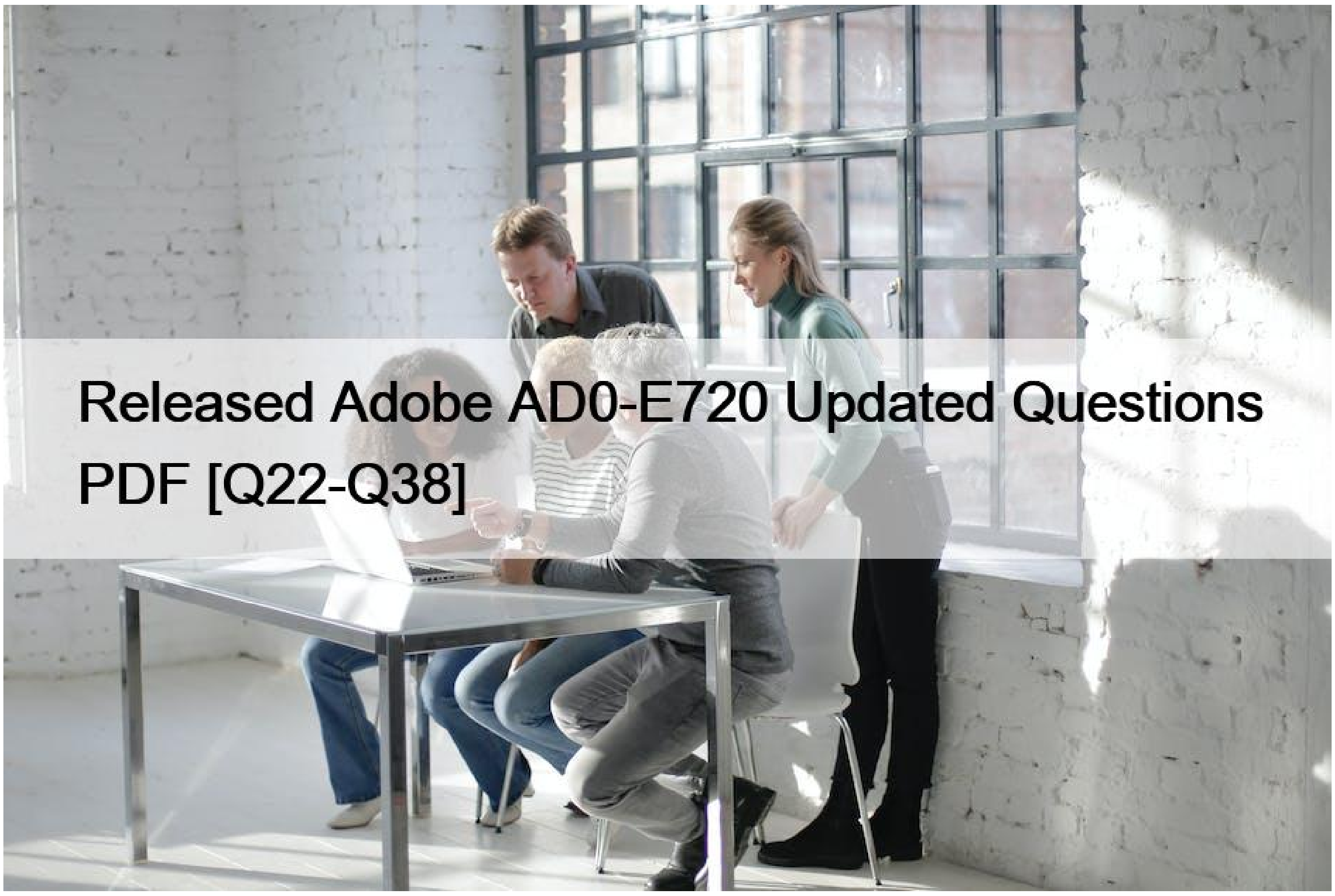
Released Adobe AD0-E720 Updated Questions PDF [Q22-Q38



Released Adobe AD0-E720 Updated Questions PDF
AD0-E720 Dumps and Practice Test (52 Exam Questions)

NO.22 Which two steps are required to delete a manually installed theme? (Choose two.)
* Remove the theme using the theme:uninstall CLI command
* Remove the directory app/design/frontend/<VendorNAME/<ThemeName>
* Disable the theme from the backend admin configuration
* Remove the theme record from the theme database table
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 thatare 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. References: [Delete a theme], [theme:uninstall]

NO.23 Adobe commerce fronted developer needs to speed up the cloud environment deployment process with predefined theme languages for reducing the number of unnecessary themes files.

