"Please note that these files may not be up to date. However, the questions will help you understand the exam format and typical question patterns."

## Question: 1

Which statement is correct about DevOps?

A. DevOps is a collection of strict guidelines that promotes the project completion over all other aspects.

B. DevOps is meant to define and restrict the development and operations tools used for a project.

C. DevOps is meant to unite the development, operations, and other teams to improve project collaborations.

D. DevOps is a defined standard written and maintained by the IEEE standards group.

Answer: C

Explanation:

DevOps is a set of practices, tools, and cultural philosophies that aims to integrate and automate the processes between software development and IT operations teams. The primary goal of DevOps is to shorten the systems development life cycle and provide continuous delivery with high software quality.

Option C is correct because DevOps fundamentally focuses on breaking down the silos between development and operations teams, fostering a collaborative environment where these teams work together throughout the entire software lifecycle. This collaboration extends to other stakeholders, including quality assurance (QA), security, and more, to ensure that the product is continuously delivered and improved based on real-time feedback.

DevOps promotes a cultural shift where teams are no longer isolated but work together to share responsibilities, which leads to increased efficiency, faster problem resolution, and a more streamlined deployment process. This culture of collaboration is supported by various automation tools and practices such as Continuous Integration (CI), Continuous Deployment (CD), Infrastructure as Code (IaC), and automated testing.

Supporting Reference:

Juniper Networks Automation and DevOps Documentation: This documentation emphasizes the importance of collaboration between development and operations teams to streamline processes and improve efficiency, aligning perfectly with the principles of DevOps.

"The DevOps Handbook" by Gene Kim, Patrick Debois, John Willis, and Jez Humble: This book provides an in-depth look into how DevOps practices enhance collaboration and lead to faster, more reliable software delivery.

IEEE and Industry Standards: While DevOps practices are widely adopted, they are not defined or maintained by IEEE or any other formal standards body, which is why option D is incorrect.

## Question: 2

Which data construct is used to guarantee that element names and data values remain unique in an XML document?

A. element

B. schema definition

C. namespace

D. dictionary

Answer: C

Explanation:

In XML documents, a namespace is the data construct used to ensure that element names and data values remain unique. Namespaces prevent naming conflicts by differentiating between elements or attributes that may have the same name but different meanings. This is particularly important in XML, where documents often incorporate elements from multiple sources.

Detailed Explanation:

XML Namespaces: A namespace is a collection of names, identified by a URI reference, which is used to distinguish between elements that may have identical names but different definitions or origins. This helps avoid ambiguity in the document.

How Namespaces Work: When a namespace is applied, each element or attribute in the XML document is associated with a prefix. This prefix, combined with the namespace URI, ensures that the element or attribute is uniquely identified, even if another element or attribute in the same document has the same local name but a different namespace.

Schema Definition vs. Namespace: Although an XML schema definition (XSD) can define the structure and type constraints of an XML document, it does not guarantee uniqueness of element names across different XML documents. That role is fulfilled by namespaces.

Practical Example:

xml

Copy code

```
<root xmlns:ns1="http://www.example.com/ns1"
      xmlns:ns2="http://www.example.com/ns2">
    <ns1:item>Item in namespace 1</ns1:item>
    <ns2:item>Item in namespace 2</ns2:item>
</root>
```

In this example, the item elements are in different namespaces (ns1 and ns2), which keeps them unique even though they have the same name.

Reference:

Juniper Automation and DevOps Documentation: These practices highlight the importance of namespaces in XML documents to maintain the integrity and uniqueness of data, which is essential in automation scripts and configuration files.

W3C XML Namespace Specification: The World Wide Web Consortium (W3C) standard for XML Namespaces defines how namespaces should be used to avoid name conflicts.

Namespaces are a crucial concept in XML, ensuring that data can be consistently managed and interpreted correctly, particularly in complex systems where multiple XML documents or schemas are involved.

## Question: 3

You are asked to develop an on-box Junos script that prevents deletion ofthe SNMP configuration.

Which type of script serves this purpose?

A. commit script

B. event script

C. op script

D. SNMP script

Answer: A

Explanation:

A commit script in Junos is used to enforce policies and configuration constraints on the device. These scripts are written in Extensible Stylesheet Language Transformations (XSLT) or Python and are executed automatically during the commit process of a configuration change.

In this context, to prevent the deletion ofthe SNMP configuration, a commit script is the appropriate choice. It can be designed to check the configuration changes being committed and reject any commit that attempts to delete or modify the SNMP configuration. This script essentially acts as a gatekeeper, ensuring that only allowable changes are committed to the device configuration.

Supporting Reference:

Juniper Networks Commit Scripts Documentation: The official Juniper documentation provides examples and use cases of commit scripts, including how they can be used to prevent unauthorized changes to the device configuration.

"Junos Automation Scripting" by Jonathan Looney: This resource gives practical examples and best practices for creating commit scripts to enforce configuration policies in Junos OS.

## Question: 4

A REST API client uses which two HTTP methods to execute RPC requests on the server? (Choose two.)

A. POST

B. HEAD

C. GET

D. CONNECT

Answer: A, C

Explanation:

REST APIs use HTTP methods to perform different operations on resources. In the context of RPC (Remote Procedure Call) requests:

GET: This method is used to retrieve data from the server. In a REST API, it is commonly used to fetch information about resources, such as the current configuration or operational state.

POST: This method is used to send data to the server to create or update a resource. In the context of RPC, POST is often used to execute a procedure on the server that may result in the modification of a resource or triggering of an action.

Options B (HEAD) and D (CONNECT) are not typically used for executing RPC requests:

HEAD is similar to GET but only retrieves the headers, not the body of the response.

CONNECT is used to establish a tunnel to the server, primarily for SSL-encrypted communication, and is not commonly associated with RESTful RPC operations.

Supporting Reference:

Juniper Networks REST API Documentation: The documentation provides detailed information about the use of HTTP methods in Juniper's RESTful services.

"RESTful Web Services" by Leonard Richardson and Sam Ruby: This book explains the principles of REST and how different HTTP methods, particularly GET and POST, are used to interact with RESTful APIs.

## Question: 5

YAML uses which two data structures? (Choose two.)

A. arrays

B. mappings

C. sequences

D. objects

Answer: B, C

Explanation:

YAML (YAML Ain't Markup Language) primarily uses two data structures:

Mappings: These are key-value pairs, similar to dictionaries or hashes in programming languages. In YAML, mappings are used to represent associative arrays or objects. They are defined with a colon (:) separating the key from the value.

Example:

key: value

name: John Doe

Sequences: These are ordered lists of items, equivalent to arrays or lists in other programming languages. Sequences in YAML are denoted by a dash (-) followed by a space and then the item.

Example:

fruits:

- Apple
- Banana
- Cherry

Detailed Explanation:

Mappings (B) allow you to define relationships between keys and values, making it possible to represent more complex data structures like dictionaries or objects.

Sequences (C) allow you to represent ordered collections, which is important for listing elements that must maintain a specific order.

YAML is often used in configuration files and data serialization in DevOps environments, such as in Ansible playbooks, Kubernetes manifest files, and CI/CD pipeline definitions. Its simplicity and human-readable format make it a popular choice for these applications.

Reference:

YAML Official Documentation: YAML's specification outlines these core data structures.

Juniper Automation and DevOps Documentation: Provides best practices for using YAML in network automation scripts and configuration management.

## Question: 6

Junos supports which two APIs for on-box scripting? (Choose two.)

