Search | Compartilhar | 1 | mais ▾ Próximo blog»   Criar um blog   Login

PHP geeks wink

Because every semi-colon is important ;)

Please shower your <3 by making +1s and subscribing via email

| Home | Chalk Talk | PHP Interview | ZCE | PHP Basics Mock Test | My Profile |

Tuesday, May 22, 2012

Zend Certified Engineer ZCE PHP Basics Mock Test

Questions

1. Choose the selection that best matches the following statements:
PHP is a _____ scripting language based on the _____ engine. It is primarily used to develop dynamic _____ content, although it can be used to generate _____ documents (among others) as well.

   A.   Dynamic, PHP, Database, HTML
   B.   Embedded, Zend, HTML, XML
   C.   Perl-based, PHP, Web, Static
   D.   Embedded, Zend, Docbook, MySQL
   E.   Zend-based, PHP, Image, HTML

2. Which of the following tags is not a valid way to begin and end a PHP code block?

   A.   <% %>
   B.   <? ?>
   C.   <?= ?>
   D.   <! !>
   E.   <?php ?>

3. Which of the following is not valid PHP code?

   A. $_10
   B. ${"MyVar"}
   C.   &$something
   D. $10_somethings
   E. $aVaR

4. What is displayed when the following script is executed?

```
<?php
    define(myvalue, "10");
    $myarray[10] = "Dog";
    $myarray[] = "Human";
    $myarray['myvalue'] = "Cat";
    $myarray["Dog"] = "Cat";
    print "The value is: ";
    print $myarray[myvalue]."\n";
?>
```

   A.   The value is: Dog
   B.   The value is: Cat
   C.   The value is: Human
   D.   The value is: 10
   E.   Dog

Followers

Subscribe via Email

Email address...   Submit

PHP Geeks Wink

About Me

charanjeet kaur
View my complete profile

Blog Archive

5. What is the difference between print() and echo()?

    A.   print() can be used as part of an expression, while echo() can't
    B.   echo() can be used as part of an expression, while print() can't
    C.   echo() can be used in the CLI version of PHP, while print() can't
    D.   print() can be used in the CLI version of PHP, while echo() can't
    E.   There's no difference: both functions print out some text!

6. What is the output of the following script?

```php
<?php
    $a = 10;
    $b = 20;
    $c = 4;
    $d = 8;
    $e = 1.0;
    $f = $c + $d * 2;
    $g = $f % 20;
    $h = $b - $a + $c + 2;
    $i = $h << $c;
    $j = $i * $e;
    print $j;
?>
```

    A.   128
    B.   42
    C.   242.0
    D.   256
    E.   342

7. Which values should be assigned to the variables $a, $b and $c in order for the following script to display the string Hello, World!?

```php
<?php
    $string = "Hello, World!";
    $a = ?;
    $b = ?;
    $c = ?;
    if($a) {
        if($b && !$c) {
            echo "Goodbye Cruel World!";
        } else if(!$b && !$c) {
            echo "Nothing here";
        }
    } else {
        if(!$b) {
            if(!$a && (!$b && $c)) {
                echo "Hello, World!";
            } else {
                echo "Goodbye World!";
            }
        } else {
            echo "Not quite.";
        }
    }
?>
```

    A. False, True, False
    B. True, True, False
    C. False, True, True
    D. False, False, True
    E. True, True, True

8. What will the following script output?

```php
<?php
$array = '0123456789ABCDEFG';
```

```php
$s = '';
for ($i = 1; $i < 50; $i++) {
    $s .= $array[rand(0,strlen ($array) - 1)];
}
echo $s;
?>
```

    A. A string of 50 random characters
    B. A string of 49 copies of the same character, because the random number generator
       has not been initialized
    C. A string of 49 random characters
    D. Nothing, because $array is not an array
    E. A string of 49 'G' characters

9. Which language construct can best represent the following series of if conditionals?

```php
<?php
    if($a == 'a') {
        somefunction();
    } else if ($a == 'b') {
        anotherfunction();
    } else if ($a == 'c') {
        dosomething();
    } else {
        donothing();
    }
?>
```

