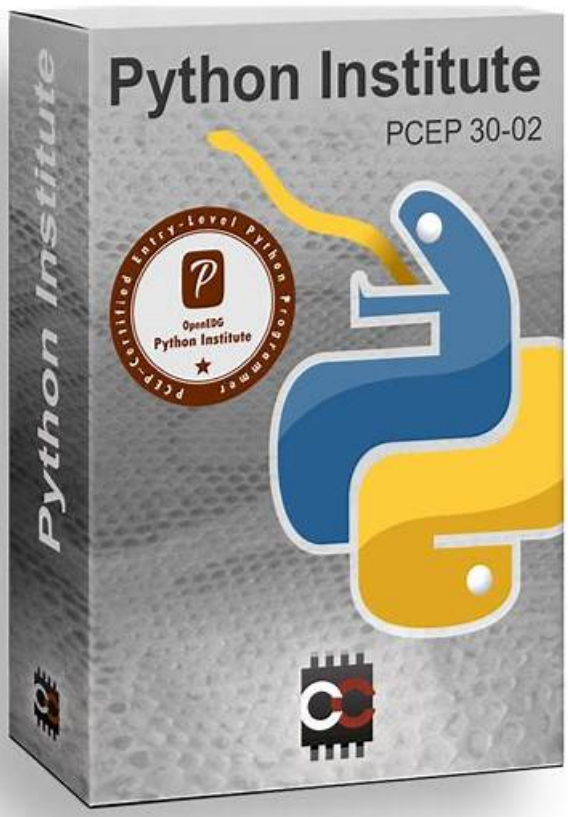
Free PDF Quiz Python Institute - PCEP-30-02–High Pass-Rate Valid Test Registration



P.S. Free & New PCEP-30-02 dumps are available on Google Drive shared by PDFTorrent: https://drive.google.com/open?id=1aO917tA60tD2W6haYoig6_0cRc4ESGTr

Contemporarily, social competitions stimulate development of modern science, technology and business, which revolutionizes our society's recognition to PCEP-30-02 exam and affect the quality of people's life. According to a recent report, those who own more than one skill certificate are easier to be promoted by their boss. To be out of the ordinary and seek an ideal life, we must master an extra skill to get high scores and win the match in the workplace. Our PCEP-30-02 Exam Question can help make your dream come true. What's more, you can have a visit of our website that provides you more detailed information about the PCEP-30-02 guide torrent.

Python Institute PCEP-30-02 Exam Syllabus Topics:

Topic	Details
Topic 1	• 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 Input • output operations.
Topic 2	• Control Flow: This section covers conditional statements such as if, if-else, if-elif, if-elif-else
Topic 3	• Functions and Exceptions: This part of the exam covers the definition of function and invocation

PCEP-30-02 Study Materials: PCEP - Certified Entry-Level Python Programmer & PCEP-30-02 Certification Training

If you use our PCEP-30-02 practice test software, you can prepare for the exam in an atmosphere that is quite similar to the PCEP-30-02 real test, which will greatly aid in your preparation. The Python Institute PCEP-30-02 desktop practice exam software keeps track of your previous tries. This feature will help you identify where you need the most improvement so you can focus your efforts and boost your score the next time you take the PCEP - Certified Entry-Level Python Programmer (PCEP-30-02) practice test.

Python Institute PCEP - Certified Entry-Level Python Programmer Sample Questions (Q12-Q17):

NEW QUESTION # 12
What is the expected output of the following code?

```
equals = 0
for i in range(2):
    for j in range(2):
        if i != j:
            equals += 1
else:
    equals += 1
print(equals)
```

- A. 0
- B. The code outputs nothing.
- C. 1
- D. 2

Answer: D

Explanation:
Explanation
The code snippet that you have sent is checking if two numbers are equal and printing the result. The code is as follows:
num1 = 1 num2 = 2 if num1 == num2: print(4) else: print(1)
The code starts with assigning the values 1 and 2 to the variables "num1" and "num2" respectively. Then, it enters an if statement that compares the values of "num1" and "num2" using the equality operator (==). If the values are equal, the code prints 4 to the screen. If the values are not equal, the code prints 1 to the screen.
The expected output of the code is 1, because the values of "num1" and "num2" are not equal. Therefore, the correct answer is C. 1.

NEW QUESTION # 13
Which of the following are the names of Python passing argument styles?
(Select two answers.)

