## Hausaufgabe: Fragen zum PCEP-30-02

Gesamtpunktzahl 305/310

Fülle das Quiz aus und gebe bitte deine Techstarter-Email an. Falls du eine Antwort nicht weißt, recherchiere die Begriffe oder verwende die Unterlagen, um dein Wissen zu vertiefen. Viel Erfolg! :)

Beim Absenden dieses Formulars wurde die E-Mail-Adresse der befragten Person (alexander.manhardt@tn.techstarter.de) gespeichert.

✓ Welcher der folgenden Begriffe beschreibt Python am besten? *	5/5
Compilierte Sprache	
Interpretierte Sprache	<b>✓</b>
Maschinensprache	
Assembly-Sprache	
Feedback	
Korrekt! Python ist eine interpretierte Hochsprache. Das bedeutet, dass Python-Code Laufzeit von einem Interpreter ausgeführt wird, ohne dass eine vorherige Kompilierum Maschinencode erforderlich ist. Diese Eigenschaft macht Python plattformunabhäng und trägt zur einfachen Lesbarkeit und Wartbarkeit des Codes bei.	ng in

<b>✓</b>	Was ist der Hauptunterschied zwischen kompilierten und interpretierten *5/5 Sprachen?
0	Kompilierte Sprachen sind immer schneller
0	Interpretierte Sprachen werden direkt in Maschinencode übersetzt
	Kompilierte Sprachen werden vor der Ausführung in Maschinencode übersetzt 🗸
0	Interpretierte Sprachen können nur auf bestimmten Betriebssystemen ausgeführt werden
Ki ge in au	eedback  orrekt! Der grundlegende Unterschied besteht darin, dass bei kompilierten Sprachen der esamte Quellcode vor der Ausführung in Maschinencode übersetzt wird. Bei terpretierten Sprachen wie Python wird der Code Zeile für Zeile während der Laufzeit usgeführt. Kompilierte Programme laufen oft schneller, während interpretierte rogramme flexibler und portabler sind.
<b>✓</b>	Welche Behauptung zur Python-Laufzeitumgebung ist KORREKT? * 5/5
0	Python-Programme benötigen immer eine grafische Benutzeroberfläche
•	Python-Programme können nur auf Systemen ausgeführt werden, auf denen Python installiert ist
0	Python-Programme können nicht in einer interaktiven Umgebung ausgeführt werden
0	Python-Programme werden immer in eine .exe-Datei kompiliert
K	eedback orrekt! Python-Programme benötigen eine Python-Laufzeitumgebung, um ausgeführt zu erden. Dies bedeutet, dass auf dem System, auf dem ein Python-Programm laufen soll,

✓ Was ist ein korrekter Weg einen Kommentar in Python zu schreiben? *	5/5
// Dies ist ein Kommentar	
/* Dies ist ein Kommentar */	
# Dies ist ein Kommentar	<b>✓</b>
Dies ist ein Kommentar	
Feedback  Korrekt! In Python werden Kommentare mit dem Hash-Zeichen (#) eingeleitet. Alles, was nach diesem Zeichen in einer Zeile steht, wird vom Interpreter ignoriert. Für mehrzeilige Kommentare werden entweder mehrere Zeilen mit # oder ein mehrzeiliger String (" oder \\\) verwendet	
✓ Welche der folgenden Variablenbezeichnungen ist in Python NICHT zulässig?	*5/5
my_var	
_var	
2var	<b>✓</b>
var_2	
Feedback	,,
Korrekt! In Python dürfen Variablennamen nicht mit einer Ziffer beginnen. Daher ist '2vai ungültig. Gültige Variablennamen müssen mit einem Buchstaben oder Unterstrich beginnen und können dann Buchstaben, Ziffern oder Unterstriche enthalten.	



✓ Was ist der Wert von x nach Ausführung des folgenden Codes? x = 10 //	/ 3 * 5/5
3	<b>✓</b>
3.33	
3.0	
O 2	
Feedback	
Korrekt! Der Operator '//' ist der Ganzzahldivisions-Operator in Python. Er teilt die linke durch die rechte und rundet das Ergebnis nach unten auf die nächste ganze Zahl ab. Hist 10 // 3 = 3.	
✓ Welcher der folgenden Ausdrücke entspricht NICHT dem booleschen Wert True?	*5/5
obool(1)	
obool('False')	
<pre>bool([])</pre>	<b>✓</b>
O bool(0.1)	
Feedback	
Korrekt! In Python werden leere Sequenzen wie die leere Liste '[]' zu False ausgewertet anderen Optionen ergeben alle True: bool(1) für Zahlen ungleich 0; bool('False') für nic leere Strings; bool(0.1) für Fließkommazahlen ungleich 0.	