A. JET

B. Puppet

C. XML

D. Chef

Answer: A, C

Explanation:

Juniper Networks' Junos operating system supports several APIs for on-box scripting, two of which are:

JET (Juniper Extension Toolkit): JET is a modern API framework that provides a programmable interface for interacting with Junos. It allows developers to create custom applications that run directly on Junos devices, enabling the automation of network operations. JET provides both a gRPC and a REST API interface, allowing for flexible integration with external systems.

XML API: The Junos XML API allows direct interaction with the Junos OS through XML-based requests. This API can be used to retrieve information, configure devices, and execute commands on Junos devices. The XML API is crucial for automation tasks as it provides a structured and consistent way to interact with the device's configuration and operational data.

Detailed Explanation:

JET (A) provides high-performance access to Junos routing, switching, and service elements via programmable interfaces. It is highly used for creating custom applications that require tight integration with the Junos OS.

XML (C), on the other hand, is a well-established method for interacting with Junos, especially for legacy systems or when working within environments where XML is the standard data format.

Other options like Puppet (B) and Chef (D) are not APIs provided by Junos for on-box scripting but are configuration management tools used externally to manage Junos devices.

Reference:

Juniper Networks JET Documentation: Provides details on how to leverage JET APIs for automation.

Junos XML Management Protocol Guide: Describes how to use XML for scripting and automating tasks in Junos.

These APIs are key components of Juniper's automation strategy, allowing for scalable, flexible, and efficient network operations.

## Question: 7

Which two statements about NETCONF layers are correct? (Choose two.)

A. NETCONF layers use the messages layer to receive RPCs from a remote NETCONF server.

B. NETCONF layers use the messages layer to send RPCs to a remote NETCONF server.

C. NETCONF layers use the operations layer to receive RPCs from a remote NETCONF server.

D. NETCONF layers use the operations layer to send RPCs to a remote NETCONF server.

Answer: B, C

Explanation:

NETCONF (Network Configuration Protocol) is a standard protocol defined for managing network devices. NETCONF operates in a layered architecture, which includes the following key layers:

Operations Layer: This layer deals with the actual operations like <get-config>, <edit-config>, <copyconfig>, and others. It receives RPC (Remote Procedure Call) requests from a remote NETCONF client and processes these requests.

Messages Layer: This layer is responsible for encoding the RPCs and sending them over the network. It handles the communication between the NETCONF client and server, ensuring that the RPC messages are correctly formatted (usually in XML) and transmitted.

Statement B is correct because the Messages layer is responsible for sending RPCs to a remote NETCONF server.

Statement C is correct because the Operations layer is where the NETCONF server receives and processes the RPCs sent by the client.

Supporting Reference:

Juniper Networks NETCONF Documentation: Provides a detailed breakdown of the NETCONF protocol layers and their functions.

RFC 6241: The official specification for NETCONF, which describes the layered architecture, including the operations and messages layers.

## Question: 8

Which two tools would you use to make REST API requests? (Choose two.)

A. cURL

B. NETCOfJF

C. Web browser

D. SSH

Answer: AC

Explanation:

REST API requests can be made using various tools that support HTTP methods. Two common tools are:

cURL: A command-line tool that allows you to send HTTP requests, including GET, POST, PUT, and DELETE. It is widely used for testing and interacting with RESTful APIs due to its simplicity and flexibility.

Web Browser: Modern web browsers can be used to send HTTP GET requests directly by entering the URL into the address bar. Additionally, browser extensions like Postman or built-in developer tools can be used to construct and send more complex REST API requests.

Option B (NETCOfJF) is incorrect as it does not refer to a standard tool for making REST API requests.

Option D (SSH) is incorrect because SSH is a protocol used for secure remote login and command execution, not for sending REST API requests.

Supporting Reference:

cURL Documentation: Official cURL documentation provides extensive information on how to use cURL to

interact with REST APIs.

Web Development and REST API Guides: Many web development resources and REST API documentation recommend using web browsers and tools like cURL for testing and interacting with APIs.

## Question: 9

Which DevOps "Three way" principle addresses technical debt?

A. feedback

B. flow

C. continuous experimentation and learning

D. continuous experimentation

Answer: A

Explanation:

In the context of the DevOps "Three Ways" principles, the feedback principle directly addresses the management of technical debt.

The "Three Ways" are core principles guiding DevOps practices, and they are as follows:

Flow: Refers to the smooth and fast flow of work through the system, from development to operations.

Feedback: Emphasizes creating effective, fast, and continuous feedback loops between teams to catch issues early, address technical debt, and ensure quality.

Continuous experimentation and learning: Encourages constant experimentation, innovation, and learning from failures to improve systems and processes over time.

Feedback and Technical Debt:

Feedback loops play a crucial role in addressing technical debt. Technical debt refers to the implied cost of additional work that arises when code or system design decisions are made for short-term gains, such as quick fixes or temporary patches. Over time, technical debt can accumulate and degrade system performance, reliability, and maintainability.

The feedback loop ensures that issues related to technical debt (such as poor code quality, design shortcuts, or performance bottlenecks) are caught early in the process, ideally before they become major problems. Continuous monitoring, testing, and reviewing help identify and resolve technical debt incrementally rather than letting it accumulate unchecked.

Automation in feedback loops: In DevOps, automated testing, continuous integration (CI), and monitoring tools provide immediate feedback to developers, highlighting areas where technical debt is increasing. This feedback is crucial for making proactive decisions about refactoring code or improving infrastructure without waiting for problems to manifest in production.

For instance, the feedback loop might expose slowdowns in application performance after each new feature is added. This would trigger a review to either refactor the feature code or improve system resources, preventing further technical debt accumulation.

Flow and Technical Debt:

While flow focuses on the smooth transition of work through the pipeline, it indirectly helps with technical debt by ensuring continuous and streamlined processes. However, feedback mechanisms are the primary tools for identifying and resolving technical debt.

Continuous Experimentation and Learning:

This principle promotes innovation and learning from failures but does not directly address technical debt. The focus here is more on risk-taking and improvement rather than managing or eliminating technical debt.

Reference from DevOps Practices:

The Phoenix Project, a book often referenced in DevOps, discusses how feedback loops are essential for maintaining system integrity and managing technical debt effectively. By improving feedback mechanisms, teams can address small issues before they become costly to fix.

The DevOps Handbook also highlights the importance of feedback in managing technical debt, emphasizing that fast feedback allows for continuous improvement and avoids the accumulation of bad practices that would otherwise lead to technical debt.

Juniper Automation and DevOps Context: Juniper's automation frameworks integrate feedback mechanisms using tools like continuous monitoring and automated testing. These tools help engineers track the health of network systems, identify configuration drifts, and resolve issues before they lead to significant technical debt.

Additional Resources:

The Phoenix Project by Gene Kim

The DevOps Handbook

## Question: 10

Which two processes are used by Junos automation? (Choose two.)

A. mod

B. jsd

C. ifd

D. kmd

Answer: BD

Explanation:

Junos automation relies on several key processes to handle various automation and API interactions. Let's break down the two key processes involved:

jsd (Junos Script Daemon):

The jsd process is responsible for handling automation scripts, including Python and SLAX scripts, as well as handling JET (Junos Extension Toolkit) API requests. This process is fundamental in the automation framework of Junos, as it deals with external and internal API requests, ensuring that the necessary scripts are executed when specific triggers or events occur.

kmd (Key Management Daemon):

The kmd process is involved in key management for IPsec and other security services. While its primary function is related to managing cryptographic keys, it plays a role in Junos automation by enabling secure communication and ensuring that automation tasks involving security services (such as automated IPsec tunnel creation) are handled securely.

Why the Other Options Are Incorrect:

A . mod: This process doesn't exist as part of the Junos automation framework. It's likely a distractor.