    A. A switch statement without a default case
    B. A recursive function call
    C. A while statement
    D. It is the only representation of this logic
    E. A switch statement using a default case

10. What is the best way to iterate through the $myarray array, assuming you want to modify the value of each element as you do?

```php
<?php
    $myarray = array ("My String",
                "Another String",
                "Hi, Mom!");
?>
```

    A. Using a for loop
    B. Using a foreach loop
    C. Using a while loop
    D. Using a do…while loop
    E. There is no way to accomplish this goal

11. Consider the following segment of code:

```php
<?php
    define("STOP_AT", 1024);
    $result = array();
    /* Missing code */
    {
        $result[] = $idx;
    }
    print_r($result);
?>
```

What should go in the marked segment to produce the following array output?

```
Array
(
    [0] => 1
    [1] => 2
    [2] => 4
    [3] => 8
    [4] => 16
    [5] => 32
    [6] => 64
```

```
            [7] => 128
            [8] => 256
            [9] => 512
       }
```

A. foreach($result as $key => $val)
B. while($idx *= 2)
C. for($idx = 1; $idx < STOP_AT; $idx *= 2)
D. for($idx *= 2; STOP_AT >= $idx; $idx = 0)
E. while($idx < STOP_AT) do $idx *= 2

12. Choose the appropriate function declaration for the user-defined function is_leap(). Assume that, if not otherwise defined, the is_leap function uses the year 2000 as a default value:

```php
<?php
/* Function declaration here */
{
    $is_leap = (!($year %4) && (($year % 100) ||
            !($year % 400)));

    return $is_leap;
}
var_dump(is_leap(1987));       /* Displays false */
var_dump(is_leap());           /* Displays true */
?>
```

A. function is_leap($year = 2000)
B. is_leap($year default 2000)
C. function is_leap($year default 2000)
D. function is_leap($year)
E. function is_leap(2000 = $year)

13. What is the value displayed when the following is executed? Assume that the code was executed using the following URL:
    testscript.php?c=25

```php
<?php
    function process($c, $d = 25)
    {
        global $e;
        $retval = $c + $d - $_GET['c'] - $e;
        return $retval;
    }
    $e = 10;
    echo process(5);
?>
```

A. 25
B. -5
C. 10
D. 5
E. 0

14. Consider the following script:

```php
<?php
    function myfunction($a, $b = true)
    {
        if($a && !$b) {
            echo "Hello, World!\n";
        }

    }
    $s = array(0 => "my",
            1 => "call",
            2 => '$function',
            3 => ' ',
            4 => "function",
            5 => '$a',
            6 => '$b',
            7 => 'a',
```

```
                    8 => 'b',
                    9 => '');
            $a = true;
            $b = false;
            /* Group A */
            $name = $s[?].$s[?].$s[?].$s[?].$s[?].$s[?];
            /* Group B */
            $name(${$s[?]}, ${$s[?]});
    ?>
```

Each ? in the above script represents an integer index against the $s array. In order to display the Hello, World! string when executed, what must the missing integer indexes be?

    A. Group A: 4,3,0,4,9,9 Group B: 7,8
    B. Group A: 1,3,0,4,9,9 Group B: 7,6
    C. Group A: 1,3,2,3,0,4 Group B: 5,8
    D. Group A: 0,4,9,9,9,9 Group B: 7,8
    E. Group A: 4,3,0,4,9,9 Group B: 7,8

15. Run-time inclusion of a PHP script is performed using the _____ construct, while compile-time inclusion of PHP scripts is performed using the _____ construct.

