Magento Front End Developer Certification

TEST QUESTIONS

v1.0

2015

Table of Contents

1. Introduction
2. The Magento Fallback System
3. Using Layout XML
4. Create & Customize Template Files
5. Block-Template Design System
6. Theming Related Directory Structure
7. Customize and Create Javascript
8. Use CSS Effectively
9. Customize Specific Magento Pages
10. Correctly Use Admin Config Scopes
11. Internationalization
12. Answers

Magento Front End Developer Certification - Test Questions

The following is a list of Magento Front End Developer Certification (M70-301) test questions that you can use to practice for the real exam. They have been separated into each section of the official Study Guide for ease of use.

This is not a pass-guide or a list of questions that will be asked in the exam. This is simply a list of questions to get you thinking about the areas you need to cover within your revision.

Introduction

The Magento Front End Developer Certification exam (M70-301) consists of 81 questions. 16 of them are unmarked - therefore only 65 are marked and count towards your percentage. You will not know which of the questions are unmarked (though it is our belief that these may be the freely available test questions).

You will have 120 minutes to complete the exam.

As of January 2015 the exam focusses on CE 1.9 - therefore it is safe to say that they try to keep the questions up to date with the newest releases.

Exam overview:

1. Use the Magento Design Fallback System inline link (7%)
2. Use Layout XML to Customize a Theme (19%)
3. Create and Customize Template Files (16%)
4. Effectively use the Magento Block-Template Design System (11%)
5. Identify Where to locate Files and Create New Files in the Theming-related Directory Structure (7%)
6. Customize and Create JavaScript Within the Magento Framework (8%)
7. Use CSS Effectively to Customize Magento Look and Feel (9%)
8. Customize the Look and Feel of Specific Magento Pages (12%)
9. Correctly Use the Admin Configuration Scopes (Default / Website / Store View Fallback) (6%)
10. Implement Internationalization of Frontend Pages (4%)
11. Answers (4%)

The Magento Fallback System

1. Use the Magento Design Fallback System (7%)

Key Areas of Study

- Understand the priority order of fallbacks (all the way from custom themes to the default) as well as what the fallback means for files and folders of the same name in each fallback.
- Understand how the admin configuration can be used to change the fallback process and the difference between `System > Design` and `System > Config > Design`.

Questions

1.1 In a Magento instance configured to use the package name "creare" and the theme name "blue", which file would override the default catalog category view template?

```
A. app/frontend/base/creare/blue/template/catalog/category/view.phtml
B. app/design/frontend/creare/blue/template/catalog/category/view.phtml
C. app/design/frontend/creare/default/template/catalog/view.phtml
D. app/design/frontend/blue/creare/template/catalog/category.phtml
E. app/frontend/default/blue/customcategory/view.phtml
```

1.2 You have a “custom/default” design package. You need to edit this file: `/app/design/frontend/base/default/template/page/html/header.phtml`. Which method is bad practice?

```
A. Create a new module in “base/default” called “custom” and copy the file there for editing.
B. Create a new package called “custom/custom” and copy the file there for editing.
C. Copy the file to “custom/default” for editing.
D. Edit the “base/default” file directly.
```

1.3 Given a configured package “mypackage” and a configured theme “mytheme”, which THREE of the following templates would be part of the design fallback? (Choose THREE.)

```
A. mypackage/default/template/my/template.phtml
B. base/mytheme/template/my/template.phtml
C. default/default/template/my/template.phtml
D. base/default/template/my/template.phtml
E. default/mytheme/template/my/template.phtml
F. mypackage/mytheme/template/my/template.phtml
```

1.4 What is System > Design used for?

```
A. Configuring the fallback system of a store view
B. Overriding the fallback system of a store view for a limited period of time
C. Activating the fallback system of a store view
D. Overriding the fallback system of a subpart (Layout, Templates, ...) of a store view
E. Configuring basic Package and Theme settings of your Magento instance
```

Using Layout XML

2. Use Layout XML to Customize a Theme (19%)

Key Areas of Study

- Ensure you are 100% confident on all the possible ways to add Scripts / CSS through layout XML for different scenarios.
- Understand `<action>` tags and how they affect blocks.

