Free Questions for PCPP-32-101 by go4braindumps

Shared by Oconnor on 24-05-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

Select the true statements related to PEP 8 programming recommendations for code writing. (Select two answers:)

Options:

A- You should use the not ... is operator (e.g. if not spam is None:), rather than the is not operator (e.g. if spam is not None:), to increase readability.

B- You should make object type comparisons using the ismstanceQ method (e.g. if isinstance (obj, int) :) instead of comparing types directly (eg if type(obj) is type(i)).

C- You should write code in a way that favors the CPython implementation over PyPy, Cython. and Jython.

D- You should not write string literals that rely on significant trailing whitespaces as they may be visually indistinguishable, and certain editors may trim them

Answer:

B, D

Explanation:

The two true statements related to PEP 8 programming recommendations for code writing areOption BandOption D.

Option B is true because PEP 8 recommends making object type comparisons using theisinstance()method instead of comparing types directly1.

Option D is true because PEP 8 recommends not writing string literals that rely on significant trailing whitespaces as they may be visually indistinguishable, and certain editors may trim them1.

Question 2

Question Type: MultipleChoice

Select the true statement related to PEP 257.

Options:

A- String literals that occur immediately after another docstring are called attribute docstrings.

B- Attribute docstrings and Additional docstrings are two types of extra docstrings that can be extracted by software tools.

C- String literals that occur in places other than the first statement in a module, function, or class definition can act as documentation They are recognized by the Python bytecode compiler and are accessible as runtime object attributes

D- String literals that occur immediately after a simple assignment at the top level of a module are called complementary docstrings

Answer:

B

Explanation:

The true statement related to PEP 257 isOption B. According to PEP 257, string literals occurring elsewhere in Python code may also act as documentation. They are not recognized by the Python bytecode compiler and are not accessible as runtime object attributes (i.e. not assigned todoc), but two types of extra docstrings may be extracted by software tools: String literals occurring immediately after a simple assignment at the top level of a module, class, orinitmethod are called "attribute docstrings".String literals occurring immediately after another docstring are called "additional docstrings"1.

Question 3

Question Type: MultipleChoice

Look at the following examples of comments and docstrings in Python Select the ones that are useful and compliant with PEP 8 recommendations (Select the two best answers.)

A)

```
def area_price(area, price=1.25):
    """Calculate the area in square meters.
       Keyword arguments:
       area -- the land area of the slot
       price -- price per sq/m (default 1.25)"""
    ...
```

B)

```
def area_price(area, price=2.25):
    """Calculate the area in square meters.


    Keyword arguments:
    area -- the land area of the slot
    price -- price per sq/m (default 2.25)
    """
    ...
```

C)

```
# Example that illustrates creating
# a two-element list, and printing
# the list contents to the screen.

my_list = [a, b]
print(my_list)
```

D)

```
price = price + 1  # Decrement price by one to compensate for loss.
```

Options:

A- Option A

B- Option B

C- Option C

D- Option D

Answer:

B, D

Explanation:

According to PEP 8 recommendations, the two best options areOption BandOption D.

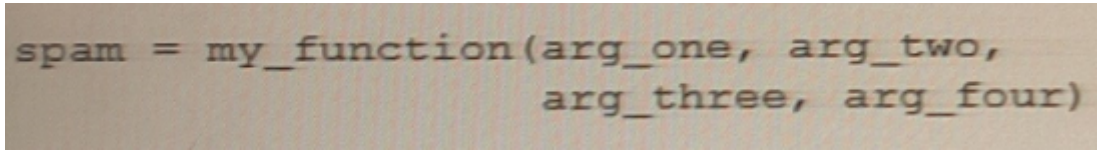
Option B follows PEP 8's suggestion that all lines should be limited to 79 characters and for longer blocks of text like docstrings or comments, the length should be limited to 72 characters1. Option D follows PEP 8's conventions for writing good documentation strings (a.k.a. "docstrings") which are immortalized in PEP 257.It suggests writing docstrings for all public modules, functions, classes, and methods2.

Question 4

Question Type: MultipleChoice

Which of the following examples using line breaks and different indentation methods are compliant with PEP 8 recommendations? (Select two answers.)

