Shwetank Education

Magento interview questions and answers for freshers and experienced

Content Ref :pcds.co.in | only for use Education and Job purpose, not for official purpose.

Questions : 1 **What Is Magento ?**

Answers : 1 Magento is a open source eCommerce platform for a complete eCommerce web site solution. Magento's open source architecture enables the user to maintain complete control the structure and functionality of a web site. One of the most impressive aspects of Magento is the use of the Zend Framework. This allows for separation of the Model-View-Controller (MVC) which separates its core operations from functionality and appearance. Magento offers a stable eCommerce environment and gives us the flexibility to completely customize the web site to suit our business needs.

Questions : 2 **Explain Magento Versions and License**

Answers : 2 There are currently two versions of Magento: the Magento Community Edition and the Magento Enterprise Edition. The Magento Community Edition is licensed under the Open Software License (OSL) v3.0. The Magento Enterprise Edition is available under a commercial license and requires an annual fee. Both editions share the same underlying framework. The Enterprise Edition includes several additional features that are not included with the Community Edition, including gift certificates, customer store credits, and strong data encryption

Questions : 3 What The History Behind Magento ?

Answers : 3 advertisements

Varien began developing Magento in January of 2007. It was their vision to establish an eCommerce platform unlike anything else on the market. Varien originally planned on building aneCommerce platform based on a fork of osCommerce, but decided to write their own system using the Zend Framework instead. During Varien's six month beta period, they had over 225,000 downloads of Magento. The first public release of Magento, version 1.0, was released on March 31, 2008. Shortly after, Magento 1.1 was released. Magento 1.1 fixed many bugs found in version 1.0, but also included a faster and more informative administrative user interface. Magento 1.2 was released on December 29, 2008. This version included several upgraded Zend Framework libraries and a new feature: downloadable products. Magento 1.3 was released on March 30, 2009. This version increased Magento’s speed in both public and administrative views. The Magento Enterprise Edition version 1.3 Enterprise was released on April 15, 2009. This version requires a service agreement with Varien and costs $8,900 USD per year and it includes technical support which is not available with the Community Edition.

In February 2011, eBay announced it had made an investment in Magento in 2010, worth a 49% ownership share of the company.

As of June 6, 2011 eBay owns 100% of Magento. eBay announced that Magento would join eBay's new X.Commerce initiative

Magento's CEO and co-founder Roy Rubin wrote on the Magento blog stating that "Magento will continue to operate out of Los Angeles, with Yoav Kutner and me as its leaders.".

As of February 2, 2012 Magento passed 4 million downloads of its software platform.

Questions : 4	Explain features use of Magento
Answers : 4	In addition to a solid architecture and framework, there are several unique reasons why Magento makes a great choice for an eCommerce solution. • One of the most amazing features about Magento is that you can design and develop multiple web sites and they store and share one administrative interface. This extremely flexible feature allows you to modify and control multiple web sites. All of your products inventory and pricing can be controlled from one central location. There is no longer a need to login to multiple locations to handle multiple web sites. Magento has the ability to control them all. • Magento supports over sixty languages, multiple currencies, and tax rates. This gives you the ability to easily expand in the global market. • Layered navigation gives users customized browsing options when viewing products by categories. You can now sort products by price, size, color, and other customizable attributes. • Magento also has built-in web services. This flexibility allows external applications to access magento’s data without changing the underlying core code. Currently, SOAP and XML-RPC protocols are available out of the box.

- Magento has Search Engine Optimization (SEO) built in from the start. It has the ability to handle friendly URL rewrites which make it easy for search engines to index your store and products.
- Not only does Magento offer real-time carrier rates and quotes, users can ship products from one order to multiple shipping address. This makes gift shopping especially easy.
- Magento also has several reporting features built in. These allows for easy view of sales reports, best-selling products, and customer reporting. They can even be exported in a CSV format to integrate with excel and other database programs.
- Magento has designed its file structure to three major sections: core, functionality, and design. This allows for easy updating of images and CSS styling without affecting the functionality of the site. Store functionality can also be easily customized without affecting the Magento's core. As a result, you can modify Magento without having to worry about upgrading to newer versions in the future.