Which place developer should add language codes?
* SCD_matrix deploy variable in .magento.env.yaml file
* relationships property in .magento.app.yaral file
* scopes section in config.php file
Explanation

The SCD_matrix deploy variable in the .magento.env.yaml file is used to speed up the cloud environment deployment process by specifying the theme languages for each theme. This reduces the number of unnecessary theme files that are generated during the static content deployment. The developer can add the language codes for each theme in the following format:

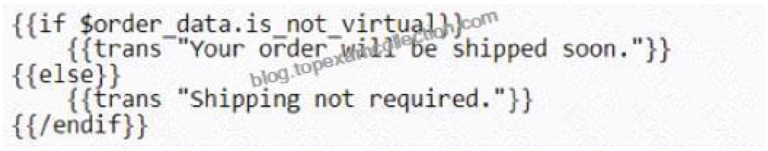
SCD_MATRIX: <theme>: <languages>

For example, to specify English and French languages for the Vendor/Orange theme, the developer can use:

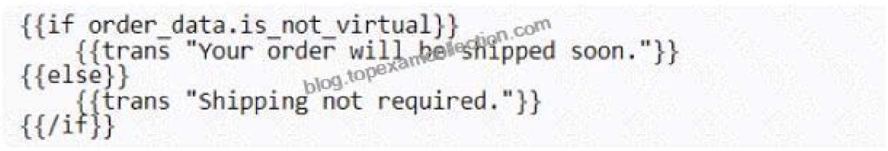
SCD_MATRIX: Vendor/Orange: en_US,fr_FR

The other two options are incorrect because they are not related to the theme languages or the static content deployment. The relationships property in the .magento.app.yaml file is used to define how services are connected to the application. The scopes section in the config.php file is used to specify the configuration scope of each module. References: Adobe Commerce Developer Documentation, Adobe Inc.

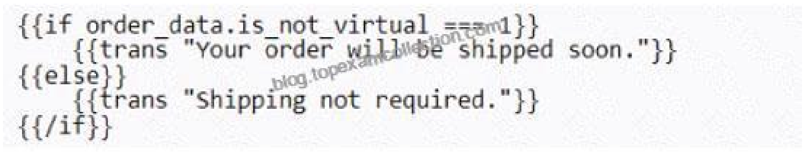
NO.24 An Adobe Commerce developer needs to add a conditional static note depending on whether the order type is virtual or not. Which option would the developer use to add the conditional text in the email template?

* 

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

* 

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

* 

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

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 &#8220;Shipping not required.&#8221; 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.

NO.25 An Adobe Commerce developer is extending a theme from Magentoblank and wants to override parent styles.

Which file does the developer need to change to override the parent theme styles?
* web/css/source/_extends.less
* web/css/source/_extend.less
* web/css/source/_theme. less
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 &#8216;source/_extend.less&#8217;; // 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. References: [Theme inheritance], [Extend parent theme styles]

NO.26 An Adobe Commerce developer wants to override the following Layout XML file in the theme ExampleCorp/orange.

app/design/frontend/ExampleCorp/blank/Vendor_Module/layout/catalog_product_view.xml What path would the developer use inside the layout directory of the theme to override the file?
* /override/ExampleCorp/blank/catalog_product_view.xml
* /override/theme/ExampleCorp/blank/catalog_product_view.xml
* /catalog_product_view.xml
Explanation

To override a layout XML file from a parent theme, the developer just needs to place the modified file in the same path relative to the layout directory of the child theme. In this case, the file would be app/design/frontend/ExampleCorp/orange/Vendor_Module/layout/catalog_product_view.xml. The override directory is not used for overriding layout files, but for overriding templates and web assets. References:

[Layout instructions], [Override templates and layout files]

NO.27 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.)
* {{translations &#8220;Lorem Ipsum is simply dummy text of the printing&#8221;}}
* {{translations &#8220;%items items&#8221; items=&#8221;numltems&#8221;}}
* {{trans &#8220;Lorem Ipsum is simply dummy text of the printing&#8221;}}
* {{trans &#8220;%items items&#8221; items=&#8221;numltems&#8221;}}
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 &#8220;Lorem Ipsum is simply dummy text of the printing&#8221;}}

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 &#8220;%items items&#8221; items=&#8221;numItems&#8221;}}

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.

