Python

Dateiverarbeitung

Lesen	`with open("file.txt", "r") as datei:` `  text = datei.read()`
Schreiben	`with open("file.txt", "w") as datei:` `  datei.write(text + "Hallo\n")`
Leerzeichen entfernen	`text.strip("\n")`

Nutzerinteraktion

input()
input("str")
	Unterbricht den Programmablauf und wartet auf Eingabe des Users. Programm wird danach fortgesetzt. Der Funktion kann ein String übergeben werde, der auf dem Bildschirm angezeigt wird.
	Die Eingabe des Users ist der Rückgabewert dieser Funktion. Rückgabewert ist immer String.
		`# Eingabeaufforderung`
		`# Eingabe des Users wird in abfrage geschrieben`
		`abfrage = input("Hier könnte Ihre Werbung stehen: ")`
		
		`# Bildschirmausgabe`
		`Hier könnte Ihre Werbung stehen:`
		`# Usereingabe`
		`Nein`
		
		`print(abfrage)`
		`# Ausgabe`
		`Nein`

Verzweigungen

```
if bmi < 19:
  print("Untergewicht")
elif bmi <= 24:
  print("Normalgewicht")
else:
  print("Uebergewicht")
```

Schleifen

while	`i = 0`
	`while i < 10:`
	`  print(i)`
	`  i += 1`

for (eigentlich foreach)	`for zahl in [1, 2, 3]:` `  print(zahl)`

Operatoren

in	Prüft ob Variable in angegebener Sequenz vorhanden ist
	vorhanden gibt "True" zurück
	nicht vorhanden gibt "False" zurück
		`#true`
		`print(2 in [2, 3, 4])`
		
		`#false`
		`erg = "bla" in ["bli", "blub"]`
		`print(erg)`

range	Funktion gibt eine Sequenz von Zahlen zurück
	`range(start, stop, step)`
	
	start = inklusiver Anfang der Sequenz, optional, default ist 0
	stop = exklusves Ende der Sequenz, benötigt
	step = Inkrement, optional, default 1
		`# 0, 1, 2, 3, 4`
		`range(5)`
		
		`# 4, 5, 6, 7`
		`range(4, 8)`
		
		`# 2, 4, 6, 8`
		`range(2, 9, 2)`

Listen

append()	hängt Objekt an das Ende der Liste	`# hängt den Wert 5 an Liste an` `liste.append(5)`

index()	gibt den ersten Index zu angegebenem Wert zurück	`liste = [2, 3, 4, 5, 6, 3, 4, 5]`
		`print(liste.index(4))`
		`# Ausgabe ist 2`

insert()	fügt Wert vor Index ein `liste.insert(index, value)`	`liste = [2, 3, 4, 5]`
		`liste.insert(1, 6)`
		`print(liste)`
		`# Ausgabe [2, 6, 3, 4, 5]`

remove()	entfernt das erste Vorkommen des Werts	`liste = [2, 3, 4, 5]`
		`liste.remove(4)`
		`print(liste)`
		`# Ausgabe [2, 3, 5]`

reverse()	Umdrehen der Liste in-place	`liste = [2, 3, 4, 5]`
		`liste.reverse()`
		`print(liste)`
		`# Ausgabe [5, 4, 3, 2]`

sort()	Sortiert die Liste in-place
	default = aufsteigend
	"reverse=True" = absteigend
		`# Sortieren in aufsteigender Reihenfolge`
		`liste = [3, 2, 5, 4]`
		`liste.sort()`
		`print(liste)`
		`# Ausgabe [2, 3, 4, 5]`
		
		`# Sortieren in absteigender Reihenfolge`
		`liste = [3, 2, 5, 4]`
		`liste.sort(reverse=True)`
		`print(liste)`
		`# Ausgabe [5, 4, 3, 2]`

Ausgabe Formatierung	Die Ausgabe von Listen kann nach folgender Notation formatiert werde `listenname[start:stop:step]`
	
	start = inklusiver Startwert, default 0
	stop = exklusiver Stopwert, default Länge
	step = Inkrement, default 1
		`liste = [2, 3, 4, 5, 6, 7, 8, 9]`
		
		`print(liste[0:4])`
		`# Ausgabe [2, 3, 4]`
		
		`print(liste[1:6:2])`
		`# Ausgabe [3, 5, 7]`
		
		`print(liste[2:6:2])`
		`# Ausgabe [4, 6]`

Umwandlung von Datentypen

> Wichtig: Alle Funktionen sind **KEINE** in-place Ersetzung

type()	Gibt den Datentyp einer Variablen zurück	`type(5)`
		`# Ausgabe <class 'int'>`
		
		`type(5.0)`
		`# Ausgabe <class 'float'>`
		
		`type(5)`
		`# Ausgabe <class 'str'>`

str()	Rückgabewert dieser Funktion ist ein String	`str(5)` `# Ausgabe '5'`
int()	Rückgabewert dieser Funktion ist ein Integer	`int("5")` `# Ausgabe 5`
float()	Rückgabewert dieser Funktion ist eine Fließkommazahl	`float(5)` `# Ausgabe 5.0`

Created 2022-08-01 08:56:26 UTC
Updated 2022-08-02 08:41:07 UTC