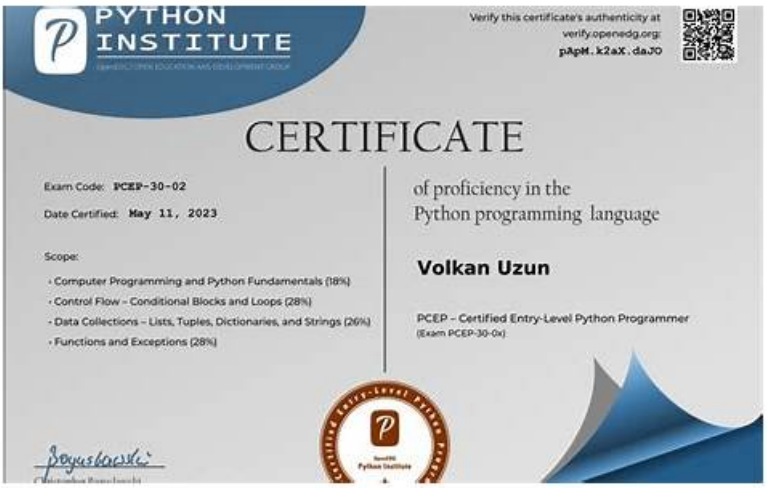
PCEP-30-02 Schulungsmaterialien & PCEP-30-02 Dumps Prüfung & PCEP-30-02 Studienguide



P.S. Kostenlose 2026 Python Institute PCEP-30-02 Prüfungsfragen sind auf Google Drive freigegeben von ExamFragen verfügbar: https://drive.google.com/open?id=12xVvTwrfPBo2BTxxvkDZkt6_K-MVjxJc

Bemühen Sie sich noch um die Python Institute PCEP-30-02 Zertifizierungsprüfung? Wollen Sie schneller Ihren Traum verwirklichen? Bitte wählen Sie die PCEP-30-02 Schulungsmaterialien von ExamFragen. Wenn Sie ExamFragen wählen, ist es kein Traum mehr, das Python Institute PCEP-30-02 Zertifikat zu erhalten.

Python Institute PCEP-30-02 Prüfungsplan:

Thema	Einzelheiten
Thema 1	• Loops: while, for, range(), loops control, and nesting of loops.
Thema 2	• Data Collections: In this section, the focus is on list construction, indexing, slicing, methods, and comprehensions; it covers Tuples, Dictionaries, and Strings.
Thema 3	• Functions and Exceptions: This part of the exam covers the definition of function and invocation
Thema 4	• parameters, arguments, and scopes. It also covers Recursion, Exception hierarchy, Exception handling, etc.

>> PCEP-30-02 Fragen Beantworten <<

Python Institute PCEP-30-02 Prüfungsmaterialien, PCEP-30-02 Lerntipps

Wir ExamFragen bieten Ihnen die umfassendsten Python Institute PCEP-30-02 Dumps mit sehr hoher Hit-Rate. Und alle Probleme, die vielleicht in aktuellen Prüfungen sind in Dumps vorhanden. Und wir aktualisieren unsere Dumps nach der Veränderung der Prüfungsinhalte. Es kann den sinnlosen Zeitaufwand vermeiden und Ihnen helfen, leichter und hocheffektiver die Python Institute PCEP-30-02 Prüfung zu bestehen. Obwohl Sie diePython Institute PCEP-30-02 Prüfung nicht bestehen, geben wir Ihnen voll Geld zurück. Deshalb können Sie keinen Verlust haben. Die Chance ist für die Leute, die gut bereit sind. Wir hoffen, dass Sie keine gut Chance verlieren.

Python Institute PCEP - Certified Entry-Level Python Programmer PCEP-30-02 Prüfungsfragen mit Lösungen (Q23-Q28):

23. Frage

What is the expected result of running the following code?

