1) Explain what is Drupal?

Drupal is a Content Management System (CMS).  It’s an open source content and free CMS framework written in PHP.  It enables you to organize, manage and publish content with ease and comes with a variety of customization option.

2) Why Drupal is considered powerful CMS compare to other CMS system?

Using content construction kit Drupal allows you to create multiple content types without any [programming](http://career.guru99.com/category/programming-2/) skills. And for each content type it allows customize theme template. You can have an article, story, slideshow, content type and so on.

3) Mention what are the key features of Drupal?

There are two key features of Drupal

* Upload Module: It enables users to attach different types of files to node types of your liking.
* Embedding: It enables to embed media within nodes that are hosted on your websites.

4) Explain how caching work in Drupal?

Through caching Drupal allows to speed up website using different configuration like page caching, block caching and lifetime for cached pages.

* Page Caching:  It enables the entire HTML of each page to be stored in the [database](http://career.guru99.com/category/database/).  It reduces the amount of queries needed
* Block Caching:  You can set the cache setting for a block in views, when block created by views
* Minimum cache lifetime:  It is the amount of time before the page cache is cleared. On each cron run, page caches are cleared.
* Expiration of cached pages: It only applies for an external mechanism, for example, your browser cache or varnish

5) What is a patch?

A patch is a file that contains a list of differences between one set of files, and another.  Through patches, all the changes in code like additions or deletions to Drupal core can be made. Also, patches can be used to make changes to another copy of the original or main file.

6) What do PDO mean?

PDO means PHP Data Object; it is a lean and consistent way to access databases. It allows developers to write portable code with ease.

7) Explain how database system of Drupal works?

In a database Drupal stores information, and each information has its own database table.  For example, the basic information regarding the nodes of your site are stored and kept in the node table, and if you use the CCK module to add fields to your nodes, the field information is stored in a separate table.

) Name some of the most commonly used PHP based CMSs?

Some of the commonly used PHP are

* Drupal
* Joomla
* WordPress
* TYPO3

9) What is DRUSH in Drupal?

DRUSH is a command line shell and UNIX scripting interface for Drupal

10) Explain what is module in Drupal? List out some of the modules used in Drupal?

Modules are like plugin for your site. It allows you to add different functions to your site such as polls, contact forms and search fields. Some of the modules recommended are

* Views
* Token
* Ctools
* Quicktabs
* Pathauto

11) Explain what Drupal distributions is and when to use?

Distributions are full copies of Drupal, which includes Drupal Core, along with additional software as modules, themes, libraries and installation profiles. You can use distributions for

* Evaluating Drupal
* Demoing Drupal
* Learning Drupal
* Quickly binding Site

12) List out the SEO modules available in Drupal?

Some of the SEO modules available in Drupal are

* Pathauto
* Meta tags/ Node words
* Service Links
* Google Analytics
* Related Links
* Search 404
* Site map
* Url list

13) Explain what is Drupal weight?

The term Drupal weight is used to describe the priority or order in which the function is processed, or block/ node is displayed.

A heavier (+10) weights will appear lower in lists while a lower weight (-10) will float to the top of lists.

14) Explain what is Drupal cron?

To execute commands or scripts automatically at specified time and date intervals, you have to use Drupal cron.

15) Why Drupal needs database? What databases are supported?

Drupal stores its information in the database like individual pages, registered users and so on.  For your Drupal site, database forms the back end; it supports MySQL /MariaDB, PostgreSQL and in Drupal 7, SQLite.

16) Explain what is render array in Drupal?

For Drupal content render arrays are the basic building blocks.  In Drupal, render arrays provide you a structured way to programmatically change the content before it is displayed.

17) Explain what is the use of Ctools in Drupal?

This suite is a set of APIs and tools, and it makes easier to handle AJAX requests and tell the client what to do with them.

18) What are the files required for theme and module?

The files required for

* Theme: style.css , page.tpl.php , template.php, block.tpl.php
* Module: modulename.info , modulename.module , optional modulename.install

19) Explain using module how you can drop the table?

To drop the table you can use db\_drop\_table in install file.

20) What are the translations available in Drupal?

Drupal is translated into all popular languages including

* Catalan
* French
* Hungarian
* Dutch

21) Does Drupal provides support?