    A. include_once, include
    B. require, include
    C. require_once, include
    D. include, require
    E. All of the above are correct

16. Under what circumstance is it impossible to assign a default value to a parameter while declaring a function?

    A. When the parameter is Boolean
    B. When the function is being declared as a member of a class
    C. When the parameter is being declared as passed by reference
    D. When the function contains only one parameter
    E. Never

17. The _____ operator returns True if either of its operands can be evaluated as True, but not both.
    Your Answer: _____

18. How does the identity operator === compare two values?

    A. It converts them to a common compatible data type and then compares the resulting values
    B. It returns True only if they are both of the same type and value
    C. If the two values are strings, it performs a lexical comparison
    D. It bases its comparison on the C strcmp function exclusively
    E. It converts both values to strings and compares them

19. Which of the following expressions multiply the value of the integer variable $a by 4? (Choose 2)

    A. $a *= pow (2, 2);
    B. $a >>= 2;
    C. $a <<= 2;
    D. $a += $a + $a;
    E. None of the above

20. How can a script come to a clean termination?

    A. When exit() is called
    B. When the execution reaches the end of the current file
    C. When PHP crashes
    D. When Apache terminates because of a system problem

For answers to above ZCE questions click here
http://www.php-geeks-wink.blogspot.in/search/label/zce-zend-certified-engineer-php-basic-mock-test-answers

Posted by charanjeet kaur at 1:51 AM   3 comments

M B t F 8 +1 Recommend this on Google

Labels: zce-zend-certified-engineer-php-basic-mock-test-answers, zce-zend-certified-engineer-php-basics-test-paper, zce-zend-certified-engineer-practice-test, zce-zend-certified-engineer-questions-answers

Sunday, May 20, 2012

Zend Certified Engineer ZCE PHP Basics Mock Test Answers

For Complete blog visit Zend Certified Engineer ZCE PHP Basics Mock Test

Answers for the above mentioned blog are

1. Looking at the answers, the only one that makes sense for every blank is B. PHP is a scripting language based on the Zend Engine that is usually embedded in HTML code. As such, it is primarily used to develop HTML documents, although it can be used just as nicely to develop other types of documents, such as XML.

2. While tags such as <% %> and <?= ?> are often forgotten in PHP programming, they are valid ways to delimit a PHP code block. The <! and !> tags, however, are not valid and, therefore, the correct answer is D. Keep in mind, in any case, that some of these tags are not always available, depending on how the php.ini file on which the PHP interpreter runs is configured.

3. PHP variables always start with a dollar sign and are a sequence of characters and numbers within the Latin alphabet, plus the underscore character. ${"MyVar"} is a valid variable name that simply uses a slightly less common naming convention, while &$something is a reference to the $something variable. Variables, however cannot start with numbers, making $10_somethings invalid and Answer D correct.

4. The important thing to note here is that the $myarray array's key value is being referenced without quotes around it. Because of this, the key being accessed is not the myvalue string but the value represented by the myvalue constant. Hence, it is equivalent to accessing $myarray[10], which is Dog, and Answer A is correct.

5. Even though print() and echo() are essentially interchangeable most of the time, there is a substantial difference between them. While print() behaves like a function with its own return value (although it is a language construct), echo() is actually a language construct that has no return value and cannot, therefore, be used in an expression. Thus, Answer A is correct.

6. Other than the simple math, the % operator is a modulus, which returns whatever the remainder would be if its two operands were divided. The << operator is a left-shift operator, which effectively multiplies an integer number by powers of two. Finally, the ultimate answer is multiplied by a floating point and, therefore, its type changes accordingly. However, the result is still printed out without any fractional part, since the latter is nil. The final output is 256 (Answer D).

7. Following the logic of the conditions, the only way to get to the Hello, World! string is in the else condition of the first if statement. Thus, $a must be False. Likewise, $b must be False. The final conditional relies on both previous conditions ($a and $b) being False, but insists that $c be True (Answer D).

8. The correct answer is C. As of PHP 4.2.0, there is no need to initialize the random number generator using srand() unless a specific sequence of pseudorandom numbers is sought. Besides, even if the random number generator had not been seeded, the script would have still outputted 49 pseudo-random characters—the same ones every time. The $array variable, though a string, can be accessed as an array, in which case the individual characters corresponding to the numeric index used will be returned. Finally, the for loop starts from 1 and continues until $i is less than 50—for a total of 49 times.