C . ifd: The ifd process is associated with the physical interfaces on the device and does not play a role in automation or script processing.

Reference from Juniper Documentation:

Junos Automation Processes

## Question: 11

Which two statements are valid regarding Junos automation? (Choose two.)

A. The jsd process handles XML API calls.

B. The mgd process handles JET API requests.

C. The jsd process handles JET API requests.

D. The mod process handles XML API calls.

Answer: C, A

Explanation:

In Junos automation, several processes handle API requests, and understanding which process handles what is crucial:

jsd Process:

XML API Calls (A): The jsd process is responsible for handling XML API calls, which are a significant part of Junos automation. XML API allows for structured and standardized communication with Junos devices, enabling automation scripts to query and configure devices.

JET API Requests (C): The jsd process also handles JET (Junos Extension Toolkit) API requests. JET provides a more modern, programmable interface for interacting with Junos OS, and jsd is the process that manages these interactions.

mgd Process (Incorrect Option):

Not for JET API Requests: The mgd process handles general management operations, such as CLI commands and managing the configuration database, but it does not handle JET API requests. That role is fulfilled by jsd.

mod Process (Incorrect Option):

Not for XML API Calls: The mod process deals with managing chassis components and is not involved in handling XML API calls.

Reference:

Juniper Networks JET and XML API Documentation: Describes the roles of jsd in handling both XML and JET API requests.

Junos Automation and DevOps Documentation: Provides insights into how different processes interact with Junos APIs.

## Question: 12

What is the difference between a list and a tuple in Python?

A. Lists are immutable objects that use square brackets, and tuples are mutable objects that use parentheses.

B. Lists are mutable objects that use square brackets, and tuples are immutable objects that use parentheses.

C. Lists are immutable objects that use parentheses, and tuples are immutable objects that use square brackets.

D. Lists are mutable objects that use parentheses, and tuples are immutable objects that use square brackets.

Answer: B

### Explanation:

In Python, the distinction between lists and tuples is essential for efficient programming:

Lists:

Mutable (B): This means that once a list is created, its elements can be changed, added, or removed. Lists are versatile and commonly used when the data is expected to change.

Square Brackets: Lists are defined using square brackets [].

Example:

```
my_list = [1, 2, 3]

my_list[0] = 10 # Modifying the first element
```

Tuples:

Immutable (B): Once a tuple is created, it cannot be altered. Tuples are used when a fixed collection of items is needed, providing more integrity to the data.

Parentheses: Tuples are defined using parentheses ().

Example:

```
my_tuple = (1, 2, 3)

# my_tuple[0] = 10 # This would raise an error because tuples are immutable
```

Reference:

Python Official Documentation: The Python Language Reference provides detailed information on

data types like lists and tuples, including their mutability and syntax.

Automation Scripts: In the context of automation, understanding when to use mutable or immutable data structures can significantly impact script performance and reliability.

## Question: 13

You want to use a Python package or module.

In this scenario, which statement would accomplish this task?

A. reap

B. dir

C. input

D. Import

Answer: D

Explanation:

In Python, to use a package or module, you use the import statement. This statement allows you to load a module into your script so that you can use its functions, classes, and variables. For example, if you wanted to use the math module, you would write import math. This makes all the functions and constants in the math module available for use in your program.

Option A (reap), B (dir), and C (input) do not serve the purpose of importing modules. dir is used to list the attributes of an object, input is used to get user input, and reap is not a valid Python command related to importing modules.

Supporting Reference:

Python Documentation on Imports: The Python documentation provides clear guidelines on how to use the import statement to include modules in your Python scripts.

## Question: 14

Using the set rest control configuration command, what are two ways to control access to the REST API running on a Junos device? (Choose two.)

A. Limit management access to only SSH

B. Limit management access to specific users.

C. Limit the number of simultaneous connections.

D. Limit access to only certain source IP addresses

Answer: C, D

Explanation:

When using the set rest control configuration command on a Junos device, you have several options to control access to the REST API. Two effective methods include:

Limiting the number of simultaneous connections: This ensures that the REST API is not overwhelmed by too many concurrent requests, which could potentially lead to performance issues or denial of service.

Limiting access to certain source IP addresses: This method restricts API access to specific IP addresses, enhancing security by ensuring that only trusted sources can interact with the REST API.

Option A (Limit management access to only SSH) is unrelated to controlling REST API access specifically.

Option B (Limit management access to specific users) might be relevant in a different context, but it is not directly tied to REST API control via the specific command mentioned.

Supporting Reference:

Juniper Networks REST API Documentation: This documentation explains how to configure and control access to the REST API on Junos devices, including connection limits and IP-based access control.

## Question: 15

Which two statements about Junos automation are correct? (Choose two.)

A. The Junos REST API client is on-box.

B. Junos automation does not allow for device provisioning throuqh the console port.

C. Junos automation allows for device provisioning through the console port.

D. The Junos REST API client is off-box.

Answer: AC

Explanation:

A. The Junos REST API client is on-box:
The Junos REST API is on-box, meaning it is hosted directly on the Junos OS device. This allows you to interact with the device through RESTful API calls without needing an external client to act as an intermediary. With the on-box REST API, users can manage and automate configuration and operational tasks directly from the Junos device itself using HTTP/HTTPS protocols. This simplifies automation and remote management since the API server is embedded within the device.

Key Automation Capabilities of On-box REST API:

Supports configuration, monitoring, and operational commands.
Allows for direct device interaction via tools like curl, or through custom-built automation scripts.

Reference: Junos Automation documentation explains that the REST API client runs natively on the device, providing a web-based interface for automation tasks.

C . Junos automation allows for device provisioning through the console port:
Junos automation does indeed allow for device provisioning through the console port, especially in the context of Zero Touch Provisioning (ZTP). When network interfaces are not initially configured, or when remote access is not possible, devices can be provisioned via the console port. This method is commonly used during the initial setup process, enabling administrators to deploy configurations even without network access. ZTP automates initial configurations, including system setup and software installation, which can be

triggered via the console.

Key Advantages of Console-based Provisioning:

Useful in environments where network interfaces are unavailable or not yet configured.

Essential for the initial bootstrapping of devices in remote locations.

Reference: Junos documentation on Zero Touch Provisioning (ZTP) and automation methods highlights that provisioning through the console port is supported and often used for initial setups.

Why the Other Options Are Incorrect:

B. Junos automation does not allow for device provisioning through the console port: This statement is incorrect because Junos automation does allow for provisioning via the console port, particularly during initial device setups.

D . The Junos REST API client is off-box: This is incorrect because the REST API client can be directly on the Junos device, providing local API functionality (on-box).

Juniper Automation in DevOps Context: Junos automation, especially with on-box REST API and console-based provisioning, enhances the flexibility and accessibility of device management in DevOps environments. These capabilities simplify remote configuration, monitoring, and device setup even in cases where direct network access is unavailable.

## Question: 16

Which two statements are correct about using the Junos REST API? (Choose two.)

A. It supports data In CSV format.

B. It must use SSH for a connection.

C. NETCONF is not supported.

D. It is a simple configuration.

Answer: A, D

Explanation:

A. It supports data in CSV format:

The Junos REST API supports multiple data formats for transferring information between systems, including XML, JSON, and CSV (Comma Separated Values). This flexibility allows for easier data

parsing, especially in environments where structured data (like CSV) is a standard. CSV is often used for bulk data export or import and reporting purposes, making it an essential format for automation tasks involving external systems or large datasets.

Example Usage in REST API:

When using the Junos REST API, a user can request configuration or operational data and specify the response format (XML, JSON, or CSV). CSV is particularly useful when integrating Junos devices with systems that require easily readable, tabular formats.