- A. The code prints 1 .
- B. The code prints 2
- C. The code raises an unhandled exception.
- D. The code prints 0

Antwort: C

Begründung:
Explanation
The code snippet that you have sent is trying to use the index method to find the position of a value in a list.
The code is as follows:
the_list = [1, 2, 3, 4, 5] print(the_list.index(6))
The code starts with creating a list called "the_list" that contains the numbers 1, 2, 3, 4, and 5. Then, it tries to print the result of calling the index method on the list with the argument 6. The index method is used to return the first occurrence of a value in a list. For example, the_list.index(1) returns 0, because 1 is the first value in the list.
However, the code has a problem. The problem is that the value 6 is not present in the list, so the index method cannot find it. This will cause a ValueError exception, which is an error that occurs when a function or operation receives an argument that has the right type but an inappropriate value. The code does not handle the exception, and therefore it will terminate with an error message.
The expected result of the code is an unhandled exception, because the code tries to find a value that does not exist in the list.
Therefore, the correct answer is C. The code raises an unhandled exception.

24. Frage
Assuming that the following assignment has been successfully executed:
Which of the following expressions evaluate to True? (Select two expressions.)

- A. the_list. index {'1'} -- 0
- B. 1.1 in the_list |1:3 |
- C. len (the list [0:2]} <3
- D. the_List.index {"1"} in the_list

Antwort: A,C

Begründung:
The code snippet that you have sent is assigning a list of four values to a variable called "the_list". The code is as follows:
the_list = ['1', 1, 1, 1]
The code creates a list object that contains the values '1', 1, 1, and 1, and assigns it to the variable "the_list".
The list can be accessed by using the variable name or by using the index of the values. The index starts from
0 for the first value and goes up to the length of the list minus one for the last value. The index can also be negative, in which case it counts from the end of the list. For example, the_list[0] returns '1', and the_list[-1] returns 1.
The expressions that you have given are trying to evaluate some conditions on the list and return a boolean value, either True or False. Some of them are valid, and some of them are invalid and will raise an exception.
An exception is an error that occurs when the code cannot be executed properly. The expressions are as follows:
A). the_List.index {"1"} in the_list: This expression is trying to check if the index of the value '1' in the list is also a value in the list. However, this expression is invalid, because it uses curly brackets instead of parentheses to call the index method. The index method is used to return the first occurrence of a value in a list. For example, the_list.index('1') returns 0, because '1' is the first value in the list. However, the_list.index
{"1"} will raise a SyntaxError exception and output nothing.
B). 1.1 in the_list |1:3 |: This expression is trying to check if the value 1.1 is present in a sublist of the list.
However, this expression is invalid, because it uses a vertical bar instead of a colon to specify the start and end index of the sublist. The sublist is obtained by using the slicing operation, which uses square brackets and a colon to get a part of the list. For example, the_list[1:3] returns [1, 1], which is the sublist of the list from the index 1 to the index 3, excluding the end index. However, the_list |1:3 | will raise a SyntaxError exception and output nothing.
C). len (the list [0:2]} <3: This expression is trying to check if the length of a sublist of the list is less than 3.
This expression is valid, because it uses the len function and the slicing operation correctly. The len function is used to return the number of values in a list or a sublist. For example, len(the_list) returns 4, because the list has four values. The slicing operation is used to get a part of the list by using square brackets and a colon. For example, the_list[0:2] returns ['1', 1], which is the sublist of the list from the index 0 to the index 2, excluding the end index. The expression len (the list [0:2]} <3 returns True, because the length of the sublist ['1', 1] is 2, which is less than 3.

