1. Rechnerübung: Variablen, Aus- und Eingaben, Listen

Zurechtfinden

Finden Sie sich in Ihrer Entwicklungsumgebung zurecht und stellen Sie sicher, dass Sie Python Kommandos ausführen können und kleine Progämmchen erstellen und laufen lassen können.

Aufgabe 1: Ausgabe von Werten mit Python

Gehen Sie die folgenden Anweisungen durch. Bevor Sie sie eingeben und laufen lassen, sagen Sie voraus, was die Ausgabe sein wird. Freuen Sie sich, wenn Sie Recht gehabt haben, und finden Sie eine Erklärung, wenn nicht. Da die Ausdrücke so kurz sind, sollten Sie unbedingt aus lerntechnischen Gründen, die Ausdrücke selbst komplett tippen und nich mit Copy/Paste arbeiten. Das gilt auch für die allermeisten zukünftigen Aufgaben.

Die folgende Zelle enthält schon zwei Anweisungen. Lassen Sie die stehen. Und fügen Sie dann auch in diese Zelle eine Anweisung nach der anderen dazu. Wenn ein Anweisung eine Fehlermeldung erzeugen sollte, dann nehmen Sie sie wieder heraus. Am Ende der Aufgabe sollen dann alle nicht-fehlererzeugenden Anweisung untereinander stehen und ausgeführt werden könen:

a)
```
x=x

print(x)
```

b)
```
3=x

print(x)
```

c)
```
1=1

print(1)
```

d)
```
x

p
```

f)
```
y=print

print(y)
```

g)
```
print(-----5)
```

h)
```
x=x+1

print(x+1)
```

i)
```

```

k)
```
x=[3,2,1]

print(x)
```

l)
```
x[1]=x[2]

print(x)
```

m)
```
y=x[:]

print(y)
```

n)
```

```

p)
```
a=[3,6,
[9,0],4][2]
[1]

print(x[a])
```

q)
```
a=7/2-7//2

print(a,2*a)
```

r)
```
print('3'*4)

print('3'*'4')
```

s)
```
pri

pri
```

u)
```
a=False or not True
print(a)
```

v)
```
b=7==7
print(b,b<=b)
```

w)
```
print(7!=7)
print(7<>7)
```

x)
```
a=
pr
```

z)
```
L=[]
print(L)
```

ä)
```
L.append([])
print(L,len(L))
```

ö)
```
L.append([1,2])
print(L,len(L))
```

ü)
```
L=L
pri
```

In [14]:
```
x=7
y=11
# hier kommen Ihre Anweisungen, die Sie aus der Aufgabenstellung oben abt
```

Aufgabe 2: Die Fibonacci-Folge

Die Fibonacci-Folge wird meist als 1,1,2,3,5,8,13,21,34,55,... dargestellt. So dass die zehnte Zahl f_{10}=55 ist. Seltener findet man Darstellungen der Art 0,1,1,2,3,5,... oder 1,2,3,... In jedem Fall aber ist jedes Glied der Folge (außer den beiden ersten) gleich der Summe seiner beiden Vorgänger.

a) Mit 10 Variablen

Schreiben Sie ein Programm, das mit den Variablen a,b,c,..j arbeitet und am Ende in j den Wert von f_{10} berechnet und ausgibt. Nirgendwo im Programm darf eine andere Zahl als 1 stehen - sonst ist es nicht berechnet. Die ersten beiden Zeilen sind schon vorgegeben:

In [15]:
```
a=1
b=1
# hier kommen Ihre Anweisungen
```

b) Mit 3 Variablen

Erreichen Sie das gleiche Ziel (die *Berechnung* von f_{10}) unter verwendung von lediglich 3 Variablen a,b und c. Außer a und b müssen alle anderen Werte der Folge berechnet werden als Summe der beiden Vorgänger. Nirgendwo im Programm darf eine andere Zahl als 1 stehen:

In [16]:
```
a=1
b=1
c=a+b
# hier kommen Ihre Anweisungen
```

c) Mit 2 Variablen

Jetzt wird's interessant, denn jetzt sollen Sie das gleiche erreichen, dürfen aber außer a und b gar keine anderen Variablen verwenden. Trotzdem soll am Ende eine print-Anweisung stehen, die 55 ausgibt. (Und ja, alle anderen Fibonacci-Zahlen müssen unterwegs auch irgendwann berechnet worden sein.)

