



Universidad Gerardo Barrios

Facultad de Ciencia y Tecnología

Fecha:
19/02/2017

Guía Practica 5: Introducción a la programación con PHP Programación Computacional IV

Objetivos:

- Manejo de variables String, numéricas
- Manejo de variables predefinidas.
- Estructuras de control y Arreglos

Manejo de Cadenas:

Los String son parte importante de cualquier lenguaje de programación, las acciones que se pueden llevar a cabo haciendo un correcto uso del contenido y de las funciones respectivas al tratamiento de cadenas son muy amplias, una muestra de la gran importancia de este tipo de información es el buscador más utilizado a nivel mundial hablamos de Google. PHP nos permite de manera fácil y sencilla definir variables de tipo cadena y operar con ellas a través de las funciones que el lenguaje integra.

Ejemplo:

```
1 <?php
2     $texto = 'Texto de prueba en php';
3     echo "Numero de caracteres incluyendo espacios en Blanco: <b>".strlen($texto)."</b>";
4     trim($texto);
5     echo "Numero de caracteres sin espacios en blanco: <b>".strlen($texto)."</b><br>";
6     echo "Texto en Mayuscula: <b>".strtoupper($texto)."</b><br>";
7     echo "Texto en Minuscula: <b>".strtolower($texto)."</b><br>";
8     echo "Primera letra Mayuscula: <b>".ucfirst($texto)."</b><br>";
9
10    $otra = str_replace(array("Prueba"), array("Reemplazo"), $texto);
11    echo "<h3>".$otra."</h3><br>";
12
13    $palabras = explode(" ", trim($texto));
14    for($i=0; $i<count($palabras); $i++):
15        echo $palabras[$i]."<br>";
16    endfor;
17 ?>
```

Manejo de Enteros:

Otro de los tipos de datos que soporta PHP es los números enteros en el siguiente ejemplo vemos como podemos de manera práctica definir variables enteras o redondear incluso variables o números con decimales.

Ejemplo con enteros:

```

1  <?php
2      echo "<h1>Manejo de Enteros</h1>";
3
4      $foo = 10;
5      $bar = 20;
6      $foobar = $foo + $bar;
7      echo $foobar."<br>";
8      $baz = (int) (5.5*3);
9      echo "16.5 es redondeo: " . $baz;
10 ?>

```

Manejo de números Flotantes.

De la misma forma que declaramos los enteros así también declaramos las variables de tipo float o de punto flotante la asignación de tipos lo hace el intérprete en tiempo de ejecución según el valor que se está asociando a la variable.

Ejemplo de flotante:

```

1  <?php
2      echo "<h1>Manejo de Float</h1>";
3
4      $foo = 10.5;
5      $bar = 20.5;
6      $foobar = $foo + $bar;
7      echo "El valor de Foobar | " . $foobar. "<br>";
8      $baz = (5.5*3);
9      echo "El valor de baz es: " . $baz;
10 ?>

```

Manejo de Booleanos:

Los valores booleanos son especiales en cualquier lenguaje de programación ya que permiten el trabajo con expresiones lógicas, que dan paso a la toma de decisiones y que el flujo de las aplicaciones sea un tanto más dinámica e interactiva.

Estas variables deben ser utilizadas de manera directa en las estructuras de repetición y de control sin la necesidad de comparar pues en programación la comparación de un valor booleano con otro es incoherente.

```

1 <?php
2 $foo = FALSE;
3 $bar = false;
4 $foobar = TRUE;
5
6 if (!$foo) {
7     echo "Valor falso<br/>";
8 }
9
10 if (!$bar) {
11     echo "Valor falso <br/>";
12 }
13
14 if ($foobar) {
15     echo "Valor Verdadero <br/>";
16 }

```

Manejo de variables predefinidas.

```

1 <?php
2 $cad = "El script que est&aacute;s ejecutando: " . $_SERVER['PHP_SELF'] . ". ";
3 $cad .= "En el servidor: " . $_SERVER["SERVER_NAME"] . ".<br>";
4 echo "<h3>" . $cad . "</h3>";
5 ?>

```

1. El siguiente ejemplo muestra el uso de las diversas variables predefinidas de PHP, algunas de ellas conocidas como matrices superglobales y otras como variables de servidor.