A)



B)

```
eggs = (1, 2, 3,
    4, 5, 6)
```

C)

```
my_list = [
    1, 2, 3,
    4, 5, 6,
    ]
```

```
foo = my_function
(
arg_one, arg_two,
arg_three, arg_four
)
```

Options:

A- Option A

B- Option B

C- Option C

D- Option D

Answer:

B, D

Explanation:

The correct answers areB. Option BandD. Option D. Both options B and D are compliant with PEP 8 recommendations for line breaks and indentation. PEP 8 recommends using 4 spaces per indentation level and breaking lines before binary operators. In option B, the arguments to theprintfunction are aligned with the opening delimiter, which is another acceptable way to format long lines according to PEP 8. In option D, the second line is indented by 4 spaces to distinguish it from the next logical line.

Question 5

Question Type: **MultipleChoice**

What is a___traceback___?

(Select two answers)

Options:

A- An attribute owned by every exception object

B- A special method delivered by the traceback module to retrieve a full list of strings describing the traceback

C- An attribute that is added to every object when the traceback module is imported

D- An attribute that holds interesting information that is particularly useful when the programmer wants to store exception details in other objects

Answer:

A, D

Explanation:

The correct answers areA. An attribute owned by every exception objectandD. An attribute that holds interesting information that is particularly useful when the programmer wants to store exception details in other objects. A traceback is an attribute of an exception object that contains a stack trace representing the call stack at the point where the exception was raised. The traceback attribute holds information about the sequence of function calls that led to the exception, which can be useful for debugging and error reporting.

Question 6

Question Type: MultipleChoice

What does the term deserialization mean? Select the best answer.

Options:

A- It is a process of creating Python objects based on sequences of bytes.

B- It is a process of assigning unique identifiers to every newly created Python object

C- It is another name for the data transmission process

D- It is a process of converting the structure of an object into a stream of bytes

Answer:

A

Explanation:

Deserialization is the process of converting data that has been serialized

For example, if you have a Python objectmy_objand you want to serialize it to a JSON string, you might do something like this:

import json

serialized_obj = json.dumps(my_obj)

To deserialize the JSON string back into a Python object, you would use thejson.loads()method:

deserialized_obj = json.loads(serialized_obj)

This would convert the JSON string back into its original Python object form.

Official Python Documentation on Serialization:https://docs.python.org/3/library/pickle.html#module-pickle

Real Python Tutorial on Serialization and Deserialization in Python:https://realpython.com/python-serialization/

Deserialization is the process of converting a sequence of bytes, such as a file or a network message, into a Python object. This is the opposite of serialization, which is the process of converting a Python object into a sequence of bytes for storage or transmission.

Question 7

Question Type: **MultipleChoice**

What is true about type in the object-oriented programming sense?

Options:

A- It is the bottommost type that any object can inherit from.

B- It is a built-in method that allows enumeration of composite objects

C- It is the topmost type that any class can inherit from

D- It is an object used to instantiate a class

Answer:

C

Explanation:

In Python,typeis the built-in metaclass that serves as the base class for all new-style classes. All new-style classes in Python, including built-in types likeintandstr, are instances of thetypemetaclass and inherit from it.

Question 8

Question Type: **MultipleChoice**

What is a static method?

Options:

A- A method that works on the class itself

B- A method decorated with the @method trait

C- A method that requires no parameters referring to the class itself

D- A method that works on class objects that are instantiated

Answer:

C

Explanation:

A static method is a method that belongs to a class rather than an instance of the class. It is defined using the@staticmethoddecorator and does not take aselforclsparameter. Static methods are often used to define utility functions that do not depend on the state of an instance or the class itself.

Question 9

Question Type: MultipleChoice

Select the true statement about the___name___attribute.

Options:

A- ___name___is a special attribute, which is inherent for both classes and instances, and it contains information about the class to which a class instance belongs.

B- ___name is a special attribute, which is inherent for both classes and instances, and it contains a dictionary of object attributes.

C- __name___is a special attribute, which is inherent for classes and it contains information about the class to which a class instance belongs.

D- __name___is a special attribute, which is inherent for classes, and it contains the name of a class.

Answer:

D

Explanation:

The true statement about the__name__attribute isD.nameis a special attribute, which is inherent for classes, and it contains the name of a class. The__name__attribute is a special attribute of classes that contains the name of the class as a string.

