



Free Questions for [AD0-E720](#) by [certsinside](#)

Shared by [Brewer](#) on [04-10-2023](#)

For More Free Questions and Preparation Resources

[Check the Links on Last Page](#)

Question 1

Question Type: MultipleChoice

An Adobe Commerce developer created a new CMS page and set the page title as "My Custom Page". The page must be accessible at the URL /custom_page.

Which CMS page configuration do they set to make the page accessible at /custom_page?

Options:

- A- Under 'Content', they set the 'Path' field as 'custom_page'.
- B- Under 'Search Engine Optimization', they set the 'URL Key' field as 'custom_page'.
- C- Under 'Page in Websites', they set the 'URL' field as 'custom.page'.

Answer:

B

Explanation:

The URL Key field under the Search Engine Optimization section is used to specify the URL suffix for the CMS page. The developer can set it as "custom_page" to make the page accessible at /custom_page. The Path field under the Content section is used to specify the

path to the template file that renders the CMS page content. The URL field under the Page in Websites section does not exist as a valid configuration option for CMS pages. Reference: [Adobe Commerce Developer Documentation], [Adobe Inc.]

Question 2

Question Type: MultipleChoice

An Adobe Commerce developer needs to add a conditional static note depending on whether the order type is virtual or not. Which option would the developer use to add the conditional text in the email template?

A)

```
{{if $order_data.is_not_virtual}}
    {{trans "Your order will be shipped soon."}}
{{else}}
    {{trans "Shipping not required."}}
{{/endif}}
```

B)

```
{{if order_data.is_not_virtual}}
    {{trans "Your order will be shipped soon."}}
{{else}}
    {{trans "Shipping not required."}}
{{/if}}
```

C)

```
{{if order_data.is_not_virtual === 1}}  
  {{trans "Your order will be shipped soon."}}  
{{else}}  
  {{trans "Shipping not required."}}  
{{/if}}
```

Options:

A- Option A

B- Option B

C- Option C

Answer:

B

Explanation:

Option B is the correct way to add a conditional static note depending on whether the order type is virtual or not in the email template. Option B uses the `trans` directive to translate the text and the `depend` directive to check the value of the `order.is_virtual` variable. If the order is virtual, the text "Shipping not required." will be displayed. Option A and Option C are incorrect because they use the wrong syntax for the `trans` and `depend` directives. Option A uses curly braces instead of parentheses for the `trans` directive and does

not use quotes for the text. Option C uses parentheses instead of curly braces for the `{{depend}}` directive and does not use a dot to access the `order.is_virtual` variable.

Question 3

Question Type: MultipleChoice

An Adobe Commerce developer wants to determine which template is rendering a specific element on the storefront. Which two methods can they use to turn on template hints? (Choose two.)

Options:

- A-** In the Admin, navigate to Stores > Configuration > Advanced > Developer > Debug.
Set Enabled Template Path Hints for Storefront to Yes
- B-** In the Admin, navigate to system > Advanced > Template > Developer > Debug.
Set Enabled Template Path Hints for Storefront to Yes
- C-** On the command line, run the command; `php bin/magento setup:enable-template-hints`
- D-** On the command line, run the command; `php bin/magento dev:template-hints:enable`

Answer:

A, D

Explanation:

Template hints are a useful tool for frontend developers to determine which template is rendering a specific element on the storefront.

Template hints can be turned on using either of these two methods:

In the Admin Panel, navigate to Stores > Configuration > Advanced > Developer > Debug. Set Enabled Template Path Hints for Storefront to Yes

On the command line, run the command; `php bin/magento dev:template-hints:enable` The other two options are incorrect because they do not exist as valid methods to turn on template hints. Reference: [Adobe Commerce Developer Documentation](#), Adobe Inc.

Question 4

Question Type: MultipleChoice

An Adobe Commerce developer needs to improve the time of first render of the pages. How would the developer achieve this?

Options:

- A- Use the quick static files deployment strategy
- B- Enable CSS critical path
- C- Enable CSS file merging
- D- Enable JavaScript minification

Answer:

B

Explanation:

CSS critical path is a feature that improves the time of first render of the pages by inlining the CSS rules that are required to render the above-the-fold content of the page. This reduces the number of requests and bytes that need to be downloaded before the page is rendered. CSS critical path can be enabled in the Admin Panel by navigating to Stores > Configuration > Advanced > Developer > CSS Settings and setting Enable CSS Critical Path to Yes. Reference:Adobe Commerce Developer Documentation,Adobe Inc.

