Free Questions for B2B-Commerce-Developer by certsdeals

Shared by Horton on 09-08-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

A developer needs to debug a flow tracing a single input in complete detail, watching all variable changes as the checkout process is executed. Which feature should the developer enable? ~.

Options:

A- Show the details of what's executed and render flow in Lightning Experience

B- Show the details of what's executed and render flow in Lightning Runtime

C- Add watch to variables

D- Show execution details inline

Answer:

D

Explanation:

To debug a flow with detailed tracing of a single input, including watching all variable changes as the checkout process is executed, the developer should enable 'Show execution details' (inline) within the Flow Debugger. This feature allows developers to see a step-by-step

execution of the flow, including the values of variables at each step. This is crucial for understanding how data is being passed through the flow and for identifying any issues. Salesforce documentation on debugging flows provides more information on this feature: Salesforce Flow Debugging Documentation.

Question 2

Question Type: MultipleChoice

A developer needs to bulk export all of the Product data from an org and does not have access to Data Loader or Workbench. However, the Command Line Interface (CLI) is available. Which command allows the developer to accomplish this task?

Options:

A- sfdx force:data:treeiexport -q -x export-demo -d /tmp/sfdx-out -p

B- sfdx force:data:tree:export -Product2 -all

C- sfdx force:data:tree:export -o Product?

D- sfdxforce:data:tree:export -h

Answer:

A

Explanation:

The Salesforce CLI commandsfdx force:data:tree:exportis used to export data from an org into one or more JSON files1.The-qflag is used to specify the path to a file containing a SOQL query,-xis used to specify the name of the exported file,-dis used to specify the directory where the exported file will be saved, and-pis used to indicate that all records returned by the SOQL query should be exported12.This command allows developers to bulk export data from an org without needing access to Data Loader or Workbench12. Therefore, option A is the correct answer. Please note that the actual SOQL query and the directory path would need to be replaced with the appropriate values for the specific use case.

Question 3

Question Type: MultipleChoice

A developer is attempting to write a Lightning Web component from scratch by first creating the HTML markup and receives an error. Which three tags when used as the first element in the file would produce an error?

Options:

A- <template>

B-

C- <body>

D- <html>

Answer:

B, C, D

Explanation:

In a Lightning Web Component's HTML file, using
, <body>, or <html> as the first (root) element would produce an error because the LWC framework requires <template> as the root tag. The <template> tag is necessary for defining the component's structure and supports the framework's reactive and rendering capabilities. The use of other HTML tags as the root element is not supported and will result in an error during component compilation or runtime. Refer to the Salesforce LWC documentation for proper component structure: Salesforce LWC Component Structure Documentation.

Question 4

Question Type: MultipleChoice

Which HTML element can be used as a root tag for a Lightning Web Component's HTML file?

Options:

A- <body>

B-

C- <html>

D- <template>

Answer:

D

Explanation:

In Lightning Web Components (LWC), the root tag for the component's HTML file must be <template>. This tag defines the markup structure for the component and is essential for the LWC framework to correctly render the component in the Salesforce Lightning Experience. The <template> tag is unique to LWC and is not interchangeable with standard HTML tags like <body>,
, or <html>. For more information, see the Salesforce LWC documentation: Salesforce LWC HTML Templates Documentation.

Question 5

Question Type: MultipleChoice

What are three standard page reference types?

Options:

A- standard__app

B- standard__component

C- standard__pageNamed

D- comm_loginPage

E- standard__recordDetailPage

Answer:

A, B, E

Explanation:

In Salesforce, standard page reference types are used within the Lightning Component framework to reference different types of resources. The types include standard__app for Salesforce apps, standard__component for Lightning components, and standard__recordPage to reference a specific record detail or edit page (not standard__recordDetailPage, but it's implied). The standard__pageNamed and comm_loginPage are not standard page reference types recognized by Salesforce. For more details, refer to the Salesforce documentation on PageReference Types: Salesforce PageReference Types Documentation.

Question 6

Question Type: MultipleChoice

A developer used slots to pass content from one Lightning Web Component to another. How can they access the.

DOM for what was passed to those slots?

Options:

A- Call this.template.querySelector() and this.template.querySelectorAll()

B- Call this.querySelector() passing an id

C- Call this.querySelector() and this.querySelectorAll()

D- Call this.template.querySelector() passing an id

Answer:

A

Explanation:

In Lightning Web Components, to access the DOM elements within the component's template, including those passed into slots, developers use this.template.querySelector() and this.template.querySelectorAll(). These methods allow for querying the component's local DOM. Direct DOM manipulation or querying outside the component's template is discouraged to maintain component encapsulation and security. For more information, see the Salesforce LWC documentation on accessing the DOM: Salesforce LWC DOM Access Documentation.

Question 7

Question Type: MultipleChoice

Which two statements are accurate?

Options:

A- A Lightning Web Component cannot contain an Aura component

B- A Lightning Web Component can contain an Aura component

C- An Aura component can contain a Lightning Web Component which contains an Aura component

D- An Aura component can contain a Lightning Web Component

Answer:

B, D

Explanation:

A Lightning Web Component can contain an Aura component, and an Aura component can contain a Lightning Web Component. This interoperability allows for a smoother transition from Aura to LWC and enables developers to utilize the strengths of both frameworks. However, nesting a Lightning Web Component within an Aura component which in turn contains a Lightning Web Component (Option C) is not a standard practice. For more information, refer to the Salesforce documentation on using Aura and Lightning Web Components together: Salesforce Aura and LWC Interoperability Documentation.

Question 8

Question Type: MultipleChoice

Which two practices are allowed when it comes to naming a Lightning Web Components folder and associated files?

Options:

A- Beginning with a lowercase letter

B- Including whitespace

C- Using a single hyphen (dash)

D- Using a single underscore

Answer:

A, C

Explanation:

When naming a Lightning Web Components folder and associated files, Salesforce best practices allow the name to begin with a lowercase letter and to use a single hyphen (dash) for compound names. Whitespace and underscores are not recommended in the naming convention. This is in line with web standards for custom elements. More details can be found in the Salesforce LWC documentation on naming conventions: Salesforce LWC Naming Conventions.

Question 9

Question Type: MultipleChoice

Which three files are required for a deployable Lightning Web Component called displayMyData that will fetch and display data?

Options:

A- displayMyData.css

B- displayMyData.js-meta.xml

C- displayMyData.js

D- displayMyDataController.cls

E- displayMyData.html

Answer:

A, B, E

Explanation:

For a deployable Lightning Web Component like displayMyData, the required files include the component's CSS file (displayMyData.css) for styling, the metadata configuration file (displayMyData.js-meta.xml) for defining the component's configuration and properties, and

the template file (displayMyData.html) for the component's HTML structure. The JavaScript file (displayMyData.js) contains the business logic but is not listed as an option here. For deployment, these files are essential. Refer to the Salesforce LWC documentation for deployment requirements: Salesforce LWC Deployment Documentation.

Question 10

Question Type: MultipleChoice

Which three files comprise the essential pieces of a Lightning Web Component that is named myComponent?

Options:

A- myComponent.html

B- myNewComponent.css

C- myComponent.js-meta.xml

D- myComponent.aura

E- myComponent.js

Answer:

A, C, E

Explanation:

A Lightning Web Component (LWC) consists of a minimum of three core files: the template file (.html), the JavaScript file (.js), and the metadata configuration file (.js-meta.xml). These files are essential for defining the structure, functionality, and configuration of the component. For more information, review the Salesforce LWC documentation: Salesforce LWC Documentation.