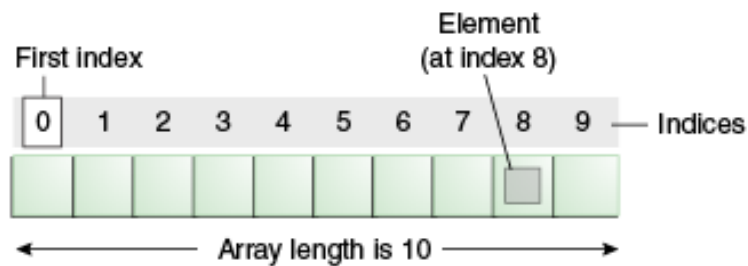
```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4 <title>Variables predefinidas</title>
5 </head>
6 <body>
7 <?php
8     printf("\t<p>En esta pagina se ilustrara como se utilizan algunas variables predefinidas\n");
9
10    printf("<h3>Ejemplos de utilizacion:</h3>\n");
11    printf("<u>\n<li>El nombre del servidor donde se esta ejecutando este script es:
12    <b><i>\$SERVER['SERVER_NAME'] = %s</i></b></li>\n", \$SERVER['SERVER_NAME']);
13    printf("<li>El directorio raiz de documentos en el servidor web es:
14    <b><i>\$SERVER['DOCUMENT_ROOT'] = %s</i></b></li>\n", \$SERVER['DOCUMENT_ROOT']);
15    printf("<li>El software que esta utilizando es: <b><i>\$SERVER['SERVER_SOFTWARE'] =
16    %s</i></b></li>\n", \$SERVER['SERVER_SOFTWARE']);
17    printf("<li>El protocolo que se esta utilizando para ver esta pagina es:
18    <b><i>\$SERVER['SERVER_PROTOCOL'] = %s</i></b></li>\n", \$SERVER['SERVER_PROTOCOL']);
19    printf("<li>El puerto de la maquina usado por el servidor web para la comunicacion
20    HTTP es: <b><i>\$SERVER['SERVER_PORT'] = %s</i></b></li>\n", \$SERVER['SERVER_PORT']);
21    printf("<li>La direccion IP de su computadora es: <b><i>\$SERVER['REMOTE_ADDR'] =
22    %s</i></b></li>\n", \$SERVER['REMOTE_ADDR']);
23    printf("<li>La ruta del script actual en el servidor es:
24    <b><i>\$SERVER['SCRIPT_NAME'] = %s</i></b></li>\n", \$SERVER['SCRIPT_NAME']);
25    printf("<li>La ruta absoluta del script que se esta ejecutando actualmente:
26    <b><i>\$SERVER['SCRIPT_FILENAME'] = %s</i></b></li>\n", \$SERVER['SCRIPT_FILENAME']);
27
28    printf("<li>La cadena de consulta de la peticion de la pagina (si existe) es:
29    <b><i>\$SERVER['QUERY_STRING'] = %s</i></b></li>\n", \$SERVER['QUERY_STRING']);
30    printf("<li>La URI que se empleo para acceder a la pagina es:
31    <b><i>\$SERVER['REQUEST_URI'] = %s</i></b></li>\n", \$SERVER['REQUEST_URI']);
32    printf("<li>El puerto empleado por el equipo del usuario para comunicarse con el
33    servidor es: <b><i>\$SERVER['REMOTE_PORT'] = %s</i></b></li>\n",
34    \$SERVER['REMOTE_PORT']);
35    printf("<li>El idioma definido por el navegador de tu computadora es:
36    <b><i>\$SERVER['HTTP_ACCEPT_LANGUAGE'] = %s</i></b></li>\n",
37    \$SERVER['HTTP_ACCEPT_LANGUAGE']);
38    printf("<li>Estas viendo esta pagina en un navegador:
39    <b><i>\$SERVER['HTTP_USER_AGENT'] = %s</i></b></li>\n", \$SERVER['HTTP_USER_AGENT']);
40    printf("<li>El puerto de comunicacion con el servidor de la computadora que estas
41    utilizando es: <b><i>\$SERVER['REMOTE_PORT'] = %s</i></b></li>\n",
42    \$SERVER['REMOTE_PORT']);
43    printf("<li>La ubicacion y el nombre del archivo que contiene el script que estas
44    ejecutando es: <b><i>\$SERVER['PHP_SELF'] = %s</i></b></li>\n</ul>",
45    \$SERVER['PHP_SELF']);
46    printf("</article>\n");
47    printf("</section>\n");
48    printf("</div>\n");
49 </body>
50 </html>
```