Question 5

Question Type: MultipleChoice

An Adobe Commerce developer needs to add CMS content above products on a specific category page via the Admin Panel. Where would the developer need to put the content in order to use Display Modes functionality and display it on the category?

Options:

A- Widget

B- CMS Page

C- CMS Block

Answer:

C

Explanation:

CMS Blocks are the best option to add CMS content above products on a specific category page via the Admin Panel. CMS Blocks are reusable pieces of content that can be inserted into any page or layout using widgets or layout XML. CMS Blocks can be assigned to specific categories using the Display Settings tab in the category edit page. The developer can choose the Display Mode for the category, which determines how the CMS Block and the products are displayed on the category page. For example, the developer can choose Static block and products to show both the CMS Block and the products, or Static block only to show only the CMS Block and no products. Reference: [Adobe Commerce Developer Documentation](#), Adobe Inc.

Question 6

Question Type: MultipleChoice

By creating a Custom_Module, an Adobe Commerce Developer has implemented a new Page Builder viewport for tablet devices but the viewport's tablet selector button is missing.

The button .svg has been properly added to the path: Custom_Module/web/css/images/switcher/switcher-tablet .svg. How the developer would implement the viewport button icon?

A)

By setting the node icon in the theme's etc/view.xml file for the respective viewport configuration data.

```
<view>
  <vars module="Magento_PageBuilder">
    <var name="breakpoints">
      <var name="tablet">
        <!-- ... -->
        <var name="icon">Custom_Module::css/images/switcher/switcher-tablet.svg</var>
        <!-- ... -->
      </var>
    </var>
  </vars>
</view>
```

B)

By setting the node button-image in the theme's etc/view.xml file for the respective viewport configuration data.

```
<view>
  <vars module="Magento_PageBuilder">
    <var name="breakpoints">
      <var name="tablet">
        <!-- ... -->
        <var name="button-image">Custom_Module::css/images/switcher/switcher-tablet.svg</var>
        <!-- ... -->
      </var>
    </var>
  </vars>
</view>
```

C) By adding the node image in the theme's etc/viewport.xml file for the respective viewport configuration data.

```
<viewport>
  <vars module="Magento_PageBuilder">
    <var name="breakpoints">
      <var name="tablet">
        <!-- ... -->
        <var name="image">PageBuilder_Breakpoints::css/images/switcher/switcher-tablet.svg</var>
        <!-- ... -->
      </var>
    </var>
  </vars>
</viewport>
```

D)

By adding the node image in the theme's etc/viewport.xml file for the respective viewport configuration data.

```
<viewport>
<vars module="Magento_PageBuilder">
  <var name="breakpoints">
    <var name="tablet">
      <!-- ... -->
      <var name="image">PageBuilder_Breakpoints::css/images/switcher/switcher-tablet.svg</var>
      <!-- ... -->
    </var>
  </var>
</vars>
</viewport>
```

Options:

- A- Option A
- B- Option B
- C- Option C
- D- Option D

Answer:

C

Explanation:

Option C is the correct way to implement the viewport button icon. The image node specifies the path to the .svg file relative to the web/css directory of the module. Option A is incorrect because there is no icon node in the viewport configuration data. Option B is incorrect because there is no button-image node in the viewport configuration data. Option D is incorrect because the image node value should not include the web/css part of the path. <https://api.flutter.dev/flutter/widgets/PageView-class.html>

Question 7

Question Type: MultipleChoice

An Adobe Commerce developer needs to apply a Knockout binding to show content under certain conditions.

Which two syntaxes would achieve this? (Choose two.)

A)

```
<div class="someClass" when="isVisible">
  <transatetext translate="'Some translatable message!'"></transatetext >
  <span html="content"></span>
</div>
```

B)

```
<div class="someClass" if(!isVisible) { 'style:display:none' }>
  <span translate="'Some translatable message!'"></span>
  <span html="content"></span>
</div>
```

C)

```
<if args="isVisible">
  <div class="someClass">
    <translate args="'Some translatable message!'" />
    <span html="content"></span>
  </div>
</if>
```

D)

```
<div class="someClass" if="isVisible">
  <span translate="'Some translatable message!'"></span>
  <span html="content"></span>
</div>
```

Options:

A- Option A

B- Option B

C- Option C

D- Option D

Answer:

A, C

Explanation:

Option A and Option C are both valid ways to apply a Knockout binding to show content under certain conditions. Option A uses the visible binding, which sets the display style of the element to none if the value is false. Option C uses the if binding, which removes or inserts the element from the DOM based on the value. Option B and Option D are incorrect because they use invalid syntax for Knockout bindings. Option B uses a colon instead of an equal sign to assign the value, and Option D uses a single quote instead of a double quote to enclose the value.

