



Free Questions for [AD0-E720](#) by [go4braindumps](#)

Shared by [Winters](#) on [15-04-2024](#)

For More Free Questions and Preparation Resources

[Check the Links on Last Page](#)

Question 1

Question Type: MultipleChoice

An Adobe Commerce developer created a module called Orange_Customer. In customer information.

Where would the developer place this file?

Options:

A- app/code/OrangG/customer/viev^/frontend/web/templates/customer-info.phtml

B- app/code/Orange/Customer/frontend/templates/customer-info.phtml

C- app/code/OrangG/Custon>Gr/viGw/frontGnd/templates/customGr-info.phtml

Answer:

C

Explanation:

To place a template file for a custom module, the developer should follow this path pattern:

app/code/<Vendor>/<Module>/view/<Area>/templates/<Template>

In this case, the vendor name is Orange, the module name is Customer, the area is frontend, and the template name is customer-info.phtml. Therefore, the correct path is:

app/code/Orange/Customer/view/frontend/templates/customer-info.phtml

The following paths are not correct and will not work:

app/code/Orange/customer/view/frontend/web/templates/customer-info.phtml: This path is incorrect because it uses web instead of templates, which is used for storing web assets like CSS, JS, and images, not template files.

app/code/Orange/Customer/frontend/templates/customer-info.phtml: This path is incorrect because it misses the view directory, which is required for separating frontend and backend templates.

Question 2

Question Type: MultipleChoice

An Adobe Commerce developer needs to pass JSON data to a JavaScript component while keeping XSS prevention strategies in mind.

Which two options would the developer use? (Choose two.)

A)

```
<script>
let settings = <?=$myJson ?>
</script>
```

```
<div data-bind='settings: <?=$escaper->escapeHtmlAttr($myJson) ?>'></div>
```

C)

```
<script>
let settings = <?=$escaper->escapeHtmlAttr($myJson) ?>
</script>
```

D)

```
<div data-bind='settings: <?=$myJson ?>'></div>
```

Options:

A- Option A

B- Option B

C- Option C

D- Option D

Answer:

A, C

Explanation:

To pass JSON data to a JavaScript component while keeping XSS prevention strategies in mind, the developer should use the following options:

Option A: Use the `x-magento-init` script tag with the `data-mage-init` attribute and the `JSON.parse` function to initialize the component with the JSON data. This option is secure because it does not use any HTML tags or attributes that can be exploited by XSS attacks.

Option C: Use the `text/x-magento-init` script tag with the `type` attribute and the `JSON.parse` function to initialize the component with the JSON data. This option is secure because it does not use any HTML tags or attributes that can be exploited by XSS attacks.

The following options are not secure and should not be used:

Option B: Use the `script` tag with the `type` attribute and the `escapeHtmlAttr` function to initialize the component with the JSON data. This option is not secure because it uses the `escapeHtmlAttr` function, which is meant for escaping HTML attributes, not JSON data. This function can introduce double quotes in the JSON data, which can break the JSON syntax and cause errors.

Option D: Use the `script` tag with the `type` attribute and the `escapeJsQuote` function to initialize the component with the JSON data. This option is not secure because it uses the `escapeJsQuote` function, which is meant for escaping JavaScript strings, not JSON data. This function can introduce backslashes in the JSON data, which can break the JSON syntax and cause errors.

Question 3

Question Type: MultipleChoice

An Adobe Commerce developer is using a view model within an existing block:

```
<referenceBlock name="blog.posts.list">
  <arguments>
    <argument name="view_model" xsi:type="object">ExampleCorp\Blog\ViewModel\MyNewViewModel</argument>
  </arguments>
</referenceBlock>
```

What are two ways to access the view model class in the template? (Choose two.)

Options:

- A- `$block->getData('view_model')`
- B- `$block->viewModel()`
- C- `$block->getViewHodel()`
- D- `$block->getData('viewModel')`

Answer:

A, D

Explanation:

To access a view model within an existing block, the developer can use either of the following ways:

`$block->getData('view_model')`: This method will return the view model object that is assigned to the argument name "view_model" in the layout XML file. For example:

```
<referenceBlock name="blog_posts_list"> ExampleObjectModel/ExampleObjectModel </referenceBlock>
```

In the template file, the developer can access the view model object by using:

```
$block->getData('view_model')
```

`$block->getData('viewModel')`: This method will return the view model object that is assigned to the argument name "viewModel" in the layout XML file. For example:

```
<referenceBlock name="blog_posts_list"> ExampleObjectModel/ExampleObjectModel </referenceBlock>
```

In the template file, the developer can access the view model object by using:

```
$block->getData('viewModel')
```

The following methods are not valid and will not work:

`$block->viewModel()`: This method does not exist and will cause an error.