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

NO.28 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?
* lib/web/css/source/lib/_buttons.less
* lib/web/less/source/lib/_buttons.less
* lib/web/css/source/lib/_button.less
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,

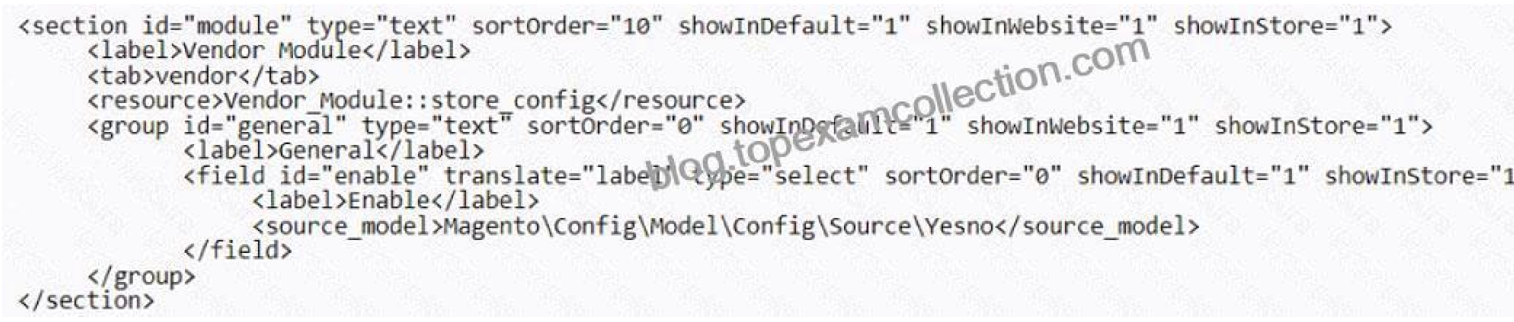
@_button-border, @_button-text-transform, @_button-box-shadow, @_button-hover-color,

@_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. References: [Buttons], [Magento UI library]

NO.29 Which Ul component property is used for cross tracking property changes?
* exports
* listens
* links
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&#8217;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

NO.30 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
                 <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?
* <block name=&#8221;block.name&#8221; template=&#8221;Vendor_Module::template.phtml=&#8221;

ifconfig=&#8221;vendor/general/enable&#8221; />
* <block name=&#8221;block.name&#8221; template=&#8221;Vendor_Module:

:template.phtml&#8221;ifconfig=&#8221;module/general/enable&#8221; />
* <block name=&#8221;block.name&#8221; template=&#8221;Vendor_Module::template.phtml&#8221;

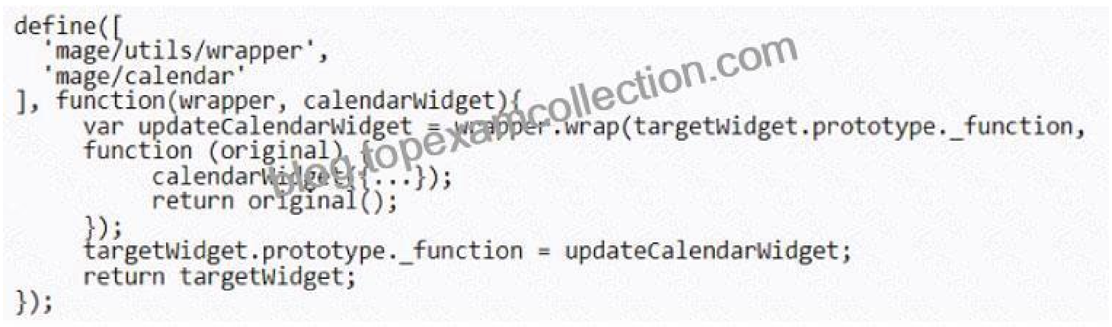
ifconfig=&#8221;general/module/enable&#8221; />
Explanation