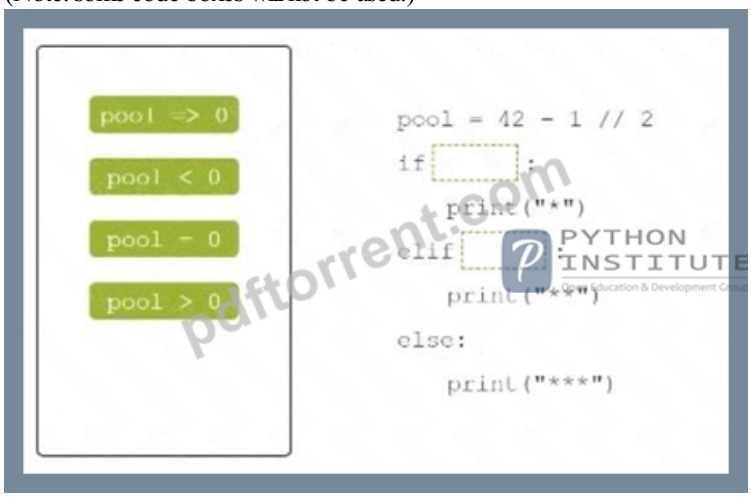
- A. indicatory
- B. reference
- C. keyword
- D. positional

Answer: C,D

Explanation:
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

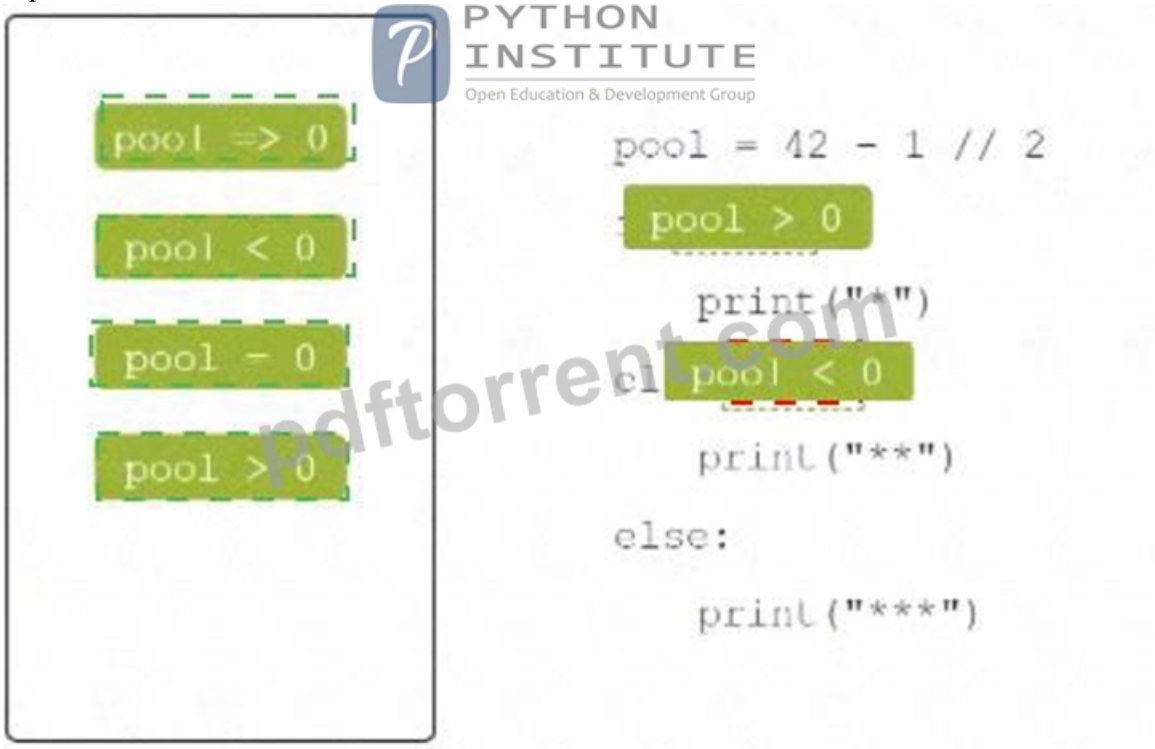
('Hello', 'World') is a function call with positional arguments. Positional arguments must be passed before any keyword arguments, and they must match the number and type of the parameters of the function2.
References: 1: 5 Types of Arguments in Python Function Definitions | Built In 2: python - What's the pythonic way to pass arguments between functions ...

NEW QUESTION # 14
Drag and drop the conditional expressions to obtain a code which outputs * to the screen.
(Note: some code boxes will not be used.)