En esta pagina se ilustrara como se utilizan algunas variables predefinidas

Ejemplos de utilizacion:

- El nombre del servidor donde se esta ejecutando este script es: `\$SERVER['SERVER_NAME'] = localhost`
- El directorio raiz de documentos en el servidor web es: `\$SERVER['DOCUMENT_ROOT'] = C:/Users/GiselaE/Documents/UGB2016/Programacion IV/USBWebserver v8.6/root`
- El software que esta utilizando es: `\$SERVER['SERVER_SOFTWARE'] = Apache/2.4.6 (Win32) PHP/5.5.15`
- El protocolo que se esta utilizando para ver esta pagina es: `\$SERVER['SERVER_PROTOCOL'] = HTTP/1.1`
- El puerto de la maquina usado por el servidor web para la comunicacion HTTP es: `\$SERVER['SERVER_PORT'] = 8080`
- La direccion IP de su computadora es: `\$SERVER['REMOTE_ADDR'] = ::1`
- La ruta del script actual en el servidor es: `\$SERVER['SCRIPT_NAME'] = /p5/predefinidas.php`
- La ruta absoluta del script que se esta ejecutando actualmente: `\$SERVER['SCRIPT_FILENAME'] = C:/Users/GiselaE/Documents/UGB2016/Programacion IV/USBWebserver v8.6/root/p5/predefinidas.php`
- La cadena de consulta de la peticion de la pagina (si existe) es: `\$SERVER['QUERY_STRING'] =`
- La URI que se empleo para acceder a la pagina es: `\$SERVER['REQUEST_URI'] = /p5/predefinidas.php`
- El puerto empleado por el equipo del usuario para comunicarse con el servidor es: `\$SERVER['REMOTE_PORT'] = 56268`
- El idioma definido por el navegador de tu computadora es: `\$SERVER['HTTP_ACCEPT_LANGUAGE'] = es-ES;q=0.8`
- Estas viendo esta pagina en un navegador: `\$SERVER['HTTP_USER_AGENT'] = Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/48.0.2564.116 Safari/537.36`
- El puerto de comunicacion con el servidor de la computadora que estas utilizando es: `\$SERVER['REMOTE_PORT'] = 56268`
- La ubicacion y el nombre del archivo que contiene el script que estas ejecutando es: `\$SERVER['PHP_SELF'] = /p5/predefinidas.php`

Acerca de los arrays:



Un array es un conjunto de datos ordenados, que asocian valores que se relacionan entre si en variables cuyo nombre solo lo diferencia el índice de acceso, puede ser utilizado como una matriz de datos real, un lista (Vector), una tabla asociativa que permita almacenar un conjunto de claves y sus valores, un diccionario, una colección de datos, una pila, cola, y muchos otros usos.

Array en PHP

Puede ser creado utilizando el constructor disponible en el lenguaje llamada "array()". Este toma un cierto número de parejas clave => valor como argumentos, cada uno de ellos separado por una coma que delimita la pareja de elementos que se esta ingresando.

```
<?php
2  $favoritecolors = array(
3      "Juan"    => "Verde",
4      "Ronald"  => "Gris",
5      "Mario"   => "Verde",
6      "Edwin"   => "Rojo"
7  );
8
9  echo $favoritecolors["Juan"];
10
11 //Las claves en realidad son opcionales
12 //Veamos el siguiente ejemplo
13 $menu= array(
14     "Inicio",
15     "Catalogo",
16     "Noticias",
17     "Acercade",
18     "Contacto"
19 );
20
21 echo "<br/>".$menu[1];
```

Array asociativo:

El primer arreglo difiere un poco del segundo ya que agregamos una clave para acceder al elemento o al valor de la segunda columna con el que se asocia, en el segundo ejemplo vemos que sin la necesidad de agregar una clave, el Intérprete lo hace de manera secuencial agregando números enteros a cada posición progresivamente, este es el array más simple.

```

24 $multiplescolores = array(
25     "Juan"    => array("rojo", "verde", "gris"),
26     "Ronald"  => array("azul", "amarillo"),
27     "Mario"   => array("azul", "rojo"),
28     "Edwin"   => array("gris", "azul")
29 );
30
31 echo "<h2>Juan</h2>";
32 echo $multiplescolores["Juan"][0]. "<br/>";
33 echo $multiplescolores["Juan"][1]. "<br/>";
34 echo $multiplescolores["Juan"][2]. "<br/>";

```

En el ejemplo anterior vemos como se pueden utilizar Arrays para definir un valor, en este caso múltiples colores que podrían gustarle a una sola persona.

