

Exam Code: AD0-E720

Exam Name: Adobe Commerce Front-End Developer Expert

Exam A

QUESTION 1

In which folder can a custom theme favicon be added?

- A. <your_theme_dir>/web/
- B. <your_theme_dir>/Magento_Theme/web/
- C. <your_theme_dir>/assets/images

Correct Answer: B

Section:

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 2

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?

- A. /Magento_Email/templates/override/html/header.html
- B. /Magento_Email/email/header.html
- C. /Magento_Theme/html/header.html

www.VCEplus.io

Correct Answer: B

Section:

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 3

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

- 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

Correct Answer: B, D

Section:

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 4

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?

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

Correct Answer: B

Section:

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 5

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?

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

Correct Answer: A

Section:

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 6

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?

- 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

Correct Answer: C

Section:

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 7

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">
```

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

Correct Answer: B

Section:

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 instead of <template>, 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 8

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.)

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

Correct Answer: A, D

Section:

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 9

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>
```

- A. Option A
- B. Option B

- C. Option C
- D. Option D

Correct Answer: A, C

Section:

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 10

An Adobe Commerce developer created a module called Orange_Customer. In customer information. Where would the developer place this file?

- A. app/code/OrangG/customer/view^/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

Correct Answer: C

Section:

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 11

An Adobe Commerce developer needs to display a URL in the template. How would the variable \$url be securely output in the template?

- A. <?php echo \$escaper->escapeUrl(\$url) ?>
- B. <?php echo \$escaper->escapeLink(\$url) ?>
- C. <?php echo \$escaper->escapeHtml(\$url) ?>

Correct Answer: A

Section:

Explanation:

To display a URL in a template securely, the developer should use the `escapeUrl` method of the `escaper` object. This method will encode any special characters in the URL that can be used for XSS attacks, such as `&`, `<`, `>`, `'`, `"`, etc. For example:

```
<?php echo $escaper->escapeUrl($url) ?>
```

The following methods are not suitable for displaying URLs and should not be used:

`<?php echo $escaper->escapeLink($url) ?>`: This method is used for escaping link attributes, not URLs. It will encode any characters that are valid in URLs but invalid in HTML attributes, such as spaces, quotes, etc.

For example:

```
<?php echo $escaper->escapeLink('https://example.com/?q=hello world') ?> // Output: https://example.com/?q=hello%20world
```

`<?php echo $escaper->escapeHtml($url) ?>`: This method is used for escaping HTML content, not URLs. It will encode any characters that are valid in URLs but invalid in HTML content, such as `&`, `<`, `>`, etc. For example:

```
<?php echo $escaper->escapeHtml('https://example.com/?q=<script>alert("XSS")</script>') ?> // Output: https://example.com/?q=<script>alert('XSS')</script>
```

QUESTION 12

An Adobe Commerce developer is trying to remove a block using the `<remove name='test.block'/>` tag in a layout XML file but it shows an error and doesn't work.

Which tag would be applied to fix the error?

- A. `US6 <referenceBlock name='test.block' remove='true'/>`
- B. Use `<remove name='Custom_Module:test.block'/>`
- C. Use `<referenceBlock name='test.block' delete='true' ?>`

Correct Answer: A

Section:

Explanation:

To remove a block using layout XML, the developer should use the `<referenceBlock>` tag with the `name` attribute specifying the name of the block and the `remove` attribute set to `true`. For example:

```
<referenceBlock name="test.block" remove="true"/>
```

This will remove the block from the layout and prevent it from rendering. The `<remove>` tag is not valid and will cause an error. The `name` attribute should not include the module name, as it is not part of the block name. The `delete` attribute is not valid and will not work.

Reference: [Layout instructions], [Remove an element]

QUESTION 13

An Adobe Commerce developer wants to remove the default Wishlist and Compare Products blocks on a category page with layered navigation. Where would this modification be placed, assuming the developer only wants to make this change?

