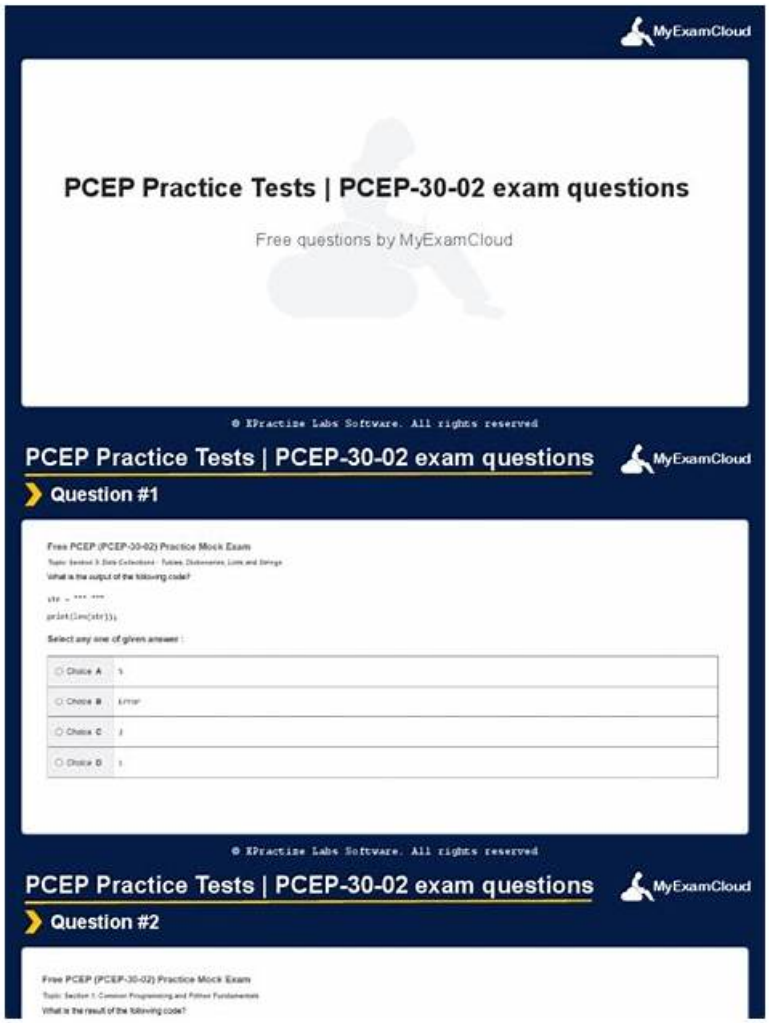
PCEP-30-02資格勉強 & PCEP-30-02日本語対策問題集



P.S.JPTestKingがGoogle Driveで共有している無料の2026 Python Institute PCEP-30-02ダンプ：https://drive.google.com/open?id=13A2-92qYcInRIzvqUcqpa5aosMlak0my

JPTestKingのPython InstituteのPCEP-30-02試験トレーニング資料は受験生が模擬試験場で勉強させます。受験生は問題を選べ、テストの時間もコントロールできます。JPTestKingというサイトで、あなたはストレスと不安なく試験の準備をすることができますから、一般的な間違いを避けられます。そうしたら、あなたは自信を得ることができて、実際の試験で経験を活かして気楽に合格します。

Python Institute PCEP-30-02 認定試験の出題範囲：

トピック	出題範囲
トピック 1	• 制御フロー: このセクションでは、if、if-else、if-elif、if-elif-elseなどの条件文について説明します。
トピック 2	• コンピュータプログラミングの基礎: この試験セクションでは、インタープリタ、コンパイラ、構文、セマンティクスなどの基本的な概念を扱います。キーワード、命令、インデント、コメント、ブール値、整数、浮動小数点数、文字列、変数、命名規則など、Pythonの基本を扱います。最後に、算術、文字列、割り当て、ビット単位、ブール値、関係、および入出力操作を扱います。
トピック 3	• ループ: while、for、range()、ループ制御、ループのネスト。

トピック４	• パラメータ、引数、スコープ。また、再帰、例外階層、例外処理などもカバーしています。

>> PCEP-30-02資格勉強 <<

PCEP-30-02日本語対策問題集、PCEP-30-02参考書勉強

PCEP-30-02試験の認証資格を取得したら、あなたは利益を得られます。あなたは試験に参加したいなら、我々のPCEP-30-02問題集はあなたの最高の復習方法です。この問題集で、あなたは気楽でPCEP-30-02試験に合格することができます。我々の資料があったら、あなたは試験の復習を心配する必要がありません。

Python Institute PCEP - Certified Entry-Level Python Programmer 認定 PCEP-30-02 試験問題 (Q39-Q44):

質問 # 39
Which of the following expressions evaluate to a non-zero result? (Select two answers.)

