



Universidad Tecnológica Nacional
Ingeniería en Sistemas de Información

Cátedra: Entornos Gráficos

Práctica Número 04: DESARROLLO WEB

Alumno: Gianluca Gatti

Email: gianlucagattiteixeira@gmail.com

Legajo: 43024

Aclaración sobre el código: todos los ejercicios han sido transcritos a PHP para verificar resultados, sin embargo, se han propuesto las soluciones antes de ver el resultado, y en caso de estar equivocado, se ha corregido.

1)

```
<?php
    function doble($i) {
        return $i*2;
    }
    $a = TRUE; //boolean
    $b = "xyz";//string
    $c = 'xyz';//string
    $d = 12;//integer
    echo " ",gettype($a); // construcción del lenguaje echo, sirve para mostrar variables
    en forma de string
    *Este párrafo describe todas las veces que aparece echo
    echo " ",gettype($b);
    echo " ",gettype($c);
    echo " ",gettype($d);
    echo "\n";
    if (is_int($d)) { //operador booleano if //funcion is_int recibe un parametro y
    devuelve si es int o no (TRUE)

        $d += 4; //operador unario aumenta el valor de la variable en una tantas unidades
    como se indique a la derecha del +=
    }
    if (is_string($a)) { //operador booleano if //funcion is_string recibe un parametro y
    devuelve si es string o no (FALSE)
        echo "Cadena: $a";
    }

    $d = $a ? ++$d : $d*3; //operación ternaria, si valida TRUE en la condición antes del ?
    se ejecuta la sentencia a la izquierda de los : si valida FALSE se ejecutará la sentencia
    del lado derecho de los :, es una forma simplificada alternativa al if/else
    echo "d= ",$d,"\n";
    $f = doble($d++); // llamado a la función doble definida al principio del código y esta
    se asigna a la variable f, el operador $d++ incrementará d en una unidad después de ser
    enviada como parámetro a doble()
    echo "d= ",$d,"\n";
    $g = $f += 10; //Doble asignación de las variables, $f toma su valor más 10, y $g toma
    el nuevo valor de $f

    echo $a," ", $b," ", $c," ", $d," ", $f," ", $g;
    //a=TRUE
    //b=xyz
    //c=xyz
    //d=17
    //f=44
    //g=44
?>
```

2)

- a) los códigos **dan el mismo resultado**, sin embargo **el funcionamiento es diferente**, ya que mientras el operador **while** evalúa la condición para acceder a código interno, la sentencia **do...while** primero ejecuta el código, si es verdadera la condición, vuelve a ejecutar hasta que evalúe falso.
- b) Los códigos escritos, en esencia presentan los mismos resultados, y son equivalente, pero algunos como el ejemplo de:

```
<?php
    for ($i = 1; $i <= 10; $i++)
    {
        print $i;
    }
?>
```

Presentan una disposición más cómoda para ser leídos, y serían recomendables en la mayoría de los casos.

- c) Nuevamente en estos casos, el resultante del código escrito es el mismo, sin embargo, los casos de aplicación de las sentencias switch y if ...else if es distinto, y dependiendo de la situación uno de estos resultará mejor que su contraparte.

3)

- a) El código de este punto crea una tabla con 5 filas y 2 columnas.
- b) El código, chequea si está seteado el submit del del post, si no lo esta muestra el formulario, si lo esta entra al if siguiente y chequea si la edad es mayor a 21, en función de esto se muestra un texto que indica la categorización de edad por mayor o menor.

4)

Muestra solo el valor de las variables que se encuentran después de la sentencia include, ya que es en ese punto cuando el intérprete las carga.

5)

El código del ejercicio se detalla perfectamente en los comentarios y se entiende perfectamente de qué manera abre el archivo, lee y trabaja con él para incrementar su valor. Cualquier análisis realizado sobre el fragmento de código sería repetitivo.

6)

Los códigos son equivalentes.

7)

Las salidas son:

- a) bar1
- b) 5942
- c) No explicita ninguna salida.

8)

- a) *Si se entra a las 15:00:00 del 19/5/2020 se mostrará:*
Has entrado en esta pagina a las 15 horas, con 0 minutos y 0 segundos, del 19/5/2020
- b) Se muestra: 5+6=11

9)

```
<?php
```

```

function comprobar_nombre_usuario($nombre_usuario){
    //compruebo que el tamaño del string sea válido.
    if (strlen($nombre_usuario)<3 || strlen($nombre_usuario)>20)
    {
        echo $nombre_usuario . " no es válido<br>";
        return false;
    }

    //compruebo que los caracteres sean los permitidos
    //Recorre el string enviado como nombre de usuario caracter por caracter

    //Si el caracter en la posición indicada no es un sub-string de longitud
    //uno de $permitidos, escribe que no es válido

    //Si recorre el string sin que lo anterior suceda retorna verdadero
    $permitidos = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789-_" ;
    for ($i=0; $i<strlen($nombre_usuario); $i++)
    {
        if (strpos($permitidos, substr($nombre_usuario,$i,1))===false)
        {
            echo $nombre_usuario . " no es válido<br>";
            return false;
        }
    }

    echo $nombre_usuario . " es válido<br>";
    return true;
}
?>

<html>
    <?php //Llamo a la función, le paso el parámetro "nombre" del post ?>
    <?php comprobar_nombre_usuario($_POST['nombre'])?>
</html>

```

Form:

```

<html>
    <form action="ComprobarNombreUsuario.php" method="post">
        <input type="text" name="nombre"/>
        <input type="submit"/>
    </form>
</html>

```