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=&#8221;block.name&#8221; template=&#8221;Vendor_Module::template.phtml&#8221; ifconfig=&#8221;vendor/general/enable&#8221; /> 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. References: [Layout instructions], [System configuration]

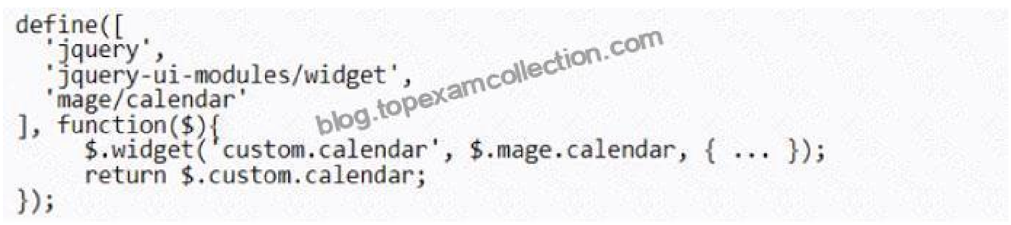
NO.31 An Adobe Commerce developer wants to add a custom widget that extends the default Calendar Widget. What would the contents of this file look like?

*

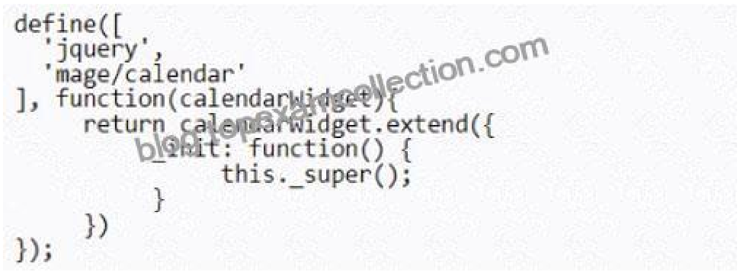


```
define([
   'mage/utils/wrapper',
   'mage/calendar'
], function(wrapper, calendarWidget){
     var updateCalendarWidget = wrapper.wrap(targetWidget.prototype._function,
     function (original) {
          calendarWidget({...});
          return original();
     });
     targetWidget.prototype._function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

*



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

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

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 &#8220;Vendor_Module/js/calendar-widget&#8221; that depends on the &#8220;jquery/ui&#8221; and &#8220;Magento_Ui/js/lib/knockout/bindings/datepicker&#8221; modules.

Returns a function that creates a new widget with the name &#8220;vendor.calendarWidget&#8221; 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.

References: [jQuery widgets], [Calendar Widget]

NO.32 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?
* /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
* Add in the styles to target the elements that require being changed.

/vendor/Theme/web/css/source/_typography.less
* Add in a lib-font-face mixin with the custom font name into the newly created file.
* Add the font-family into the <head></head> of the email within the email template.
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(&#8216;custom-font&#8217;, &#8216;@{baseDir}fonts/custom-font.ttf&#8217;, &#8216;truetype&#8217;); 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: &#8216;@{baseDir}fonts/custom-font&#8217;, @font-weight:

normal, @font-style: normal);

h1 { .lib-font-face(@family-name: @custom-font, @font-path: &#8216;@{baseDir}fonts/custom-font&#8217;,

@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. References: [Custom fonts],

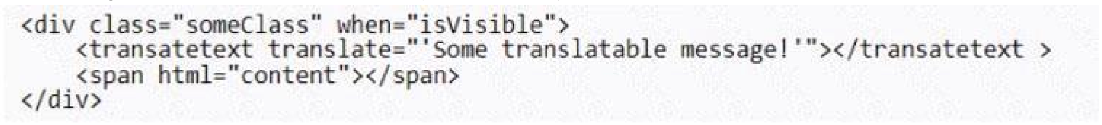
[Email templates overview]

NO.33 Where are the Magento Ul library LESS files located?
* Magento_Ui/web/css/source/
* Magento_Lib/web/css/source
* lib/web/css/source/lib
Explanation

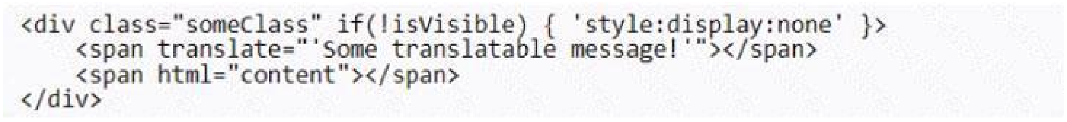
This directory contains various LESS files that define variables, mixins, functions, and styles for common UI elements and components. The Magento_Ui/web/css/source and lib/web/css/source/lib directories are not valid and do not contain the Magento UI library LESS files. References: [Magento UI library],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

NO.34 An Adobe Commerce developer needs to apply a Knockout binding to show content under certain conditions.

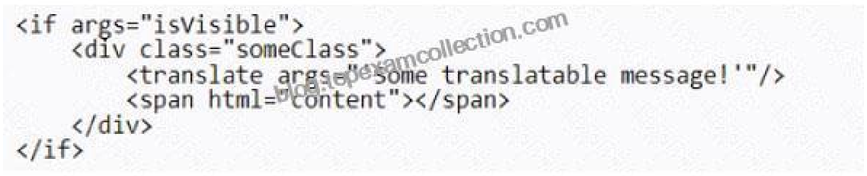
Which two syntaxes would achieve this? (Choose two.)

*


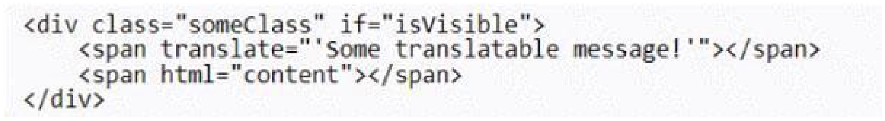
```