- Magento has a huge community backing. In addition to a public forum and bug tracking, Magento also has its own public repository of extensions called Magento Connect. Magento Connect features both free and commercial extensions to enhance the functionality of your web site.
- Since Magento is released under the Open Software License (OSL), the Magento Community Edition is available at no cost. In turn, this allows web site developers and eCommerce web site owners to cut down on software costs

Questions : 5 what is EAV in Magento?

Answers : 5 EAV means Entity Attribute Value this allows us to add unlimited columns or attributes to our table virtually. Means, the fields which is represented in "column" way in a regular table, is represented in a "row" (records) way in EAV. In EAV, we have one table which holds all the "attribute" (table field names) data, and other tables which hold the "entity" (id or primary id) and value (value for that id) against each attribute. In Magento, there is one table to hold attribute values called *eav_attribute* and 5-6 tables which holds entity and data in fully normalized form, eav_entity, eav_entity_int (for holding Integer values),
- eav_entity_varchar (for holding Varchar values),
- eav_entity_datetime (for holding Datetime values),
- eav_entity_decimal (for holding Decimal/float values),
- eav_entity_text (for holding text (mysql Text type) values).

EAV is expensive and should only be used when you are not sure about number of fields in a table which can vary in future. To just get one single record, Magento joins 4-5 tables to get data in EAV. But this doesn't mean that EAV only has drawbacks. The main advantage of EAV is when you may want to add table field in future, when there are thousands or millions of records already present in your table. In regular table, if you add table field with these amount of data, it will screw up your table, as for each empty row also some bytes will be allocated as per data type you select. While in EAV, adding the table column will not affect the previously saved records (also the extra space will not get allocated!) and all the new records will seamlessly have data in these columns without any problem

Questions : 6 How Magento's MVC architecture works:

Answers : 6 MVC means Model, View and Controller. so any architecture use to separates their business logic and user interface known as MVC it two type convention-based in this just need to follow the instructions of the core system to get our module ready in just few lines. Magento is 2nd type **configuration-based,** i.e. we need to specify each and every thing to our module's config file in order to get it work. Magento has Controller (for Routing), Block (for Business Logic), Model (for DB access, sql) and Template file (for Presentation i.e. View).

It work like below

1. When you enter the URL like http://www.pcds.co.in/frontname/controller/method/param1/value1/param2/value2), this URL is intercepted by one PHP file called index.php which instantiates Magento application
2. Magento application instantiates Front Controller object
3. Further, front controller instantiates Router objects (specified in module's config.xml, global tag)
4. Now, Router is responsible to "match" the frontname which is in our URL
5. If "match" is found, it sees controller name and method name in the URL, which is finally called.
6. Now depending on what is written in action name (method name), it is executed. If any models are called in it, the controller method will instantiate that model and call the method in it which is requested.
7. Then the controller action (method) instantiate the Layout object, which calls Block specified for this action (method) name (Each controller action name have block and template file associated with it, which can be found at app/design/frontend or adminhtml/namespace/module/layout/module.xml file, name of layout file (module.xml) can be found in config.xml of that module, in layout updates tag).
8. Template file (.phtml) now calls the corresponding block for any method request. So, if you write $this->methodName in .phtml file, it will check "methodName" in the block file which is associated in module.xml file.
9. Block contains PHP logic. It references Models for any data from DB.
10. If either Block, Template file or Controller need to get/set some data from/to database, they can call Model directly like Mage::getModel('modulename/modelname').

Questions : 7 How Magento ORM works?

Answers : 7 ORM stands for Object Relational Mapping. It's a programming technique used to convert different types of data to Objects and vice versa.

In Magento, ORM is shown as Model (based on Zend Framework’s Zend_Db_Adapter), which further breaks down to two types of Models.