Reference: Junos REST API documentation confirms support for CSV format alongside XML and JSON for RESTful interactions.

D. It is a simple configuration:

The Junos REST API is designed to be relatively simple to configure. Once the REST API service is enabled on the Junos device, it can be accessed via HTTP or HTTPS, making it an easy entry point for automation and management tasks. Unlike more complex protocols (such as NETCONF), the REST API is lightweight and easier to use for simple configuration changes and retrieving operational data.

Configuration Example:

To enable the REST API, you can add the following configuration:

set system services rest http

set system services rest https

After enabling the service, API requests can be made to interact with the device for automation tasks, without needing the complexity of SSH or NETCONF configuration.

Reference: The Junos REST API is well-documented as an easy-to-configure and use API interface, making it accessible even for those who are new to Junos automation.

Why the Other Options Are Incorrect:

B . It must use SSH for a connection: This is incorrect. The Junos REST API uses HTTP or HTTPS for communication, not SSH. While SSH is commonly used for NETCONF, it is not required for REST API connections. REST APIs operate over standard web protocols.

C . NETCONF is not supported: This is incorrect. Junos supports both REST API and NETCONF for automation and configuration management. NETCONF is an XML-based protocol used for device configuration, which operates over SSH. The REST API and NETCONF can coexist on the same device, offering multiple avenues

for automation.

Juniper Automation in DevOps Context: The simplicity and flexibility of the Junos REST API make it ideal for DevOps automation tasks. It allows teams to easily interact with Junos devices using lightweight RESTful methods, integrating with external systems through formats like CSV. The ease of configuration supports rapid deployment and scaling of automated management tasks.

Reference from Juniper Documentation:

Junos REST API Documentation

## Question: 17

What is the correct Python script syntax to prompt for input?

A. hostIP = input("Device IP address: ")

B. hostIP = input{Device IP address: }

C. hostIP = input"Device IP address: "

D. input("Device IP address: ") = hostIP

Answer: A

Explanation:

In Python, the correct syntax to prompt the user for input and store that input in a variable is:

input(prompt): The input() function is used to take input from the user. The string provided as an argument (inside the parentheses) is displayed as a prompt to the user. The input provided by the user is returned as a string and can be stored in a variable.

Example:

hostIP = input("Device IP address: ")

In this example, "Device IP address: " is the prompt displayed to the user, and the user's input will be stored in the variable hostIP.

Options B, C, and D are syntactically incorrect in Python.

Reference:

Python Official Documentation: Describes the use of the input() function for getting user input.

Python Tutorials: Various tutorials demonstrate how to properly use the input() function in scripts.

## Question: 18

Given the following Python script:

```
a = [1,2,3,4,5,6,7,8,9]

print(a[0])
```

What is the output of this print command?

A. 1

B. 2

C. 7

D. 9

Answer: A

Explanation:

In Python, lists are zero-indexed, meaning the first element of the list is at index 0. The given script is:

```
pythona = [1, 2, 3, 4, 5, 6, 7, 8, 9]

print(a[0])
```

a[0] refers to the first element in the list a, which is 1.

So, the output of the print(a[0]) command is 1.

Option A is correct because Python indexing starts at 0, making the first element of the list at index 0.

Reference:

Python Official Documentation: Covers list indexing and operations.

Python Programming Tutorials: Provide examples of list indexing.

## Question: 19

Which Junos API supports direct modification of the Ephemeral database?

A. JET

B. WebSocket

C. SOAP

D. REST

Answer: A

Explanation:

In Junos, the JET (Junos Extension Toolkit) API supports direct modification of the Ephemeral database. The Ephemeral database is a temporary configuration database used in Junos OS, allowing for changes that do not persist after a reboot unless explicitly committed to the permanent configuration.

JET API: Allows for high-performance interactions with Junos, including the ability to make changes to the Ephemeral database, which is useful for temporary configurations, dynamic policies, and other operational tasks.

Other options like WebSocket, SOAP, and REST do not provide direct access to the Ephemeral database in Junos.

Reference:

Juniper Networks JET Documentation: Details how JET API interacts with the Ephemeral database.
Junos Automation and DevOps Documentation: Discusses the use of JET for automation and dynamic configuration.

## Question: 20

Which two statements about the REST API are correct? (Choose two.)

A. The TCP session state is maintained by the server.

B. The REST API application is stateless.

C. The TCP session state is maintained by the client

D. The REST API application is stateful.

Answer: B, C

Explanation:

REST (Representational State Transfer) is an architectural style for designing networked applications, and its key principles include:

Statelessness (B): Each request from the client to the server must contain all the information needed to understand and process the request. The server does not store any session state between requests, meaning each request is independent and does not rely on previous ones.

TCP Session State (C): While REST itself is stateless, the underlying TCP connection's state, such as keeping the connection alive or managing retries, is handled by the client. The server does not retain information about the TCP connection beyond the processing of the individual request.

Options A and D are incorrect because they imply that the REST API is stateful, which contradicts the stateless nature of REST.

Reference:

REST API Design Principles: Describes the stateless nature of REST and the responsibility of clients in managing session state.

Web Development Documentation: Discusses how REST APIs operate, focusing on statelessness and client-server interaction.

## Question: 21

Which two standard logical operators does XPath support? (Choose two.)

A. IOR

B. NOT

C. AND

D. MAMD

Answer: B, C

Explanation:

XPath is a query language used for selecting nodes from an XML document. It supports various logical operators that can be used to create complex queries. The two standard logical operators supported by XPath are:

NOT: This operator negates a condition, returning true if the condition is false, and vice versa.

AND: This operator is used to combine two conditions, and it returns true only if both conditions are true.

Option A (IOR) and Option D (MAMD) are not standard XPath operators.

Supporting Reference:

XPath Documentation: The W3C XPath specification lists the standard operators supported in XPath, including AND and NOT.

## Question: 22

Which statement about the NETCONF content layer is true?

A. It uses YAML for RPC request and response payloads.

B. It uses XML for RPC request and response payloads.

C. It uses JSON for RPC request and response payloads.

D. It uses HTML for RPC request and response payloads.

Answer: B

Explanation:

The NETCONF protocol, used for network management, utilizes XML for encoding the RPC (Remote Procedure Call) requests and responses. XML is chosen because of its flexibility and ability to represent hierarchical data structures, making it well-suited for representing network configurations and states.

Option B is correct because XML is the standard format used for NETCONF RPC payloads.

Options A (YAML), C (JSON), and D (HTML) are incorrect because these formats are not used by NETCONF for its RPC payloads.

Supporting Reference:

RFC 6241 - NETCONF Protocol: This RFC describes the use of XML for encoding NETCONF messages.

## Question: 23

You are asked to use the REST API to retrieve interface configuration information from your Junos device. You decide to use a cURL HTTP GET command to retrieve this information.

In this scenario, which statement is correct?

A. The request ishandled by the mod process running on the Junos device.

B. The request ishandled by the isrpd process running on the Junos device.

C. The request ishandled by the rpd process running on the Junos device.

D. The request ishandled by the isd process running on the Junos device

Answer: B

Explanation:

When using the REST API on a Junos device, the isrpd (Integrated Service Routing Process Daemon) process is responsible for handling REST API requests. This process listens for incoming HTTP requests and processes them accordingly, including retrieving interface configuration information when a GET request is made.

Option B is correct because the isrpd process handles the REST API requests on a Junos device.

Options A (mod process), C (rpd process), and D (isd process) are incorrect in this context as they either do not exist or serve different purposes on a Junos device.