Questions

2.1 You want to remove the left structural block from a page. Which code should you use?

```
A. <remove_block name="left"/>
B. <remove type="left">
C. <remove name="left"/>
D. <remove_structure name="left"/>
```

2.2 Which three of the following action method(s) are valid ways to add custom JavaScript files?

```
A. <action method="addItem" ><type>js</type><name>myscript.js</name></action>
B. <action method="addJs"><script>myscript.js</script></action>
C. <action method="addJavascript"><type>js</type><src>myscript.js</src></action>
D. <action method="addJs"><type>skin_js</type><name>myscript.js</name></action>
E. <action method="addItem"><type>skin_js</type> <name>myscript.js</name></action>
F. <action method="appendScript"><type>js</type> <script>myscript.js</script></action>
G. <action method="addScript"><script>myscript.js</script></action>
```

2.3 You need to add a custom structural block template to a block that will render all children automatically. Which three attributes are NOT required in the block’s XML definition? (Choose THREE.)

```
A. after
B. name
C. template
D. before
E. ifModule
F. type
```

2.4 What is the proper layout update handle for adding a CSS file to the product page?

```
A. <product>
B. <catalog_product>
C. <catalog_product_view>
D. <default>
```

Create & Customize Template Files

3. Create and Customize Template Files (16%)

Key Areas of Study

- Understand how to set your own custom template using Layout XML in any given scenario
- Understand how Magento uses PHP within `template.phtml` files

Questions

3.1 The correct way to call a child template from within a template file would be to use:

```
A. <?php echo $this->getChildHtml('name_of_block') ?>
B. <?php echo $this->getTemplate('/my/template.phtml') ?>
C. <?php echo $this->getChildTemplate('name_of_block') ?>
D. <?php echo $template->getChild('name_of_block') ?>
```

3.2 How would you replace the root template of a product page using layout.xml?

```
A.
<default>
    <reference name="product_view">
        <action method="setTemplate">
            <template>page/4-columns.phtml</template>
        </action>
    </reference>
</default>

B.
<catalog_product_view>
    <reference name="root">
        <action method="setTemplate">
            <template>page/4-columns.phtml</template>
        </action>
    </reference>
</catalog_product_view>

C.
<catalog_product_view>
    <reference name="root">
        <template>page/4-columns.phtml</template>
    </reference>
</catalog_product_view>

D.
<catalog_product_view>
    <reference name="root">
        <action method="setRootTemplate">
            <template>page/4-columns.phtml</template>
        </action>
    </reference>
</catalog_product_view>
```

3.3 Why does Magento use different constructs in its `.phtml` files e.g. `if (...) : endif;` ?

```
A. if(...){ ... } doesn't work in .phtml file types
B. To make it easier to distinguish starting and ending tags within HTML/PHP mixed files
C. if(...) : endif; is Magento specific
D. There is no reason
```

3.4 What would the following code render?

```
<?php echo $this->getChildHtml(); ?>

A. It wouldn't render anything
B. It would render any child block without a "name" attribute associated
C. It would render all child blocks that have not yet been rendered
D. You will receive a PHP error
```

Block-Template Design System

4. Effectively use the Magento Block-Template Design System (11%)

Key Areas of Study

- Understand how Layout XML, `.phtml` template files and `.php` Block files relate to each other
- Know your block types (structural, content - and specific types)

Questions

4.1 Which is the correct type attribute for a structural block?

```
A. page/html
B. core/html_wrapper
C. core/template
D. core/text_list
```

4.2 What does $this refer to in template files?

```
A. $this refers to the template and variables set within the template file
B. $this refers to the controller file and allows access to controller methods
C. $this refers to the block instance of the template we are currently using
D. $this refers to the Magento database
```

Theming Related Directory Structure

5. Identify Where to Locate Files and Create New Files in the Theming-related Directory Structure (7%)

Key Areas of Study

- Understand the 'names' of the specific folders in the fallback path (e.g. area/package/theme) so if referenced in a question you know where they should sit in relation to each other.
- Understand the use of Skin and all the subfolders within