- A. 1 * 4 // 2 ** 3
- B. 4 / 2 * * 3 - 2
- C. 1 * * 3 / 4 - 1
- D. 2 ** 3 / A - 2

正解：B、D

解説：
In Python, the ** operator is used for exponentiation, the / operator is used for floating-point division, and the // operator is used for integer division. The order of operations is parentheses, exponentiation, multiplication /division, and addition/subtraction. Therefore, the expressions can be evaluated as follows:
A). 2 ** 3 / A - 2 = 8 / A - 2 (assuming A is a variable that is not zero or undefined) B. 4 / 2 * * 3 - 2 = 4 / 8 - 2 = 0.5 - 2 = -1.5 C. 1 * * 3 / 4 - 1 = 1 / 4 - 1 = 0.25 - 1 = -0.75 D. 1 * 4 // 2 ** 3 = 4 // 8 = 0 Only expressions A and B evaluate to non-zero results.
Reference: [Python Institute - Entry-Level Python Programmer Certification]

質問 # 40
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[-10]
- B. my_List- [0:1]
- C. my_list|my_Li1st | 3| I
- D. my list [6]

正解：B、C

解説：
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].

質問 # 41
Drag and drop the literals to match their data type names.

正解：

解説：
Explanation
One possible way to drag and drop the literals to match their data type names is:
STRING: "All The King's Men"
BOOLEAN: False
INTEGER: 42
FLOAT: -6.62607015E-34
A literal is a value that is written exactly as it is meant to be interpreted by the Python interpreter. A data type is a category of values that share some common characteristics or operations. Python has four basic data types:
string, boolean, integer, and float.
A string is a sequence of characters enclosed by either single or double quotes. A string can represent text, symbols, or any other information that can be displayed as text. For example, "All The King's Men" is a string literal that represents the title of a novel.
A boolean is a logical value that can be either True or False. A boolean can represent the result of a comparison, a condition, or a logical operation. For example, False is a boolean literal that represents the opposite of True.
An integer is a whole number that can be positive, negative, or zero. An integer can represent a count, an index, or any other quantity that does not require fractions or decimals. For example, 42 is an integer literal that represents the answer to life, the universe, and everything.
A float is a number that can have a fractional part after the decimal point. A float can represent a measurement, a ratio, or any other quantity that requires precision or approximation. For example,
-6.62607015E-34 is a float literal that represents the Planck constant in scientific notation.
You can find more information about the literals and data types in Python in the following references:
[Python Data Types]
[Python Literals]

[Python Basic Syntax]

質問 # 42
How many hashes (+) does the code output to the screen?

- A. three
- B. five
- C. zero (the code outputs nothing)
- D. one

正解：B

解説：
The code snippet that you have sent is a loop that checks if a variable "floor" is less than or equal to 0 and prints a string accordingly. The code is as follows:
floor = 5 while floor > 0: print("+") floor = floor - 1
The code starts with assigning the value 5 to the variable "floor". Then, it enters a while loop that repeats as long as the condition "floor > 0" is true. Inside the loop, the code prints a "+" symbol to the screen, and then subtracts 1 from the value of "floor". The loop ends when "floor" becomes 0 or negative, and the code exits.
The code outputs five "+" symbols to the screen, one for each iteration of the loop. Therefore, the correct answer is C. five.
Reference: [Python Institute - Entry-Level Python Programmer Certification]

質問 # 43
What is true about tuples? (Select two answers.)

- A. Tuples are immutable, which means that their contents cannot be changed during their lifetime.
- B. An empty tuple is written as { } .
- C. Tuples can be indexed and sliced like lists.
- D. The len { } function cannot be applied to tuples.

正解：A、C

解説：
Explanation
Tuples are one of the built-in data types in Python that are used to store collections of data. Tuples have some characteristics that distinguish them from other data types, such as lists, sets, and dictionaries. Some of these characteristics are:
Tuples are immutable, which means that their contents cannot be changed during their lifetime. Once a tuple is created, it cannot be modified, added, or removed. This makes tuples more stable and reliable than mutable data types. However, this also means that tuples are less flexible and dynamic than mutable data types. For example, if you want to change an element in a tuple, you have to create a new tuple with the modified element and assign it to the same variable12 Tuples are ordered, which means that the items in a tuple have a defined order and can be accessed by using their index. The index of a tuple starts from 0 for the first item and goes up to the length of the tuple minus one for the last item. The index can also be negative, in which case it counts from the end of the tuple. For example, if you have a tuple t = ("a", "b", "c"), then t[0] returns "a", and t[-1] returns "c"12 Tuples can be indexed and sliced like lists, which means that you can get a single item or a sublist of a tuple by using square brackets and specifying the start and end index. For example, if you have a tuple t
= ("a", "b", "c", "d", "e"), then t[2] returns "c", and t[1:4] returns ("b", "c", "d"). Slicing does not raise any exception, even if the start or end index is out of range. It will just return an empty tuple or the closest possible sublist12 Tuples can contain any data type, such as strings, numbers, booleans, lists, sets, dictionaries, or even other tuples. Tuples can also have duplicate values, which means that the same item can appear more than once in a tuple. For example, you can have a tuple t = (1, 2, 3, 1, 2), which contains two 1s and two
2s12
Tuples are written with round brackets, which means that you have to enclose the items in a tuple with parentheses. For example, you can create a tuple t = ("a", "b", "c") by using round brackets. However, you can also create a tuple without using round brackets, by just separating the items with commas. For example, you can create the same tuple t = "a", "b", "c" by using commas. This is called tuple packing, and it allows you to assign multiple values to a single variable12 The len() function can be applied to tuples, which means that you can get the number of items in a tuple by using the len() function. For example, if you have a tuple t = ("a", "b", "c"), then len(t) returns 312 An empty tuple is written as (), which means that you have to use an empty pair of parentheses to create a tuple with no items. For example, you can create an empty tuple t = () by using empty parentheses.
However, if you want to create a tuple with only one item, you have to add a comma after the item, otherwise Python will not