- A. `app/design/frontend/Vendor/Theme/Magento_LayeredNavigation/layout/override/catalog_category_view_type_layered.xml`
- B. `app/design/frontend/Vendor/Theme/Magento_Layered.Navigation/layout/catalog_category_view_type_layered.xml`
- C. `app/design/frontend/Vendor/Theme/Magento_Catalog/layout/catalog_category_view.xml`

Correct Answer: B

Section:

Explanation:

To remove the default Wishlist and Compare Products blocks on a category page with layered navigation, the developer should place the modification in the `app/design/frontend/Vendor/Theme/Magento_LayeredNavigation/layout/catalog_category_view_type_layered.xml` file. This file is specific to the category pages with layered navigation and will override the default layout file from the `Magento_LayeredNavigation` module. The modification should use the `<referenceBlock>` tag with the `name` attribute specifying the name of the block and the `remove` attribute set to `true`. For example:

```
<referenceBlock name="catalog.compare.sidebar" remove="true"/> <referenceBlock name="wishlist_sidebar" remove="true"/>
```

The `app/design/frontend/Vendor/Theme/Magento_LayeredNavigation/layout/override/catalog_category_view_type_layered.xml` file is not valid and will not work, as it is not a valid override path. The `app/design/frontend/Vendor/Theme/Magento_Catalog/layout/catalog_category_view.xml` file is not specific to the category pages with layered navigation and will affect all category pages.

Reference: [Layout override], [Remove an element]

QUESTION 14

An Adobe Commerce developer has created a system configuration field:

```
<section id="module" type="text" sortOrder="10" showInDefault="1" showInWebsite="1" showInStore="1">
  <label>Vendor Module</label>
  <tab>vendor</tab>
  <resource>Vendor_Module::store_config</resource>
  <group id="general" type="text" sortOrder="0" showInDefault="1" showInWebsite="1" showInStore="1">
    <label>General</label>
    <field id="enable" translate="label" type="select" sortOrder="0" showInDefault="1" showInStore="1" showInWebsite="1">
      <label>Enable</label>
      <source_model>Magento\Config\Model\Config\Source\Yesno</source_model>
    </field>
  </group>
</section>
```

Using Layout XML, how can the visibility of a block be controlled by a system configuration?

- A. <block name='block.name' template='Vendor_Module::template.phtml' ifconfig='vendor/general/enable' />
- B. <block name='block.name' template='Vendor_Module::template.phtml' ifconfig='module/general/enable' />
- C. <block name='block.name' template='Vendor_Module::template.phtml' ifconfig='general/module/enable' />

Correct Answer: A

Section:

Explanation:

To control the visibility of a block using a system configuration, the developer should use the ifconfig attribute in the <block> tag. The ifconfig attribute should specify the path to the system configuration field that determines whether the block is visible or not. For example:

```
<block name="block.name" template="Vendor_Module::template.phtml" ifconfig="vendor/general/enable" />
```

This will render the block only if the vendor/general/enable system configuration field is set to true. The module/general/enable and general/module/enable paths are not valid and will not work, as they do not match the system configuration field defined in the image.

Reference: [Layout instructions], [System configuration]

QUESTION 15

An Adobe Commerce developer has been asked to add text to an email template that supports translations. Which two options would they use during their implementation? (Choose two.)

- A. {{translations 'Lorem Ipsum is simply dummy text of the printing'}}
- B. {{translations '%items items' items='numItems'}}
- C. {{trans 'Lorem Ipsum is simply dummy text of the printing'}}
- D. {{trans '%items items' items='numItems'}}

Correct Answer: B, D

Section:

Explanation:

To add text to an email template that supports translations, the developer should use the {{trans}} directive with the text enclosed in double quotes. For example:

```
{{trans "Lorem Ipsum is simply dummy text of the printing"}}
```

This will render the text as it is, or translate it if a translation file is available for the current locale. If the text contains a variable, the developer should use a placeholder with a percent sign and pass the variable name as an argument. For example:

```
{{trans "%items items" items="numItems"}}
```

This will render the text with the value of numItems replacing the %items placeholder, or translate it if a translation file is available for the current locale. The {{translations}} directive is not valid and will not work.