The__name__attribute is a special attribute in Python that is available for all classes, and it contains the name of the class as a string. The__name__attribute can be accessed from both the class and its instances using the dot notation.

Official Python documentation on Classes:https://docs.python.org/3/tutorial/classes.html#class-objects

Question 10

Question Type: MultipleChoice

Which sentence about the property decorator is false?

Options:

A- The property decorator should be defined after the method that is responsible for setting an encapsulated attribute.

B- The @property decorator designates a method which is responsible for returning an attribute value

C- The property decorator marks the method whose name will be used as the name of the instance attribute

D- The property decorator should be defined before the methods that are responsible for setting and deleting an encapsulated attribute

Answer:

A

Explanation:

The@propertydecorator should be defined after the method that is responsible for setting an encapsulated attribute is a false sentence. In fact, the@propertydecorator should be defined before the method that is used to set the attribute value. The@propertydecorator and the setter and deleter methods work together to create an encapsulated attribute, which is used to provide control over the attribute's value.

Official Python documentation on Property:https://docs.python.org/3/library/functions.html#property

The@propertydecorator is used to designate a method as a getter for an instance attribute. The method decorated with@propertyshould be defined before any setter or deleter methods for the same attribute.

Question 11

Question Type: **MultipleChoice**

Which function or operator should you use to obtain the answer True or False to the question: "Do two variables refer to the same object?"

Options:

A- The = operator

B- The isinstanceO function

C- The id () function

D- The is operator

Answer:

D

Explanation:

To test whether two variables refer to the same object in memory, you should use theisoperator. Theisoperator returnsTrueif the two variables point to the same object in memory, andFalseotherwise.

For example:

a = [1, 2, 3]

b = a

c = [1, 2, 3]

print(a is b) # True

print(a is c) # False

In this example,aandbrefer to the same list object in memory, soa is breturnsTrue. On the other hand,aandcrefer to two separate list objects with the same values, soa is creturnsFalse.

Official Python documentation on Comparisons:https://docs.python.org/3/reference/expressions.html#not-in

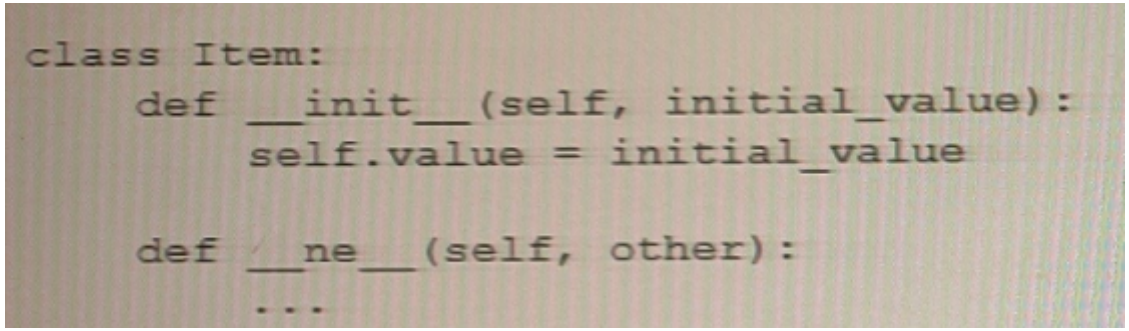
Official Python documentation on Identity comparisons:https://docs.python.org/3/reference/expressions.html#is

Theisoperator is used to test whether two variables refer to the same object in memory. If two variablesxandyrefer to the same object, the expressionx is ywill evaluate toTrue. Otherwise, it will evaluate toFalse.

Question 12

Question Type: MultipleChoice

Analyze the code and choose the best statement that describes it.



```
class Item:
    def __init__(self, initial_value):
        self.value = initial_value

    def __ne__(self, other):
        ...
```

Options:

A- ___ne___() is not a built-in special method

B- The code is erroneous

C- The code is responsible for the support of the negation operator e.g. a = - a.

D- The code is responsible for the support of the inequality operator i.e. i =

Answer:

D

Explanation:

The correct answer isD. The code is responsible for the support of the inequality operator i.e. i != j. In the given code snippet, the__ne__method is a special method that overrides the behavior of the inequality operator!=for instances of theMyClassclass. When the inequality operator is used to compare two instances ofMyClass, the__ne__method is called to determine whether the two instances are unequal.