In [17]:
```
a=1
b=1
# hier kommen Ihre Anweisungen
```

d) Mit einer Liste

Jetzt dürfen Sie sogar nur eine Variable a verwenden. Am Ende des Prgrämmchens muss die Anweisung `print(a[-1])` stehen und die 55 ausgeben. Davor müssen Ihre Anweisungen dafür sorgen, dass eine Liste mit den ersten 10 Fibonacci-Zahlen aufgebaut wird, dass also a den Wert `[1,1,2,3,5,8,13,21,34,55]` hat bevor die print-Anweisung das letzte Element ausgibt. Und ja, auch dieses Mal dürfen Sie keine beliebigen Zahlen verwenden, nur -1 und -2.

In [18]:
```
a=[1,1]
print(a[-1])
# hier kommen Ihre Anweisungen
```

```
1
```

Aufgabe 3 (Ein-/Ausgaben)

Mit `x=input("Gib was ein: ")` kann man den Benutzer auffordern über die Tastatur eine Eingabe zu machen. Der Text `"Gib was ein: "` kann natürlich auch ein anderer sein, und statt das Eingegebene der Variablen `x` zuzuweisen, kann man natürlich auch eine andere Variable verwenden. Aber vorsicht: Das, was eingegeben wird, ist ein Text, hier probieren Sie mal:

In [19]:
```
x=input("Gib was ein:")
print(x,x*3)
```

```
Test TestTestTest
```

Wenn Sie wollen, dass das Eingegebene wie eine Zahl behandelt wird, dann muss der Text in eine Zahl gewandlet werden, zum Beispiel so:

In [21]:
```
x=int(input("Gib was ein:"))
print(x,x*3)
y=float(input("Gib noch was ein:"))
print(y,y*3)
```

```
2 6
2.5 7.5
```

Jetzt, da Sie wissen, wie man Eingaben macht, können Sie auch die folgenden Prgrämmchen basteln:

a) Namentliche Begrüßung

Der Benutzer wird aufgefordert, seinen Namen einzugeben (z.B. `Hase`). Auf dem Bildschirm erscheint dann eine Grußmeldung, die den Namen enthält (z.B. `Hallo Hase!`).

In [22]:
```
# hier kommen Ihre Anweisungen
```

b) Quadrat der eingegebenen Zahl ausgeben

Der Benutzer wird aufgefordert, eine Zahl einzugeben. Auf dem Bildschirm erscheint dann eine Meldung in der das Quadrat der Zahl ausgegeben wird.

In [23]:
```
# hier kommen Ihre Anweisungen
```

Aufgabe 4 (Textausgaben formatieren)

Oft ist es umständlich `print` Anweisungen mit nur durch Komma getrennte Stückchen zu formatieren. Python bietet daher die Möglichkeit an, einen String (= einen Text) zu formatieren, bevor er weiter verwendet wird, zum Beispiel so:

In [24]:
```
s = "Hallo {}".format("Hase")
print(s)
s = "Die Aktie von {} ist am {} um {} % {}.".format("Hasiland AG","14.02.
print(s)
```

```
Hallo Hase
Die Aktie von Hasiland AG ist am 14.02. um 10 % gestiegen.
```

a) Schreiben Sie ein Prograämmchen, das den Benutzer auffordert sein Geburtsjahr einzugeben, danach soll er seinen Geburtsmonat eingeben und dann denn Tag (also drei `input` Anweisungen). Auf dem Bildschirm soll dann so etwas erscheinen wie: `Dein Geburtstag ist der 11.03.2002`.

In [25]:
```
# hier kommen Ihre Anweisungen
```

b) Man kann Zahlen addieren und man kann Texte addieren (=verketten). Man kann aber nicht Zahlen und Texte zusammen addieren (zumindest nicht in Python). Wenn man das will, muss man erst die Zahl in einen Text umwandeln. Das geht z.B. mit `s=str(x)`. Probieren Sie es mal aus:

In [26]:
```
x=13
s="Dreizehn = "+x  # Geht schief
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call las
t)
Cell In[26], line 2
      1 x=13
----> 2 s="Dreizehn = "+x

TypeError: can only concatenate str (not "int") to str
```

In [27]:
```
x=13
s="Dreizehn = "+str(x)
print(s)
```

```
Dreizehn = 13
```

Machen Sie das Geburtstagsprogrämmchen aus der a) nochmal, aber geben in der `print`-Anweisung nur ein Argument an (also keine Kommas). Sorgen Sie halt dafür,

dass der auszugebende String schon vorher korrekt zusammengebastelt ist.

In [28]:
```
# hier kommen Ihre Anweisungen
```