu dem wie in den meisten Programmiersprachen programmiert wird. Option (3) ist eine Python-Schnellversion, aber ungeeignet wenn man den Elementen tatsächlich andere Werte zuweisen möchte.**

```
test_list = [1, 5, 10, 20, 400]

# (1): So verstehen sie das Konzept

i = 0
while i < len(test_list):
    print(test_list[i])
    i = i+1

# (2): So wird in den meisten Programmiersprachen programmiert

for i in range(len(test_list)):
    print(test_list[i])

# (3): So wird in Python oft programmiert

for item in test_list:
    print(item)
```

Aufgabe Hello Basic-IO

Letzte Änderung: 15. June 2023, 12:38 Uhr

7 Punkte — [im Detail](#)

Ansicht:  | 

Hello World

Die gewünschte Ausgabe dieser Aufgabe.

In dieser Aufgabe gehen Sie erste Schritte in unserem hauseigenen, auf Einfachheit ausgelegten Grafikframework **Basic-IO**. Basic-IO erlaubt es Ihnen einfache Formen und Farben auf den Bildschirm zu bringen, sowie Tastatureingaben zu verarbeiten.

1) Lesen und verstehen Sie die bereitgestellte Dokumentation [basic_io_Handbuch.pdf](#) oder die automatisch generierte [Online-Dokumentation](#) auf Englisch.

Bemerkung: Auf Windows ist es Notwendig ihren Code in einen sog. [Main-Guard](#) zu schreiben, da ansonsten das Programm mehrfach ausgeführt wird.

```
if __name__ == "__main__":  
    Hier kommt ihr Code rein
```

2) Versuchen Sie sich an einem ersten Schritt mit dem Programm und zeigen Sie den Text "Hello World" in der Farbe Schwarz auf der Mitte des Ausgabebildschirms an und umrahmen Sie den Text mit einem roten Rechteck.

Aufgabe Variable Swap

Letzte Änderung: 10. May 2022, 11:38 Uhr

5 Punkte — [im Detail](#)

Ansicht:  | 

Beim Schreiben von Programmcode kann ein*e Programmierer*In auf die Aufgabe treffen zwei Variablen, sagen wir `x` und `y` zu vertauschen. Wenn man aber nicht darüber nachdenkt, kann einem hier leicht ein Fehler unterlaufen. Seien beispielsweise:

```
x = 5  
y = 15
```

1) Warum tauscht der folgende Programmcode die Werte der beiden Variablen nicht aus?



```
x = y  
y = x
```

2) Wie muss man den Programmcode verändern damit das gewünschte Ergebniss erzielt wird? Geben sie auf jeden Fall auch eine Lösung an, die ohne Tuple-Assignment arbeitet da dies eine Python spezifisches Feature ist und in anderen Programmiersprachen oft nicht verfügbar ist.

Aufgabe Hier kommt die Sonne

Letzte Änderung: 14. April 2022, 14:10 Uhr

14 Punkte — [im Detail](#)

Ansicht:  |  Im 21. Jahrhundert möchte man immer mehr Dinge automatisieren, sogar Liedtexte.

Zunächst möchten wir uns mit einem Klassiker des Deutschrocks befassen.

1a) Schreiben Sie ein Programm, das eine Schleife über die Zahlen 1-10 iteriert. Für die Zahlen 0-9 soll es die Zahl selbst mit dem `print`-Befehl ausgeben und für die Zahl 10 soll es das Wort `aus` ausgeben.

1b) Schreiben Sie ein Programm, das eine Schleife über die Zahlen 1-4 iteriert. Für die Zahlen 1, 2 und 4 soll die Zahl gefolgt vom Text ", hier kommt die Sonne" ausgegeben werden. Für die Zahl 3 soll die Zahl gefolgt vom text ", sie ist der hellste Stern von allen".

Nun möchten wir ein altes englisches [Weihnachtslied](#) nachprogrammieren. Jede Strophe besteht aus der der letzten Strophe und einer neuen Zeile. Der Anfang geht folgendermaßen:

```
On the first day of Christmas my true love sent to me
A partridge in a pear tree.

On the second day of Christmas my true love sent to me
Two turtle doves,
And a partridge in a pear tree.

On the third day of Christmas my true love sent to me
Three French hens,
Two turtle doves,
And a partridge in a pear tree.
```

die weiteren Zeilen fügen folgende Geschenke hinzu:

- four calling birds
- five gold rings
- six geese a-laying
- seven swans a-swimming
- eight maids a-milking
- nine ladies dancing
- ten lords a-leaping
- eleven pipers piping
- twelve drummers drumming

2) Schreiben Sie ein Programm, dass das oben gezeigte Weihnachtslied mithilfe von `for` Schleifen ausgibt; Zahlen können durch die entsprechenden Ziffern ersetzt werden. Kopieren Sie dafür die folgende Liste in ihren Code:

```
gifts = ['partridge in a pear tree', 'turtle doves', 'French hens', 'calling birds', 'gold rings', 'geese a-
laying', \
'swans a-swimming',      'maids a-milking', 'ladies dancing', 'lords a-leaping', 'pipers piping', 'drummers
drumming',]
```

Aufgabe Mittelwert

Letzte Änderung: 15. June 2023, 12:38 Uhr

14 Punkte — [im Detail](#)

Ansicht:  | 

Es sei wieder eine Liste von Fließkommanzahlen gegeben. Wir nehmen an, daß dies gemessene Daten sind, und möchten den Mittelwert und die Standardabweichung berechnen (zwei wichtige Kenngrößen aus der Statistik, die eine Schätzung für den Erwartungswert und die zu erwartende Variabilität der Daten erfassen). Die mathematischen Definitionen lauten wie folgt:

Datenvektor: $a = (a_1, \dots, a_n) \in \mathbb{R}^n, \quad n \in \mathbb{N}, \quad n \geq 1$

Mittelwert: $\mu = \frac{1}{n} \sum_{i=1}^n a_i$

Standardabweichung: $\frac{1}{n} \sum_{i=1}^n (a_i - \mu)^2$

1) Schreiben Sie ein Programm, das den Mittelwert einer Liste berechnet. Berechnen Sie den Mittelwert der unten angegebenen Testliste, kopieren Sie diesen dazu an den Anfang ihres Codes.

2) Schreiben Sie ein Programm, das die Varianz einer Liste berechnet.

Tipp: Verwenden Sie den `**2` Operator zur Quadrierung. Berechnen Sie die Varianz der unten angegebenen Testliste.

```
test_list = [7,20,5,-1,3,11,32,17,42]
```