Reference: [Translate email templates], [Email template syntax]

QUESTION 16

An Adobe Commerce developer wants to completely overwrite `_module.less` of `Orange_Checkout` module, in their theme. Where would the developer place the file?

- A. `Custom/theme/Orange_Checkout/frontend/web/css/_module.less`
- B. `Custom/theme/web/css/source/Orange_Checkout/_module.less`
- C. `Custom/theme/Orange_Checkout/web/css/source/_module.less`

Correct Answer: C

Section:

Explanation:

To completely overwrite `_module.less` of `Orange_Checkout` module in a custom theme, the developer should place the file in the `Custom/theme/Orange_Checkout/web/css/source` directory. This will override the default `_module.less` file from the `Orange_Checkout` module and apply the custom styles to the theme. The `Custom/theme/Orange_Checkout/frontend/web/css/_module.less` and `Custom/theme/web/css/source/Orange_Checkout/_module.less` paths are not valid and will not work, as they do not follow the theme structure or the module naming convention.

Reference: [Theme structure], [Module naming convention]

QUESTION 17

Where are the Magento UI library LESS files located?

- A. `Magento_Ui/web/css/source/`
- B. `Magento_Lib/web/css/source`
- C. `lib/web/css/source/lib`

Correct Answer: B

Section:

Explanation:

This directory contains various LESS files that define variables, mixins, functions, and styles for common UI elements and components. The `Magento_Ui/web/css/source` and `lib/web/css/source/lib` directories are not valid and do not contain the Magento UI library LESS files.

Reference: [Magento UI library], [Magento UI library source files]

QUESTION 18

An Adobe Commerce developer has found following code:

```
.animation {
  transition: 300ms ease-in-out;
  -moz-transition: 300ms ease-in-out;
  -webkit-transition: 300ms ease-in-out;
  -o-transition: 300ms ease-in-out;
}

.block {
  .animation();
  width: 100%;
}
```

After compiling the `.less` file into a `.css` file, what will be the results of the code above?

A)

```
.animation {
  transition: 300ms ease-in-out;
  -moz-transition: 300ms ease-in-out;
  -webkit-transition: 300ms ease-in-out;
  -o-transition: 300ms ease-in-out;
}
.block {
  width: 100%;
}
```

B)

```
.block {
  transition: 300ms ease-in-out;
  -moz-transition: 300ms ease-in-out;
  -webkit-transition: 300ms ease-in-out;
  -o-transition: 300ms ease-in-out;
  width: 100%;

  .animation {
    transition: 300ms ease-in-out;
    -moz-transition: 300ms ease-in-out;
    -webkit-transition: 300ms ease-in-out;
    -o-transition: 300ms ease-in-out;
  }
}
```

C)

```
.animation {
  transition: 300ms ease-in-out;
  -moz-transition: 300ms ease-in-out;
  -webkit-transition: 300ms ease-in-out;
  -o-transition: 300ms ease-in-out;
}
.block {
  transition: 300ms ease-in-out;
  -moz-transition: 300ms ease-in-out;
  -webkit-transition: 300ms ease-in-out;
  -o-transition: 300ms ease-in-out;
  width: 100%;
}
```

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

Correct Answer: B

Section:

Explanation:

After compiling the .less file into a .css file, the result of the code above will be option B. This is because the .less file uses a mixin called .animation() that takes two parameters: the name of the animation and the duration. The mixin defines a set of vendor-prefixed properties for the animation and assigns them the values of the parameters. For example:

```
.animation(@name; @duration) { -webkit-animation-name: @name; -webkit-animation-duration: @duration; -moz-animation-name: @name; -moz-animation-duration: @duration; animation-name: @name; animation-duration: @duration; }
```

When the mixin is called with the values "fade" and "2s", it will generate the following CSS code:

```
-webkit-animation-name: fade; -webkit-animation-duration: 2s; -moz-animation-name: fade; -moz-animation-duration: 2s; animation-name: fade; animation-duration: 2s;
```

Option A is not correct because it does not use the vendor prefixes for the animation properties. Option C is not correct because it uses the wrong values for the animation name and duration.