Like with all open-source projects there is no commercial support available. Though, there is a vast online community for Drupal available via Forums and IRC online chat who are more than eager to answer your queries.

**22) Name and describe the five conceptual layers in a Drupal system.**

The five layers, starting from the bottom layer, are as follows:

1. Data (nodes, etc.). Before anything can be displayed on the site, it must be input as data.
2. Modules. Modules are functional plugins that are either part of the Drupal core or are contributed modules that build on Drupal’s core functionality.
3. Blocks and menus. Blocks can be used to present anything, so just about any piece of content on a Drupal site (other than main content, breadcrumbs, and primary/secondary menus) is usually a block. Blocks are an extensible core feature of Drupal with a simple API provided by the block module. Blocks are similar to “widgets” in content management systems, but are highly generalized. Menus are a collection of links (menu items) used to navigate a website. The Menu module provides an interface to control and customize the menu system that comes with Drupal. By default, new menu items are placed inside a built-in menu labeled Navigation, but administrators can also create custom menus.
4. User permissions. User permissions are defined for various roles and users are assigned to these roles in order to grant them the defined permissions.
5. Themes and templates. The top conceptual layer of the drupal architecture is the theme. This consists primarily of XHTML and CSS, with some PHP variables intermixed, so Drupal-generated content can go in the appropriate spots. Also included with each theme is a set of functions that can be used to override standard functions in the modules in order to provide complete control over how the modules generate their markup at output time. A theme may contain one or more templates, depending on the complexity of the site and the way it has been designed.

**23) What is Drupal’s taxonomy system and what are some of its key features?**

Drupal comes with a built in taxonomy system that allows for categorization of the nodes on a site.

The taxonomy system allows for arbitrary definition of terms, as well as arbitrary organization of those terms into vocabularies. There is no limit to the number of vocabularies that can be created, nor is there any limit to the number of terms that can be included in a vocabulary.

A vocabulary can also have free tagging which means that, instead of entering specific terms ahead of time, users may enter tags freely at the time the content is created and those tags automatically become terms in that vocabulary.

Drupal’s taxonomy system is one if its most powerful and flexible features.

**24) Describe the Field API that was introduced into core in Drupal 7.**

The Field API allows custom data fields to be attached to Drupal entities and takes care of storing, loading, editing, and rendering field data. Any entity type (node, user, etc.) can use the Field API to make itself “fieldable” and thus allow fields to be attached to it. Other modules can provide a user interface for managing custom fields via a web browser as well as a wide and flexible variety of data type, form element, and display format capabilities.

The Field API defines two primary data structures, Field and Instance, and the concept of a Bundle. A Field defines a particular type of data that can be attached to entities. A Field Instance is a Field attached to a single Bundle. A Bundle is a set of fields that are treated as a group by the Field Attach API and is related to a single fieldable entity type.

For example, suppose a site administrator wants Article nodes to have a subtitle and photo. Using the Field API or Field UI module, the administrator creates a field named ‘subtitle’ of type ‘text’ and a field named ‘photo’ of type ‘image’. The administrator (again, via a UI) creates two Field Instances, one attaching the field ‘subtitle’ to the ‘node’ bundle ‘article’ and one attaching the field ‘photo’ to the ‘node’ bundle ‘article’. When the node system uses the Field Attach API to load all fields for an Article node, it passes the node’s entity type (which is ‘node’) and content type (which is ‘article’) as the node’s bundle. field\_attach\_load() then loads the ‘subtitle’ and ‘photo’ fields because they are both attached to the ‘node’ bundle ‘article’.

Field definitions are represented as an array of key/value pairs.

Note that the above answer has been excerpted from the [Drupal API Documentation](https://api.drupal.org/api/drupal/modules!field!field.module/group/field/7), where more information can be found.

**25) Explain the concept of “nodes” in a Drupal system.**

All content on a Drupal website is stored and treated as “nodes”. A node is any piece of individual content (e.g., a page, article, forum topic, blog entry, etc.). Note, though, that omments are not stored as nodes but are always connected to a node.

The ability to create different “content types” is a way Drupal allows you to have different kinds of nodes for different purposes. For example, an “article” is one content type, a “book page” is another, and a “blog entry” yet another. You can also create new content types of your own.

Treating all content as nodes provides a great deal of flexibility that facilitates and simplifies creating new types of content. It also makes it easy to apply new features or changes to all content of a particular type.

**26) Describe the features and uses of the Views module.**

Using the Views module, you can fetch content from the database of your site and present it to the user as lists, posts, galleries, tables, maps, graphs, menu items, blocks, reports, forum posts etc. Different content types including nodes, users, and other bundles can be displayed.

Views UI, a submodule within Views, provides a graphical interface underneath which lies a powerful SQL query builder that can access virtually any information in your database and display it in any format.

Different displays can present the query results as pages with fixed URLs on your site (or URLs accepting arguments), blocks, feeds, or panel panes.

You can also use Views to present related content or implement contextual filters. For example, you can display a list of users along with links to the content they have created and/or you can display customized content to a user according to their user ID.

More information is available in the [Views documentation](https://www.drupal.org/project/views) on the Drupal site.

**27) What are appropriate use cases for Drupal as opposed to a lower-level framework like Ruby on Rails?**

Rails is a general purpose web application framework for Ruby. It’s designed to help programmers be more productive in building web sites of all types, not just CMSs. And note that it’s for “programmers”. Unless you intend to write server-side code, you can’t hope to get much done with just Rails alone (but if you do intend to write code, both Ruby and Rails are known to be very productive).

Drupal is a Content Management System, of which there are hundreds. It happens to be written in PHP, but it’s modular design and large collection of available modules and themes make it possible to design and implement a large variety of web sites without writing a line of code. Most importantly, though, it is fundamntally a CMS. The further your site’s needs are from the CMS sweet-spot, the less likely it is that Drupal will be your best choice.

**28) What is a theme override?**

In Drupal, a **theme** function is a PHP function that is used to output the HTML of any Drupal object. These functions are prefixed with theme\_. All the functions that produce HTML are themeable and are invoked using **theme**() rather than being directly called. We can **override theme** functions in template.php.

**29) What's the difference between a theme override and a template file?**

In drupal, a theme handles the appearance of the site and a template handles how your content is rendered.Think of it like this: the template is used to render the content, then the theme is applied on that content.

**30) How to make my Drupal site offline to public, while it is under construction?**

**Answer: -** You can set your Drupal site in off-line mode, while it is being developed. Just click Administer » Site maintenance. There you can set the status to off-line. If you wants, you can also set your own custom off-line message. When set to Off-line, only users with the administer site configuration permission will be able to access your site to perform maintenance; all other visitors will see the site off-line message configured there. Authorized users can log in during Off-line mode directly via the user login page.

**31) How can I reset my Drupal admin password**

**Answer:-** In such a worst scenario, drupal admin password can easily be reset from the Drupal database. The Drupal database can be accessed through phpMyAdmin tool available with your web hosting account. Follow the below steps to get your drupal admin password reset quickly:

· Login to cPanel -> Databases box -> phpMyAdmin;

·Select the Druapl database folder from the left navigation bar. The page will refresh and and the Drupal database’s tables will be displayed on it.

· Click on the SQL tab.

·In the text field write the following SQL query:

update users set pass=md5(’NEWPASS’) where uid = 1; where “NEWPASS” is your new Drupal administrative password.

Click the GO button to submit the query. If the query is executed correctly and no errors are displayed then you should be able to login with the new password.

**32) How can I change the favicon in my Drupal Site?**

**Answer:-** A favicon (short for favorites icon), also known as a website icon or bookmark icon is a 16×16 pixel square icon that appears near the address bar and in the bookmarks folder in a visitor’s browser. By default, a drupal site shows that water drop kinda drupal logo as favicon. If you want to make your site unique by all means, you can easily change the favicon to your customized icon by following the steps below.

·         Create your own favicon.ico file using any graphic tools or with the help of any online favicon generator tools like dnamicdrive.

·         Navigate to admin » site building » themes and click the configure link next to your current theme. This will bring up the theme configuration page.

·         Here you will see a section titled Shortcut icons settings. You can either upload your favicon file or specify the path to your customized icon file.

The changes may not appear immediately in your browser, you need to clear your browser’s cache and reload the page. If you have bookmarked your site, you may need to delete the bookmark and then recreate it again so that the new favicon will appear in the bookmarks menu.

**Technical Question of php**

**1 - $color = array('white', 'green', 'red'')  
Write a PHP script which will display the colors in the following way :**

***Output :***  
white, green, red,

* green
* red
* white

***Code :***

<?php

$color = array('white', 'green', 'red');

foreach ($color as $c)

{

echo "$c, ";

}

sort($color);

echo "<ul>";

foreach ($color as $y)

{

echo "<li>$y</li>";

}

echo "</ul>";

?>

**2 - $ceu = array( "Italy"=>"Rome", "Luxembourg"=>"Luxembourg", "Belgium"=> "Brussels", "Denmark"=>"Copenhagen", "Finland"=>"Helsinki", "France" => "Paris", "Slovakia"=>"Bratislava", "Slovenia"=>"Ljubljana", "Germany" => "Berlin", "Greece" => "Athens", "Ireland"=>"Dublin", "Netherlands"=>"Amsterdam", "Portugal"=>"Lisbon", "Spain"=>"Madrid", "Sweden"=>"Stockholm", "United Kingdom"=>"London", "Cyprus"=>"Nicosia", "Lithuania"=>"Vilnius", "Czech Republic"=>"Prague", "Estonia"=>"Tallin", "Hungary"=>"Budapest", "Latvia"=>"Riga", "Malta"=>"Valetta", "Austria" => "Vienna", "Poland"=>"Warsaw") ;  
  
Create a PHP script which display the capital and country name from the above array $ceu. Sort the list by the name of the country.**

***Sample Output :***  
The capital of Netherlands is Amsterdam  
The capital of Greece is Athens  
The capital of Germany is Berlin

***Code :***

<?php

$ceu = array( "Italy"=>"Rome", "Luxembourg"=>"Luxembourg",

"Belgium"=> "Brussels", "Denmark"=>"Copenhagen",

"Finland"=>"Helsinki", "France" => "Paris",

"Slovakia"=>"Bratislava", "Slovenia"=>"Ljubljana",

"Germany" => "Berlin", "Greece" => "Athens",

"Ireland"=>"Dublin", "Netherlands"=>"Amsterdam",

"Portugal"=>"Lisbon", "Spain"=>"Madrid",

"Sweden"=>"Stockholm", "United Kingdom"=>"London",

"Cyprus"=>"Nicosia", "Lithuania"=>"Vilnius",

"Czech Republic"=>"Prague", "Estonia"=>"Tallin",

"Hungary"=>"Budapest", "Latvia"=>"Riga","Malta"=>"Valetta",

"Austria" => "Vienna", "Poland"=>"Warsaw") ;

asort($ceu) ;

foreach($ceu as $country => $capital)

{

echo "The capital of $country is $capital <br>" ;

}

?>

**3 - $x = array(1, 2, 3, 4, 5);  
  
Delete an element from the above array. After deleting the element, integer keys must be normalized.**

***Sample Output :***  
array(5) { [0]=> int(1) [1]=> int(2) [2]=> int(3) [3]=> int(4) [4]=> int(5) }   
array(4) { [0]=> int(1) [1]=> int(2) [2]=> int(3) [3]=> int(5) }

***Code :***

<?php

$x = array(1, 2, 3, 4, 5);

var\_dump($x);

unset($x[3]);

$x = array\_values($x);

echo '

';

var\_dump($x);

?>

**4 - $color = array(4 => 'white', 6 => 'green', 11=> 'red');  
  
Write a PHP script to get the first element of the above array.**

***Expected result :*** white

***Code :***

<?php

$color = array(4 => 'white', 6 => 'green', 11=> 'red');

echo reset($color);

?>

5 - **Write a PHP script which decode the following JSON string.**

***Sample JSON code :***  
{"Title": "The Cuckoos Calling",  
"Author": "Robert Galbraith",  
"Detail": {  
"Publisher": "Little Brown"  
}}  
*Expected Output :*  
Title : The Cuckoos Calling  
Author : Robert Galbraith  
Publisher : Little Brown

***Code:***  
<?php

function w3rfunction($value,$key)

{

echo "$key : $value<br>";

}

$a = '{"Title": "The Cuckoos Calling",

"Author": "Robert Galbraith",

"Detail":

{

"Publisher": "Little Brown"

}

}';

$j1 = json\_decode($a,true);

array\_walk\_recursive($j1,"w3rfunction");

?>

**6 - Write a PHP script that insert a new item in an array on any position.**

***Expected Output :***  
Original array :   
1 2 3 4 5   
After inserting '$' the array is :  
1 2 3 $ 4 5

***Code:***  
<?php

$original = array( '1','2','3','4','5' );

echo 'Original array : <br>';

foreach ($original as $x)

{echo "$x ";}

$inserted = '$';

array\_splice( $original, 3, 0, $inserted );

echo " <br> After inserting '$' the array is :<br>";

foreach ($original as $x)

{echo "$x ";}

?>

**7 - Write a PHP script to sort the following associative array :**

array("Sophia"=>"31","Jacob"=>"41","William"=>"39","Ramesh"=>"40") in   
a) ascending order sort by value  
b) ascending order sort by Key  
c) descending order sorting by Value  
d) descending order sorting by Key

***Code :***   
<?php

echo "

Associative array : Ascending order sort by value

";

$array2=array("Sophia"=>"31","Jacob"=>"41","William"=>"39","Ramesh"=>"40"); asort($array2);

foreach($array2 as $y=>$y\_value)

{

echo "Age of ".$y." is : ".$y\_value."

";

}

echo "

Associative array : Ascending order sort by Key

";

$array3=array("Sophia"=>"31","Jacob"=>"41","William"=>"39","Ramesh"=>"40"); ksort($array3);

foreach($array3 as $y=>$y\_value)

{

echo

"Age of ".$y." is : ".$y\_value."

";

}

echo "

Associative array : Descending order sorting by Value

";

$age=array("Sophia"=>"31","Jacob"=>"41","William"=>"39","Ramesh"=>"40");

arsort($age);

foreach($age as $y=>$y\_value)

{

echo "Age of ".$y." is : ".$y\_value."

";

}

echo "

Associative array : Descending order sorting by Key

";

$array4=array("Sophia"=>"31","Jacob"=>"41","William"=>"39","Ramesh"=>"40"); krsort($array4);

foreach($array4 as $y=>$y\_value)

{

echo "Age of ".$y." is : ".$y\_value."

";

}

?>

8 - **Write a PHP script to calculate and display average temperature, five lowest and highest temperatures.**

**Recorded temperatures : 78, 60, 62, 68, 71, 68, 73, 85, 66, 64, 76, 63, 75, 76, 73, 68, 62, 73, 72, 65, 74, 62, 62, 65, 64, 68, 73, 75, 79, 73**

***Expected Output :***  
Average Temperature is : 70.6   
List of seven lowest temperatures : 60, 62, 63, 63, 64,   
List of seven highest temperatures : 76, 78, 79, 81, 85,

***Code :***  
<?php

$month\_temp = "78, 60, 62, 68, 71, 68, 73, 85, 66, 64, 76, 63, 81, 76, 73,

68, 72, 73, 75, 65, 74, 63, 67, 65, 64, 68, 73, 75, 79, 73";

$temp\_array = explode(',', $month\_temp);

$tot\_temp = 0;

$temp\_array\_length = count($temp\_array);

foreach($temp\_array as $temp)

{

$tot\_temp += $temp;

}

$avg\_high\_temp = $tot\_temp/$temp\_array\_length;

echo "Average Temperature is : ".$avg\_high\_temp."

";

sort($temp\_array);

echo "<br> List of seven lowest temperatures : ";

for ($i=0; $i< 5; $i++)

{

echo $temp\_array[$i].", ";

}

echo "<br>List of seven highest temperatures : ";

for ($i=($temp\_array\_length-5); $i< ($temp\_array\_length); $i++)

{

echo $temp\_array[$i].", ";

}

?>

**9 - Write a PHP program to sort an array of positive integers using the Bead Sort Algorithm.[Go to the editor](http://www.w3resource.com/php-exercises/php-array-exercises.php" \l "editorr)  
According to Wikipedia "Bead sort is a natural sorting algorithm, developed by Joshua J. Arulanandham, Cristian S. Calude and Michael J. Dinneen in 2002. Both digital and analog hardware implementations of bead sort can achieve a sorting time of O(n); however, the implementation of this algorithm tends to be significantly slower in software and can only be used to sort lists of positive integers".**  
  
*Input array* : Array ( [0] => 5 [1] => 3 [2] => 1 [3] => 3 [4] => 8 [5] => 7 [6] => 4 [7] => 1 [8] => 1 [9] => 3 )   
*Expected Result* : Array ( [0] => 8 [1] => 7 [2] => 5 [3] => 4 [4] => 3 [5] => 3 [6] => 3 [7] => 1 [8] => 1 [9] => 1 )

***Code:***  
<?php

function columns($uarr)

{

$n=$uarr;

if (count($n) == 0)

return array();

else if (count($n) == 1)

return array\_chunk($n[0], 1);

array\_unshift($uarr, NULL);

$transpose = call\_user\_func\_array('array\_map', $uarr);

return array\_map('array\_filter', $transpose);

}

function bead\_sort($uarr)

{

foreach ($uarr as $e)

$poles []= array\_fill(0, $e, 1);

return array\_map('count', columns(columns($poles)));

}

echo 'Original Array : '.'

';

print\_r(array(5,3,1,3,8,7,4,1,1,3));

echo '

'.'After Bead sort : '.'

';

print\_r(bead\_sort(array(5,3,1,3,8,7,4,1,1,3)));

?>

**10 - Write a PHP program to merge (by index) the following two arrays.**

***Sample arrays* :**  
$array1 = array(array(77, 87), array(23, 45));  
$array2 = array("w3resource", "com");

***Code:***

<?php

$array1 = array(array(77, 87), array(23, 45));

$array2 = array("w3resource", "com");

function merge\_arrays\_by\_index($x, $y)

{

$temp = array(); $temp[] = $x; if(is\_scalar($y))

{

$temp[] = $y;

}

else

{

foreach($y as $k => $v)

{

$temp[] = $v;

}

}

return $temp;

}

echo '<pre>'; print\_r(array\_map('merge\_arrays\_by\_index',$array2, $array1));

?>

**11- Write a PHP function to change the following array's all values to upper or lower case.**

***Sample arrays* :**  
$Color = array('A' => 'Blue', 'B' => 'Green', 'c' => 'Red');  
*Expected Output* :   
Values are in lower case.  
Array ( [A] => blue [B] => green [c] => red )   
Values are in upper case.  
Array ( [A] => BLUE [B] => GREEN [c] => RED )

***Code:***  
<?php

function array\_change\_value\_case($input, $ucase)

{

$case = $ucase;

$narray = array();

if (!is\_array($input))

{

return $narray;

}

foreach ($input as $key => $value)

{

if (is\_array($value))

{

$narray[$key] = array\_change\_value\_case($value, $case);

continue;

}

$narray[$key] = ($case == CASE\_UPPER ? strtoupper($value) : strtolower($value));

}

return $narray;

}

$Color = array('A' => 'Blue', 'B' => 'Green', 'c' => 'Red');

echo 'Actual array <br>';

print\_r($Color);

echo '<br>Values are in lower case.<br>';

$myColor = array\_change\_value\_case($Color,CASE\_LOWER);

print\_r($myColor);

echo '<br>Values are in upper case.<br>';

$myColor = array\_change\_value\_case($Color,CASE\_UPPER);

print\_r($myColor);

?>

**12 - Write a PHP script which display all the numbers between 200 and 250 that are divisible by 4.**

**Note : Do not use any PHP control statement.**  
  
***Expected Output* :** 200,204,208,212,216,220,224,228,232,236,240,244,248

***Code :***   
<?php

echo implode(",",range(200,250,4));

?>

Write a PHP script to get the shortest/longest string length from an array.

*Sample arrays* : ("abcd","abc","de","hjjj","g","wer")  
*Expected Output* : The shortest array length is 1. The longest array length is 4.

<?php

$my\_array = array("abcd","abc","de","hjjj","g","wer");

$new\_array = array\_map('strlen', $my\_array);

// Show maximum and minimum string length using max() function and min() function

echo "The shortest array length is " . min($new\_array) .

". The longest array length is " . max($new\_array).'.';

?>

**13 - Write a PHP script to generate unique random numbers within a range.**

*Sample Range*: (11, 20)   
  
*Sample Output* : 17 16 13 20 14 19 18 15 11 12

***Code :***   
<?php

$n=range(11,20);

shuffle($n);

for ($x=0; $x< 10; $x++)

{

echo $n[$x].' ';

}

?>

**14- Write a PHP script to get the largest key in an array.**

***Code:***

<?php

$ceu = array( "Italy"=>"Rome", "Luxembourg"=>"Luxembourg", "Belgium"=> "Brussels",

"Denmark"=>"Copenhagen", "Finland"=>"Helsinki", "France" => "Paris", "Slovakia"=>"Bratislava",

"Slovenia"=>"Ljubljana", "Germany" => "Berlin", "Greece" => "Athens", "Ireland"=>"Dublin",

"Netherlands"=>"Amsterdam", "Portugal"=>"Lisbon", "Spain"=>"Madrid", "Sweden"=>"Stockholm",

"United Kingdom"=>"London", "Cyprus"=>"Nicosia", "Lithuania"=>"Vilnius", "Czech Republic"=>"Prague", "Estonia"=>"Tallin", "Hungary"=>"Budapest", "Latvia"=>"Riga", "Malta"=> "Valetta","Austria" => "Vienna", "Poland"=>"Warsaw") ;

$max\_key = max( array\_keys( $ceu) );

echo $max\_key;

?>

**15 - Write a PHP function that returns the lowest integer that is not 0.**

***Code:***

<?php

function min\_values\_not\_zero(Array $values)

{

return min(array\_diff(array\_map('intval', $values), array(0)));

}

print\_r(min\_values\_not\_zero(array(-1,0,1,12,-100,1)));

?>

**16 - Write a PHP function to floor decimal numbers with precision.**

Note: Accept three parameters number, precision and $separator  
*Sample Data* :   
1.155, 2, "."  
100.25781, 4, "."  
-2.9636, 3, "."  
 *Expected Output* :   
1.15   
100.2578   
-2.964

***Code:***

<?php

function floorDec($number, $precision, $separator)

{

$number\_part=explode($separator, $number);

$number\_part[1]=substr\_replace($number\_part[1],$separator,$precision,0);

if($number\_part[0]>=0)

{$number\_part[1]=floor($number\_part[1]);}

else

{$number\_part[1]=ceil($number\_part[1]);}

$ceil\_number= array($number\_part[0],$number\_part[1]);

return implode($separator,$ceil\_number);

}

print\_r(floorDec(1.155, 2, ".").'<br>');

print\_r(floorDec(100.25781, 4, ".").'<br>');

print\_r(floorDec(-2.9636, 3, "."));

?>

**17 - Write a PHP program which iterates the integers from 1 to 100. For multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".**

***Code:***

<?php

for ($i = 1; $i <= 100; $i++)

{

if ( $i%3 == 0 && $i%5 == 0 )

{

echo $i . " FizzBuzz".'<br />' ;

}

else if ( $i%3 == 0 )

{

echo $i. " Fizz".'<br />';

}

else if ( $i%5 == 0 )

{

echo $i. " Buzz".'<br />';

}

else

{

echo $i.'<br />';

}

}

?>

**18 - Write a PHP script that creates the following table (use for loops).**

***Code:***

<?php

echo "<table border =\"1\" style='border-collapse: collapse'>";

for ($row=1; $row <= 10; $row++) {

echo "<tr> \n";

for ($col=1; $col <= 10; $col++) {

$p = $col \* $row;

echo "<td>$p</td> \n";

}

echo "</tr>";

}

echo "</table>";

?>

**19-Write a PHP script using nested for loop that creates a chess board as shown below.**

**Use table width="270px" and take 30px as cell height and width.**

***Code:***

<!DOCTYPE html>

<html>

<head>

<title></title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

</head>

<body>

<h3>Chess Board using Nested For Loop</h3>

<table width="270px" cellspacing="0px" cellpadding="0px" border="1px">

<!-- cell 270px wide (8 columns x 60px) -->

<?php

for($row=1;$row<=8;$row++)

{

echo "<tr>";

for($col=1;$col<=8;$col++)

{

$total=$row+$col;

if($total%2==0)

{

echo "<td height=30px width=30px bgcolor=#FFFFFF></td>";

}

else

{

echo "<td height=30px width=30px bgcolor=#000000></td>";

}

}

echo "</tr>";

}

?>

</table>

</body>

</html>

</html>

**20- Write a PHP script that creates the following table using for loops. Add cellpadding="3px" and cellspacing="0px" to the table tag.**

***Code:***

<html>

<body>

<table align="left" border="1" cellpadding="3" cellspacing="0">

<?php

for($i=1;$i<=6;$i++)

{

echo "<tr>";

for ($j=1;$j<=5;$j++)

{

echo "<td>$i \* $j = ".$i\*$j."</td>";

}

echo "</tr>";

}

?>

</table>

</body>

</html>

**21- Write a PHP function that checks if a string is all lower case.**

***Code:***

<?php

function is\_str\_lowercase($str1)

{

for ($sc = 0; $sc < strlen($str1); $sc++) {

if (ord($str1[$sc]) >= ord('A') &&

ord($str1[$sc]) <= ord('Z')) {

return false;

}

}

return true;

}

var\_dump(is\_str\_lowercase('abc def ghi'));

var\_dump(is\_str\_lowercase('abc dEf ghi'));

?>

**22-Write a function to calculate the factorial of a number (non-negative integer). The function accept the number as a argument.**

***Code:***

<?php

function factorial\_of\_a\_number($n)

{

if($n ==0)

{

return 1;

}

else

{

return $n \* factorial\_of\_a\_number($n-1);

}

}

print\_r(factorial\_of\_a\_number(4));

?>

**23- Write a PHP function that checks whether a passed string is palindrome or not?**

***Code:***

<?php

function check\_palindrome($string)

{

if ($string == strrev($string))

return 1;

else

return 0;

}

echo check\_palindrome('madam');

?>

**17 - Write a PHP script to print "second" and Red from the following array.**  
*Sample Data* :   
$color = array ( "color" => array ( "a" => "Red", "b" => "Green", "c" => "White"),  
"numbers" => array ( 1, 2, 3, 4, 5, 6 ),  
"holes" => array ( "First", 5 => "Second", "Third"));

***Code:***

<?php

$color = array ( "color" => array ( "a" => "Red", "b" => "Green", "c" => "White"),

"numbers" => array ( 1, 2, 3, 4, 5, 6 ),

"holes" => array ( "First", 5 => "Second", "Third"));

echo $color["holes"][5].'<br>'; // prints "second"

echo $color["color"]["a"]; // prints "Red"

?>

Create a script to construct the following pattern, using nested for loop.

\*

\*\*

\*\*\*

<?php

for($x=1;$x<=5;$x++)

{

for ($y=1;$y<=$x;$y++)

{

echo "\*";

if($y< $x)

{

echo " ";

}

}

echo "<br>";

}

?>

**18 - Create a script using a for loop to add all the integers between 0 and 30 and display the sum..**

***Code:***

<?php

$sum = 0;

for($x=1; $x<=30; $x++)

{

$sum +=$x;

}

echo "The sum of the numbers 0 to 30 is $sum";

?>

**19 : Create a script to construct the following pattern, using a nested for loop.**

1

11

111

1111

11111

1111

111

11

1

***Code:***

<?php

$n=5;

for($i=1; $i<=$n; $i++)

{

for($j=1; $j<=$i; $j++)

{

echo ' 1 ';

}

echo '<br>';

}

for($i=$n; $i>=1; $i--)

{

for($j=1; $j<=$i; $j++)

{

echo ' 1 ';

}

echo '<br>';

}

?>

**20 - Write a program to calculate and print the factorial of a number using a for loop. The factorial of a number is the product of all integers up to and including that number, so the factorial of 4 is 4\*3\*2\*1= 24.**

***Code:***

<?php

$n = 6;

$x = 1;

for($i=1;$i<=$n-1;$i++)

{

$x\*=($i+1);

}

echo "The factorial of $n = $x";

?>

**21-Write a program which will give you all of the potential combinations of a two-digit decimal combination, printed in a comma delimited format.**  
  
Sample output :  
00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99,

***Code:***

<?php

for($a=0; $a< 10; $a++)

{

for($b=0; $b< 10; $b++)

{

echo $a.$b.", ";

}

}

?>

**22- Write a program which will count the "r" characters in the text "w3resource".**

***Code:***

<?php

$text="w3resource";

$search\_char="r";

$count="0";

for($x="0"; $x< strlen($text); $x++)

{

if(substr($text,$x,1)==$search\_char)

{

$count=$count+1;

}

}

echo $count;

?>