Supporting Reference:

Juniper Networks REST API Documentation: Provides insights into how REST API requests are managed and processed by the isrpd process on Junos devices.

## Question: 24

Which two statements about XML schema definition (XSD) files are correct? (Choose two.)

A. XSD files define all the elements in an XML document and the document XML hierarchy.

B. Every XML document must have an XSD file defined for it.

C. An XSD file is not an XML document.

D. XSD files ensure that everyone working with the XML document uses a common set of tags.

Answer: A, D

Explanation:

XML Schema Definition (XSD) files are used to define the structure and data types of an XML document. They ensure that the XML document adheres to a specific structure by defining the allowed elements, attributes, and their data types, thereby enforcing a consistent format.

Option A is correct because XSD files define the elements, attributes, and structure (hierarchy) of an XML document.

Option D is correct because XSD files provide a standardized format, ensuring that all parties working with the XML document use the same set of tags and structure.

Option B (Every XML document must have an XSD file defined for it) is incorrect; not every XML document requires an XSD file, although it's beneficial for validation.

Option C (An XSD file is not an XML document) is incorrect because XSD files themselves are written in XML.

Supporting Reference:

W3C XML Schema Definition (XSD) Documentation: Explains the purpose and structure of XSD files, including their role in defining XML document schemas.

## Question: 25

Which process is responsible for JET automation requests?

A. jsrpd

B. mgd

C. rpd

D. jsd

Answer: D

Explanation:

The jsd (Junos Script Daemon) process is responsible for handling JET (Junos Extension Toolkit) automation requests. The JET framework allows for more programmable interactions with Junos devices, using APIs to manage and automate network functions. The jsd process handles these API requests, executes the relevant scripts, and interacts with the Junos configuration and operational states accordingly.

JET enables developers to write applications that interact with the Junos OS through APIs, providing a powerful tool for network automation, monitoring, and configuration management. The jsd process ensures that all JET-related operations are processed and executed properly.

Why the Other Options Are Incorrect:

A . jsrpd: This is not a process associated with JET automation requests.

B . mgd: The mgd process is responsible for handling management tasks like CLI commands and configuration changes but is not directly responsible for JET automation requests.

C . rpd: The rpd (Routing Protocol Daemon) manages routing protocols like BGP and OSPF but has no involvement in JET automation.

Reference from Juniper Documentation:

Juniper's documentation on JET and the jsd process outlines how this daemon manages script and API

request processing for automation tasks.

## Question: 26

What are two important aspects of the DevOps culture? (Choose two.)

A. communication

B. separation of duties

C. use of specific tools

D. people

Answer: A, D

Explanation:

Two important aspects of the DevOps culture are:

Communication (A): In a DevOps environment, continuous and effective communication between development, operations, and other stakeholders is crucial. It ensures that everyone is aligned, reduces misunderstandings, and facilitates faster and more efficient workflows.

People (D): DevOps is as much about the people as it is about the processes and tools. The culture emphasizes collaboration, shared responsibility, and a mindset focused on continuous improvement. Empowering people to work together across different disciplines is at the heart of DevOps.

Options B and C are not primary aspects of the DevOps culture. While tools are essential in implementing DevOps practices, the culture emphasizes communication and collaboration among people more than the use of specific tools or rigid separation of duties.

Reference:

DevOps Handbook: Discusses the cultural aspects of DevOps, with a focus on communication and

people.

DevOps Best Practices: Highlights the importance of fostering a culture that prioritizes collaboration and shared goals.

## Question: 27

You are asked to use the REST API to retrieve interface configuration information from your Junos device. You decide to use a cURL HTTP GET command to retrieve this information.

In this scenario, which two statements are correct? (Choose two.)

A. You can retrieve this data in HTML or JSON formats.

B. You must have SSH enabled on the Junos device.

C. You can retrieve this data in XML or JSON formats.

D. You must Include the authentication Information with each request.

Answer: C, D

### Explanation:

When using the REST API to retrieve interface configuration information from a Junos device:

Data Formats (C): The information can be retrieved in XML or JSON formats. These are the two standard data formats supported by the Junos REST API for representing configuration and operational data.

Authentication (D): For each HTTP request, especially when using tools like cURL, authentication information must be included. This is typically done using basic authentication (username and password) or an authentication token.

Option A is incorrect because HTML is not a supported format for REST API data retrieval in Junos, and Option B is incorrect because SSH is not required for REST API requests; the REST API typically uses HTTP/HTTPS.

### Reference:

Junos REST API Documentation: Details the data formats (XML, JSON) supported by the Junos REST API and the need for authentication.

cURL Usage with REST API: Provides examples of how to use cURL with Junos REST API, including the necessity of providing authentication.

## Question: 28

Which two programming languages are used for Junos on-box scripting? (Choose two.)

A. Perl

B. Ruby

C. SLAX

D. XSLT

Answer: C, D

Explanation:

Junos on-box scripting supports the following programming languages:

SLAX (C): SLAX (Structured Language for XML) is a scripting language designed specifically for Junos devices. It allows for easy manipulation of XML data, making it ideal for creating Junos scripts that interact with device configurations.

XSLT (D): XSLT (Extensible Stylesheet Language Transformations) is another language used for transforming XML documents into other formats. It is commonly used in Junos for transforming XML data into different views or outputs.

Options A (Perl) and B (Ruby) are not used for Junos on-box scripting. While these languages are popular in other contexts, Junos scripting relies heavily on XML-based languages like SLAX and XSLT.

Reference:

Junos XML API and Scripting Guide: Describes the use of SLAX and XSLT for on-box scripting.

Juniper Networks Automation Documentation: Provides examples and best practices for using SLAX and XSLT in Junos scripting.

## Question: 29

Which process is responsible for XML automation requests?

A. jsrpd

B. mgd

C. rpd

D. jsd

Answer: B

Explanation:

The mgd (Management Daemon) process in Junos is responsible for handling XML automation requests. This daemon manages the configuration and operational commands received via NETCONF, which uses XML for data exchange. The mgd process parses the XML data and applies the necessary configuration or retrieves the requested information.

Option B is correct because mgd is the process that handles XML-based requests in Junos.

Options A (jsrpd), C (rpd), and D (jsd) are incorrect because they are responsible for different functions, such as routing protocols and services, not XML automation.

Supporting Reference:

Juniper Networks Management Daemon (mgd) Documentation: Provides an overview of the responsibilities of the mgd process, including handling XML requests.

## Question: 30

Which Junos configuration database is updated by PyEZ by default?

A. shared

B. dynamic

C. private

D. ephemeral

Answer: C

Explanation:

An event script is used to automate responses to system events in Junos, such as an interface going down. These scripts are triggered automatically when a specified event occurs, making them suitable for tasks like monitoring interface status and executing actions when the status changes.

Option B (event) is correct because event scripts are designed for reacting to system events like an interface going down.

Option A (commit) is used for configuration changes, Option C (operation) is used for operational tasks, and Option D (SNMP) is not applicable in this context.

Supporting Reference:

Juniper Networks Event Scripts Documentation: Details how event scripts are used to automate responses to specific system events in Junos

## Question: 31

You are asked to write an on-box script that will be triggered when a specific interface on a Junos device goes down.

Which type of on-box script should you use to accomplish this task?

A. commit
B. event
C. operation
D. SNMP

Answer: B

Explanation:

An event script is used to automate responses to system events in Junos, such as an interface going down. These scripts are triggered automatically when a specified event occurs, making them suitable for tasks like monitoring interface status and executing actions when the status changes.

Option B (event) is correct because event scripts are designed for reacting to system events like an interface going down.

