DUMPSBOSS.

PCPP - Certified Professional in Python Programming 1

Python Institute PCPP-32-101

Version Demo

Total Demo Questions: 10

Total Premium Questions: 45

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

QUESTION NO: 1

Select the true statements about the sqirte3 module. (Select two answers.)

A. The sqlite3 module provides an interface compliant with the DB-API 2.0.
The sqlite3 module in python provides an interface compliant to the DB-API 2.0. Thus, it follows a standard performance metric that allows for consistency in database programming with python.

B. The special name memory is used to create a database in RAM.
The special name 'memory' is used to create a database in RAM using the sqlite3 module. Thus, when you use it as the name of the database file while opening a connection, it creates a temporary database that exists only in memory.
Reference: Official Python documentation on sqlite3: https://docs.python.org/3/library/sqlite3.html

C. The sqhte3 module does not support transactions.

D. The fetchall method returns an empty list when no rows are available

ANSWER: A B

Explanation:

A. The sqlite3 module in python provides an interface compliant to the DB-API 2.0. Thus, it follows a standard performance metric that allows for consistency in database programming with python.

B. The special name 'memory' is used to create a database in RAM using the sqlite3 module. Thus, when you use it as the name of the database file while opening a connection, it creates a temporary database that exists only in memory.

Reference: Official Python documentation on sqlite3: https://docs.python.org/3/library/sqlite3.html

QUESTION NO: 2

Look at the following code snippets and decide which ones follow PEP 8 recommendations for whitespaces in expressions and statements (Select two answers.)

A)

No whitespace immediately before the opening parenthesis that starts the list of arguments of a function call:

```
# Correct:
foo(5)
# Wrong:
foo (5)
```

B)

A whitespace immediately before a comma, semicolon, and colon:

```
# Correct:
if x == 2 : print x , y ; x , y = y , x
# Wrong:
if x == 2: print x, y; x, y = y, x
```

C)

No whitespace between a trailing comma and a following closing parenthesis:

```
# Correct:
spam = (1,)
# Wrong:
spam = (1, )
```

D)

A whitespace immediately after the opening parenthesis that starts indexing or slicing:

```
# Correct:
my_dict ['key'] = my_list [index]
# Wrong:
my_dict['key'] = my_list[index]
```

A. Option A

B. Option B

C. Option C

D. Option D

ANSWER: A C

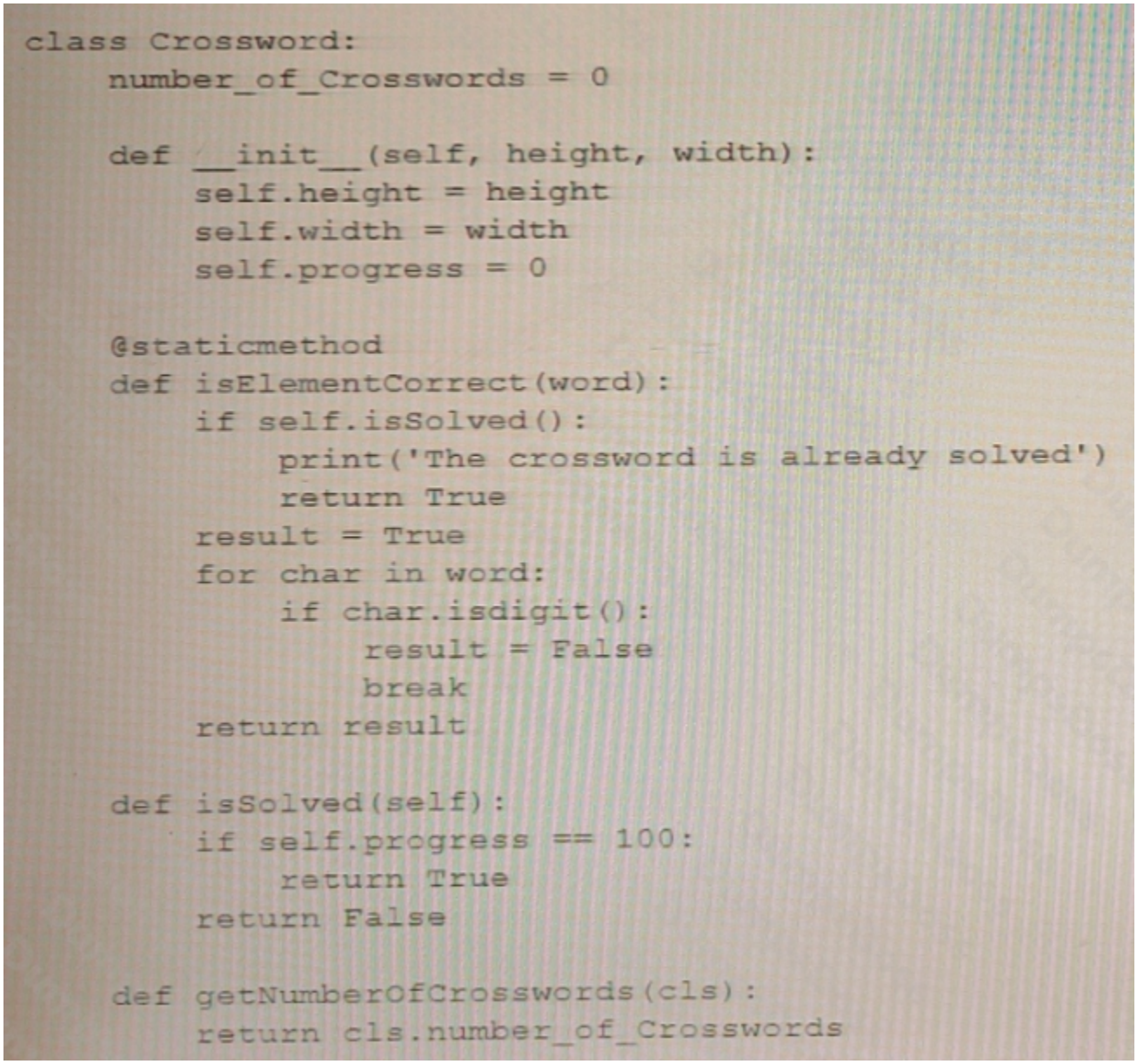
Explanation:

Option A is true because PEP 8 recommends avoiding extraneous whitespace immediately inside parentheses, brackets or braces 1.

Option C is true because PEP 8 recommends avoiding extraneous whitespace between a trailing comma and a following close parenthesis 1.

QUESTION NO: 3

Analyze the following snippet and decide whether the code is correct and/or which method should be distinguished as a class method.



```
class Crossword:
    number_of_Crosswords = 0

    def __init__(self, height, width):
        self.height = height
        self.width = width
        self.progress = 0

    @staticmethod
    def isElementCorrect(word):
        if self.isSolved():
            print('The crossword is already solved')
            return True
        result = True
        for char in word:
            if char.isdigit():
                result = False
                break
        return result

    def isSolved(self):
        if self.progress == 100:
            return True
        return False

    def getNumberOfCrosswords(cls):
        return cls.number_of_Crosswords
```

A. There is only one initializer, so there is no need for a class method.

B. The getNumberofCrosswords () method should be decorated With @classmethod.
The getNumberofCrosswords() method should be decorated with @classmethod.
This is because the getNumberofCrosswords() method is intended to access the class-level variable numberofcrosswords, but it is defined as an instance method, which requires an instance of the class to be created before it can be called. To make it work as a class-level method, you can define it as a class method by adding the @classmethod decorator to the function.
Here's an example of how to define getNumberofCrosswords() as a class method:
class Crossword:
numberofcrosswords = 0

def __init__(self, author, title):
self.author = author
self.title = title
Crossword.numberofcrosswords += 1

@classmethod
def getNumberofCrosswords(cls):
return cls.numberofcrosswords
In this example, getNumberofCrosswords() is defined as a class method using the @classmethod decorator, and the cls parameter is used to access the class-level variable numberofcrosswords.
Reference:

C. The code is erroneous.

D. The gexNumberOfcrosswords () and issrived methods should be decorated with @classzoechod.

ANSWER: B

Explanation:

The correct answer is B. The getNumberofCrosswords() method should be decorated with @classmethod. In the given code snippet, the getNumberofCrosswords method is intended to be a class method that returns the value of the numberofcrosswords class variable. However, the method is not decorated with the @classmethod decorator and does not take a cls parameter representing the class itself. To make getNumberofCrosswords a proper class method, it should be decorated with @classmethod and take a cls parameter as its first argument.

B. The getNumberofCrosswords() method should be decorated with @classmethod.

This is because the getNumberofCrosswords() method is intended to access the class-level variable numberofcrosswords, but it is defined as an instance method, which requires an instance of the class to be created before it can be called. To make it work as a class-level method, you can define it as a class method by adding the @classmethod decorator to the function.