Questions

5.1 Which three of the following folders should NOT be publicly accessible? (Choose THREE.)

```
A. /app/design/frontend/package/theme/layout
B. /app/locale
C. /var/cache
D. /js/varien
E. /skin/frontend/package/theme/js
F. /skin/frontend/package/theme/css
```

5.2 Which files are typically available within the /skin/ directory? (Choose THREE.)

```
A. Layout
B. Javascript
C. Stylesheet
D. Images
E. Translation
F. Block
```

Customize and Create Javascript

6. Customize and Create JavaScript Within the Magento Framework (8%)

Key Areas of Study

- Know how to perform standard Javascript/Prototype functions - not just those uses within the frontend of Magento
- Understand how to use prototype to extend core JS classes within Magento (in an upgrade safe manner)

Questions

6.1 Using prototype.js, which selector would be used to target a group of paragraphs with the class of green?

```
A. $$('p.green')
B. $('div').children('p.green')
C. $('.green')
D. $$('p > .green')
```

6.2 Given that the prototype library is included - what would be the result of the following code?

```
<script type="text/javascript">
    Person = Class.create();

    Person.prototype = {
         initialize: function(name)
         {
              this.name = name;
              this.sayHello('Hello');
         },

         sayHello: function (message)
         {
              return(this.name+' says '+message);
         }
    };

    Person.prototype.sayHello = Person.prototype.sayHello.wrap( function (parentMethod, message, flag) {
         if (flag){
            return(message+' Shouted '+this.name);
        } else {
            return parentMethod(message);
        }
     });

    var Jimmy = new Person('Jimmy');

    alert(Jimmy.sayHello('Hi!', true));
    alert(Jimmy.sayHello('Hi!', false));
</script>

A. "Undefined" followed by "[Object]"
B. "Jimmy says Hi!" followed by "Hi! Shouted Jimmy"
C. "Hi! Shouted Jimmy" followed by "Jimmy says Hi!"
D. "Undefined" followed by "Jimmy says Hi!"
```

Use CSS Effectively

7. Use CSS Effectively to Customize Magento Look and Feel (9%)

Key Areas of Study

- Understand as much CSS2 as you can.
- 1.9 comes with a responsive theme - know your media queries.

Questions

7.1 Given the folling HTML code - what colour would the `<span>` text be within the paragraph? The browser width has been set to 600px.

```
<html>
    <head>
        <style>
            .urgent {
                color: yellow;
            }

            @media (min-width: 600px) {
                .urgent {
                    color: blue;
                }
            }

            @media (max-width: 599px) {
                .urgent {
                    color: red;
                }
            }
        </style>
    </head>
    <body>
        <p>This is just a <span class="urgent">paragraph</span></p>
    </body>
</html>

A. Yellow
B. Blue
C. Red
D. Black
```

7.2 How many `<li>` elements would show in the first row of this grid?

```
<html>
    <head>
        <style>
            ul {
                width: 100%;
            }
            ul li {
                display: block;
                float: left;
                width: 20%;
                padding: 0 2.5%;
                border: 1px solid grey;
            }
        </style>
    </head>
    <body>
        <div>
            <ul>
                <li>Block</li>
                <li>Block</li>
                <li>Block</li>
                <li>Block</li>
            </ul>
        </div>
    </body>
</html>

A. One
B. Two
C. Three
D. Four
```

7.3 Based on the code below, what color will the background colour be of the div tag?

```
<style>
    div { background: #000; }
    #darker { background: #333; }
    .lighter { background: #FFF; }
</style>

<div class="lighter" id="darker" style="background: #F00"></div>

A. black
B. grey
C. white
D. red
```

7.4 What would be the expected outcome of the following code?

```
<html>
    <head>
        <style>
            .container {
                width: 100%;
            }
            .left {
                float: left;
                clear: left;
            }
        </style>
    </head>
    <body>
        <div class="container">
            <div class="left">BOX1</div>
            <div class="left" style="clear:right">BOX2</div>
            <div class="left" style="clear:none">BOX3</div>
        </div>
    </body>
</html>

A.
BOX1
BOX2BOX3

B.
BOX1BOX2BOX3

C.
BOX1
BOX2
BOX3

D.
BOX1BOX2
BOX3
```

Customize Specific Magento Pages

8. Customize the Look and Feel of Specific Magento Pages (12%)

Key Areas of Study

- Understand the main Magento template files and what PHP calls and blocks are available.