✓ Was ist die Ausgabe des folgenden Codes? print(type(2**3)) *	5/5
<pre><class 'int'=""></class></pre>	<b>✓</b>
<pre><class 'float'=""></class></pre>	
<pre><class 'str'=""></class></pre>	
<pre><class 'bool'=""></class></pre>	
Feedback  Korrekt! Der Ausdruck 2**3 berechnet 2 hoch 3 (8). Da beide Operanden Integer sind, ist das Ergebnis auch ein Integer. Die Funktion type() gibt den Datentyp zurück.	
✓ Was ist der Wert von z nach Ausführung des folgenden Codes? x = 5; y = 2; z = x % y	*5/5
1	<b>✓</b>
$\bigcirc$ 2	

Feedback

Korrekt! Der Operator '%' ist der Modulo-Operator in Python und gibt den Rest der Division zurück. 5 % 2 = 1, da 5 geteilt durch 2 gleich 2 mit einem Rest von 1 ist.

✓ V	Velcher der folgenden Ausdrücke führt zu einem TypeError? *	5/5
	2' + 2	<b>✓</b>
	2' * 2	
O i	nt('2') + 2	
O 9	str(2) + '2'	
Fee	dback	
2 w	rekt! In Python kann man Strings und Integers nicht direkt addieren. Der Ausdruck '2 ürde einen TypeError erzeugen. Die anderen Ausdrücke sind gültig: '2' * 2 = '22'; int(' 4; str(2) + '2' = '22'.	
✓ V	Vas ist die Ausgabe des folgenden Codes? print(10 / 5) *	5/5
$\bigcirc$ 2	2	

✓ Was ist die Ausgabe des folgenden Codes? print(10 / 5) *	5/5
O 2	
2.0	<b>✓</b>
<u> </u>	
5.0	
Feedback	
Korrekt! Der Operator '/' führt in Python 3 immer eine Fließkommadivision durch. Daher ergibt $10/5 = 2.0$ und nicht 2. Dies ist eine Änderung gegenüber Python 2.	

:30	Hausaufgabe: Fragen zum PCEP-30-02	
<b>✓</b>	Welche der folgenden Optionen erzeugt einen Syntaxfehler in Python? *	5/5
0	x = 5	
•	5 = x	<b>✓</b>
0	x y = 10 20	
0	x + 5 = y	
F	eedback	
V	orrekt! In Python muss auf der linken Seite einer Zuweisungsoperation ein gültiger ariablenname stehen. '5 = x' ist ungültig, da eine Zahl kein gültiges Ziel für eine uweisung ist.	
	Was ist der Wert von v nach dem folgenden Code? v = 'aho': v = 3 * v *	5/5

5/5
<b>✓</b>
t

✓ Welcher der folgenden Ausdrücke wandelt den String '3.14' in eine Fließkommazahl um?	<b>*</b> 5/5
int('3.14')	
float('3.14')	<b>✓</b>
str('3.14')	
3.14' + 0.0	
Feedback  Korrekt! Die Funktion float() konvertiert einen String in eine Fließkommazahl. int('3.14' würde einen ValueError erzeugen; str('3.14') gibt den String zurück; '3.14' + 0.0 würde TypeError erzeugen.	
✓ Was ist der Wert von x nach folgendem Code? x = True + True + False *	5/5
<ul><li>0</li><li>1</li><li>2</li></ul>	<b>✓</b>
O 3	
Feedback  Korrekt! In Python können boolesche Werte in numerischen Operationen als 1 (True)  (False) verwendet werden. Daher ist True + True + False = 1 + 1 + 0 = 2.	und 0

✓ Welcher der folgenden Datentypen ist in Python unveränderlich (immutable)?	<b>*</b> 5/5
Liste	
Dictionary	
Tuple	<b>✓</b>
Set	
Feedback  Korrekt! In Python sind Strings, Zahlen und Tupel unveränderlich (immutable). Lister Dictionaries und Sets sind veränderlich (mutable). Nach der Erstellung kann der Inheeines Tupels nicht mehr geändert werden.	
✓ Welcher Ausdruck ist gleichwertig zu not (x > 5 and x < 10)? *	5/5
x <= 5  and  x >= 10	
x <= 5  or  x >= 10	<b>✓</b>
onot x > 5 or not x < 10	
onot x > 5 and not x < 10	
Feedback	
Korrekt! Nach den De Morgan'schen Regeln ist not (A and B) äquivalent zu (not A) o. B). Also ist not ( $x > 5$ and $x < 10$ ) äquivalent zu not ( $x > 5$ ) or not ( $x < 10$ ), was wieden $x < 5 > 5$ or $x > 10$ entspricht.	