Reference: [LESS Mixins], [CSS Animations]

QUESTION 19

An Adobe Commerce developer is extending a theme from Magento\blank and wants to override parent styles. Which file does the developer need to change to override the parent theme styles?

- A. web/css/source/_extends.less
- B. web/css/source/_extend.less
- C. web/css/source/_theme.less

Correct Answer: B

Section:

Explanation:

To override the parent theme styles, the developer needs to change the web/css/source/_extend.less file in the child theme. This file is used to import and extend the parent theme styles without modifying the original files. The developer can use the @import directive to import the parent theme styles and then use the .lib-css() mixin to override the CSS properties. For example:

```
@import 'source/_extend.less'; // Import parent theme styles .lib-css(color, red); // Override color property
```

The web/css/source/_extends.less and web/css/source/_theme.less files are not valid and will not work, as they do not follow the theme structure or the naming convention.

Reference: [Theme inheritance], [Extend parent theme styles]

QUESTION 20

An Adobe Commerce developer has been asked to implement a custom font specifically for emails. The Adobe Commerce developer has already added their font into the file system.

Keeping best practice in mind, which two files would need to be implemented to show the custom font in the email?

- A. /Vendor/Theme/web/css/source/_extend.less Use the ^import font function with the url of the custom font from the theme. /Vendor/Theme/web/css/source/_email.less file
- B. Add in the styles to target the elements that require being changed. /vendor/Theme/web/css/source/_typography.less
- C. Add in a lib-font-face mixin with the custom font name into the newly created file.
- D. Add the font-family into the <head></head> of the email within the email template.

Correct Answer: A, B

Section:

Explanation:

To implement a custom font specifically for emails, the developer needs to do the following steps:

Add the custom font file to the web/fonts directory of the custom theme.

Use the @import font function with the url of the custom font from the theme in the /Vendor/Theme/web/css/source/_extend.less file. This will import the custom font and make it available for use in other LESS files. For example:

```
@import font('custom-font', '{@baseDir}fonts/custom-font.ttf', 'truetype');
```

Add in the styles to target the elements that require being changed in the /Vendor/Theme/web/css/source/_email.less file. This file is used to define the styles for email templates. The developer can use the .lib-font-face() mixin to apply the custom font to specific selectors. For example:

```
.lib-font-face( @family-name: @custom-font, @font-path: '{@baseDir}fonts/custom-font', @font-weight: normal, @font-style: normal );
```

```
h1 { .lib-font-face( @family-name: @custom-font, @font-path: '{@baseDir}fonts/custom-font', @font-weight: normal, @font-style: normal ); }
```

The /vendor/Theme/web/css/source/_typography.less file is not suitable for implementing a custom font for emails, as it is used for defining global typography styles for web pages. The <head></head> tag is not used for adding fonts in email templates, as it is not supported by most email clients.

Reference: [Custom fonts], [Email templates overview]

QUESTION 21

An Adobe Commerce developer is customizing buttons for a custom theme that inherits Magento/blank theme and needs to override the default values. Where would the default values for the buttons be located?

- A. lib/web/css/source/lib/_buttons.less
- B. lib/web/less/source/lib/_buttons.less
- C. lib/web/css/source/lib/_button.less

Correct Answer: A

Section:

Explanation:

To find the default values for the buttons, the developer needs to look at the lib/web/css/source/lib/_buttons.less file. This file contains various variables, mixins, and styles for defining and customizing buttons.

The developer can override these values in their custom theme by using the .lib-button() mixin or by creating their own mixins or classes. For example:

```
.lib-button( @_button-selector, @_button-type, @_button-shape, @_button-color, @_button-background, @_button-border, @_button-text-transform, @_button-box-shadow, @_button-hover-color, @_button-hover-background, @_button-hover-border, @_button-hover-box-shadow );
```

The lib/web/less/source/lib/_buttons.less and lib/web/css/source/lib/_button.less files are not valid and do not exist.