Option A (commit) is used for configuration changes, Option C (operation) is used for operational tasks, and Option D (SNMP) is not applicable in this context.

Supporting Reference:

Juniper Networks Event Scripts Documentation: Details how event scripts are used to automate responses to specific system events in Junos

## Question: 32

What are two Junos PyEZ configuration object methods? (Choose two.)

A. commie ()
B. device ()
C. lockO
D. config ()

Answer: D, C

Explanation:

In Junos PyEZ, the Config object provides various methods for interacting with device configurations. Two of the key methods are:

lock(): This method locks the candidate configuration database to prevent other users or processes from making changes while you are modifying the configuration.

config(): This method is used to create a Config object that represents the configuration database, allowing you to load, modify, and commit configuration changes.

Option C (lock) and Option D (config) are correct because they are valid methods provided by the PyEZ

Config object.

Option A (commie) and Option B (device) are incorrect as they are not methods of the Config object.

Supporting Reference:

Junos PyEZ Documentation: Details the methods available in the Config object, including lock() and config().

## Question: 33

Junos PyEZ tables are formatted using which file type?

A. SON

B. YAML

C. txt

D. IXML

Answer: B

Explanation:

Junos PyEZ uses YAML (YAML Ain't Markup Language) files to define the format for tables and views when working with operational and configuration data. YAML is a human-readable data format that is commonly used for configuration files, making it suitable for defining data structures in PyEZ.

Option B (YAML) is correct because PyEZ tables are defined using YAML files.

Options A (JSON), C (txt), and D (IXML) are incorrect in this context, as YAML is the standard format used.

Supporting Reference:

Junos PyEZ Tables Documentation: Explains the use of YAML files for formatting tables and views in Junos PyEZ.

## Question: 34

What is the correct sequence for Python script execution?

A. The code is translated to byte code, the byte code is executed in runtime, and then the code is interpreted.

B. The code is interpreted, the code is translated to byte code, and then the byte code is executed in runtime.

C. The code is translated to byte code, the code is interpreted, and then the byte code is executed in runtime.

D. The byte code is executed in runtime, the code is interpreted, and then the code is translated to byte code.

Answer: B

Explanation:

Python follows a specific execution flow when a script is run:

The code is interpreted:

Python is an interpreted language, meaning that the Python interpreter reads the code line by line. When a Python script is executed, the interpreter first reads the source code.

The code is translated to bytecode:

After interpreting the source code, Python translates it into bytecode. Bytecode is an intermediate representation of the source code that is portable and efficient for execution by the Python Virtual Machine (PVM).

The bytecode is executed in runtime:

Finally, the Python Virtual Machine (PVM) executes the bytecode. The PVM is a part of the Python runtime environment, responsible for interpreting the bytecode into machine-specific instructions for execution.

Hence, the correct sequence is: interpreted → translated to bytecode → bytecode executed in runtime.

Why the Other Options Are Incorrect:

Options A, C, and D present an incorrect order of the script execution process, especially in how bytecode is generated and executed.

Reference:

Python's documentation on the interpreter and its execution model explains this standard process.

## Question: 35

Which type of on-box automation script is designed to run every time a user executes a configuration change?

A. event

B. SNMP

C. commit

D. operation

Answer: C

Explanation:

In Junos OS, a commit script is an on-box automation script that runs every time a configuration change is committed. Commit scripts are used to enforce configuration policies, validate configuration changes, or make automatic adjustments to configurations when certain conditions are met.

Commit Script (C): Executes automatically during the commit process, ensuring that the new configuration adheres to specific rules or conventions before it is applied to the system.

Event, SNMP, and operation scripts are used for other purposes in Junos automation but do not run automatically with every configuration change.

Reference:

Junos OS Automation Scripts Guide: Provides details on different types of scripts, including commit scripts, and their use cases.

Juniper Networks Documentation: Offers examples and best practices for creating and using commit scripts.

## Question: 36

You must use Junos PyEZ to configure unique IP addresses on individual machines.

Which two features will permit this requirement? (Choose). Ian SCP module

A. an SCP module

B. a BSON data file

C. a YAML data file

D. a Jinja2 template

Answer: C, D

Explanation:

To configure unique IP addresses on individual machines using Junos PyEZ, you can use the following features:

YAML Data File (C): YAML files are used to store configuration data in a human-readable format. They are often used in combination with Jinja2 templates to provide the data necessary for template rendering.

Jinja2 Template (D): Jinja2 is a templating engine for Python that allows you to create dynamic templates. When used with Junos PyEZ, a Jinja2 template can be filled with data (such as IP addresses from a YAML file) to generate configuration snippets that are applied to different devices.

Options A (SCP module) and B (BSON data file) are not typically used with Junos PyEZ for this purpose.

Reference:

Junos PyEZ Documentation: Discusses the use of YAML files and Jinja2 templates for generating configurations.

Jinja2 Templating Documentation: Provides details on how to create and use templates in Python scripts.

## Question: 37

Which two statements are true about an XML schema document? (Choose two.)

A. It is formatted as an XLT file.

B. It cannot be examined in the Junos CLI.

C. It is an authoritative source for operational and configuration XML.

D. lit is formatted as an XSD file.

Answer: C, D

Explanation:

An XML schema document (XSD) is a key component in defining the structure and constraints of XML data used in various applications, including Junos:

Authoritative Source (C): An XML schema document serves as the authoritative definition of the structure, content, and semantics of XML documents. It ensures that the XML data adheres to specific rules and formats, which is essential for both operational and configuration XML.

XSD Format (D): XML schema documents are typically written in the XSD (XML Schema Definition) format, which provides a formal description of the XML document's structure.

Option A is incorrect because XML schemas are not formatted as XLT files (which are related to XSLT transformations), and Option B is incorrect because XML schemas can indeed be examined in the Junos CLI using appropriate commands.

Reference:

W3C XML Schema Definition Language (XSD) Documentation: Provides comprehensive information on the XSD format.

Juniper Networks Documentation: Discusses the role of XML schemas in managing Junos configurations.

## Question: 38

You want to make a list in Python to store data.

Which statement is the correct way to accomplish this task?

A. L = "0, 1, 2, 3, 4, 5"

B. L = {0, 1, 2, 3, 4, 5}

C. L = [0, 1, 2, 3, 4, 5]

D. L = (0, 1, 2, 3, 4, 5)

Answer: C

Explanation:

In Python, to create a list, you use square brackets []. The correct syntax to create a list containing the numbers 0 through 5 is:

L = [0, 1, 2, 3, 4, 5]

This statement creates a list object that stores the specified integers.

Other options are incorrect:

A defines a string, not a list.

B defines a set, which is an unordered collection with no duplicate elements.

D defines a tuple, which is an immutable sequence, not a list.

Reference:

Python Official Documentation: Discusses lists, sets, tuples, and their syntaxes.

Python Data Structures Guide: Provides examples of creating and manipulating lists.

## Question: 39

Which two statements about NETCONF are correct? (Choose two.)

A. The default port for NETCONF is port 930.

B. The default port for NETCONF is port 830.

C. NETCONF cannot use the default SSH port.

D. NETCONF can use the default SSH port.

Answer: B, D

Explanation:

NETCONF (Network Configuration Protocol) is used for network device management and can operate over SSH. The following are true about NETCONF:

Default Port 830 (B): By default, NETCONF uses port 830 for communication over SSH. This is the standard port reserved for NETCONF sessions.

Use of Default SSH Port (D): NETCONF can also operate over the standard SSH port (port 22) if configured to do so. This allows flexibility in network management scenarios where port 830 might not be available or used.

