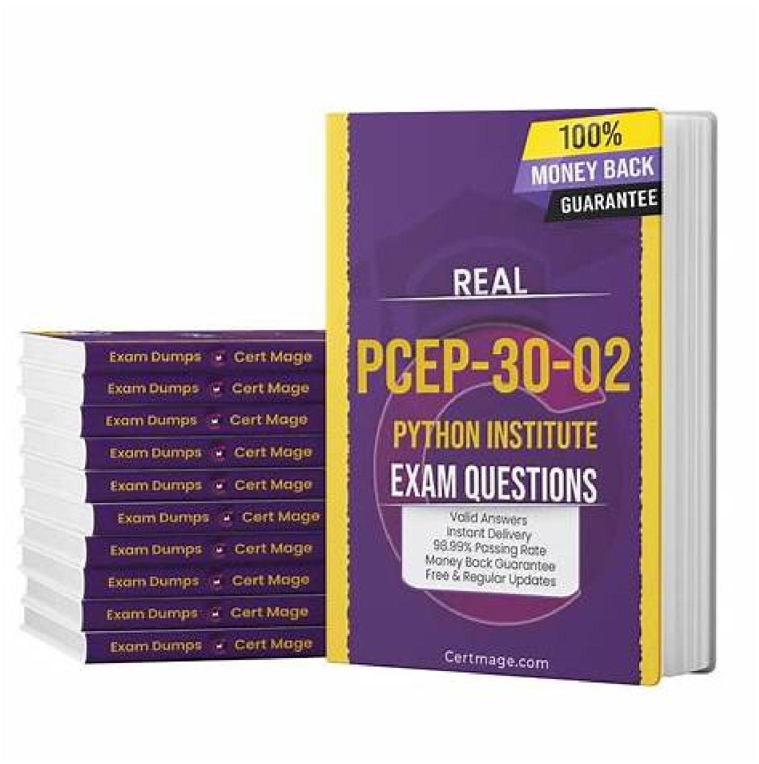
PCEP-30-02考古題分享 -最新PCEP-30-02考題



從Google Drive中免費下載最新的PDFExamDumps PCEP-30-02 PDF版考試題庫：https://drive.google.com/open?id=1KCehJoiGq9EdhiqP6Tg7R20-aIYpUzQG

我們PDFExamDumps不僅僅提供優質的產品給每位PCEP-30-02考生，而且提供完善的售後服務給每位考生，如果你使用了我們的產品，我們將讓你享受一年免費的更新，並且在第一時間回饋給每位考生，讓你及時得到更新的最新的考試資料，以最大效益的服務給每位PCEP-30-02考生。

Python Institute PCEP-30-02 考試大綱：

主題	簡介
主題 1	• Control Flow: This section covers conditional statements such as if, if-else, if-elif, if-elif-else
主題 2	• parameters, arguments, and scopes. It also covers Recursion, Exception hierarchy, Exception handling, etc.
主題 3	• 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.
主題 4	• 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	• Loops: while, for, range(), loops control, and nesting of loops.

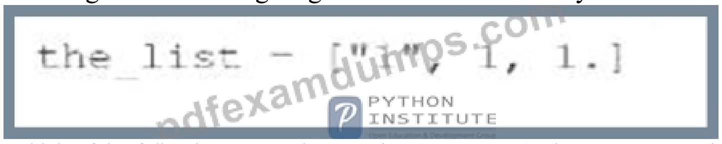
>> PCEP-30-02考古題分享 <<

可靠的PCEP-30-02考古題分享 |高通過率的考試材料|值得信賴的PCEP-30-02：PCEP - Certified Entry-Level Python Programmer

作為一位 Python Institute PCEP-30-02 考生而言，作好充分的準備可以幫助您通過考試。首先您必須去當地考試中心咨詢相關考試信息，然后挑選最新的 PCEP-30-02 考試題庫，因為擁有了最新的 PCEP-30-02 考試題庫可以有利的提高通過考試的機率。使用PDFExamDumps 的題庫可以節省您寶貴的時間，保證你順利通過 PCEP-30-02 考試。既能幫您節省時間，又可以順利幫助您通過考試，這將是您的最佳選擇。

最新的 Python Institute PCEP PCEP-30-02 免費考試真題 (Q30-Q35):

問題 #30
Assuming that the following assignment has been successfully executed:



```
the_list - ["1", 1, 1.]
```

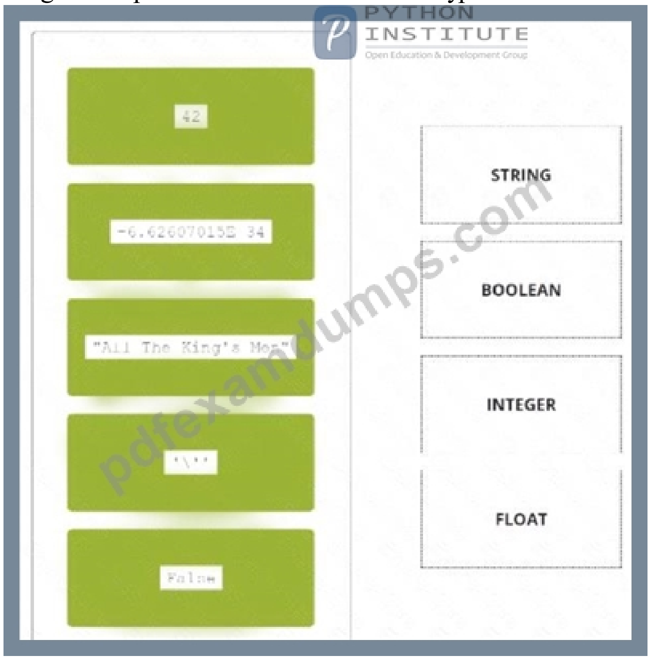
Which of the following expressions evaluate to True? (Select two expressions.)

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

答案：B,C

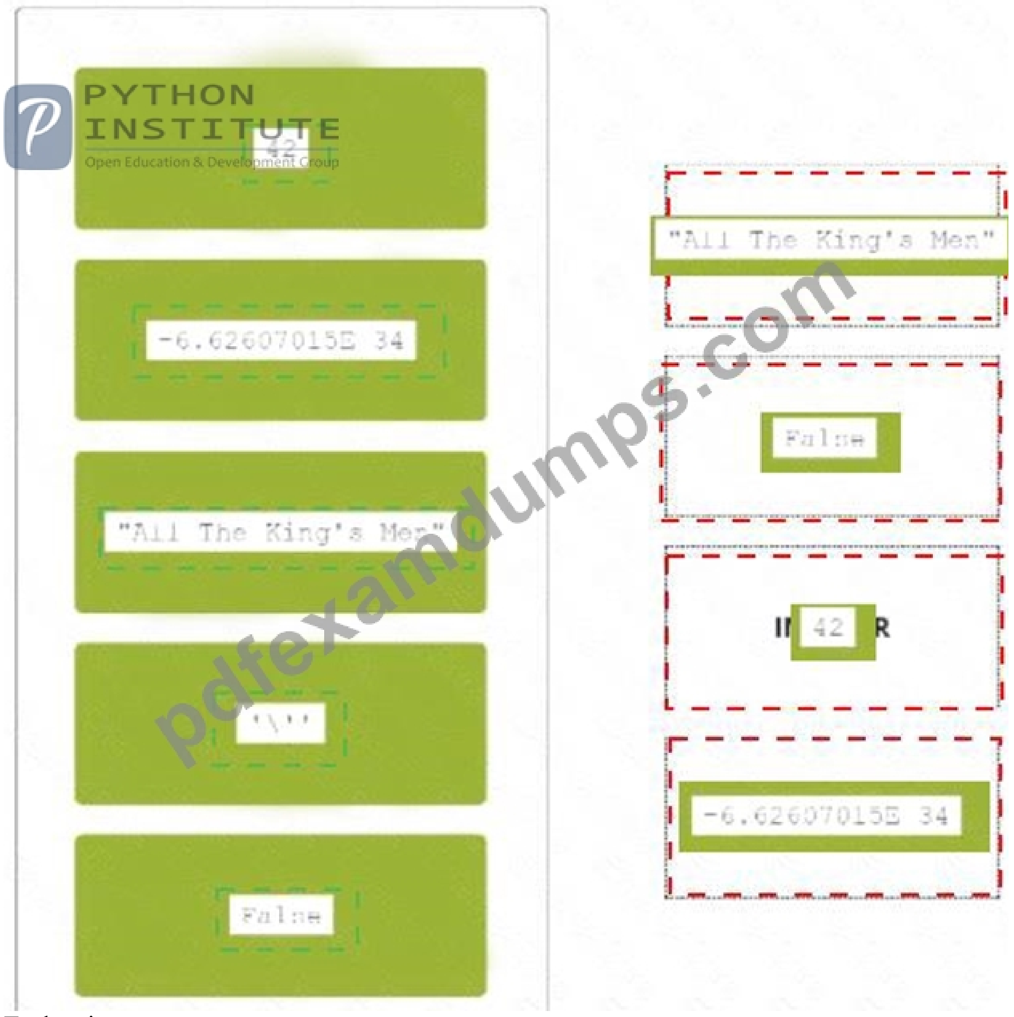
解題說明：
The code snippet that you have sent is assigning a list of four values to a variable called "the_list". The code is as follows:
the_list = ['1', 1, 1, 1]
The code creates a list object that contains the values '1', 1, 1, and 1, and assigns it to the variable "the_list".
The list can be accessed 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

問題 #31
Drag and drop the literals to match their data type names.