Al ejecutar el script verás la siguiente ejecución:

Juan

```

rojo
verde
gris

```

Para la lectura de arrays asociativos una de las estructuras de control que se ha incluido en casi todos los lenguajes de programación es el **foreach**. Esta estructura de interacción especial para los arrays la estudiaremos posteriormente en el apartado de estructuras de repetición, con ello se hará más fácil el trabajo con arreglos.

Estructuras de Control.

En PHP tenemos las estructuras básicas a las que ya estamos acostumbrados como:

- **if.**
- **switch.**
- **for.**
- **while.**
- **do while.**
- **foreach.**

Todos poseen una sintaxis alternativa aconsejable de utilizar por su estética a las llaves, la cual es aconsejable utilizar porque brinda mayor estética al código y lo hace más entendible cuando se mezclan muchos bloques de este tipo de estructuras.

De manera especial veremos el **operador ternario** muy utilizado en ejemplos que se distribuyen por la Web.

Estructuras de Control:

Permiten que los lenguajes de programación puedan evaluar expresiones Lógicas y a partir de un resultado booleano realizar una determinada tarea previamente programada por el desarrollador de la aplicación.

Esto nos permite modificar el flujo de ejecución de las aplicaciones brindando así mayor dinamismo interno ya que ciertos fragmentos de código deberán o no ejecutarse dependiendo del momento de ejecución, variables de entorno o contexto de la aplicación específico.

IF:

La estructura IF nos permite comparar un valor y a partir del resultado de la expresión si el valor es TRUE se ejecuta el primer bloque de código si el resultado es FALSE se ejecuta el segundo bloque de código ELSE.

```
1 <?php
2 $edad = 10;
3 if($edad <= 12):
4     echo "Eres un Niño";
5 else:
6     echo "No eres un niño";
7 endif;
```

SWITCH:

Permite evaluar una expresión o valor y definir un conjunto de casos para que al cumplirse uno de ellos se ejecute dicho fragmento de código en PHP la sintaxis es la siguiente:

```
9 $seleccion = 1;
10 switch ($seleccion):
11
12     case 1:
13         echo "Elejiste la Opcion 1";
14         break;
15
16     case 2:
17         echo "Elejiste la Opcion 2";
18         break;
19
20     case 3:
21         echo "Elejiste la Opcion 3";
22         break;
23
24     default:
25         echo "Elejiste una opcion diferente";
26         break;
27 endswitch;
```

Estructuras de Repetición o de iterativas:

Permiten repetir un bloque de instrucciones mientras una condición se cumple:

FOR:

```
for($i=0; $i<=10; $i++):  
    echo "Ciclo actual es igual a: ".$i."<br/>";  
endfor;
```

Generalmente es utilizada para leer un array aunque actualmente los lenguajes de programación han integrado la estructura **foreach** que simplifica el proceso, a excepción de algunos casos en los cuales se desean utilizar condicionales dentro del for para evaluar el índice lo mejor es utilizar el foreach.

WHILE:

```
$x = 18;  
while($x<20):  
    echo "While se repite<br/>";  
    $x++;  
endwhile;
```

Repite un conjunto de instrucciones mientras la condición se cumpla, en el ejemplo anterior el **echo** se imprime 2 veces ya que cuando \$x es igual a 20 la expresión lógica devuelve false.

En el While es posible que ninguna vez se ejecute o se entre al bloque de instrucciones si la condición no se cumple:

DO WHILE:

```
$x=20;  
do{  
    echo "While se repite al menos una vez";  
}while ($x<20);
```

A diferencia del **While**, el **do while** asegura que el bucle se ejecutará al menos una vez, en este caso aunque \$x = 20 la instrucción del **echo** siempre se ejecuta pero al final al comprobar que \$x no es menor que 20 ya no se vuelve a ejecutar.