Reference: [Buttons], [Magento UI library]

QUESTION 22

an Adobe commerce developer wants to override the core Magento UI library dropdowns in your theme. Which is the correct way to achieve this?

- A. /web/css/source/_dropdowns.less
- B. lib/web/css/source/.dropdowns.less
- C. /web/css/source/lib/.dropdowns.less

Correct Answer: A

Section:

Explanation:

To override the core Magento UI library dropdowns in a custom theme, the developer needs to create a file named _dropdowns.less in the /web/css/source directory of the theme. This file will override the default _dropdowns.less file from the lib/web/css/source/lib directory and apply the custom styles to the dropdown elements. The lib/web/css/source/_dropdowns.less and /web/css/source/lib/_dropdowns.less files are not valid and will not work, as they do not follow the theme structure or the naming convention.

Reference: [Dropdowns], [Theme structure]

QUESTION 23

What is the difference between styles-l.less and styles-m.less ?

- A. styles-i.less is used to generate basic and mobile-specific styles and styles-m.less is used to generate desktop-specific styles.
- B. styles-i.less is used to generate desktop-specific styles and styles-m.less is used to generate only mobile-specific styles.
- C. styles-i.less is used to generate desktop-specific styles and styles-m.less is used to generate basic and mobile-specific styles.

Correct Answer: C

Section:

Explanation:

The styles-l.less and styles-m.less files are two main LESS files that are used to generate CSS files for different devices and screen sizes. The styles-l.less file is used to generate styles-l.css, which contains desktop-specific styles that are applied only when the screen width is greater than 768px. The styles-m.less file is used to generate styles-m.css, which contains basic and mobile-specific styles that are applied for all devices and screen sizes. The other options are not correct and do not reflect the purpose of these files.

Reference: [Stylesheets], [Responsive web design]

QUESTION 24

www.VCEplus.io

In which mode would the stylesheet customizations display immediately after you reload a page in a browser?

- A. server- side compilation mode
- B. less-side compilation mode
- C. client-side compilation mode

Correct Answer: C

Section:

Explanation:

In client-side compilation mode, the stylesheet customizations will display immediately after reloading a page in a browser. This is because in this mode, the LESS files are compiled into CSS files by the browser using JavaScript. This mode is useful for development and debugging purposes, as it allows quick changes and previews of the styles. However, this mode is not recommended for production, as it can affect the performance and compatibility of the site. The other modes, server-side compilation and LESS compilation, require running commands or tools to compile the LESS files into CSS files on the server side, which can take some time and delay the display of the customizations.

Reference: [Compilation modes], [Compile LESS]

QUESTION 25

An Adobe Commerce developer wants to add a custom widget that extends the default Calendar Widget. What would the contents of this file look like?

A)

```
define([
    'mage/utils/wrapper',
    'mage/calendar'
], function(wrapper, calendarWidget){
    var updateCalendarWidget = wrapper.wrap(targetWidget.prototype._function,
    function (original) {
        calendarWidget({...});
        return original();
    });
    targetWidget.prototype._function = updateCalendarWidget;
    return targetWidget;
});
```

B)

```
define([
    'jquery',
    'jquery-ui-modules/widget',
    'mage/calendar'
], function($){
    $.widget('custom.calendar', $.mage.calendar, { ... });
    return $.custom.calendar;
});
```

C)

www.VCEplus.io

```
define([
    'jquery',
    'mage/calendar'
], function(calendarWidget){
    return calendarWidget.extend({
        _init: function() {
            this._super();
        }
    });
});
```

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

Correct Answer: B

Section:

Explanation:

To add a custom widget that extends the default Calendar Widget, the contents of the file would look like option B. This is because option B follows the correct syntax and structure for defining a jQuery widget in Magento. The code does the following steps:

Defines a module with the name "Vendor_Module/js/calendar-widget" that depends on the "jquery/ui" and "Magento_Ui/js/lib/knockout/bindings/datepicker" modules.