答案:

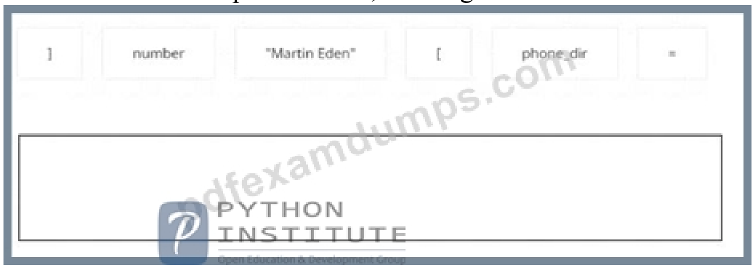
解題說明:



Explanation

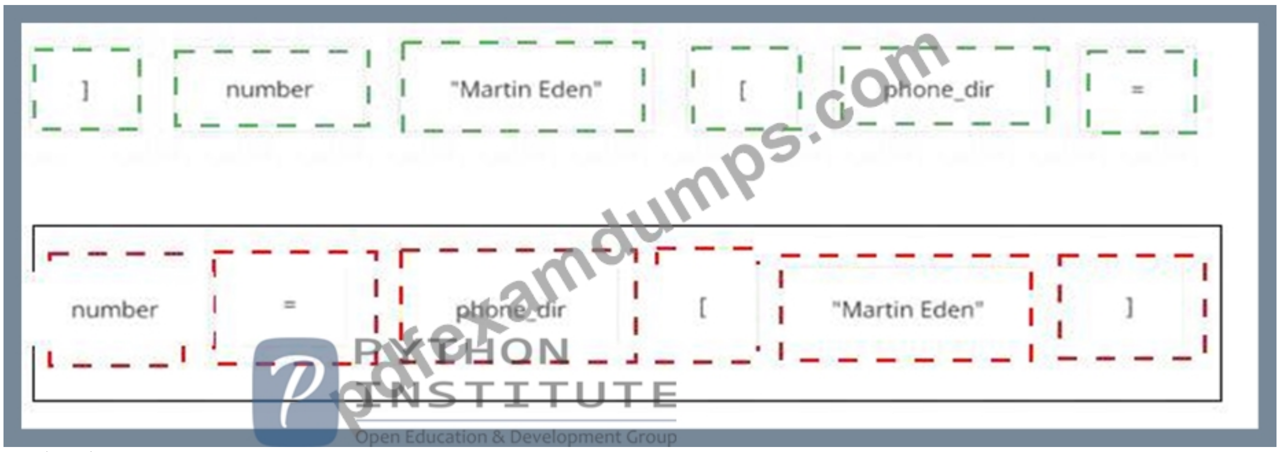
One possible way to drag and drop the literals to match their data type names is:
STRING: "All The King's Men"
BOOLEAN: False
INTEGER: 42
FLOAT: -6.62607015E-34
A literal is a value that is written exactly as it is meant to be interpreted by the Python interpreter. A data type is a category of values 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]

問題 #32
Assuming that the phonc_dir dictionary contains namemumber pairs, arrange the code boxes to create a valid line of code which retrieves Martin Eden's phone number, and assigns it to the number variable.

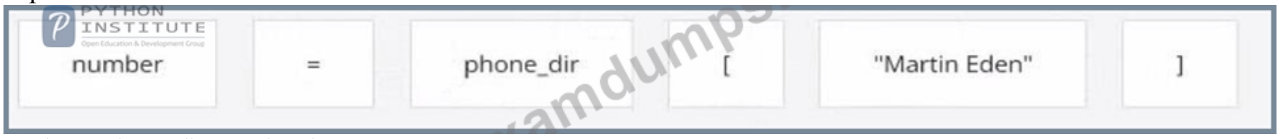


答案:

解題說明:



Explanation



number = phone_dir["Martin Eden"]
This code uses the square brackets notation to access the value associated with the key "Martin Eden" in the phone_dir dictionary. The value is then assigned to the variable number. A dictionary is a data structure that stores key-value pairs, where each key is

unique and can be used to retrieve its corresponding value. You can find more information about dictionaries in Python in the following references:
[Python Dictionaries - W3Schools]
[Python Dictionary (With Examples) - Programiz]
[5.5. Dictionaries - How to Think Like a Computer Scientist ...]

