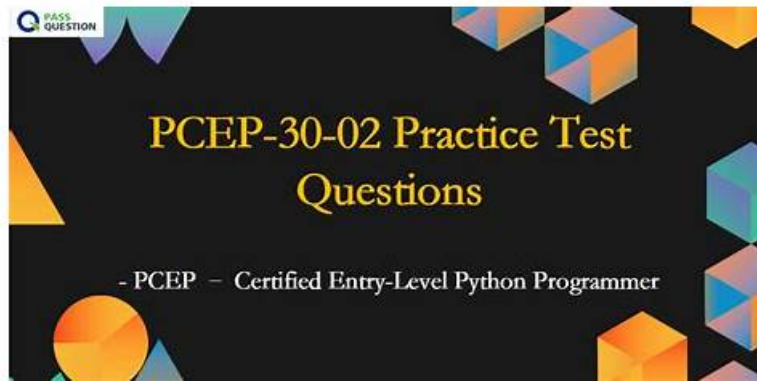


PCEP-30-02考試指南， PCEP-30-02學習筆記



順便提一下，可以從雲存儲中下載Fast2test PCEP-30-02考試題庫的完整版：https://drive.google.com/open?id=1z4mbTu9clwBpWfD7kSsrhYFDLs_u6xxv

如果你要參加Python Institute的PCEP-30-02認定考試，Fast2test的PCEP-30-02考古題是你最好的準備工具。這個資料可以幫助你輕鬆地通過考試。這是一個評價很高的資料，有了它，你就不用再擔心你的考試了。因為這個考古題可以解決你在準備考試時遇到的一切難題。在購買Fast2test的PCEP-30-02考古題之前，你還可以下載免費的考古題樣本作為試用。這樣你就可以自己判斷這個資料是不是適合自己。

Python Institute PCEP-30-02 考試大綱：

主題	簡介
主題 1	<ul style="list-style-type: none">Control Flow: This section covers conditional statements such as if, if-else, if-elif, if-elif-else
主題 2	<ul style="list-style-type: none">Functions and Exceptions: This part of the exam covers the definition of function and invocation
主題 3	<ul style="list-style-type: none">Loops: while, for, range(), loops control, and nesting of loops.
主題 4	<ul style="list-style-type: none">Data Collections: In this section, the focus is on list construction, indexing, slicing, methods, and comprehensions; it covers Tuples, Dictionaries, and Strings.
主題 5	<ul style="list-style-type: none">Computer Programming Fundamentals: This section of the exam covers fundamental concepts such as interpreters, compilers, syntax, and semantics. It covers Python basics: keywords, instructions, indentation, comments in addition to Booleans, integers, floats, strings, and Variables, and naming conventions. Finally, it covers arithmetic, string, assignment, bitwise, Boolean, relational, and Inputoutput operations.

>> PCEP-30-02考試指南 <<

完全覆蓋的PCEP-30-02考試指南和資格考試和熱門的PCEP-30-02學習筆記的領導者

PCEP-30-02 認證是 Python Institute 認證體系中增長最快的領域，也是一個國際性的廠商中比較難的認證考試。不過不用擔心，Fast2test 就是一個能使 PCEP-30-02 認證考試的通過率提高的一個網站，我們的 Python Institute PCEP-30-02 考題指南由我們的專業團隊破解PCEP-30-02 考試系統數據包，經過資深IT認證講師和技術專家精心編輯整理。包括了當前 PCEP-30-02 考試所有單選題、複選題、實作題、拖拉題等題型。可以幫助考生順利通過考試。

最新的 Python Institute PCEP PCEP-30-02 免費考試真題 (Q37-Q42):

問題 #37

A set of rules which defines the ways in which words can be coupled in sentences is called:

- A. lexis
- **B. syntax**
- C. dictionary
- D. semantics

答案： B

解題說明：

Syntax is the branch of linguistics that studies the structure and rules of sentences in natural languages. Lexis is the vocabulary of a language. Semantics is the study of meaning in language. A dictionary is a collection of words and their definitions, synonyms, pronunciations, etc.

Reference: [Python Institute - Entry-Level Python Programmer Certification]

問題 #38

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[0:1]**
- **B. my_list|my_list | 3| 1**
- C. my_list[-10]
- D. my_list[6]

答案： A,B

解題說明：

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_list | 3| 1: 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_list | 3| 1: 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_List | 3]` I and D. `my_List- [0:1]`.