Returns a function that creates a new widget with the name "vendor.calendarWidget" that extends the base calendar widget class.

Overrides the init function of the base calendar widget class to add custom logic or functionality to the widget.

Option A is not correct because it does not use the correct syntax for defining a jQuery widget. It uses a script tag instead of a define function, which is not valid for creating a module. It also uses an incorrect name for the widget, which should use a dot instead of a slash. Option C is not correct because it does not use the correct syntax for extending a widget. It uses an extend function instead of a widget function, which is not valid for creating a new widget. It also does not return anything from the module, which will cause an error.

Reference: [jQuery widgets], [Calendar Widget]

QUESTION 26

An Adobe Commerce developer wants to apply a knockout binding to a to run a function, onClick(), when it's clicked. Which two solutions would the developer use to achieve this? (Choose two.)

- A. <x/div> B <!-- ko click: 'onClick' --> <!-- /ko -->
- B.
- C.

Correct Answer: A, C

Section:

Explanation:

To apply a knockout binding to a to run a function, onClick(), when it's clicked, the developer can use either of the following solutions:

A) : This solution will use the click binding to execute the onClick() function as an expression when the is clicked. The function will be called with the current binding context as the first parameter and the event object as the second parameter.

C) : This solution will use the click binding to execute the onClick function as a reference when the is clicked. The function will be called with the current binding context as the first parameter and the event object as the second parameter.

The following solutions are not valid and will not work:

B) <!-- ko click: 'onClick' --><!-- /ko -->: This solution will use the virtual element syntax to apply the click binding, but it will pass a string literal instead of a function expression or reference. This will cause an error, as the click binding expects a function value.

D) : This solution will use a custom attribute instead of a data-bind attribute to apply the click binding. This is not supported by knockout and will not have any effect.

QUESTION 27

An Adobe commerce developer wants to initialize a JavaScript component using a data attribute. Which option would initialize the JavaScript component?

- A. `<nav data-bind='{<component_name>: {...}}></nav>`
- B. `<nav data-init='{<component_name>: {...}}></nav>`
- C. `<nav data-mage-init='{<component_name>: {...}}></nav>`

Correct Answer: C

Section:

Explanation:

To initialize a JavaScript component using a data attribute, the developer should use the data-mage-init attribute. This attribute allows the developer to specify the name and configuration of the component in a JSON format. For example:

```
<nav data-mage-init='{<Vendor_Module/js/nav>: {"option1": "value1", "option2": "value2"}}'></nav>
```

This will initialize the nav component from the Vendor_Module/js/nav file with the given options. The data-bind and data-init attributes are not valid and will not work, as they are not supported by Magento.

Reference: [JavaScript initialization], [data-mage-init]

QUESTION 28

An Adobe Commerce developer is working on a custom knockout UI component and they need to add the text Happy Birthday. to be translated inside an .html template.

How would the developer add the text?

- A. ``
- B. ``
- C. `<!-- ko i18n = 'Happy Birthday.' --><!-- /ko -->`

Correct Answer: B

Section:

Explanation:

To add the text Happy Birthday. to be translated inside an .html template, the developer should use the i18n binding. This binding allows the developer to specify the text as a string literal and translate it using the Magento translation mechanism. For example:

```
<span data-bind="i18n: 'Happy Birthday.'"></span>
```

This will render the text as it is, or translate it if a translation file is available for the current locale. The i18n binding can also accept variables or expressions as arguments. For example:

```
<span data-bind="i18n: name + ' Happy Birthday.'"></span>
```

This will render the text with the value of name variable, or translate it if a translation file is available for the current locale. The Mil8n and il8n bindings are not valid and will not work, as they are misspelled and do not match the knockout binding syntax.

Reference: [Knockout bindings], [i18n binding]

QUESTION 29

An Adobe Commerce developer needs to alias URLs and third party libraries inside a require js-config.js file. Which configuration would the developer use?