- First is the “simple” i.e. **Regular** Models which is nothing but flat table or our regular table structure.
- Second Model is **EAV** (Entity Attribute Value), which is quite complicated and expensive to query.

All Magento Models interacting with database are inherited from Mage_Core_Model_Abstract class, which is further inherited from Varien_Object.

Difference between two Models is, Simple Model is inherited from *Mage_Core_Model_Resource_Db_Abstract* class,
while EAV is inherited from *Mage_Eav_Model_Entity_Abstract*.

For those who don’t know what EAV is, please read my 3rd answer below.

So, to end up this question,
when you want to get some data in Magento, you call it like this:

```
Mage::getModel('module/model')->load(1);
```

where 1 is the primary key id for some Regular/Simple table, while in EAV so many tables are joined to fetch just single row of data.

Questions : 8 what is Difference between Magento function getSingleton() and getModel()

Answers : 8 The difference between Mage:getSingleton() and Mage::getModel() is that the former one does not create an object if the object for same class is already created, while the later creates new objects every time for the class when it’s called.

Mage::getSingleton() uses the “**singleton design pattern**” of PHP. If the object is not created, it will create it.

Mage::getSingleton() is mostly used when you want to create an object once, modify it and later fetch from it. Popular example is session, you first create a session object, and then add/remove values from session across different pages, so that it retains your values (e.g. cart values, logged in customer details, etc.) and doesn’t create new session object losing your last changes.

Mage::getModel() is used when you want to have the fresh data from the database. Example is when you want to show records from database.

Questions : 9 How to improve magento performance?

Answers : 9 Enabled magento caching

MySQL Query caching

Enable Gzip Compression

Disable any unused modules

Disable the Magento log

Optimise your images
Combine external CSS/JS into one file
Enable Apache KeepAlives:
Make sure your Apache configuratio
has KeepAlives enabled.

Questions : 10 What is benefit of namespace (package) in magento?

Answers : 10 you can have more than one module with same name but they should be placed in different namespaces. All magento core modules are contained in mage namespace. And Magento search module in order by local then community then base module

Questions : 11 How to include CMS block in template file(.phtml)?

Answers : 11 Access blocks content from .phtml template file by :

```
echo $this->getLayout()->createBlock("cms/block")
->setBlockId("static_block_id")->toHTML();
```

Questions : 12 How to add an external javascript or css file to Magento?

Answers : 12 To add an external javascript or css file to Magento is too much easy as we can add this in any of below tow ways in xml layout file in given like in <catelog_product_view> part we an add in

<referense name="head">

```
<action method="addJs"><script>js/yourfile.js</script></action>
 <action
method="addCss"><stylesheet>css/yourstyle.css</stylesheet></action>
 or
 <action method="addItem"><type>skin_js</type><name>js/
yourfile.js</name></action>
 <action method="addItem"><type>skin_css</type><name>css/yourstyle.
css</name></action>
```

Questions : 13 What are handles in magento (layout)?

Answers : 13 Handles are basically used for controlling the structure of the page like which block will be displayed and where. First level child elements of the node are called layout handles. Every page request can have several unique Handles. The handle is called for every page. handle for products belongs to virtual product type, PRODUCT_TYPE_simple is called for product details page of simple product type and PRODUCT_TYPE_virtual is called for the virtual product detail page and customer_logged_in handle is called only if customer is logged in. The muster_index_index handle is created by combining the frontName (muster), Action Controller (index), and Action Controller Action Method (index) into a single string and this handle will be called only when /muster/index/index url is accessed.

```
 <layout version="0.1.0">
 <muster_index_index>
 <reference name="root">
 <block type="page/html" name="root" output="toHtml"
template="../somepage.phtml" />
 </reference>
 </muster_index_index>
 </layout>
```