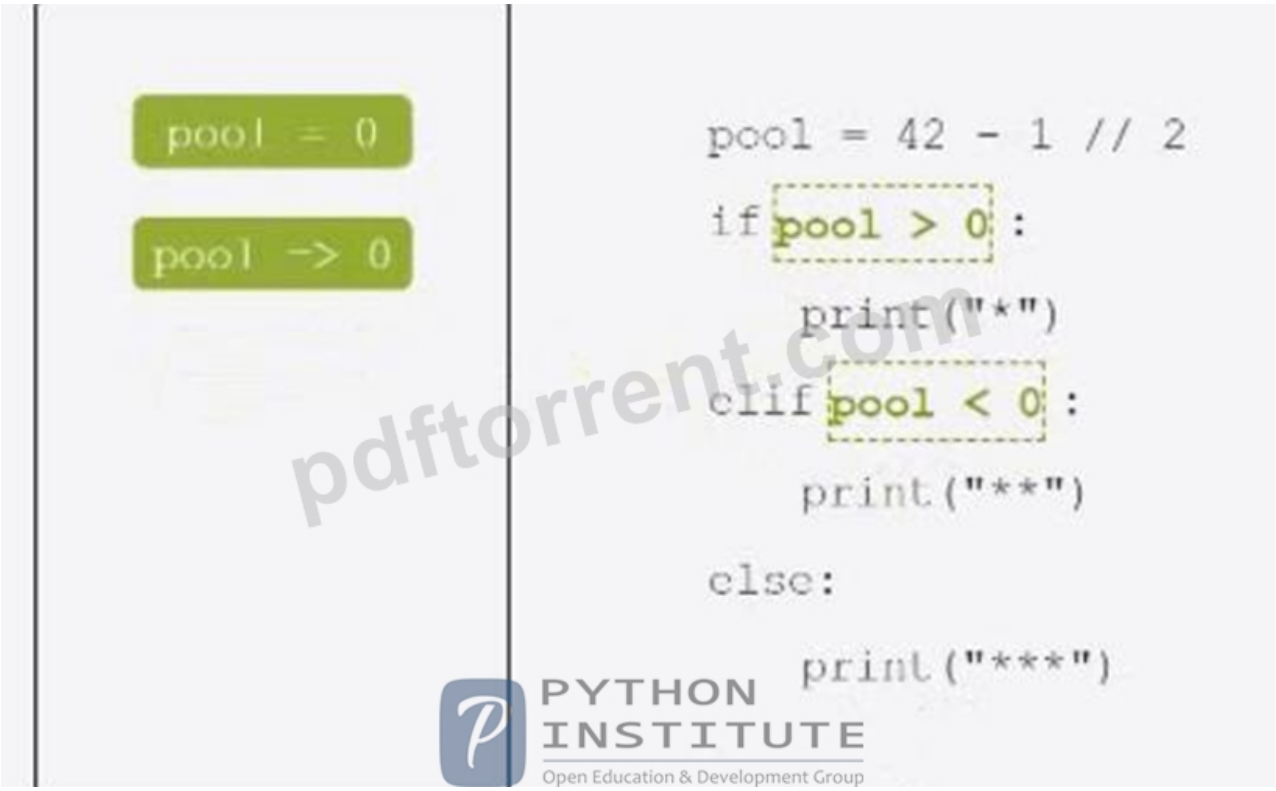


Answer:

Explanation:



Explanation



One possible way to drag and drop the conditional expressions to obtain a code which outputs * to the screen is:

```
if pool > 0:
print("*")
elif pool < 0:
print("**")
else:
print("***")
```

This code uses the if, elif, and else keywords to create a conditional statement that checks the value of the variable pool. Depending on whether the value is greater than, less than, or equal to zero, the code will print a different pattern of asterisks to the screen. The print function is used to display the output. The code is indented to show the blocks of code that belong to each condition. The code will output * if the value of pool is positive, ** if the value of pool is negative, and *** if the value of pool is zero.

You can find more information about the conditional statements and the print function in Python in the following references:

[Python If ... Else]

[Python Print Function]

[Python Basic Syntax]

NEW QUESTION # 15

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

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

Answer: B,C

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.
Reference: Python Tuples - W3SchoolsTuples in Python - GeeksforGeeks

NEW QUESTION # 16
What is true about tuples? (Select two answers.)

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

Answer: B,C

Explanation:
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.

NEW QUESTION # 17
.....

It is undeniable that a secure investment can bring many benefits to candidates who want to pass the PCEP-30-02 exam, without worrying that their money is wasted on useless exam materials, and the most important thing is to pass PCEP-30-02 exams. In addition, after the purchase, the candidate will be entitled to a one-year free update, which will help the candidate keep the latest news feeds, and will not leave any opportunity that may lead them to fail the PCEP-30-02 Exam. We also provide a 100% refund policy for all users who purchase our questions. If for any reason, any candidates fail in the Python Institute PCEP-30-02 certification exam, we can help you to refund your money and ensure your investment is absolutely safe.