A)

```
paths: {  
  'thirdpartylib': 'library/file',  
  'thirdpartyurl': 'https://some-library.com/file'  
}
```

B)

www.VCEplus.io

```
alias: {
  'thirdpartylib': 'library/file',
  'thirdpartyurl': 'https://some-library.com/file'
}
```

C)

```
map: {
  'thirdpartylib': 'library/file',
  'thirdpartyurl': 'https://some-library.com/file'
}
```

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

Correct Answer: B

Section:

Explanation:

To alias URLs and third party libraries inside a requirejs-config.js file, the developer should use the paths configuration option. This option allows the developer to map module names to URLs or paths relative to the baseUrl. For example:

```
var config = { paths: { 'jquery': 'https://code.jquery.com/jquery-3.6.0.min', 'custom': 'Vendor_Module/js/custom' } };
```

This will map the module name 'jquery' to the URL of the jQuery library and the module name 'custom' to the path of the custom module. The developer can then use these module names in other modules or files without specifying the full URL or path.

Option A is not correct because it uses the shim configuration option, which is used to define dependencies and exports for non-AMD modules, not to alias URLs or paths. Option C is not correct because it uses the map configuration option, which is used to map module names to other module names for specific contexts, not to alias URLs or paths.

Reference: [RequireJS configuration], [RequireJS paths]

www.VCEplus.io

QUESTION 30

An Adobe Commerce developer was asked to customize a JavaScript component which is written as a function. How would the developer extend the native JavaScript function?

A)

```
define([
  'mage/utils/wrapper'
], function (wrapper) {
  'use strict';

  return function (yourCustomFunction) {
    return wrapper.wrap(yourCustomFunction, function (originalFunction) {
      originalFunction();

      // add extended functionality here
    });
  };
});
```

B)


```
define([
  'mage/utils/wrapper'
], function (wrapper) {
  'use strict';

  return function (yourCustomFunction) {
    return wrapper.wrap(yourCustomFunction, function (originalFunction) {
      this._super();

      // add extended functionality here
    });
  });
});
```

C)

```
define([
  'mage/utils/wrapper'
], function (wrapper) {
  'use strict';

  return function (yourCustomFunction) {
    return wrapper.wrap(yourCustomFunction, function (originalFunction, config, element) {
      originalFunction(config, element);

      // add extended functionality here
    });
  });
});
```

www.VCEplus.io

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

Correct Answer: A

Section:

Explanation:

To customize a JavaScript component that is written as a function, the developer can use option A. This option will use the prototype property of the function to extend its functionality and add new methods or properties. For example:

```
function Component() { // Component logic }
```

```
Component.prototype.customMethod = function() { // Custom method logic };
```

This will create a new method called customMethod on the prototype of the Component function, which can be accessed by any instance of the Component object. The developer can also override existing methods or properties on the prototype by reassigning them with new values.

Option B is not correct because it will not extend the native JavaScript function, but create a new function that wraps the original function. This will not allow the developer to access or modify the properties or methods of the original function. Option C is not correct because it will not extend the native JavaScript function, but create a new object that inherits from the original function. This will not allow the developer to customize the original function itself, but only its instances.

QUESTION 31

Which UI component property is used for cross tracking property changes?

- A. exports

- B. listens
- C. links

Correct Answer: B

Section:

Explanation:

The listens property is used for cross tracking property changes in the UI component. The listens property defines the dependencies between the properties of different UI components. It allows one UI component to listen to the changes of another UI component's property and react accordingly. For example, the listens property can be used to update the value of a text field based on the selection of a dropdown menu

QUESTION 32

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'  
});
```

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

Correct Answer: C

Section:

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/>

QUESTION 33

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>
```

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

Correct Answer: B

Section:

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 34

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>
```

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

Correct Answer: A

Section:

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-fedeveloper0623.html?lang=en>

QUESTION 35

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>
```

- A. Option A

- B. Option B
- C. Option C
- D. Option D

Correct Answer: A, C

Section:

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>

www.VCEplus.io