<b>✓</b>	Was ist die Ausgabe des folgenden Codes? for i in range(3): print(i) *	5/5
•	012	<b>✓</b>
0	123	
0	0123	
0	01	
K fo	eedback forrekt! Die range(3)-Funktion erzeugt eine Sequenz von 0 bis 2 (3 ausgeschlossen). Die or-Schleife iteriert über jedes Element und gibt es aus. Daher werden nacheinander die Verte 0, 1 und 2 ausgegeben.	
<b>~</b>	Was bewirkt die break-Anweisung in einer Schleife? *	5/5
0	Sie beendet nur die aktuelle Iteration und fährt mit der nächsten fort	
•	Sie beendet die Schleife vollständig und fährt mit dem Code nach der Schleife fort	<b>✓</b>
0	Sie überspringt den restlichen Code in der Schleife und beginnt die nächste Iteration	
0	Sie beendet das gesamte Programm	
K S	eedback Torrekt! Die break-Anweisung beendet die Schleife vollständig, unabhängig davon, ob die Tichleifenbedingung noch wahr ist oder nicht. Nach dem break wird die Ausführung mit Tem Code fortgesetzt, der direkt nach der Schleife steht.	



<b>✓</b>	Was bewirkt die continue-Anweisung in einer Schleife? *	5/5
0	Sie beendet die Schleife vollständig	
0	Sie pausiert die Schleife für eine bestimmte Zeit	
•	Sie überspringt den restlichen Code in der aktuellen Iteration und beginnt die nächste Iteration	<b>✓</b>
0	Sie überspringt die nächste Iteration und fährt mit der übernächsten fort	
Ko ur	eedback orrekt! Die continue-Anweisung überspringt den restlichen Code in der aktuellen Iteratio nd springt zum Anfang der Schleife für die nächste Iteration. Im Gegensatz zu break eendet continue die Schleife nicht vollständig.	on
<b>✓</b>	Wofür wird die pass-Anweisung in Python verwendet? *	5/5
✓ ○	Wofür wird die pass-Anweisung in Python verwendet? *  Um eine Funktion zu überspringen	5/5
✓ ○ ●		5/5
<ul><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li><!--</td--><td>Um eine Funktion zu überspringen</td><td>5/5</td></li></ul>	Um eine Funktion zu überspringen	5/5
<ul><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li><!--</td--><td>Um eine Funktion zu überspringen Als Platzhalter für Code, der später implementiert werden soll</td><td>5/5</td></li></ul>	Um eine Funktion zu überspringen Als Platzhalter für Code, der später implementiert werden soll	5/5



Welcher der folgenden Codeblöcke ist ein Beispiel für verschachtelte Kontrollstrukturen?	*5/5
for i in range(5): print(i)	
if x > 0: print('Positiv'); else: print('Nicht positiv')	
for i in range(3): if i > 0: print(i)	<b>✓</b>
while True: break	
Feedback  Korrekt! Verschachtelte Kontrollstrukturen bedeuten, dass eine Kontrollstruktur (wie if, forwhile) innerhalb einer anderen platziert ist. In Option C ist eine if-Anweisung innerhalb einer for-Schleife verschachtelt.	or,
✓ Was ist der Ternäre Operator in Python? *	5/5
✓ Was ist der Ternäre Operator in Python? * if-else	5/5
	5/5
if-else	5/5
if-else	5/5
if-else for-in ?:	5/5
<ul> <li>if-else</li> <li>for-in</li> <li>?:</li> <li>x if y else z</li> </ul> Feedback	5/5
<ul> <li>if-else</li> <li>for-in</li> <li>?:</li> <li>x if y else z</li> </ul>	<b>✓</b>



✓ Welche der folgenden Optionen ist KEINE gültige Schleife in Python? *	5/5
of for i in range(5):	
while i < 10:	
o do: while i < 5:	<b>✓</b>
of for i in [1, 2, 3]:	
Feedback  Korrekt! In Python gibt es keine do-while-Schleife. Die gültigen Schleifentypen sind for Schleifen (für Iterationen über Sequenzen) und while-Schleifen (für Iterationen, solang eine Bedingung wahr ist).	
✓ Was ist die Ausgabe des folgenden Codes? i = 0; while i < 5: if i == 3: break; print(i); i += 1	*5/5
01234	
0 1 2	<b>✓</b>
O 1 2 3	
012345	
Feedback	
Korrekt! Die Schleife beginnt mit i = 0 und prüft, ob i < 5 (wahr). In der Schleife wird ge ob i == 3 (falsch für i = 0), also wird 0 ausgegeben und i auf 1 erhöht. Der Prozess wiederholt sich für i = 1 und i = 2. Bei i = 3 ist die Bedingung i == 3 wahr, also wird die break-Anweisung ausgeführt und die Schleife beendet. Daher wird nur '0 1 2' ausgege	