Options A and C are incorrect because they refer to incorrect or non-applicable port numbers for NETCONF.

Reference:

IETF RFC 6241: Specifies the use of NETCONF over SSH, including port details.

Juniper Networks NETCONF Documentation: Discusses the configuration and operation of NETCONF on Junos devices, including port usage.

## Question: 40

Which two PyEZ object methods are included by default when using a Python context manager? (Choose two.)

A. lock () and unlock (>

B. open() and close()

C. |load() and commit ()

D. pdiff() and diff()

Answer: A, B

Explanation:

When using a Python context manager with Junos PyEZ, two key methods are automatically included:

open() and close(): These methods are used to establish and terminate a connection to a Junos device. When you use a context manager (the with statement), open() is called when entering the block, and close() is called when exiting, ensuring the connection is properly managed.

lock() and unlock(): These methods are used to lock the configuration database to prevent other users from making changes while you are working on it. When using a context manager, lock() is called at the start of the block, and unlock() is called at the end, ensuring safe configuration changes.

Supporting Reference:

Junos PyEZ Documentation: The documentation explains how context managers work in PyEZ, including the automatic invocation of open(), close(), lock(), and unlock() methods.

## Question: 41

Which two data structures are used in JSON? (Choose two.)

A. tuples

B. objects

C. arrays

D. dictionaries

Answer: B, C

Explanation:

In JSON (JavaScript Object Notation), the two primary data structures are:

Objects: These are collections of key-value pairs, where each key is a string, and the value can be a string,

number, array, boolean, or another object. In Python, this structure is analogous to a dictionary.

Arrays: These are ordered lists of values, where each value can be of any data type, including another array or object. In Python, this structure is similar to a list.

Option A (tuples) and Option D (dictionaries) refer to Python-specific data structures and are not directly used in JSON.

Supporting Reference:

JSON Documentation and Tutorials: JSON objects and arrays are the standard data structures used in this format, as described in many tutorials and the official JSON documentation.

## Question: 42

Which Python operator is used to test if two variables are equal?

A. !=

B. ==

C. %

D. =

Answer: B

Explanation:

In Python, the == operator is used to test whether two variables are equal. It returns True if the variables are equal and False if they are not.

Option B (==) is correct because it is the equality operator in Python.

Option A (!=) is used for inequality, Option C (%) is the modulus operator, and Option D (=) is used for assignment, not for testing equality.

Supporting Reference:

Python Documentation on Operators: The official Python documentation covers the use of == for equality checks.

## Question: 43

Which statement is valid regarding YAML and JSON?

A. YAML and JSON use indentation.

B. White space is ignored in YAML and JSON.

C. Comments are available in YAML and JSON.

D. YAML and JSON are case-sensitive.

Answer: D

Explanation:

Both YAML and JSON are case-sensitive, meaning that the distinction between uppercase and lowercase characters matters. For example, in JSON or YAML, Key and key would be considered different.

Option D (case-sensitive) is correct because both YAML and JSON treat keys and values with different cases as distinct.

Option A is incorrect because, while JSON does not use indentation, YAML does rely on indentation to define structure. Option B is incorrect because whitespace can be significant in YAML for structure, and Option C is incorrect because JSON does not support comments, while YAML does.

Supporting Reference:

YAML and JSON Documentation: The official specifications for both YAML and JSON emphasize their case sensitivity.

## Question: 44

Which two statements are correct about a Python dictionary data type? (Choose two.)

A. The data contained in a dictionary data type cannot be removed once the dictionary has been created.

B. The data stored in a dictionary data type is sequenced and indexed.

C. The data contained in a dictionary data type is a key/value pair.

D. The data stored in a dictionary data type is not sequenced or indexed.

Answer: C, D

Explanation:

A Python dictionary is a data type that stores data in the form of key/value pairs. It has the following characteristics:

Key/Value Pair (C): Each entry in a dictionary is a pair consisting of a unique key and a value. The key is used to access the corresponding value.

Not Sequenced or Indexed (D): Unlike lists or tuples, dictionaries do not maintain order for their entries (in versions prior to Python 3.7). Even though Python 3.7+ maintains insertion order, dictionaries are not considered indexed or sequenced in the traditional sense like lists, where elements are accessed via positional index.

Option A is incorrect because dictionary entries can be added, modified, or removed after the dictionary is created. Option B is incorrect because dictionaries are not accessed by a numeric index but rather by their keys.

Reference:

Python Official Documentation: Details the nature of dictionaries, including their mutability and key/value structure.

Python Data Structures Guide: Explains dictionary operations and characteristics.

## Question: 45

Which two programming languages have a NETCONF library supported by Juniper Networks? (Choose two.)

A. Ruby

B. Python

C. Go

D. SLAX

Answer: B, C

Explanation:

Juniper Networks supports NETCONF libraries for several programming languages, including:

Python (B): Python has a well-supported NETCONF library called ncclient, which is widely used for automating network configurations across Junos devices.

Go (C): Go also has a NETCONF library (go-netconf), which provides similar functionalities for managing Junos devices.

Ruby (A) and SLAX (D) do not have widely recognized or supported NETCONF libraries directly from Juniper Networks, making Python and Go the correct choices.

Reference:

Juniper Networks NETCONF Documentation: Lists supported programming languages and libraries for interacting with NETCONF on Junos devices.

ncclient Documentation: The primary Python library for working with NETCONF.

## Question: 46

Which two statements are correct about a Python list data type? (Choose two.)

A. The data contained in a list data type can be modified.

B. The data contained in a list data type is sequenced and indexed starting from 0.

C. The data contained in a list data type cannot be modified.

D. The data contained in a list data type is not sequenced or indexed.

Answer: A, B

Explanation:

Python lists have the following characteristics:

Modifiable Data (A): Lists are mutable, meaning you can change, add, or remove elements after the list has

been created.

Sequenced and Indexed (B): Lists maintain the order of their elements and are indexed starting from 0. This means you can access elements by their position in the list.

Option C is incorrect because lists are mutable, allowing modifications. Option D is incorrect because lists are indeed sequenced and indexed, unlike dictionaries.

Reference:

Python Official Documentation: Covers the properties of lists, including mutability and indexing.

Python Data Structures Guide: Explains list operations and how to manipulate them.

## Question: 47

Which two statements are correct about the Junos REST API Explorer? (Choose two.)

A. The REST API Explorer is enabled by default on all Junos devices.

B. The REST API Explorer returns data only in XML format.

C. The REST API Explorer supports both GET and POST calls.

D. The REST API Explorer supports multiple RPC calls.

Answer: C, D

Explanation:

The Junos REST API Explorer provides an interactive environment to explore and execute REST API calls. The correct statements are:

Supports GET and POST Calls (C): The REST API Explorer allows users to make both GET and POST requests, enabling retrieval and submission of data to the Junos device.

Supports Multiple RPC Calls (D): The REST API Explorer can execute multiple RPC (Remote Procedure Call) commands, allowing a wide range of operations to be performed directly through the interface.

Option A is incorrect because the REST API Explorer is not enabled by default; it must be enabled manually. Option B is incorrect because the REST API Explorer returns data in both XML and JSON formats, not just XML.

Reference:

Junos REST API Explorer Documentation: Provides details on the supported operations and how to use the Explorer for different types of requests.

Juniper Networks Documentation: Covers the setup and usage of the REST API Explorer.

## Question: 48

Which two statements about NETCONF are true? (Choose two.)