Here's an example of how to define getNumberofCrosswords() as a class method:

class Crossword:

numberofcrosswords = 0

def __init__(self, author, title):

self.author = author

self.title = title

Crossword.numberofcrosswords += 1

@classmethod

def getNumberofCrosswords(cls):

return cls.numberofcrosswords

In this example, getNumberofCrosswords() is defined as a class method using the @classmethod decorator, and the cls parameter is used to access the class-level variable numberofcrosswords.

Reference:

QUESTION NO: 4

What is true about the unbind () method? (Select two answers.)

A. It is invoked from within the events object

B. It is invoked from within a widget's object

C. It needs a widget's object as an argument

D. It needs the event name as an argument

ANSWER: B D

Explanation:

Option B is true because the unbind() method is invoked from within a widget's object 1.

Option D is true because the unbind() method needs the event name as an argument 1.

The unbind() method in Tkinter is used to remove a binding between an event and a function. It can be invoked from within a widget's object when a binding is no longer needed. The method requires the event name as an argument to remove the binding for that specific event. For example:

button = tk.Button(root, text="Click me")

button.bind("", callback_function) # bind left mouse click event to callback_function

button.unbind("") # remove the binding for the left mouse click event

QUESTION NO: 5

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

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:

Answer- Deserialization is the process of converting data that has been serialized or encoded in a specific format, back into its original form as an object or a data structure in memory. In Python, this typically involves creating Python objects based on sequences of bytes that have been serialized using a protocol such as JSON, Pickle, or YAML.

For example, if you have a Python object my_obj and 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 the json.loads() method:

deserialized_obj = json.loads(serialized_obj)

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

Reference:

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 NO: 6

Select the true statement related to PEP 257.

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 is Option 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 to doc), 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, or init method are called “attribute docstrings”. String literals occurring immediately after another docstring are called “additional docstrings”1.

QUESTION NO: 7

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.
```

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 are Option B and Option 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 NO: 8

Analyze the following snippet and select the statement that best describes it.