`$block->getViewHodel()`: This method is misspelled and will cause an error.

Question 4

Question Type: MultipleChoice

An Adobe Commerce developer wants to override the template assigned to a block named existing, product, block. This relationship is defined in the catalog_product_view.xml layout in the Magento_Catalog module. They cannot simply override the file in their theme, as this change is part of a feature that is being released to the marketplace as a module called "Orange_CustomProduct".

The developer has already created the desired template at app/code/Orange/CustomProduct/view/frontend/templates/custom-product-block.phtml.

What can they add to app/code/Orange/CustomProduct/view/frontend/layout/catalog_product_view.xml in their module to accomplish this?

A)

```
<referenceBlock name="existing.product.block">
  <arguments>
    <argument name="template" xsi:type="string">Orange_CustomProduct::custom-product-block.phtml</argument>
  </arguments>
</referenceBlock>
```

B)


```
<referenceBlock name="existing.product.block">
  <template path="Orange_CustomProduct::custom-product-block.phtml" />
</referenceBlock>
```

C)

```
<referenceBlock name="existing.product.block">
  <action method="setTemplate">
    <argument name="template" xsi:type="string">Orange_CustomProduct::custom-product-block.phtml</argument>
  </action>
</referenceBlock>
```

D)

```
<referenceBlock name="existing.product.block" template="Orange_CustomProduct::custom-product-block.phtml">
```

Options:

A- Option A

B- Option B

C- Option C

D- Option D

Answer:

B

Explanation:

To override the template assigned to a block in a module, the developer needs to use the `<referenceBlock>` layout instruction with the `name` attribute specifying the name of the block and the `template` attribute specifying the path to the new template file. In this case, the code would be:

```
<referenceBlock name="existing.product.block" template="Orange_CustomProduct::custom-product-block.phtml"/>
```

Option A is not valid because it uses `<block>` instead of `<referenceBlock>`, which would create a new block instead of referencing an existing one. Option C is not valid because it uses `<template>` instead of `<referenceBlock>`, which would not change the template of the block. Option D is not valid because it uses an incorrect syntax for the `template` attribute, which should use two colons instead of a slash. Reference: [Layout instructions], [Override templates and layout files]

Question 5

Question Type: MultipleChoice

An Adobe Commerce Developer is adding a new page layout to the theme directory within a custom theme. Which file needs to be created to register the new page layout?

Options:

- A- app/design/frontend/<VendorName>/<ThemeName>/layouts. xml
- B- app/design/frontend/<VendorName>/<ThemeName>/Magento_Theme/layouts.xml
- C- app/design/frontend/<VendorName>/<ThemeName>/Magento_Theme/layout/layouts.xml

Answer:

C

Explanation:

To register a new page layout in a custom theme, the developer needs to create a layouts.xml file in the app/design/frontend/<VendorName>/<ThemeName>/Magento_Theme/layout directory. The layouts.xml file should contain the <layout> element with the id, label, and file attributes. The id attribute is used to reference the layout in other layout files, the label attribute is used to display the layout name in the admin panel, and the file attribute is used to specify the path to the layout file relative to the web directory of the theme. The app/design/frontend/<VendorName>/<ThemeName>/layouts.xml and app/design/frontend/<VendorName>/<ThemeName>/Magento_Theme/layouts.xml files are not valid and will not work. Reference: [Create a new page layout], [layouts.xml]

Question 6

Question Type: MultipleChoice

An Adobe Commerce developer has been asked to move a block called country from the container sidebar to the container content, the block has to be the last child on the content container.

Which layout instruction would be used to move the block?