<https://knockoutjs.com/documentation/binding-syntax.html>

<https://knockoutjs.com/documentation/binding-context.html>

Question 8

Question Type: MultipleChoice

An Adobe Commerce developer wants to initialize a JS component via Layout XML in custom reference block test.component. Which Layout XML instruction would be used to initialize this JS component?

A)

```
<referenceBlock name="test.component">
  <arguments>
    <argument name="jsComponents" xsi:type="array">
      <!-- Your customization will be here -->
      ...
    </argument>
  </arguments>
</referenceBlock>
```

B)

```
<referenceBlock name="test.component">
  <arguments>
    <argument name="components" xsi:type="array">
      <!-- Your customization will be here -->
      ...
    </argument>
  </arguments>
</referenceBlock>
```

C)

```
<referenceBlock name="test.component">
  <arguments>
    <argument name="jsLayout" xsi:type="array">
      <!-- Your customization will be here -->
      ...
    </argument>
  </arguments>
</referenceBlock>
```

Options:

- A- Option A
- B- Option B
- C- Option C

Answer:

A

Explanation:

Option A is the correct way to initialize a JS component via Layout XML in a custom reference block. The `x-magento-init` tag is used to specify the component name and the options. Option B is incorrect because it uses the `data-mage-init` attribute, which is only valid for HTML elements. Option C is incorrect because it uses the `x-magento-component` tag, which is deprecated and should not be used.

<https://developer.adobe.com/commerce/frontend-core/>

<https://experienceleague.adobe.com/docs/certification/program/technical-certifications/ac/ac-expert/ac-e-fedevolver0623.html?lang=en>

Question 9

Question Type: MultipleChoice

An Adobe Commerce developer wants to create a sticky widget.

How would the widget be initialized using the data-mage-init attribute?

A)

```
<div class="block-item-summary"
  data-mage-init='{ "sticky": ".main-item-form" }'>
  [...]
</div>
```

B)

```
<div class="block-item-summary"
  data-mage-init='{ "sticky": { "container": ".main-item-form" } }'>
  [...]
</div>
```

C)

```
<div class="block-item-summary"  
  data-mage-init='{ "sticky": {"element": ".main-item-form"} }'>  
  [...]  
</div>
```

Options:

A- Option A

B- Option B

C- Option C

Answer:

B

Explanation:

Option B is the correct way to initialize a widget using the data-mage-init attribute. The attribute value should be a JSON object with the widget name as the key and the widget options as the value. Option A is incorrect because it uses a string instead of an object for the attribute value. Option C is incorrect because it uses a colon instead of a comma to separate the widget name and the options.

Question 10

Question Type: MultipleChoice

An Adobe Commerce developer is implementing a sticky sidebar using a jQuery widget. How would the developer initialize the block in a JavaScript file?

A)

```
$('.sticky-sidebar').sticky({  
  container: '.sticky-parent'  
});
```

```
$.mage.widget({  
  sticky: '.sticky-sidebar'  
});
```

C)

```
$('.sticky-sidebar').stickyWidget({  
  container: '.sticky-parent'  
});
```

Options:

A- Option A

B- Option B

C- Option C

Answer:

C

Explanation:

Option C is the correct way to initialize a jQuery widget in a JavaScript file. The widget name should be prefixed with "mage" and the options should be passed as an object literal. Option A is incorrect because it uses a dot notation instead of a colon to separate the widget name and the options. Option B is incorrect because it uses a string instead of an object literal to pass the options.

<https://experienceleague.adobe.com/docs/certification/program/technical-certifications/ac/ac-expert/ac-e-fedevolver0623.html?lang=en>

<https://developer.adobe.com/commerce/docs/>

To Get Premium Files for AD0-E720 Visit

<https://www.p2pexams.com/products/ad0-e720>

For More Free Questions Visit

<https://www.p2pexams.com/adobe/pdf/ad0-e720>

