DUMPSARENA

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://dumpsarena.co

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

QUESTION NO: 1

Which sentence about the ©property decorator is false?

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

Reference:

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

QUESTION NO: 2

Select the true statements about the json.-dumps () function. (Select two answers.)

A. It returns a JSON string.
It returns a JSON string.
This statement is true because the json.dumps () function takes a Python object as its argument and returns a JSON-formatted string that represents the object. For example, json.dumps ([1, 2, 3]) returns '[1, 2, 3]'.

B. It returns a Python entity.

C. It takes a JSON string as its argument

D. It takes Python data as its argument.
It takes Python data as its argument.
This statement is true because the json.dumps () function accepts any Python object that can be serialized into JSON, such as lists, dictionaries, strings, numbers, booleans, or None. For example, json.dumps ({"name": "Alice", "age": 25}) returns '{"name": "Alice", "age": 25}'.

ANSWER: A D

Explanation:

The json.dumps() function is used to convert a Python object into a JSON string 1. It takes Python data as its argument, such as a dictionary or a list, and returns a JSON string.

Reference: Official Python documentation on json.dumps(): https://docs.python.org/3/library/json.html#json.dumps

A. It returns a JSON string.

This statement is true because the json.dumps () function takes a Python object as its argument and returns a JSON-formatted string that represents the object. For example, json.dumps ([1, 2, 3]) returns '[1, 2, 3]'.

D. It takes Python data as its argument.

This statement is true because the json.dumps () function accepts any Python object that can be serialized into JSON, such as lists, dictionaries, strings, numbers, booleans, or None. For example, json.dumps ({"name": "Alice", "age": 25}) returns '{"name": "Alice", "age": 25}'.

QUESTION NO: 3

Which of the following constants will be used if you do riot define the quoting argument in the writer method provided by the csv module?

A. csv.QUOTE_MINIMAL

B. csv.QUOTE_NONE

C. svQUOTE_ALL

D. csv.QUOTE_NONNUMERIC

ANSWER: A

Explanation:

If you do not define the quoting argument in the writer method provided by the csv module, the default quoting behavior is set to QUOTE_MINIMAL. This means that fields containing special characters such as the delimiter or newline character will be quoted, while fields that do not contain special characters will not be quoted.

Reference: Official Python documentation on the csv module: https://docs.python.org/3/library/csv.html

QUESTION NO: 4

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: 5

Select the true statements about the sqlite3 module. (Select two answers.)

A. The fetchalt method returns None when no rows are available

B. The execute method allows you to perform several queries at once

C. The execute method is provided by the Cursor class
The execute method is provided by the Cursor class
This statement is true because the execute method is one of the methods of the Cursor class in the sqlite3 module. The Cursor class represents an object that can execute SQL statements and fetch results from a database connection. The execute method takes an SQL query as an argument and executes it against the database. For example, cur = conn.cursor (); cur.execute ("SELECT * FROM table") creates and executes a cursor object that selects all rows from a table.

D. The fetchone method returns None when no rows are available
The fetchone method returns None when no rows are available
This statement is true because the fetchone method is another method of the Cursor class in the sqlite3 module. The fetchone method fetches the next row of a query result set and returns it as a single tuple or None if no more rows are available. For example, row = cur.fetchone () fetches and returns one row from the cursor object or None if there are no more rows.

ANSWER: C D

Explanation:

C. The execute method is provided by the Cursor class

This statement is true because the execute method is one of the methods of the Cursor class in the sqlite3 module. The Cursor class represents an object that can execute SQL statements and fetch results from a database connection. The execute method takes an SQL query as an argument and executes it against the database. For example, cur = conn.cursor (); cur.execute ("SELECT * FROM table") creates and executes a cursor object that selects all rows from a table.