9. A series of if…else if code blocks checking for a single condition as above is a perfect place to use a switch statement:

```php
<?php
    switch($a) {
        case 'a':
            somefunction();
            break;
        case 'b':
            anotherfunction();
            break;
        case 'c':
            dosomething();
            break;
        default:
            donothing();
    }
?>
```

Because there is a catch-all else condition, a default case must also be provided for that situation. Answer E is correct.

10. Normally, the foreach statement is the most appropriate construct for iterating through an array. However, because we are being asked to modify each element in the array, this option is not available, since foreach works on a copy of the array and would therefore result in added overhead. Although a while loop or a do…while loop might work, because the array is sequentially indexed a for statement is best suited for the task, making Answer A correct:

```php
<?php
    $myarray = array ("My String", "Another String", "Hi, Mom!");
    for($i = 0; $i < count($myarray); $i++)
    {
        $myarray[$i] .= " ($i)";
    }
?>
```

11. As it is only possible to add a single line of code to the segment provided, the only statement that makes sense is a for loop, making the choice either C or D. In order to select the for loop that actually produces the correct result, we must first of all revisit its structural elements. In PHP, for loops are declared as follows:

```
for (<init statement>; <continue until statement>;
    <iteration statement>)
```

where the <init statement> is executed prior to entering the loop. The for loop then begins executing the code within its code block until the <continue until> statement evaluates to False. Every time an iteration of the loop is completed, the <iteration statement> is executed. Applying this to our code segment, the correct for statement is:

```
for ($idx = 1; $idx < STOP_AT; $idx *= 2)
```

or answer C.

12. Of the five options, only two are valid PHP function declarations (A and D). Of these two declarations, only one will provide a default parameter if none is passed—Answer A.

13. This question is designed to test your knowledge of how PHP scopes variables when dealing with functions. Specifically, you must understand how the global statement works to bring global variables into the local scope, and the scope-less nature of superglobal arrays such as $_GET, $_POST, $_COOKIE, $_REQUEST and others. In this case, the math works out to 5 + 25 - 25 – 10, which is -5, or answer B.

14. Functions can be called dynamically by appending parentheses (as well as any parameter needed) to a variable containing the name of the function to call. Thus, for Group A the appropriate index combination is 0, 4, 9, 9, 9, 9, which evaluates to the string myfunction. The parameters, on the other hand, are evaluated as variables dynamically using the ${} construct. This means the appropriate indexes for group B are 7 and 8, which evaluate to ${'a'} and ${'b'}—meaning the variables $a and $b respectively. Therefore, the correct answer is D.

15. In recent versions of PHP, the only difference between require() (or require_once()) and include() (or include_once()) is in the fact that, while the former will only throw a warning and allow the script to continue its execution if the include file is not found, the latter will throw an error and halt the script. Therefore, Answer E is correct.

16. When a parameter is declared as being passed by reference you cannot specify a default value for it, since the interpreter will expect a variable that can be modified from within the function itself. Therefore, Answer C is correct.

17. The right answer here is the exclusive-or (xor) operator.

18. The identity operator works by first comparing the type of both its operands, and then their values. If either differ, it returns False—therefore, Answer B is correct.

19. The correct answers are A and C. In Answer A, the pow function is used to calculate 22, which corresponds to 4. In Answer C, the left bitwise shift operator is used to shift the value of $a by two bits to the left, which corresponds to a multiplication by 4.

20. The only answer that really fits the bill is A. A script doesn't necessarily terminate when it reaches the end of any file other than the main one—so the "current" file could be externally included and not cause the script to terminate at its end. As far as PHP and Apache crashes, they can hardly be considered "clean" ways to terminate a script.

Posted by charanjeet kaur at 1:59 AM   0 comments

MᴮtF +1 Recommend this on Google

Labels: zce-zend-certified-engineer-php-basic-mock-test-answers, zce-zend-certified-engineer-php-basics-test-paper, zce-zend-certified-engineer-practice-test, zce-zend-certified-engineer-questions-answers

Home   Older Posts

Subscribe to: Posts (Atom)

Ethereal template. Powered by Blogger.