Questions : 14	How to upgrade to the latest version using Magento Connect?
Answers : 14	Upgrading Magento to the latest version is a fairly simple task. Copy and Paste this key magento-core/Mage_All_Latest VIA Magento Connect where it states Paste extension key to install:. This will upgrade Magento to the newest version.
Questions : 15	Why magento is slow ?
Answers : 15	Magento is slow due to below thing 1. EAV structure of magento database, even for retrieving single entity the query becomes very complex . 2. Magento's *template system involves a lot of recursive rendering* *3. Huge XML trees built up for layout configuration, application configuration settings*

Questions : 16	**What are the addAttributeToFilter Conditionals in Magento?**
Answers : 16	In Magento we can use addAttributeToFilter Conditions same as WHERE in SQL Below are the all condtions

Equals: eq

```
$_products-
>addAttributeToFilter('status'
, array('eq' => 1));
```

Not Equals - neq

-

```
$_products->addAttributeToFilter('sku', array('neq' => 'test-
product'));
```

Like - like

-

```
$_products->addAttributeToFilter('sku', array('like' => 'UX%'));
```

One thing to note about like is that you can include SQL wildcard characters such as the percent sign.

Not Like - nlike

-

```
$_products->addAttributeToFilter('sku', array('nlike' => 'err-
prod%'));
```

In - in

-

```
$_products->addAttributeToFilter('id', array('in' =>
array(1,4,98)));
```

When using in, the value parameter accepts an array of values.

Not In - nin

```
$_products->addAttributeToFilter('id', array('nin' =>
array(1,4,98)));
```

NULL - null

-

```
$_products->addAttributeToFilter('description', 'null');
```

Not NULL - notnull

-

```
$_products->addAttributeToFilter('description', 'notnull');
```

Greater Than - gt

-

```
$_products->addAttributeToFilter('id', array('gt' => 5));
```

Less Than - lt

-

```
$_products->addAttributeToFilter('id', array('lt' => 5));
```

Greater Than or Equals To- gteq

-

```
$_products->addAttributeToFilter('id', array('gteq' => 5));
```

Less Than or Equals To - lteq

-

```
$_products-
>addAttributeToFilter('id',
array('lteq' => 5));
```

Questions : 17	**What permissions required for file and folder In Magento ?**
Answers : 17	A security Issue happen if Setting all the files & folders to 777 will make them writable by everyone.!

Normal Operation

For the normal operation or installation of a Magento store, only 2 folders need to be writable:

/media - for web accessible files, such as product images
/var - for temporary (cache, session) and import/export files

Installation

During installation Magento Install Wizard will create

app/etc/local.xml file which contains database configuration and global encryption key for your Magento copy.

Meaning **app/etc** folder has to be writable to web service.

Web based MagentoConnect manager

If you wish to use web based downloader for installation, upgrades or MagentoConnect extensions installation, you will need to have write permissions on ALL magento files including the root Magento folder.

Files and folders will need to be returned to 655 non-writable permissions after installation.

Questions : 18	**How to change Currency in INR in magento ?**
Answers : 18	In Magento, Currency change is very easy just we need to follow below step

1. From the Admin panel, select **System** > **Configuration**.
2. From the Configuration panel on the left, under General, select the Currency Setup tab.
3. Click to expand the **Currency Options** section. Then, do the following:
    1. In the **Base Currency** list, select the primary currency that is used for store transactions.
    2. In the **Default Display Currency** list, select the primary currency that is used to display pricing in your store.
    3. In the **Allowed Currencies** list, select all currencies that are accepted as payment by your store.
4. When finished, click the **Save Config** button.

To customize currency symbols:

Currency symbols appear in the Admin panel of your store, and are also used throughout your storefront in orders, customer checkout, and so on.

1. From the Admin panel, select **System** > **Manage Currency** > **Symbols**.

   Each enabled currency for your store appears in the **Currency** list.

2. Enter a custom symbol to use for each currency, or select the standard symbol selecting the **Use Standard** checkbox to the right of each currency.
3. When finished, click the **Save Currency Symbols** button.

Questions : 19 How To Add Blog In Magento?