```
def f1(*arg, **args):
    pass
```

A. The code is syntactically correct despite the fact that the names of the function parameters do not follow the naming convention

B. The *arg parameter holds a list of unnamed parameters
The *args parameter holds a list of unnamed parameters, while the **kwargs parameter holds a dictionary of named parameters.
Reference:

The arg parameter holds a list of unnamed parameters. In the given code snippet, the f1 function takes two arguments: *arg and **kwarg. The *arg syntax in the function signature is used to pass a variable number of non-keyword (positional) arguments to the function. Inside the function, arg is a tuple containing the positional arguments passed to the function. The **kwarg syntax in the function signature is used to pass a variable number of keyword arguments to the function. Inside the function, kwarg is a dictionary containing the keyword arguments passed to the function.

C. The code is missing a placeholder for unnamed parameters.

D. The code is syntactically incorrect - the function should be defined as def f1 (*args, **kwargs) :

ANSWER: B

Explanation:

The provided code snippet defines a function f1 that accepts variable-length arguments using the *args and **kwargs syntax. The *args parameter allows for an arbitrary number of unnamed arguments to be passed to the function as a tuple, while the **kwargs parameter allows for an arbitrary number of named arguments to be passed to the function as a dictionary.

Therefore, the correct statement that best describes the code is:

B. The *args parameter holds a list of unnamed parameters, while the **kwargs parameter holds a dictionary of named parameters.

Reference:

The arg parameter holds a list of unnamed parameters. In the given code snippet, the f1 function takes two arguments: *arg and **kwarg. The *arg syntax in the function signature is used to pass a variable number of non-keyword (positional) arguments to the function. Inside the function, arg is a tuple containing the positional arguments passed to the function. The **kwarg syntax in the function signature is used to pass a variable number of keyword arguments to the function. Inside the function, kwarg is a dictionary containing the keyword arguments passed to the function.

QUESTION NO: 9

Select the true statements about sockets. (Select two answers)

A. A socket is a connection point that enables a two-way communication between programs running in a network.
A socket is a connection point that enables a two-way communication between programs running in a network.
This statement is true because a socket is a software structure that serves as an endpoint for sending and receiving data across a network. A socket is defined by an application programming interface (API) for the networking architecture, such as TCP/IP. A socket can be used to establish a communication channel between two programs running on the same or different network nodes12.

B. A socket is always the secure means by which computers on a network can safely communicate, without the risk of exposure to an attack
A socket is always the secure means by which computers on a network can safely communicate, without the risk of exposure to an attack.
This statement is false because a socket by itself does not provide any security or encryption for the data transmitted over the network. A socket can be vulnerable to various types of attacks, such as eavesdropping, spoofing, hijacking, or denial-of-service. To ensure secure communication, a socket can use additional protocols or mechanisms, such as SSL/TLS, SSH, VPN, or firewall3.

C. A socket is a connection point that enables a one-way communication only between remote processes
A socket is a connection point that enables a one-way communication only between remote processes.
This statement is false because a socket can enable both one-way and two-way communication between processes running on the same or different network nodes. A socket can be used for connection-oriented or connectionless communication, depending on the type of protocol used. For example, TCP is a connection-oriented protocol that provides reliable and bidirectional data transfer, while UDP is a connectionless protocol that provides unreliable and unidirectional data transfer12.

D. A socket can be used to establish a communication endpoint for processes running on the same or different machines.
A socket can be used to establish a communication endpoint for processes running on the same or different machines.
This statement is true because a socket can be used for inter-process communication (IPC) within a single machine or across different machines on a network. A socket can use different types of addresses to identify the processes involved in the communication, such as IP address and port number for network sockets, or file name or path for Unix domain sockets12.
References:
1: https://en.wikipedia.org/wiki/Network_socket 2: https://www.geeksforgeeks.org/socket-in-computer-network/ 3: https://www.tutorialspoint.com/what-is-a-network-socket-computer-networks

ANSWER: A D

Explanation:

A. A socket is a connection point that enables a two-way communication between programs running in a network.

This statement is true because a socket is a software structure that serves as an endpoint for sending and receiving data across a network. A socket is defined by an application programming interface (API) for the networking architecture, such as TCP/IP. A socket can be used to establish a communication channel between two programs running on the same or different network nodes12.

B. A socket is always the secure means by which computers on a network can safely communicate, without the risk of exposure to an attack.

This statement is false because a socket by itself does not provide any security or encryption for the data transmitted over the network. A socket can be vulnerable to various types of attacks, such as eavesdropping, spoofing, hijacking, or denial-of-service. To ensure secure communication, a socket can use additional protocols or mechanisms, such as SSL/TLS, SSH, VPN, or firewall3.

C. A socket is a connection point that enables a one-way communication only between remote processes.

This statement is false because a socket can enable both one-way and two-way communication between processes running on the same or different network nodes. A socket can be used for connection-oriented or connectionless communication, depending on the type of protocol used. For example, TCP is a connection-oriented protocol that provides reliable and bidirectional data transfer, while UDP is a connectionless protocol that provides unreliable and unidirectional data transfer12.

D. A socket can be used to establish a communication endpoint for processes running on the same or different machines.

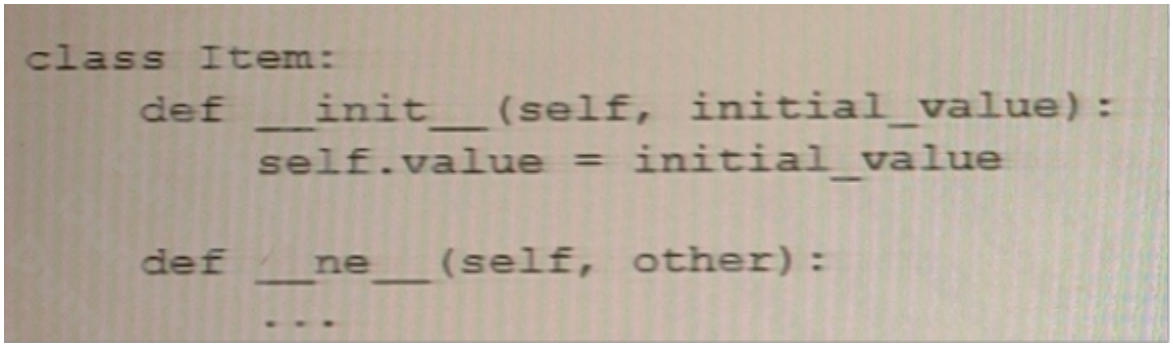
This statement is true because a socket can be used for inter-process communication (IPC) within a single machine or across different machines on a network. A socket can use different types of addresses to identify the processes involved in the communication, such as IP address and port number for network sockets, or file name or path for Unix domain sockets12.

References:

1: https://en.wikipedia.org/wiki/Network_socket 2: https://www.geeksforgeeks.org/socket-in-computer-network/ 3: https://www.tutorialspoint.com/what-is-a-network-socket-computer-networks

QUESTION NO: 10

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



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 is D. 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 the MyClass class. When the inequality operator is used to compare two instances of MyClass, the __ne__ method is called to determine whether the two instances are unequal.