✓ Welche der folgenden Datensammlungen in Python ist geordnet und *5/5 veränderbar (mutable)?
Tupel (tuple)
Dictionary (dict)
Set (set)
Liste (list)
Feedback  Korrekt! Listen in Python sind geordnete und veränderbare (mutable) Sammlungen. Das bedeutet, dass die Reihenfolge der Elemente beibehalten wird und Elemente nach der Erstellung hinzugefügt, geändert oder entfernt werden können. Tupel sind geordnet, aber unveränderlich (immutable); Dictionaries sind veränderbar, aber ungeordnet (in Python 3.7+ wird die Einfügereihenfolge beibehalten); Sets sind veränderbar, aber ungeordnet und erlauben keine Duplikate.
✓ Welche der folgenden Aussagen über Tupel ist FALSCH? * 5/5
Tupel werden mit runden Klammern () definiert
Tupel sind unveränderlich (immutable)
Tupel können nicht sortiert werden
<ul> <li>Tupel-Elemente können nach der Erstellung geändert werden</li> </ul>
Feedback  Korrekt! Tupel sind unveränderlich (immutable), was bedeutet, dass ihre Elemente nach



Welche Methode wird verwendet, um ein Element aus einer Liste zu entfernen, wenn der Index bekannt ist?	*5/5
remove()	
delete()	
pop()	<b>✓</b>
discard()	
orrekt! Die pop()-Methode entfernt ein Element an einem bestimmten Index aus einer ste und gibt es zurück. Wenn kein Index angegeben wird, wird das letzte Element atfernt. Die remove()-Methode entfernt das erste Vorkommen eines bestimmten Werts.	
Was ist die Ausgabe des folgenden Codes? my_set = {1, 2, 3, 2, 1}; print(my_set)	*5/5
{1, 2, 3, 2, 1}	
{1, 2, 3}	<b>✓</b>
[1, 2, 3]	
Error	
orrekt! Sets in Python sind Sammlungen von eindeutigen Elementen. Beim Erstellen eine ets werden Duplikate automatisch entfernt. Daher enthält my_set nur die eindeutigen	es
	remove()  delete()  pop()  discard()  eedback  orrekt! Die pop()-Methode entfernt ein Element an einem bestimmten Index aus einer iste und gibt es zurück. Wenn kein Index angegeben wird, wird das letzte Element internt. Die remove()-Methode entfernt das erste Vorkommen eines bestimmten Werts. elete() ist keine Listenmethode, und discard() ist eine Methode für Sets, nicht für Listen.  Was ist die Ausgabe des folgenden Codes? my_set = {1, 2, 3, 2, 1};  print(my_set)  {1, 2, 3, 2, 1}  {1, 2, 3}



•	Welche der folgenden Optionen ist ein gültiger Weg, um über ein Dictionary zu iterieren und sowohl die Schlüssel als auch die Werte zu erhalten?	*5/5
0	for key in my_dict: print(key, my_dict[key])	
0	for key, value in my_dict: print(key, value)	
	for key, value in my_dict.items(): print(key, value)	<b>✓</b>
0	for item in my_dict: print(item[0], item[1])	
K al	eedback forrekt! Die items()-Methode eines Dictionaries gibt eine Ansicht der Schlüssel-Wert-Paa Is Tupel zurück. Mit der for key, value in my_dict.items()-Syntax können wir direkt über iese Paare iterieren und auf die Schlüssel und Werte zugreifen.	are
<b>✓</b>	Was ist die Ausgabe des folgenden Codes? my_tuple = (1, 2, 3); my_tuple[0] = 0; print(my_tuple)	*5/5
		*5/5
<ul><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li><!--</th--><th>my_tuple[0] = 0; print(my_tuple)</th><th>*5/5</th></li></ul>	my_tuple[0] = 0; print(my_tuple)	*5/5
<ul><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li><!--</th--><th>my_tuple[0] = 0; print(my_tuple) (0, 2, 3)</th><th>*5/5</th></li></ul>	my_tuple[0] = 0; print(my_tuple) (0, 2, 3)	*5/5
<ul><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li><!--</th--><th>my_tuple[0] = 0; print(my_tuple)  (0, 2, 3)  (1, 2, 3)</th><th>*5/5</th></li></ul>	my_tuple[0] = 0; print(my_tuple)  (0, 2, 3)  (1, 2, 3)	*5/5



✓ Welche Methode wird verwendet, um alle Elemente aus einer Liste zu entfernen?	*5/5
clear()	<b>✓</b>
remove_all()	
O delete()	
empty()	
Feedback  Korrekt! Die clear()-Methode entfernt alle Elemente aus einer Liste. Nach dem Aufruf vo my_list.clear() wird die Liste leer sein: [].	n
Was ist die Ausgabe des folgenden Codes? a = {1, 2, 3}; b = {3, 4, 5}; print(a   b)	*0/5
[1, 2, 3, 4, 5]	×
<b>(</b> 1, 2, 3, 4, 5)	
<b>(</b> 1, 2, 4, 5)	
Richtige Antwort	
<b>(</b> 1, 2, 3, 4, 5)	
Feedback  Falsch. Der Operator   führt eine Vereinigung (union) von zwei Sets durch. Das Ergebnis ein neues Set, das alle eindeutigen Elemente aus beiden Sets enthält. In diesem Fall sin die eindeutigen Elemente aus a und b: 1, 2, 3, 4 und 5.	