問題 #33
What is true about exceptions and debugging? (Select two answers.)

- A. One try-except block may contain more than one except branch.
- B. The default (anonymous) except branch cannot be the last branch in the try-except block.
- C. A tool that allows you to precisely trace program execution is called a debugger.
- D. If some Python code is executed without errors, this proves that there are no errors in it.

答案： A,C

解題說明：
Explanation
Exceptions and debugging are two important concepts in Python programming that are related to handling and preventing errors. Exceptions are errors that occur when the code cannot be executed properly, such as syntax errors, type errors, index errors, etc. Debugging is the process of finding and fixing errors in the code, using various tools and techniques. Some of the facts about exceptions and debugging are:
A tool that allows you to precisely trace program execution is called a debugger. A debugger is a program that can run another program step by step, inspect the values of variables, set breakpoints, evaluate expressions, etc. A debugger can help you find the source and cause of an error, and test possible solutions. Python has a built-in debugger module called pdb, which can be used from the command line or within the code. There are also other third-party debuggers available for Python, such as PyCharm, Visual Studio Code, etc12 If some Python code is executed without errors, this does not prove that there are no errors in it. It only means that the code did not encounter any exceptions that would stop the execution. However, the code may still have logical errors, which are errors that cause the code to produce incorrect or unexpected results. For example, if you write a function that is supposed to calculate the area of a circle, but you use the wrong formula, the code may run without errors, but it will give you the wrong answer. Logical errors are harder to detect and debug than syntax or runtime errors, because they do not generate any error messages. You have to test the code with different inputs and outputs, and compare them with the expected results34 One try-except block may contain more than one except branch. A try-except block is a way of handling exceptions in Python, by using the keywords try and except. The try block contains the code that may raise an exception, and the except block contains the code that will execute if an exception occurs. You can have multiple except blocks for different types of exceptions, or for different actions to take. For example, you can write a try-except block like this:
try: # some code that may raise an exception except ValueError: # handle the ValueError exception except ZeroDivisionError: # handle the ZeroDivisionError exception except: # handle any other exception This way, you can customize the error handling for different situations, and provide more informative messages or alternative solutions5 The default (anonymous) except branch can be the last branch in the try-except block. The default except branch is the one that does not specify any exception type, and it will catch any exception that is not handled by the previous except branches. The default except branch can be the last branch in the try-except block, but it cannot be the first or the only branch. For example, you can write a try-except block like this:
try: # some code that may raise an exception except ValueError: # handle the ValueError exception except: # handle any other exception This is a valid try-except block, and the default except branch will be the last branch. However, you cannot write a try-except block like this:
try: # some code that may raise an exception except: # handle any exception This is an invalid try-except block, because the default except branch is the only branch, and it will catch all exceptions, even those that are not errors, such as KeyboardInterrupt or SystemExit. This is considered a bad practice, because it may hide or ignore important exceptions that should be handled differently or propagated further. Therefore, you should always specify the exception types that you want to handle, and use the default except branch only as a last resort5 Therefore, the correct answers are A. A tool that allows you to precisely trace program execution is called a debugger. and C. One try-except block may contain more than one except branch.

問題 #34
What is true about exceptions in Python? (Select two answers.)

- A. Python's philosophy encourages developers to make all possible efforts to protect the program from the occurrence of an exception.
- B. Not more than one except branch can be executed inside one try-except block.

- C. According to Python terminology, exceptions are raised
- D. According 10 Python terminology, exceptions are thrown

答案： B,C

問題 #35
......

PDFExamDumps題供了不同培訓工具和資源來準備Python Institute的PCEP-30-02考試，編制指南包括課程，實踐的檢驗，測試引擎和部分免費PDF下載，我們的考題及答案反應的問題問Python Institute的PCEP-30-02考試。

最新 PCEP-30-02考題: https://www.pdfexamdumps.com/PCEP-30-02_valid-braindumps.html

從Google Drive中免費下載最新的PDFExamDumps PCEP-30-02 PDF版考試題庫：https://drive.google.com/open?id=1KCehJoiGq9EdhiqP6Tg7R20-aIYpUzQG