recognize it as a tuple. For example, you can create a tuple with one item t = ("a",) by using a comma12 Therefore, the correct answers are A.
Tuples are immutable, which means that their contents cannot be changed during their lifetime. and D. Tuples can be indexed and sliced like lists.

質問 # 44
.....

JPTestKingが提供しておりますのは専門家チームの研究したPCEP-30-02問題と真題で弊社の高い名誉はたぶり信頼をうけられます。安心で弊社の商品を使うために無料なPCEP-30-02サンプルをダウンロードしてください。

PCEP-30-02日本語対策問題集: https://www.jptestking.com/PCEP-30-02-exam.html

- PCEP-30-02日本語復習赤本 □ PCEP-30-02日本語復習赤本 □ PCEP-30-02日本語受験教科書 □ ➥ www.xhs1991.com □から簡単に□ PCEP-30-02 □を無料でダウンロードできますPCEP-30-02試験概要
- PCEP-30-02復習問題集 □ PCEP-30-02資格準備 □ PCEP-30-02入門知識 □ ☀ www.goshiken.com □☀□で ▸ PCEP-30-02 ◂を検索して、無料で簡単にダウンロードできますPCEP-30-02入門知識
- PCEP-30-02テスト資料 □ PCEP-30-02認定デベロッパー □ PCEP-30-02資格認定 □ 《 www.it-passports.com 》には無料の⇛ PCEP-30-02 ⇚問題集がありますPCEP-30-02専門知識内容
- PCEP-30-02ミシュレーション問題 □ PCEP-30-02テスト資料 □ PCEP-30-02日本語受験教科書 □ ✔ www.goshiken.com □✔□を開き、➽ PCEP-30-02 □を入力して、無料でダウンロードしてくださいPCEP-30-02復習問題集
- 素敵PCEP-30-02｜完璧なPCEP-30-02資格勉強試験｜試験の準備方法PCEP - Certified Entry-Level Python Programmer日本語対策問題集 □ □ www.passtest.jp □は、（ PCEP-30-02 ）を無料でダウンロードするのに最適なサイトですPCEP-30-02テスト資料
- PCEP-30-02資格準備 □ PCEP-30-02日本語受験教科書 ✔□ PCEP-30-02復習問題集 □ 【 www.goshiken.com 】に移動し、▹ PCEP-30-02 ◃を検索して、無料でダウンロード可能な試験資料を探しますPCEP-30-02テスト資料
- 試験の準備方法-効率的なPCEP-30-02資格勉強試験-完璧なPCEP-30-02日本語対策問題集 □ 時間限定無料で使える➤ PCEP-30-02 □の試験問題は□ www.goshiken.com □サイトで検索PCEP-30-02トレーニング資料
- 試験の準備方法-効率的なPCEP-30-02資格勉強試験-完璧なPCEP-30-02日本語対策問題集 □ [www.goshiken.com]から☀ PCEP-30-02 □☀□を検索して、試験資料を無料でダウンロードしてください PCEP-30-02専門知識内容
- 便利なPCEP-30-02資格勉強 - 合格スムーズPCEP-30-02日本語対策問題集 | 効果的なPCEP-30-02参考書勉強 □ 「 www.passtest.jp 」サイトにて{ PCEP-30-02 }問題集を無料で使おうPCEP-30-02テスト資料
- PCEP-30-02試験の準備方法｜正確的なPCEP-30-02資格勉強試験｜検証するPCEP - Certified Entry-Level Python Programmer日本語対策問題集 □ 検索するだけで【 www.goshiken.com 】から☀ PCEP-30-02 □☀□を無料でダウンロードPCEP-30-02ミシュレーション問題
- 真実的なPCEP-30-02資格勉強 - 合格スムーズPCEP-30-02日本語対策問題集 | 完璧なPCEP-30-02参考書勉強 □ 今すぐ「 www.it-passports.com 」で▸ PCEP-30-02 ◂を検索して、無料でダウンロードしてください PCEP-30-02模擬トレーリング
- www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, Disposable vapes

BONUS！！！ JPTestKing PCEP-30-02ダンプの一部を無料でダウンロード：https://drive.google.com/open?id=13A2-92qYcInRIzvqUcqpa5aosMlak0my