<b>✓</b>	Welche der folgenden Methoden kann NICHT auf ein Tupel angewendet werden?	<b>*</b> 5/5
0	count()	
0	index()	
	append()	<b>✓</b>
0	len()	
Ko ar be	eedback orrekt! Die append()-Methode ist eine Listenmethode und kann nicht auf Tupel ngewendet werden, da Tupel unveränderlich sind. Die count()-Methode zählt, wie oft ein estimmter Wert im Tupel vorkommt, index() gibt den Index des ersten Vorkommens ein ferts zurück, und len() ist eine eingebaute Funktion, die die Länge des Tupels zurückgib	es
<b>✓</b>	Was ist die Ausgabe des folgenden Codes? my_list = [1, 2, 3]; print(my_list * 2)	*5/5
✓ •		*5/5
•	print(my_list * 2)	*5/5
<ul><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li><!--</td--><td>print(my_list * 2) [1, 2, 3, 1, 2, 3]</td><td>*5/5</td></li></ul>	print(my_list * 2) [1, 2, 3, 1, 2, 3]	*5/5
•	print(my_list * 2) [1, 2, 3, 1, 2, 3] [2, 4, 6]	*5/5
	print(my_list * 2)  [1, 2, 3, 1, 2, 3]  [2, 4, 6]  [1, 1, 2, 2, 3, 3]	*5/5
<ul><li>✓</li><li>✓</li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li><td>print(my_list * 2)  [1, 2, 3, 1, 2, 3]  [2, 4, 6]  [1, 1, 2, 2, 3, 3]</td><td>*5/5</td></li></ul>	print(my_list * 2)  [1, 2, 3, 1, 2, 3]  [2, 4, 6]  [1, 1, 2, 2, 3, 3]	*5/5
Ko er	print(my_list * 2)  [1, 2, 3, 1, 2, 3]  [2, 4, 6]  [1, 1, 2, 2, 3, 3]  Error	<b>~</b>



✓ Welche der folgenden Optionen ist KEINE gültige Methode, um ein neu Element zu einem Dictionary hinzuzufügen?	es *5/5
my_dict['new_key'] = 'new_value'	
my_dict.add('new_key', 'new_value')	<b>✓</b>
<pre>my_dict.update({'new_key': 'new_value'})</pre>	
my_dict.setdefault('new_key', 'new_value')	
Feedback  Korrekt! Dictionaries haben keine add()-Methode. Um ein neues Element zu einem Dictionary hinzuzufügen, kann man direkt zuweisen (my_dict['new_key'] = 'new_value update()-Methode verwenden oder setdefault(), das einen Wert nur setzt, wenn der Schlüssel noch nicht existiert.	'), die
Was ist die Ausgabe des folgenden Codes? my_set = {1, 2, 3}; my_set.add(2); print(my_set)	*5/5
[1, 2, 3, 2]	
<b>(</b> 1, 2, 3, 2 <b>)</b>	
<b>(</b> 1, 2, 3)	<b>✓</b>
○ Error	
Feedback  Korrekt! Sets enthalten nur eindeutige Elemente. Wenn man versucht, ein Element hinzuzufügen, das bereits im Set vorhanden ist (wie 2 in diesem Fall), bleibt das Set unverändert. Daher bleibt my_set {1, 2, 3}.	



✓ Welches der folgenden Elemente kann NICHT als Schlüssel in einem Dictionary verwendet werden?	<b>*</b> 5/5
Integer	
String	
Liste	<b>~</b>
○ Tupel	
Feedback  Korrekt! In Python können nur unveränderliche (immutable) Typen als Dictionary-Scoverwendet werden. Listen sind veränderlich (mutable) und können daher nicht als Schlüssel verwendet werden. Integer, Strings und Tupel (solange sie nur unverände Elemente enthalten) können als Schlüssel verwendet werden.	
Was ist die Ausgabe des folgenden Codes? my_dict = {'a': 1, 'b': 2}; pri in my_dict)	nt('c' *5/5
True	
False	<b>✓</b>
None	
○ KeyError	
Feedback  Korrekt! Der Ausdruck 'c' in my_dict prüft, ob der Schlüssel 'c' im Dictionary my_dict vorhanden ist. Da 'c' nicht in my_dict enthalten ist, ist das Ergebnis False.	



✓ Welche der folgenden Aussagen über Sets ist FALSCH? *	5/5
Sets können nur eindeutige Elemente enthalten	
Sets sind ungeordnet	
Sets können mit geschweiften Klammern {} erstellt werden	
Sets können verschachtelte Sets enthalten	<b>✓</b>
Feedback  Korrekt! Sets können keine veränderlichen (mutable) Objekte enthalten, daher könne keine verschachtelten Sets enthalten, da Sets selbst veränderlich sind. Die anderen Aussagen sind korrekt: Sets enthalten nur eindeutige Elemente, sind ungeordnet unwerden mit geschweiften Klammern erstellt.	
✓ Was ist die Ausgabe des folgenden Codes? my_dict = {'a': 1, 'b': 2}; print(list(my_dict.keys()))	*5/5
	*5/5
print(list(my_dict.keys()))	*5/5
print(list(my_dict.keys())) <pre>     ['a', 'b']</pre>	*5/5
print(list(my_dict.keys()))  ( ['a', 'b'] ( [1, 2]	*5/5
print(list(my_dict.keys()))  ['a', 'b']  [1, 2]  [('a', 1), ('b', 2)]  {'a': 1, 'b': 2}	*5/5
print(list(my_dict.keys())) <ul> <li>['a', 'b']</li> <li>[1, 2]</li> <li>[('a', 1), ('b', 2)]</li> </ul>	

	Welcher der folgenden Datentypen unterstützt Indizierung mit ganzen Zahlen?	*5/5
0	Dictionary	
0	Set	
	Liste	<b>✓</b>
0	Alle der oben genannten	
Ki si no	eedback  orrekt! Listen und Tupel unterstützen Indizierung mit ganzen Zahlen (z.B. my_list[0]), da e geordnete Sammlungen sind. Dictionaries werden mit Schlüsseln indiziert (nicht otwendigerweise ganze Zahlen), und Sets unterstützen überhaupt keine Indizierung, da e ungeordnet sind.	
<b>/</b>	Was ist die Ausgabe des folgenden Codes? a = {1, 2, 3}; b = {3, 4, 5}; print(a - b)	<b>*</b> 5/5
0		
	[1, 2]	
•	[1, 2] {1, 2}	<b>✓</b>
•		<b>✓</b>
<ul><li> </li><li> <td>{1, 2}</td><td><b>✓</b></td></li></ul>	{1, 2}	<b>✓</b>

<b>✓</b>	Welche der folgenden Methoden wird verwendet, um ein bestimmtes Element aus einem Set zu entfernen und einen Fehler zu werfen, wenn das Element nicht vorhanden ist?	*5/5
	remove()	<b>✓</b>
0	discard()	
0	pop()	
0	delete()	
Ko ei er Di	perekt! Die remove()-Methode entfernt ein bestimmtes Element aus einem Set und wirft nen KeyError, wenn das Element nicht im Set vorhanden ist. Die discard()-Methode atfernt ein Element, wenn es vorhanden ist, und tut nichts, wenn es nicht vorhanden ist. de pop()-Methode entfernt und gibt ein beliebiges Element zurück. delete() ist keine Set- ethode.	
<b>✓</b>	Was ist die Ausgabe des folgenden Codes? a = [1, 2, 3]; b = a; b.append(4); print(a)	*5/5
0	[1, 2, 3]	
•	[1, 2, 3, 4]	<b>✓</b>
0	[1, 2, 3, [4]]	
0	Error	
Ko ve zu	eedback orrekt! In Python werden Listen als Referenzen übergeben. Wenn wir b = a zuweisen, erweisen sowohl a als auch b auf dieselbe Liste. Wenn wir b.append(4) aufrufen, wird 4 or Liste hinzugefügt, auf die beide Variablen verweisen. Daher enthält a auch den neuen fert und die Ausgabe ist [1, 2, 3, 4].	

<b>✓</b>	Welcher der folgenden Ausdrücke prüft korrekt, ob ein Schlüssel in einem Dictionary vorhanden ist?	<b>*</b> 5/5
	key in my_dict	<b>✓</b>
0	key in my_dict.keys()	
0	my_dict.has_key(key)	
0	my_dict.contains(key)	
K) D	eedback orrekt! Der Ausdruck key in my_dict prüft, ob key ein Schlüssel im Dictionary my_dict ist. ie zweite Option funktioniert auch, ist aber weniger effizient. Die has_key()-Methode urde in Python 3 entfernt, und contains() ist keine Dictionary-Methode.	
<b>✓</b>	Was ist der Zweck der Einrückung in Python? *	5/5
0	Lediglich zur besseren Lesbarkeit des Codes	
	Um Codeblöcke zu definieren und die Struktur des Programms festzulegen	<b>✓</b>
0	Um Fehler bei der Ausführung zu vermeiden	
0	Um Kommentare vom Code zu unterscheiden	
Ki al Si	eedback orrekt! In Python wird die Einrückung verwendet, um Codeblöcke zu definieren. Anders is in vielen anderen Programmiersprachen, die geschweifte Klammern oder ochlüsselwörter verwenden, bestimmt in Python die Einrückung, welcher Code zu einem estimmten Block (wie einer if-Anweisung oder Schleife) gehört.	



✓ Was beschreibt den Begriff 'Laufzeitfehler' (runtime error) am besten? * 5/5
Ein Fehler der während der Kompilierung auftritt
Ein Fehler in der Syntax des Programms
■ Ein Fehler der während der Ausführung des Programms auftritt
Ein logischer Fehler im Programmdesign
Feedback  Korrekt! Ein Laufzeitfehler (runtime error) tritt während der Ausführung eines Programms auf, nachdem die Syntax bereits überprüft wurde. Beispiele sind Division durch Null, Zugriff auf nicht existierende Dateien oder Verwendung undefinierter Variablen. Im Gegensatz dazu treten Syntaxfehler während der Parsing-Phase auf, bevor das Programm ausgeführt wird.
✓ Welches der folgenden Schlüsselwörter ist in Python reserviert und kann *5/5 nicht als Variablenname verwendet werden?
O variable
count
name
Feedback  Korrekt! In Python gibt es eine Reihe von reservierten Schlüsselwörtern, die für spezielle Zwecke reserviert sind und nicht als Variablennamen verwendet werden können. 'class' ist ein solches Schlüsselwort, das zur Definition von Klassen verwendet wird. 'variable', 'count' und 'name' sind keine reservierten Schlüsselwörter und können als Variablennamen verwendet werden.



✓ Was ist ein Python-Modul? *	5/5
Eine Klasse mit Methoden	
Eine Funktion die andere Funktionen aufruft	
Eine Datei mit Python-Code die importiert werden kann	<b>✓</b>
Ein spezieller Datentyp für große Zahlen	
Feedback  Korrekt! Ein Python-Modul ist eine Datei, die Python-Code enthält und in anderen Python-Programmen importiert werden kann. Module ermöglichen die Wiederverwendung von Code und die Organisation von Programmen in separate, verwaltbare Dateien. Die Standardbibliothek von Python besteht aus vielen Modulen, die verschiedene Funktionalitäten bieten.	
✓ Welche der folgenden Anweisungen zu Python-Skripten ist KORREKT? *	5/5
<ul> <li>✓ Welche der folgenden Anweisungen zu Python-Skripten ist KORREKT? *</li> <li>● Python-Skripte haben immer die Dateiendung .py</li> </ul>	5/5
	5/5
Python-Skripte haben immer die Dateiendung .py	5/5
<ul><li>Python-Skripte haben immer die Dateiendung .py</li><li>Python-Skripte können nur unter Windows ausgeführt werden</li></ul>	5/5



•	Was macht die Shebang-Zeile '#!/usr/bin/env python3' am Anfang eines * Python-Skripts?	5/5
0	Sie importiert alle notwendigen Module	
$\bigcirc$	Sie kommentiert den Code für bessere Lesbarkeit	
	Sie gibt dem Betriebssystem an welcher Interpreter zum Ausführen verwendet werden soll	/
0	Sie definiert die Python-Version die für die Syntax überprüft wird	
Ko we de pla	errekt! Die Shebang-Zeile am Anfang eines Python-Skripts gibt dem Betriebssystem an, elcher Interpreter zum Ausführen des Skripts verwendet werden soll. In diesem Fall wird er Pfad zum Python 3-Interpreter über den Befehl 'env' ermittelt, was attformübergreifende Kompatibilität gewährleistet. Diese Zeile ist besonders wichtig in nix/Linux-Umgebungen, wenn Skripte direkt ausgeführt werden sollen.	
<b>✓</b>	Welche der folgenden Aussagen zum Zen of Python ist FALSCH? *	5/5
<b>/</b>	Welche der folgenden Aussagen zum Zen of Python ist FALSCH? *  'Einfach ist besser als komplex'	5/5
<ul><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li><!--</td--><td></td><td>5/5</td></li></ul>		5/5
<ul><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li><!--</td--><td>'Einfach ist besser als komplex'</td><td>5/5</td></li></ul>	'Einfach ist besser als komplex'	5/5
<ul><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li><!--</td--><td>'Einfach ist besser als komplex' 'Komplex ist besser als kompliziert'</td><td>5/5</td></li></ul>	'Einfach ist besser als komplex' 'Komplex ist besser als kompliziert'	5/5



✓ Wie werden Blöcke in Python definiert? *	5/5
Ourch geschweifte Klammern { }	
<ul><li>Durch Einrückung und Doppelpunkte</li></ul>	~
Ourch BEGIN und END Schlüsselwörter	
Ourch runde Klammern ( )	
Feedback  Korrekt! In Python werden Codeblöcke durch Einrückung definiert. Ein Beiner Zeile, die mit einem Doppelpunkt endet (wie bei if-, for-, while-Anw Funktionsdefinitionen), und alle nachfolgenden eingerückten Zeilen geh Block. Dies unterscheidet Python von vielen anderen Programmiersprac geschweifte Klammern oder BEGIN/END-Schlüsselwörter verwenden.	veisungen oder hören zu diesem
✓ Was ist PEP 8? *	5/5
✓ Was ist PEP 8? * ○ Ein Python-Erweiterungspaket	5/5
	5/5
Ein Python-Erweiterungspaket	5/5
<ul><li>Ein Python-Erweiterungspaket</li><li>Der Python-Paketmanager</li></ul>	5/5