問題 #39

Arrange the code boxes in the correct positions in order to obtain a loop which executes its body with the counter variable going through values 1, 3, and 5 (in the same order)



答案:

解題說明:

`for counter in range(1, 7, 2):`

Explanation:

* for

* counter

* in

* range

* (

* 1

*,

* 7

*,

* 2

*)

Arrange the code boxes in this order:

This will loop counter through: 1 # 3 # 5

問題 #40

What is the expected result of the following code?

```
rates = (1.2, 1.4, 1.0)
new = rates[3:]
for rate in rates[-2:]:
    new.append(rate)
print(len(new))
```

- A. 0
- B. The code will cause an unhandled
- C. 1
- D. 2

答案: B

解題說明:

The code snippet that you have sent is trying to use a list comprehension to create a new list from an existing list. The code is as follows:

`my_list = [1, 2, 3, 4, 5] new_list = [x for x in my_list if x > 5]`

The code starts with creating a list called "my_list" that contains the numbers 1, 2, 3, 4, and 5. Then, it tries to create a new list called "new_list" by using a list comprehension. A list comprehension is a concise way of creating a new list from an existing list by applying some expression or condition to each element. The syntax of a list comprehension is:

```
new_list = [expression for element in old_list if condition]
```

The expression is the value that will be added to the new list, which can be the same as the element or a modified version of it. The element is the variable that takes each value from the old list. The condition is an optional filter that determines which elements will be included in the new list. For example, the following list comprehension creates a new list that contains the squares of the even numbers from the old list:

```
old_list = [1, 2, 3, 4, 5, 6] new_list = [x ** 2 for x in old_list if x % 2 == 0] new_list = [4, 16, 36]
```

The code that you have sent is trying to create a new list that contains the elements from the old list that are greater than 5. However, there is a problem with this code. The problem is that none of the elements in the old list are greater than 5, so the condition is always false. This means that the new list will be empty, and the expression will never be evaluated. However, the expression is not valid, because it uses the variable x without defining it. This will cause a NameError exception, which is an error that occurs when a variable name is not found in the current scope. 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 use an undefined variable in an expression that is never executed. Therefore, the correct answer is D. The code will cause an unhandled exception.

Reference: Python - List Comprehension - W3Schools Python - List Comprehension - GeeksforGeeks Python Exceptions: An Introduction - Real Python

問題 #41

Arrange the code boxes in the correct positions to form a conditional instruction which guarantees that a certain statement is executed when the temperature variable is equal to 0.0.



答案:

解題說明:

```
if temperature == 0.0:
```

Explanation:

```
* if
* temperature
* ==
* 0.0
* :
```

Arrange the boxes in this order:

This checks if temperature is exactly 0.0, and if so, runs the code inside the if block.

問題 #42

.....

如果你想在短時間內，以最小的努力，達到最有效果的結果Python Institute的PCEP-30-02考試準備，你可以使用Fast2test Python Institute的PCEP-30-02考試培訓資料，Fast2test的培訓資料是通過實踐檢驗了的，也是通過眾多考生證明了它確實可以百分百通過考試，利用了它，你將達到你的目的，得到最佳的效果。

PCEP-30-02學習筆記: <https://tw.fast2test.com/PCEP-30-02-premium-file.html>

- PCEP-30-02題庫更新 PCEP-30-02題庫資料 PCEP-30-02考證 立即在《www.testpdf.net》上搜尋「PCEP-30-02」並免費下載PCEP-30-02認證資料
- 最新的PCEP-30-02考試指南 - Python Institute PCEP-30-02學習筆記: PCEP - Certified Entry-Level Python Programmer成功通過 透過> www.newdumpspdf.com <輕鬆獲取“PCEP-30-02”免費下載新版PCEP-30-02題庫
- 值得信賴的PCEP-30-02考試指南和資格考試中的領先提供者與最新的PCEP-30-02學習筆記 立即打開（www.newdumpspdf.com）並搜索✪ PCEP-30-02 ✪ 以獲取免費下載PCEP-30-02最新題庫
- PCEP-30-02指南 PCEP-30-02最新試題 PCEP-30-02最新試題 在（www.newdumpspdf.com）上搜索“PCEP-30-02”並獲取免費下載PCEP-30-02最新試題
- PCEP-30-02考證 PCEP-30-02題庫資料 PCEP-30-02最新試題 免費下載 PCEP-30-02 只需進入