<div class="someClass" when="isVisible">
    <transatetext translate="'Some translatable message!'"></transatetext >
    <span html="content"></span>
</div>
```

*


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

* 

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

* 

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

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

NO.35 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?
* Vendor/theme/etc/images.xml
* Vendor/theme/etc/view.xml
* Vendor/theme/etc/theme.xml

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&#8217;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. References:

[view.xml], [theme.xml]

NO.36 The merchant needs to create a new website, and is need modify a template the third party vendor&#8217;s, because the customer is different. The file is found in a module here: app/code/Vendor/Module Keep it simple in your mind!
* Create another layout for the new website and configure new file.phtml.

app/code/Vendor/Module/view/frontend/templates/file.phtml
* Create a new module for extends layout.xml and include new file.phtml.

app/code/Vendor/Module_Two/view/frontend/templates/file.phtml
* Create a new theme, define a new website and customize in app/design.

app/design/frontend/Custom/Theme/Vendor_Module/templates/file.phtml
Explanation

The best way to customize a template file from a third-party module is to create a new theme that inherits from the parent theme and override the template file in the app/design/frontend/Custom/Theme/Vendor_Module/templates directory. This way, the customization is isolated from the original module and can be applied to a specific website or store view. Creating another layout file or a new module would not be as simple or flexible as creating a new theme. References: Frontend development guide, [Create a theme], [Theme inheritance]

NO.37 An Adobe Commerce developer is working on a custom knockout Ul component and they need to add the text Happy Birthday. to be translated inside an .html template.

How would the developer add the text?
* <span data-bind=Mil8n: &#8216;Happy Birthday.'&#8221;></span>
* <span data-bind=&#8221;il8n: Happy Birthday.&#8221;></span>
* <!&#8211; ko il8n = &#8216;Happy Birthday.&#8217; &#8211;><!&#8211; /ko &#8211;>
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=&#8221;i18n: &#8216;Happy Birthday.'&#8221;></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=&#8221;i18n: name + &#8216; Happy Birthday.'&#8221;></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. References: [Knockout bindings], [i18n binding]

NO.38 An Adobe Commerce developer wants to completely overwrite _module. less of Orange_Checkout module, in their theme. Where would the developer place the file?
* Custom/theme/Orange_Checkout/frontend/web/css/_module.less
* Custom/theme/web/css/source/Orange_Checkout/_module.less
* Custom/theme/Orange_Checkout/web/css/source/_module.less
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. References: [Theme structure], [Module naming convention]

AD0-E720 Exam Dumps Pass with Updated 2024 Certified Exam Questions: https://www.topexamcollection.com/AD0-E720-vce-collection.html]