Options:

A- `<move element = 'country' destination='content' after='-'/>`

B- `<move element='country-element' destination='content-element'/>`

C- `<move element='country' destination='content' after="last-child"/>`

Answer:

A

Explanation:

To move a block from one container to another, the developer needs to use the `<move>` layout instruction with the `element` attribute specifying the name of the block and the `destination` attribute specifying the name of the container. The `after` attribute can be used to position the block relative to other blocks in the same container. The value `"-"` means that the block will be placed after all other blocks, i.e., as the last child. The value `"last-child"` is not valid for the `after` attribute and will not work. The `element` and `destination` attributes

should use the names of the blocks and containers, not their aliases or classes. Reference: [Layout instructions], [Move an element]

Question 7

Question Type: MultipleChoice

An Adobe Commerce developer needs to modify the width and height of all product images inside the theme Vendor/theme. What file inside the theme is responsible for these changes?

Options:

- A- Vendor/theme/etc/images.xml
- B- Vendor/theme/etc/view.xml
- C- Vendor/theme/etc/theme.xml

Answer:

B

Explanation:

To modify the width and height of all product images inside a theme, the developer needs to edit the view.xml file inside the etc directory of the theme. The view.xml file contains the configuration for the theme's images, fonts, and layout. The images.xml file does not exist by default and is not used for configuring images. The theme.xml file is used for specifying the parent theme and other metadata of the theme. Reference: [view.xml], [theme.xml]

Question 8

Question Type: MultipleChoice

Which two steps are required to delete a manually installed theme? (Choose two.)

Options:

- A- Remove the theme using the theme:uninstall CLI command
- B- Remove the directory app/design/frontend/<VendorNAME/<ThemeName>
- C- Disable the theme from the backend admin configuration
- D- Remove the theme record from the theme database table

Answer:

B, D

Explanation:

To delete a manually installed theme, the developer needs to remove the theme directory from the app/design/frontend directory and also delete the corresponding record from the theme table in the database. The theme:uninstall CLI command is only used for deleting themes that are installed as Composer packages. Disabling the theme from the backend admin configuration does not delete the theme files or records, but only makes it unavailable for use. Reference: [Delete a theme], [theme:uninstall]

Question 9

Question Type: MultipleChoice

An Adobe Commerce developer is building a theme Vendor/Orange and needs to customize the header of email templates. Where in the theme does the developer need to place the new template for this customization?

Options:

A- /Magento_Email/templates/override/html/header.html

B- /Magento_Email/email/header.html

C- /Magento_Theme/html/header.html

Answer:

B

Explanation:

To customize the header of email templates, the developer needs to place the new template in the /Magento_Email/email/header.html path of the theme. This will override the default header template from the Magento_Email module. The /Magento_Email/templates/override/html/header.html path is not valid and will not work. The /Magento_Theme/html/header.html path is used for customizing the header of web pages, not emails. Reference: [Customize email templates], [Email templates overview]

Question 10

Question Type: MultipleChoice

In which folder can a custom theme favicon be added?

Options:

A- <your_theme_dir>/web/

B- <your_theme_dir>/Magento_Theme/web/

C- <your_theme_dir>/assets/images

Answer:

B

Explanation:

The favicon can be added to the <your_theme_dir>/Magento_Theme/web/ directory of a custom theme. The favicon should have the name favicon.ico and replace the default one from the Magento_Theme module. The <your_theme_dir>/web/ directory is used for storing web assets that are not specific to any module. The <your_theme_dir>/assets/images directory does not exist by default and is not used for storing favicons. Reference: [Add a favicon], [Theme structure]

Question 11

Question Type: MultipleChoice

An Adobe Commerce developer wants to create a new theme Vendor_Orange which extends from MagentoMum

a. Which file is responsible for specifying the parent theme?

Options:

A- view.xml

B- registration.php

C- theme.xml

Answer:

C

Explanation:

The theme.xml file is responsible for specifying the parent theme of a custom theme. The file should contain the `parent` element with the value of the parent theme's directory, such as MagentoMuma. The view.xml file is used to configure the theme's images, fonts, and layout. The registration.php file is used to register the theme in the system. Reference: [Create a theme], [theme.xml]

Question 12

Question Type: MultipleChoice

An Adobe Commerce developer needs to create translations for the Orange/custom theme. Which directory would the developer place the translations?

Options:

- A- Orange/custom/etc
- B- Orange/custom/translations
- C- Orange/custom/il8n

Answer:

C

Explanation:

To create translations for a theme, the developer needs to place the translation files in the il8n directory of the theme. The translation files should have the format <language code>_<country code>.csv, such as en_US.csv or fr_FR.csv. The etc and translations directories are not used for storing translation files. Reference: [Translations overview], [Translate theme strings]

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>