D). the_list. index {'1'} - 0: This expression is trying to check if the index of the value '1' in the list is equal to 0. This expression is valid, because it uses the index method and the equality operator correctly. The index method is used to return the first occurrence of a value in a list. For example, the_list.index('1') returns 0, because '1' is the first value in the list. The equality operator is used to compare two values and return True if they are equal, or False if they are not. For example, 0 == 0 returns True, and 0 == 1 returns False. The expression the_list. index {'1'} - 0 returns True, because the index of '1' in the list is 0, and 0 is equal to 0.
Therefore, the correct answers are C. len (the list [0:2]} <3 and D. the_list. index {'1'} - 0.
Reference: Python List Methods - W3Schools5. Data Structures - Python 3.11.5 documentationList methods in Python - GeeksforGeeks

25. Frage
Which of the following are the names of Python passing argument styles?
(Select two answers.)

- A. keyword
- B. positional
- C. indicatory
- D. reference

Antwort: A,B

Begründung:
Keyword arguments are arguments that are specified by using the name of the parameter, followed by an equal sign and the value of the argument. For example, print (sep='-', end='!') is a function call with keyword arguments. Keyword arguments can be used to pass arguments in any order, and to provide default values for some arguments1.
Positional arguments are arguments that are passed in the same order as the parameters of the function definition. For example, print ('Hello', 'World') is a function call with positional arguments. Positional arguments must be passed before any keyword arguments, and they must match the number and type of the parameters of the function2.
References: 1: 5 Types of Arguments in Python Function Definitions | Built In 2: python - What's the pythonic way to pass arguments between functions ...

26. Frage
Insert the code boxes in the correct positions in order to build a line of code which asks the user for an Integer value and assigns it to the counter variable.
(Note: some code boxes will not be used.)

Antwort:

Begründung:

27. Frage
Assuming that the following assignment has been successfully executed:
My_list - [1, 1, 2, 3]
Select the expressions which will not raise any exception.
(Select two expressions.)

- A. my_list|my_Li1st | 3| I
- B. my list [6]
- C. my_List- [0:1]
- D. my_list[-10]

Antwort: A,C

Begründung:
Explanation
The code snippet that you have sent is assigning a list of four numbers to a variable called "my_list". The code is as follows:
my_list = [1, 1, 2, 3]
The code creates a list object that contains the elements 1, 1, 2, and 3, and assigns it to the variable "my_list".

The list can be accessed by using the variable name or by using the index of the elements. The index starts from 0 for the first element and goes up to the length of the list minus one for the last element. The index can also be negative, in which case it counts from the end of the list. For example, my_list[0] returns 1, and my_list[-1] returns 3.
The code also allows some operations on the list, such as slicing, concatenation, repetition, and membership.
Slicing is used to get a sublist of the original list by specifying the start and end index. For example, my_list[1:3] returns [1, 2].
Concatenation is used to join two lists together by using the + operator. For example, my_list + [4, 5] returns [1, 1, 2, 3, 4, 5].
Repetition is used to create a new list by repeating the original list a number of times by using the * operator. For example, my_list * 2 returns [1, 1, 2, 3, 1, 1, 2, 3].
Membership is used to check if an element is present in the list by using the in operator. For example, 2 in my_list returns True, and 4 in my_list returns False.
The expressions that you have given are trying to access or manipulate the list in different ways. Some of them are valid, and some of them are invalid and will raise an exception. An exception is an error that occurs when the code cannot be executed properly. The expressions are as follows:
A). my_list[-10]: This expression is trying to access the element at the index -10 of the list. However, the list only has four elements, so the index -10 is out of range. This will raise an IndexError exception and output nothing.
B). my_list|my_Li1st | 3| I: This expression is trying to perform a bitwise OR operation on the list and some other operands. The bitwise OR operation is used to compare the binary representation of two numbers and return a new number that has a 1 in each bit position where either number has a 1. For example, 3 | 1 returns 3, because 3 in binary is 11 and 1 in binary is 01, and 11 | 01 is 11. However, the bitwise OR operation cannot be applied to a list, because a list is not a number. This will raise a TypeError exception and output nothing.
C). my list [6]: This expression is trying to access the element at the index 6 of the list. However, the list only has four elements, so the index 6 is out of range. This will raise an IndexError exception and output nothing.
D). my_List- [0:1]: This expression is trying to perform a subtraction operation on the list and a sublist. The subtraction operation is used to subtract one number from another and return the difference. For example, 3 - 1 returns 2. However, the subtraction operation cannot be applied to a list, because a list is not a number. This will raise a TypeError exception and output nothing.
Only two expressions will not raise any exception. They are:
B). my_list|my_Li1st | 3| I: This expression is not a valid Python code, but it is not an expression that tries to access or manipulate the list. It is just a string of characters that has no meaning. Therefore, it will not raise any exception, but it will also not output anything.
D). my_List- [0:1]: This expression is a valid Python code that uses the slicing operation to get a sublist of the list. The slicing operation does not raise any exception, even if the start or end index is out of range. It will just return an empty list or the closest possible sublist. For example, my_list[0:10] returns [1, 1, 2, 3], and my_list[10:20] returns []. The expression my_List- [0:1] returns the sublist of the list from the index 0 to the index 1, excluding the end index. Therefore, it returns [1]. This expression will not raise any exception, and it will output [1].
Therefore, the correct answers are B. my_list|my_Li1st | 3| I and D. my_List- [0:1].

28. Frage
.....

- PCEP-30-02 Testantworten □ PCEP-30-02 Prüfungsaufgaben ➡□ PCEP-30-02 Buch □ Sie müssen nur zu 「 www.itzert.com 」 gehen um nach kostenloser Download von " PCEP-30-02 " zu suchen □PCEP-30-02 Fragen Und Antworten
- Die seit kurzem aktuellsten PCEP - Certified Entry-Level Python Programmer Prüfungsunterlagen, 100% Garantie für Ihen Erfolg in der Python Institute PCEP-30-02 Prüfungen! □ Sie müssen nur zu 《 www.zertfragen.com 》 gehen um nach kostenloser Download von ⇛ PCEP-30-02 ⇚ zu suchen □PCEP-30-02 Online Prüfungen
- PCEP-30-02 Prüfungsfragen Prüfungsvorbereitungen, PCEP-30-02 Fragen und Antworten, PCEP - Certified Entry-Level Python Programmer □ Öffnen Sie die Website ➤ www.itzert.com □ Suchen Sie 「 PCEP-30-02 」 Kostenloser Download □PCEP-30-02 Examsfragen
- PCEP-30-02 Mit Hilfe von uns können Sie bedeutendes Zertifikat der PCEP-30-02 einfach erhalten! □ Öffnen Sie die Website 《 www.echtefrage.top 》 Suchen Sie ▹ PCEP-30-02 ◃ Kostenloser Download □PCEP-30-02 Musterprüfungsfragen
- Neueste PCEP - Certified Entry-Level Python Programmer Prüfung pdf - PCEP-30-02 Prüfung Torrent □ Suchen Sie auf ➥ www.itzert.com □ nach kostenlosem Download von □ PCEP-30-02 □ □PCEP-30-02 Zertifikatsfragen
- PCEP-30-02 Torrent Anleitung - PCEP-30-02 Studienführer - PCEP-30-02 wirkliche Prüfung □ Suchen Sie auf ▹ www.zertpruefung.de ◃ nach 【 PCEP-30-02 】 und erhalten Sie den kostenlosen Download mühelos □PCEP-30-02 Examsfragen
- sidneyzrie097590.wikifordummies.com, behindvlsi.com, socialbuzztoday.com, keithiuvp625556.liberty-blog.com, kalesiwm040384.blogginaway.com, allennjjh189968.bloggip.com, idamgcz906139.wikiap.com, bookmarkfox.com, www.stes.tyc.edu.tw, mayaiooe283168.activablog.com, Disposable vapes

Laden Sie die neuesten ExamFragen PCEP-30-02 PDF-Versionen von Prüfungsfragen kostenlos von Google Drive herunter: https://drive.google.com/open?id=12xVvTwrfPBo2BTxxvkDZkt6_K-MVjxJc