✓ Was bedeutet der Begriff 'Duck Typing' in Python? *	5/5
<ul> <li>Eine Methode zur schnellen Fehlersuche</li> <li>Ein Designmuster für grafische Benutzeroberflächen</li> <li>Ein Prinzip bei dem der Typ eines Objekts durch sein Verhalten bestimmt wir</li> </ul>	d ./
Eine spezielle Art der Variablendeklaration	u 🗸
Feedback  Korrekt! Duck Typing ist ein Programmierkonzept, bei dem der Typ eines Objekts anhaseines Verhaltens (Methoden und Eigenschaften) und nicht anhand seiner Klasse od Vererbungsstruktur bestimmt wird. Der Name leitet sich vom Sprichwort ab: 'Wenn eine Ente aussieht, schwimmt wie eine Ente und quakt wie eine Ente, dann ist es wahrscheinlich eine Ente.' In Python bedeutet dies, dass der tatsächliche Typ eines Oweniger wichtig ist als sein Verhalten.	er s wie
✓ Was ist die korrekte Methode um das Paket 'numpy' in Python zu installieren?	<b>*</b> 5/5
python install numpy	
<ul><li>python -m pip install numpy</li><li>python -m install numpy</li><li>pip numpy install</li></ul>	<b>✓</b>
Feedback	
Korrekt! Der korrekte Befehl zum Installieren von Paketen in Python ist 'python -m pip install paketname'. Dies führt den pip-Paketmanager über das Python-Modul-System und installiert das angegebene Paket (in diesem Fall numpy). Dieser Ansatz stellt sic dass das Paket für die richtige Python-Installation installiert wird, wenn mehrere Vers auf dem System vorhanden sind.	aus her,

✓ Welcher der folgenden Befehle startet die interaktive Python-Shell? * 5/5	'
O python run	
python -interactive	
python	
opython shell	
Feedback	
Korrekt! Der Befehl 'python' ohne weitere Argumente startet die interaktive Python-Shell, in der Python-Code direkt eingegeben und ausgeführt werden kann. Die anderen Optionen sind keine gültigen Python-Befehle für diesen Zweck.	
✓ Welcher der folgenden Ausdrücke ergibt ein Ergebnis vom Typ float? * 5/5	
5+3	
5 // 3	
5 % 3	
Feedback	
Korrekt! Der Operator '/' führt in Python 3 immer eine Fließkommadivision durch. Daher ist $5/3 = 1.6666$ ein float. Die anderen Operationen ergeben Integer, wenn die Operanden Integer sind: $5 + 3 = 8$ , $5//3 = 1$ (Ganzzahldivision) und $5\%3 = 2$ (Modulo).	



✓ Was ist die Ausgabe des folgenden Codes? print('Hello' + 'World') * 5/	5
O Hello World	
HelloWorld	
Hello + World	
Fehler: Strings können nicht addiert werden	
Feedback  Korrekt! In Python werden Strings mit dem +-Operator verkettet (konkateniert). Der Ausdruck 'Hello' + 'World' fügt die beiden Strings zusammen, ohne ein Leerzeichen dazwischen einzufügen, was 'HelloWorld' ergibt.	
✓ Welche der folgenden Aussagen ist FALSCH in Bezug auf Python-Strings? *5/	5
Strings sind unveränderlich	
String-Indizierung beginnt bei 0	
Negative Indizes können verwendet werden	
Strings können mit der +-Methode verkettet werden	
Feedback  Korrekt! Strings werden mit dem +-Operator verkettet, nicht mit einer +-Methode. Die anderen Aussagen sind korrekt: Strings sind unveränderlich (immutable); die Indizierung beginnt bei 0; negative Indizes können verwendet werden, um von hinten auf Elemente zuzugreifen.	



✓ Was ist die Ausgabe des folgenden Codes? print(10 - 4 * 2) *	5/5
O 12	
2	<b>✓</b>
O 0	
○ 8	
Feedback	
Korrekt! Durch die Operatorrangfolge wird $4*2=8$ zuerst ausgeführt und dann $10-8$ Multiplikation hat eine höhere Priorität als Subtraktion.	= 2.

Dieses Formular wurde bei techstarter.de erstellt. - <u>Eigentümer dieses Formulars kontaktieren</u>
Sieht dieses Formular verdächtig aus? <u>Bericht</u>

## Google Formulare