Answers : 19 To add Blog in Magento we just need to get magento connect extension key and follow below step

In admin panel

system>magento connect>Magento connect >Magento connect manager

and then below input type

Paste extension key to install:

need to add this key

http://connect20.magentocommerce.com/community/AW_Blog

there then click on Install button after that need to proceed

Just it you will get a link in admin panel name **blog**

Questions : 20 How to add review list on product page of Magento?

Answers : 20 To Add Review List on mageto product page we have to do below few step

First of all we need to open catalog.xml in Layout folder of used design theme and then we have to add

below code in **catalog_product_view**. section

```
<block type="review/product_view_list"
name="product.info.product_additional_data"
 as="reviews"
template="review/product/view/list.phtml"/>
```

and then next step is to just put a single line of code in file **catalog/product/view.phtml** in template

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

after that must refresh cache management through admin section

That all the story of adding review list on magento product page

Questions : 21	How to run custom MySql query in Magento ?
Answers : 21	To write custom SQl query in magento is very easy we have to just need to all

```
$db = Mage::getSingleton('core/resource')-
>getConnection('core_write');
```

```
$result=$db->query("SELECT * FROM PCDSTable");
```

Questions : 22	How to add Custom javascript and css file in magento
Answers : 22	It's much important for Magento theme developer to add their own external js and css file so that they can use custom JavaScript and css and it much easier in Magento to add this. as just you need to add local.xml in your own theme and add below structure code in that file in local.xml under layout folder of your own theme /or default theme that currently in use

```
<?xml version="1.0" encoding="UTF-8"?>
<layout>
<default>
<reference name="head">
<action method="addCss">
<stylesheet>css/pcds_style.css</stylesheet>
</action>
<action method="addJs">
<script>pcds/jquery.js</script>
</action>
</reference>
</default>
</layout>
```

here you can replace name of js and css file accordingly yours and folder name too just keep in mind

your css file would be add in below path
/skin/frontend/default/pcds/css/

and js file directly
js/pcds/

folder here PCDS is our own theme name or folder name of default or current theme

Questions : 23	**How To Create a Featured Product?**

We are gooing to add Fueture product on our magento store as The Featured Product is a product with an attribute added from the administrative UI. When the administrator selects Yes in the Featured attribute, that product will be displayed in a content block on the category page.

Step 1) Create new "Featured" attribute

Create a new attribute by going to Catalog > Attributes > Manage Attributes > Add New Attribute.

Attribute Properties

- Attribute Identifier: **featured**
- Scope: **Store View**
- Catalog Input Type for Store Owner: **Yes/No**
- Unique Value (not shared with other products): **No**
- Values Required: **No**
- Input Validation for Store Owner: **None**
- Apply To: **All Product Types**

Front End Properties

- Use in quick search: **No**
- Use in advanced search: **Yes**
- Comparable on Front-end: **No**
- Use In Layered Navigation (Can be used only with catalog input type ‘Dropdown’): **No**
- Visible on Catalog Pages on Front-end: **Yes**

Manage Label/Options

- Default: **Featured Product**

- English: **Featured Product**

Save the new attribute and go to Catalog → Attributes → Manage Attributes Sets to add the attribute to the default feature set.