FOREACH:

```
$arr = array(1,2,3,4,"hola","x", "y");  
foreach ($arr as $key):  
    echo $key."<br/>";  
endforeach;
```


El **foreach** nos permite de manera fácil recorrer un array sin la necesidad de declarar variables o condicionales como pasa con la estructura for, una de las mayores ventajas es que funciona tanto para **arrays escalares** o en formato de llave incremental o en **arrays asociativos** en los cuales utilizamos los nombres comunes para designar las claves o índices de los valores.

Ejercicios.

Indicación:

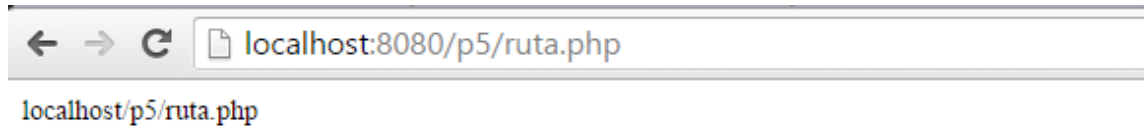
- ✓ Cree una carpeta con el nombre practica5_smis[000000]
- ✓ Para cada ejercicio debe crear un script el cual deberá guardar en la carpeta que creo.
- ✓ Para los ejercicios que requieran de una respuesta teórica elaborar un documento en Word con la explicación.
- ✓ Al finalizar la clase práctica deberá subir la carpeta creada en un archivo zip que contenga los scripts y documento de texto creados.

1. Explique qué hace el siguiente script.

```
1 <html>
2 <head>
3 <title>Tabla condicional</title>
4 </head>
5 <body>
6 <h1>Tabla condicional 1</h1>
7 <?php
8 /* Crearemos una tabla de valores de seno y coseno de 0 a 2
9 en incrementos de 0.01. Los valores negativos que resulten los queremos
10 mostrar en rojo, y los valores positivos en azul */
11 /* En primer lugar vamos a crear una funcion con las condicionales
12 Aqui estamos usando la sentencion de FUNCTION al que le pasamos un valor
13 dependiendo de este valor, ejecutamos una condicion IF..ELSE...
14 y asignamos un color al tipo de letra para generar la tabla
15 */
16 function muestra($valor) {
17 if ($valor < 0.5)
18 $color = "red";
19 else
20 $color = "blue";
21 echo "<td><font color='$color'>$valor</font></td>\n";
22 }
23 ?>
24 <table border="1">
25 <?
26 for ($x=0; $x<=2; $x+=0.01){
27 echo "<tr>";
28 muestra($x);
29 muestra(sin($x));
30 muestra(cos($x));
31 echo "</tr>";
32 }
33 ?>
34 </body>
35 </html>
```

- ### 2. Definir tres variables enteras. Luego definir un string que incorpore dichas variables y las sustituya en tiempo de ejecucion. Recordar que una variable se sustituye cuando el string esta encerrado por comillas dobles: \$precio = 90; Echo "La computadora tiene un precio de \$precio";

3. Realice un script PHP que utilizando variables predefinidas de matrices superglobales muestre la dirección completa del script; es decir, el nombre del servidor junto a la ruta completa del script que se está ejecutando. Imprima la dirección completa de esta forma:



4. Realice un script PHP que muestre mediante la utilización de variables sus datos personales: nombre completo, lugar de nacimiento (departamento y país, si es extranjero), edad y Código de Estudiante. Muestre estos datos en una tabla.
5. Escriba un array que calcule el promedio de las siguientes notas: 8,5,6,9,8,7,8.5
6. A partir de la siguiente tabla:
- | | | |
|--------------|-------------------|-------|
| Finanzas | Roberto Gonzalez | 3,500 |
| Contabilidad | Alberto Canizales | 2,000 |
| Finanzas | Maria Gomez | 2,500 |
| Gerencia | Anastacia Rivera | 3,000 |
| Contabilidad | Andrea Robles | 3,000 |
| R.R H.H | Carmen Rosales | 1,500 |

Se pide:

- Ordenar los datos por nombre
- Calcular el sueldo promedio