Questions

8.1 How would you retrieve the customers' name in a transactional email template?

```
A. {{var customer.name}}
B. {{var customer_name}}
C. {{customer name}}
D. {{var getCustomerName()}}
```

8.2 How can a template be included in a CMS block?

```
A. {{block template="catalog/product/view.phtml"}}
B. {{block id="catalog/product/view.phtml"}}
C. {{block type="core/template" template="catalog/product/view.phtml"}}
D. {{block type="core/template" name="product.info" } }
```

Correctly Use Admin Config Scopes

9. Correctly Use the Admin Configuration Scopes (Default / Website / Store View Fallback) (6%)

Key Areas of Study

- Understand the admin configuration area - what's available in each area and what it does.
- Understand the hierarchy of Magento Global / Website / Store and Store View settings.

Questions

9.1 Where would you change the timezone settings in Magento?

```
A. Index.php
B. System > Configuration > Admin > Timezone
C. System > Configuration > General
D. System > General
```

9.2 You have two website views, ChairsRUs and TablesRUs. You want to disable the flat rate shipping method for the English Store View within ChairsRUs. Which of the following would be the most appropriate way to do this?

```
A. Disable within Default scope
B. Disable within the ChairsRUs website scope
C. Disable within the English Store View within ChairsRUs as it's more specific
D. Disable within Default scope and enable for the TablesRUs website view
```

9.3 Which four of the following page elements can be updated from the Magento Admin in `System->Configuration->Catalog->Catalog`? (Choose FOUR.)

```
A. Sitemap Generation Frequency
B. Backorders
C. Price Scope
D. Use Javascript Calendar
E. Search Type
F. Enable Terms & Conditions
G. Image Placeholders
```

Internationalization

10. Implement Internationalization of Frontend Pages (4%)

Key Areas of Study

- Understand the priority order of the different methods of translation

Questions

10.1 Which is the correct order of the three types of translations by priority (1 = highest, 3 = lowest)?

```
A.
1. Database (Inline) translation
2. Module translation
3. Theme translation
B.
1. Database (Inline) translation
2. Theme translation
3. Module translation
C.
1. Module translation
2. Database (In line) translation
3. Theme translation
D.
1. Theme translation
2. Database (In line) translation
3. Module translation
```

10.2 Where would you find the order_new.html transaction template within the Magento file structure?

```
A. Inside your themes 'locale' folder
B. Directly inside app/core/locale/en_US/
C. Directly inside app/locale/template/email/sales
D. Directly inside app/locale/template/email
```

Answers

Remember it's not about knowing the answers to the above questions - it's about knowing the 'why' in the answer to the given scenario.

- Use the Magento Design Fallback System
  - 1.1 B
  - 1.2 D
  - 1.3 ADF
  - 1.4 B
- Use Layout XML to Customize a Theme
  - 2.1 C
  - 2.2 ABE
  - 2.3 ADE
  - 2.4 C
- Create and Customize Template Files
  - 3.1 A
  - 3.2 B
  - 3.3 B
  - 3.4 C
- Effectively use the Magento Block-Template Design System
  - 4.1 D
  - 4.2 C
- Identify Where to Locate Files and Create New Files in the Theming-related Directory Structure
  - 5.1 ABC
  - 5.2 BCD
- Customize and Create JavaScript Within the Magento Framework
  - 6.1 A
  - 6.2 C
- Use CSS Effectively to Customize Magento Look and Feel
  - 7.1 B
  - 7.2 C
  - 7.3 D
  - 7.4 B
- Customize the Look and Feel of Specific Magento Pages
  - 8.1 B
  - 8.2 C
- Correctly Use the Admin Configuration Scopes (Default / Website / Store View Fallback)
  - 9.1 C
  - 9.2 B
  - 9.3 CDEG
- Implement Internationalization of Frontend Pages
  - 10.1 B
  - 10.2 C