Step 3) Create new block class that will instantiate the featured product

Create a new file, and directories: app/code/local/MyCompany/Catalog/Block/Product/Featured.php

```
<?php
class MyCompany_Catalog_Block_Product_Featured extends Mage_Catalog_Block_Product_Abstract
    {
        public function getFeaturedProduct()
        {

            // instantiate database connection object
            $storeId = Mage::app()->getStore()->getId();
            $categoryId = $this->getRequest()->getParam('id', false);
            $resource = Mage::getSingleton('core/resource');
            $read = $resource->getConnection('catalog_read');
            $categoryProductTable = $resource->getTableName('catalog/category_product');
            //$productEntityIntTable = $resource->getTableName('catalog/product_entity_int'); // doesn't work :(
            $productEntityIntTable = (string)Mage::getConfig()->getTablePrefix() . 'catalog_product_entity_int';
            $eavAttributeTable = $resource->getTableName('eav/attribute');
            // Query database for featured product
            if ($categoryId){
            $select = $read->select()
                        ->from(array('cp'=>$categoryProductTable))
                        ->join(array('pei'=>$productEntityIntTable), 'pei.entity_id=cp.product_id', array())
                        ->joinNatural(array('ea'=>$eavAttributeTable))
                        ->where('cp.category_id=?', $categoryId)
                        ->where('pei.value=1')
                        ->where('ea.attribute_code="featured"');}
              else {

                $select = $read->select()
                        ->from(array('cp'=>$categoryProductTable))
                        ->join(array('pei'=>$productEntityIntTable), 'pei.entity_id=cp.product_id', array())
                        ->joinNatural(array('ea'=>$eavAttributeTable))
                        ->where('pei.value=1')
                        ->where('ea.attribute_code="featured"');
              }
            $featuredProductData = $read->fetchAll($select);
            $i=0;
            $product=array();
            $productid=array();
            foreach ($featuredProductData as $row) {

              // instantiate the product object
              //$productid[$i] = Mage::getModel('catalog/product')->load($row['product_id']);
              $productid[$i] = $row['product_id'];

              // if the product is a featured product, return the object
              // if ($product->getData('featured')) {

              //}
              $i++;
            }
          $productid=array_unique($productid);
          $i=0;
          foreach($productid as $id){
            $product[$i] = Mage::getModel('catalog/product')->load($id);
            $i++;
          }
          return $product;
          }
}
?>
```

We’re almost there!

Step 4) Extend Mage_Catalog_Block_Category_View

Create a new file, and directories, called app/code/local/MyCompany/Catalog/Block/Category/View.php. We’re extending the core class here so our module will be separate from the core code base. When upgrading, we won’t have to worry about our code not working or having to patch files.

```
<?php
  class MyCompany_Catalog_Block_Category_View extends Mage_Catalog_Block_Category_View
  {
    public function getFeaturedProductHtml()
    {
            return $this->getBlockHtml('product_featured');
        }
    }
?>
```

Step 5) Modify the templates

Edit app/design/frontend/default/default/template/catalog/category/view.phtml and add the following code:

```
<?=$this->getFeaturedProductHtml()?>
```

right above this line:

```
<?=$this->getProductListHtml()?>
```

Create app/design/frontend/default/default/template/catalog/product/featured.phtml and add some product info HTML to show the featured product. Here is an example that simply displays a link to the product:

```
<?php $_product=$this->getFeaturedProduct() ?>
Check this out: <a href="<?php echo $_product->getProductUrl() ?>"><?php echo $this->htmlEscape($_product->getName()) ?></a>
```

Step 6) Add new blocks to the app/etc/local.xml

Add the following inside the config global tag:

```
<blocks>
    <catalog>
        <rewrite>
            <product_featured>MyCompany_Catalog_Block_Product_Featured</product_featured>
        </rewrite>
        <rewrite>
            <category_view>MyCompany_Catalog_Block_Category_View</category_view>
        </rewrite>
    </catalog>
</blocks>
```

Questions : 24	**How too reset admin password in magento ?**
Answers :	If we forget admin password and want to update this admin password then we have to just do litil bet

24	effort just need to run below query on our sql

database

SELECT * FROM admin_user;

Then, find the username we want to modify in the listing provided ‘pcds’ in this example. Then, to update the password, type

UPDATE admin_user SET password=CONCAT(MD5('Pcinfotech'), ':Pc') WHERE username='pcds';

‘Pc’ would be changed to whatever we want it to be and same goes for ‘infotech’

Questions : 25	**What is the latest version of magento?**
Answers : 25	Latest version of magento is **1.9.0.1** - Added on May 16, 2014

For any query---

Visit-www.shwetankeducation.com