A. It uses the operations layer to lock the configuration of a Junos device.

B. It uses the messages layer to commit the configuration of a Junos device.

C. It uses the messages layer to lock the configuration of a Junos device.

D. It uses the operations layer to commit the configuration of a Junos device.

Answer: A, D

Explanation:

NETCONF (Network Configuration Protocol) operates through different layers, with the operations layer being particularly important for managing configurations:

Operations Layer (A & D): This layer is responsible for actions like locking and committing the configuration on a Junos device. The lock operation prevents other sessions from modifying the configuration, and the commit operation applies the configuration changes to the device.

Options B and C are incorrect because the messages layer handles the communication aspects, such as exchanging data between the client and server, not performing configuration operations like locking and committing.

Reference:

IETF RFC 6241 (NETCONF): Describes the protocol layers and their functions, with a focus on the operations layer.

Juniper Networks NETCONF Documentation: Provides insights into how NETCONF operations are managed in Junos

## Question: 49

Which feature is used in XML to ensure that all attributes and elements have unique names?

A. selectors

B. predicate

C. namespace

D. XPath

Answer: D

Explanation:

## Question: 50

Exhibit.

W Exhibit

```
[edit oystcm services) userGrouter# show rest (
    https { server-certificate certl;
```

Referring to the exhibit, which statement about REST is correct?

A. The device is configured to allow only two simultaneous REST connections.

B. The device is configured to allow connections from the REST API Explorer

C. The device is configured to allow HTTP connections on port 3030.

D. The device is configured to allow HTTPS connections on port 3443.

Answer: D

Explanation:

## Question: 51

What is an example of correct XML syntax?

A. <Device3><ho3tname>vMX1</hostname><Devices/>

B. <Devices><hostname>vMX1<De vices/X/hostname>

C. <Devices><hostname>vMX1<Devices/>

D. <Devices><hostname>vMX1</ho3tnameX/Device3>

Answer: B

Explanation:

## Question: 52

You want to perform a dry run on the myPlays playbook and use a custom inventory file called myRouters.ini.

Which Ansible command would you use in this scenario?

A. ansible-playbook myPlays --check -i myRouters.ini

B. ansible-playbook myPlays —extra-vars "inventory_file=myRouters .ini"

C. ansible-playbook myPlays —extra-vars "dry run=True" myRouters.ini

D. ansible-playbook myPlays —limit myRouters

Answer: B

Explanation:

## Question: 53

What will determine which data serialization format is used for interfacing with a system?

A. the system's API

B. the operating system

C. programming language version

D. IDE specifications

Answer: A

Explanation:

## Question: 54

Which two statements about Ansible are correct? (Choose two.)

A. Ansible can use a single playbook to configure devices from multiple vendors.

B. Ansible modules must be written in Go.

C. Ansible is an open source project.

D. Ansible requires a license to unlock full functionality.

Answer: A, C

Explanation:

## Question: 55

Why is a REST API considered stateless?

A. The client requests to the server do not include state information.

B. The REST API is an international API.

C. No client context is stored on the server between requests.

D. The state of the server is not important when making requests.

Answer: A

Explanation:

## Question: 56

Exhibit.

```
rest { http (
        port 3000;

    traceoptions ( flag all;

    enable-explorer;

web-management { http;
    https ( system-generated-certificate;


user@router> show configuration interfaces ge-0/0/0 {
    unit 0 {
        family inet6 ( address 2001:db8:0:1::/64;

}
```

You have the configuration shown in the exhibit, but are unable to access the REST API Explorer.

Which configuration is necessary to solve this problem?

A. Configure the rest service to be active on port 80.
B. Configure an IPv4 address on ge-0/0/0 0.
C. Configure a secondary IPv6 address on ge-0/0/0.0.
D. Configure a firewall filter to explicitly permit SSH.

Answer: C

Explanation:

## Question: 57

Exhibit.

```
*l/usr/bin/python

from jnpr.junw import Device
from pprint impart pprint
from Ixml import etree

    with Device ("172.25.11.1", user="lnb", pa33wd="labl23") as dev: config_datn =
dev.rpc.get_config(filter_xml=”interfaces", options-("format": "json"))

        print("Configured interfaces:")
        for interface in configdata("configuration"]("interfaces"] (’’interface”) :
            print(interface("name"])
```

Referring to the exhibit, which two statements about the script are correct? (Choose two.)

A. The script prints the name of each configured interface.

B. The script retrieves the interface configuration in XML

C. The script prints interface information for each interface name.

D. The script retrieves the interface configuration in JSON.

Answer: A, D

Explanation:

## Question: 58

What is the default port for NETCONF connections over SSH?

A. 22

B. 8080

C. 830

D. 433

Answer: C

Explanation:

https://www.juniper.net/documentation/us/en/software/junos/netconf/topics/topic-map/netconf-ssh-connection.html

The IANA-assigned port for NETCONF-over-SSH sessions is 830.

## Question: 59

Which two programing languages would be used for on-box scripting with Junos devices? (Choose two.)

A. Python

B. Puppet

C. Ansible

D. XSLT

Answer: A, D

Explanation:

## Question: 60

Which HTTP status code indicates a response to a successful request?

A. 500

B. 302

C. 200

D. 400

Answer: C

Explanation:

Explanation

Reference: https://www.juniper.net/documentation/en_US/junos-space-sdk/13.1/apiref/com.juniper.junos_space.sdk.help/html/reference/Commonbehav.html

## Question: 61

Your organization is developing an application to automate management of Junos network appliances. You want to use the existing PyEZ libraries to improve the development process. Which API would satisfy this requirement?

A. REST API

B. JETAPI

C. RPC API

D. XML API

Answer: A

Explanation:

## Question: 62

Exhibit.

Referring to the exhibit, which XPath statement would retrieve the commit-user attribute in the candidate configuration?


A. configuration/@ junos: commit-user

B. configuration:@ junos: commit-user

C. configuration/junos: commit-user

D. configuration/commit-user

Answer: B

Explanation:

## Question: 63

Which statement is correct about Ansible playbooks?

A. A playbook is a configuration file that defines the Ansible related parameters

B. A playbook contains one or more tasks written in XML

C. A playbook can contain multiple tasks and execute multiple Python modules

D. A playbook is a specific Python module that is executed on a host

Answer: C

Explanation:

## Question: 64

Which development model is the classic approach to software development?

A. Waterfall

B. Scrum

C. Kanban

D. Lean

Answer: A

Explanation:

## Question: 65

Exhibit.

```
<rpc-reply xmlns:junos="http://xml.juniper.net/junos/20.1R0/junos">
  ■(configuration junos:commit—seconds="1589295982" junos:commit—localtime="2020—05—12 15:06:22 UTC"
junos:commit-user="user">
            <interfaces>
                  <interface>
                     <name>ge-0/0/l</name>
                     <unit>
                        <name>O</name>
```

```
                        <family>
                           <inet>
                              <address>
                                 <name>192.168.10.1/24</name>
                              </address>
                           </inet>
                        </family>
                     </unit>
                  </interface>
                  <interface>
                     <name>fxp0</name>
                     <unit>
                        <name>0</name>
                        <family>
                              <inet>
                                 <address>
                                 <name>172.25.11.1/24</name>
                              </address>
                           </inet>
                        </family>
                     </unit>
                  </interface>
               </interfaces>
            </configuration>
         </rpc-reply>
```

Referring to the exhibit, which XML XPath expression will only show the IP address XML elements?

A. //address/name

B. //name

C. /name

D. /address/name

Answer: A

Explanation:

https://www.w3schools.com/xml/xpath_syntax.asp

Question: 66

You are using the curl tool to include multiple RPCs in a REST API request.

Which HTTP method must be used for this request?

A. GET

B. PUT

C. POST

D. HEAD

Answer: C

Explanation: