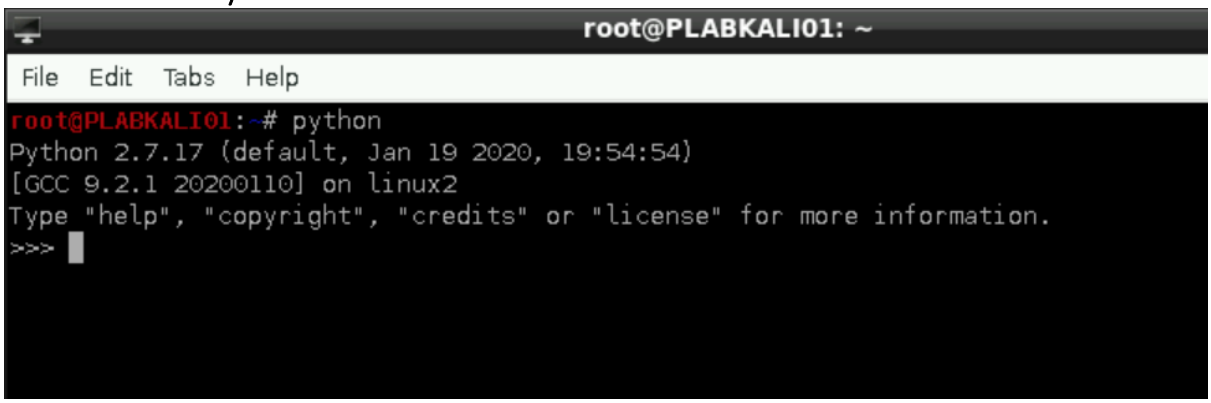
Übung – Einführung in Python

Einführung

Um die Übungen durchzufüren, starten Sie bitte ein Terminal und geben einfach nur *python*
ein (und drücken die Enter-Taste).

Das öffnet die Python-Konsole:



Python als Taschenrechner

Nutzen Sie den Interpreter wie einen Taschenrechner. Sie geben einen Ausdruck ein, und der Interpreter berechnet das Ergebnis.

Die Syntax der Ausdrücke ist genau wie bei einem Taschenrechner: die Operatoren +, -, * und / wirken genauso. Klammern können zur Gruppierung verwendet werden.

Beispiel:

```
>>> 1+1
2
>>> #Ein Kommentar
... 3+2
5
>>> 7/9
0
>>> 9/7
1
>>> #Ganzzahldivision liefert ein abgerundetes Ergebnis!
... #Klammern werden zum Gruppieren verwendet:
... (50-10*8)/4
-8
>>>
```

Das Gleichheitszeichen kann benutzt werden um einer Variablen einen Wert zuzuweisen. Danach wird nicht direkt ein Ergebnis an der interaktiven Eingabeaufforderung gezeigt:

```
>>> breite=20
>>> hoehe=50
>>> breite*hoehe
1000
>>>
```

Man muss Variablen deklarieren/definieren, bevor sie benutzt werden können. Sonst kommt es zu einer Fehlermeldung, wie hier:

```
>>> #Deklarieren einer Variablen:
... zahl=5
>>> zahl*2
10
>>> #Versuch des Aufrufens einer undeklarierten / undefinierten Variablen:
... neuezahl
Traceback (most recent call last):
  File "<stdin>", line 2, in <module>
NameError: name 'neuezahl' is not defined
>>>
```

Fließkommazahlen werden voll unterstützt:

```
>>> #Fliesskommazahlen:
... 3.5*2.8
9.799999999999999
>>> 3.0/2
1.5
>>>
```

Zahlen werden entweder implizit konvertiert - also automatisch:

```
>>> #Implizite Typkonvertierung:
... 7.0/2
3.5
>>>
```

hier wurde die 2 automatisch in 2.0 konvertiert, für die Berechnung von 7.0 / 2.0

Oder explizit, indem man den Datentyp angibt und die zu konvertierende Variable in Klammer einträgt, wie hier die Variable „a“:

```
>>> #Explizite Typkonvertierung:
... a = 2
>>> 7.0/float(a)
3.5
>>>
```

Der zuletzt ausgegebene Ausdruck im interaktiven Modus der Python-Konsole wird immer der Variablen _ zugewiesen. Das ist hilfreich wenn man Python als Taschenrechner benutzen möchte:

```
>>> steuern = 12.5/100
>>> preis = 20.9
>>> preis * steuern
2.6125
>>> preis + _
23.5125
>>> round(_, 2)
23.51
>>>
```

Statt Zahlen kann man auch Zeichenketten / Texte darstellen („Strings“):

```
>>> neuertext = "Dies ist eine Zeichenkette!"
>>> langertext = "Ein Text kann auch mehrzeilig sein,\n\
... indem man das n-Escapezeichen wie hier am Ende der Zeile dargestellt\n\
... verwendet."
>>> #Mit print() gibt man den Text (und anderes) wieder aus:
... print(neuertext)
Dies ist eine Zeichenkette!
>>> print(langertext)
Ein Text kann auch mehrzeilig sein,
indem man das n-Escapezeichen wie hier am Ende der Zeile dargestellt
verwendet.
>>>
```

Mit dem + Operator kann man Zeichenketten miteinander verbinden:

```
>>> #Verbinde die beiden Zeichenketten miteinander, und gebe sie aus:
... text = neuertext + langertext
>>> print(text)
Dies ist eine Zeichenkette!Ein Text kann auch mehrzeilig sein,
indem man das n-Escapezeichen wie hier am Ende der Zeile dargestellt
verwendet.
>>>
```

Und man kann mit der Funktion input() User-Eingaben…

```
>>> #Eine User-Eingabe:
... usertext = input('Geben Sie einen Text ein:')
Geben Sie einen Text ein:
```

…in die Zeichenketten einbetten, indem man sie im folgenden Format mit dem %-Zeichen in den print()- Befehl integriert:

```
>>> #Eine User-Eingabe:
... usertext = input('Geben Sie einen Text ein:')
Geben Sie einen Text ein:"Mein Text"
>>> print("Der eingegebene Text war:  %s." % usertext)
Der eingegebene Text war:  Mein Text.
>>>
```

Anmerkung: es gibt noch viel mehr weitere Möglichkeiten um Texte zu kombinieren oder zu formatieren.