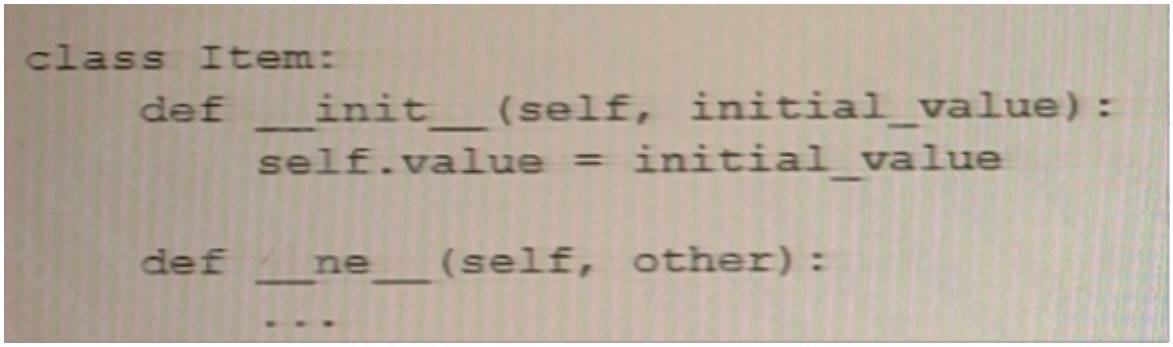
D. The fetchone method returns None when no rows are available

This statement is true because the fetchone method is another method of the Cursor class in the sqlite3 module. The fetchone method fetches the next row of a query result set and returns it as a single tuple or None if no more rows are

available. For example, row = cur.fetchone () fetches and returns one row from the cursor object or None if there are no more rows.

QUESTION NO: 6

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):
        ...
```

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.

QUESTION NO: 7

In the JSON processing context, the term serialization:

A. names a process in which Python data is turned into a JSON string.

B. names a process in which a JSON string is turned into Python data.

C. refers to nothing, because there is no such thing as JSON serialization.

D. names a process in which a JSON string is remodeled and transformed into a new JSON string

ANSWER: A

Explanation:

In the JSON processing context, the term serialization: A. names a process in which Python data is turned into a JSON string.

Reference: Official Python documentation on json: https://docs.python.org/3/library/json.html#json-serialization

QUESTION NO: 8

Select the true statements about the connection-oriented and connectionless types of communication. (Select two answers.)

A. In the context of TCP/IP networks, the communication side that initiates a connection is called the client, whereas the side that answers the client is called the server
In the context of TCP/IP networks, the communication side that initiates a connection is called the client, whereas the side that answers the client is called the server.
This statement is true because TCP/IP networks use a client-server model to establish connection-oriented communications. The client is the device or application that requests a service or resource from another device or application, which is called the server. The server responds to the client's request and provides the service or resource. For example, when you browse a website using a web browser, the browser acts as a client and sends a request to the web server that hosts the website. The web server acts as a server and sends back the requested web page to the browser1.

B. Connectionless communications are usually built on top of TCP
Connectionless communications are usually built on top of TCP.
This statement is false because TCP (Transmission Control Protocol) is a connection-oriented protocol that requires establishing and terminating a connection before and after sending data. Connectionless communications are usually built on top of UDP (User Datagram Protocol), which is a connectionless protocol that does not require any connection setup or teardown. UDP simply sends data packets to the destination without checking if they are received or not2.

C. Using walkie-talkies is an example of a connection-oriented communication
Using walkie-talkies is an example of a connection-oriented communication.
This statement is false because using walkie-talkies is an example of a connectionless communication. Walkie-talkies do not establish a dedicated channel or connection between the sender and receiver before transmitting data. They simply broadcast data over a shared frequency without ensuring that the receiver is ready or available to receive it. The sender does not know if the receiver has received the data or not3.

D. A phone call is an example of a connection-oriented communication
A phone call is an example of a connection-oriented communication.
This statement is true because a phone call is an example of a connection-oriented communication. A phone call requires setting up a circuit or connection between the caller and callee before exchanging voice data. The caller and callee can hear each other's voice and know if they are connected or not. The phone call also requires terminating the connection when the conversation is over4.
References:
1: https://www.techtarget.com/searchnetworking/definition/client-server 2: https://www.javatpoint.com/connection-oriented-vs-connectionless-service 3: https://en.wikipedia.org/wiki/Walkie-talkie 4: https://en.wikipedia.org/wiki/Telephone_call
A is true because in the context of TCP/IP networks, the communication side that initiates a connection is called the client, and the side that answers the client is called the server. This is the basis for establishing a connection-oriented communication.
D is true because a phone call is an example of a connection-oriented communication. Like TCP/IP, a phone call establishes a connection between two devices (in this case, two phones) before communication can occur.

A is true because in the context of TCP/IP networks, the communication side that initiates a connection is called the client, and the side that answers the client is called the server. This is the basis for establishing a connection-oriented communication.
D is true because a phone call is an example of a connection-oriented communication. Like TCP/IP, a phone call establishes a connection between two devices (in this case, two phones) before communication can occur.
B is false because connectionless communications are usually built on top of UDP, not TCP. UDP is a connectionless protocol that does not establish a connection before sending data.

C is false because using walkie-talkies is an example of a connectionless communication. Walkie-talkies do not establish a connection before communication begins, and messages are simply broadcasted to all devices within range.
Here is a sample code in Python using the socket module to create a TCP server and client to demonstrate the connection-oriented communication:
Server-side code:

```
import socket

HOST = '127.0.0.1'
PORT = 8080

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
s.bind((HOST, PORT))
s.listen()
conn, addr = s.accept()
with conn:
print('Connected by', addr)
while True:
data = conn.recv(1024)
if not data:
break
conn.sendall(data)
```

Client-side code:

```
import socket

HOST = '127.0.0.1'
PORT = 8080

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
s.connect((HOST, PORT))
s.sendall(b'Hello, world')
data = s.recv(1024)

print('Received', repr(data))
```

The server listens for incoming connections on port 8080, and when a connection is established, it prints the address of the client that has connected. The server then continuously receives data from the client and sends it back to the client until the connection is closed.
The client establishes a connection with the server and sends the message "Hello, world" encoded as bytes. It then waits for a response from the server and prints the data it receives.

ANSWER: A D

Explanation:

A. In the context of TCP/IP networks, the communication side that initiates a connection is called the client, whereas the side that answers the client is called the server.

This statement is true because TCP/IP networks use a client-server model to establish connection-oriented communications. The client is the device or application that requests a service or resource from another device or application, which is called the server. The server responds to the client's request and provides the service or resource. For example, when you browse a website using a web browser, the browser acts as a client and sends a request to the web server that hosts the website. The web server acts as a server and sends back the requested web page to the browser1.

B. Connectionless communications are usually built on top of TCP.

This statement is false because TCP (Transmission Control Protocol) is a connection-oriented protocol that requires establishing and terminating a connection before and after sending data. Connectionless communications are usually built on top of UDP (User Datagram Protocol), which is a connectionless protocol that does not require any connection setup or teardown. UDP simply sends data packets to the destination without checking if they are received or not2.

C. Using walkie-talkies is an example of a connection-oriented communication.

This statement is false because using walkie-talkies is an example of a connectionless communication. Walkie-talkies do not establish a dedicated channel or connection between the sender and receiver before transmitting data. They simply broadcast data over a shared frequency without ensuring that the receiver is ready or available to receive it. The sender does not know if the receiver has received the data or not3.

D. A phone call is an example of a connection-oriented communication.

This statement is true because a phone call is an example of a connection-oriented communication. A phone call requires setting up a circuit or connection between the caller and callee before exchanging voice data. The caller and callee can hear each other's voice and know if they are connected or not. The phone call also requires terminating the connection when the conversation is over4.

References:

1: https://www.techtarget.com/searchnetworking/definition/client-server 2: https://www.javatpoint.com/connection-oriented-vs-connectionless-service 3: https://en.wikipedia.org/wiki/Walkie-talkie 4: https://en.wikipedia.org/wiki/Telephone_call

A is true because in the context of TCP/IP networks, the communication side that initiates a connection is called the client, and the side that answers the client is called the server. This is the basis for establishing a connection-oriented communication.

D is true because a phone call is an example of a connection-oriented communication. Like TCP/IP, a phone call establishes a connection between two devices (in this case, two phones) before communication can occur.

A is true because in the context of TCP/IP networks, the communication side that initiates a connection is called the client, and the side that answers the client is called the server. This is the basis for establishing a connection-oriented communication.

D is true because a phone call is an example of a connection-oriented communication. Like TCP/IP, a phone call establishes a connection between two devices (in this case, two phones) before communication can occur.

B is false because connectionless communications are usually built on top of UDP, not TCP. UDP is a connectionless protocol that does not establish a connection before sending data.

C is false because using walkie-talkies is an example of a connectionless communication. Walkie-talkies do not establish a connection before communication begins, and messages are simply broadcasted to all devices within range.

Here is a sample code in Python using the socket module to create a TCP server and client to demonstrate the connection-oriented communication:

Server-side code:

```
import socket

HOST = '127.0.0.1'

PORT = 8080

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:

s.bind((HOST, PORT))
```

```
s.listen()

conn, addr = s.accept()

with conn:

print('Connected by', addr)

while True:

data = conn.recv(1024)

if not data:

break

conn.sendall(data)
```

Client-side code:

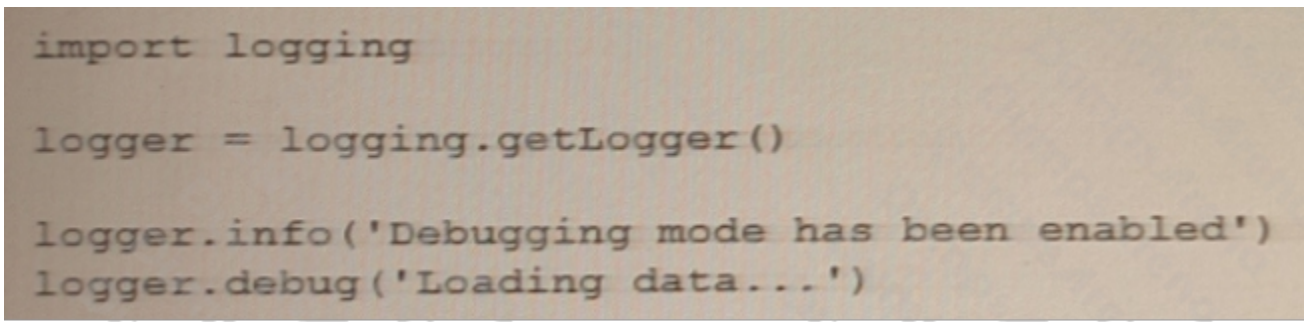
```
import socket

HOST = '127.0.0.1'

PORT = 8080

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:

s.connect((HOST, PORT))

s.sendall(b'Hello, world')

data = s.recv(1024)

print('Received', repr(data))
```

The server listens for incoming connections on port 8080, and when a connection is established, it prints the address of the client that has connected. The server then continuously receives data from the client and sends it back to the client until the connection is closed.

The client establishes a connection with the server and sends the message "Hello, world" encoded as bytes. It then waits for a response from the server and prints the data it receives.

QUESTION NO: 9

What is the result of the following code?



What is the result of the following code?

A. Nothing will be displayed

B. Loading data...

C. Debugging mode has been enabled

D. Debugging mode has been enabled Loading data...

ANSWER: B

Explanation:

This statement is true because the code uses the logging module to create a logger object and set its level to logging.INFO. The logging module provides a way of reporting events that occur during the execution of a program. The logging level determines which events are reported and which are ignored. The logging module defines five levels of severity: DEBUG, INFO, WARNING, ERROR, and CRITICAL. The lower the level, the more events are reported.

The code then uses the logger object to log two messages: one with the level logging.DEBUG and one with the level logging.INFO. The logger object only reports the messages that have a level equal or higher than its own level. Therefore, the message with the level logging.DEBUG is ignored, while the message with the level logging.INFO is reported. The default format for reporting messages is “level name: message”. Therefore, the output of the code is:

INFO: Loading data…

QUESTION NO: 10

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