PHP

Arreglos y cadenas

Arreglos

Definición

Un arreglo es un grupo de elementos relacionados entre sí por medio de índices.

Es posible crear arreglos unidimensionales y multidimensionales.

ej1.php

```
<HTML>
<TITLE>Ej 3.1 Arreglos</TITLE>
<META http-equiv="Content-type" content="text/html; charset=utf-8" />
<BODY>
<?PHP
        // Inicializacion de un arreglo unidimensional
        // que define los días de la semana
        $dia[0] = "Domingo";
        $dia[1] = "Lunes";
        $dia[2] = "Martes";
        $dia[3] = "Miércoles";
        $dia[4] = "Jueves";
        $dia[5] = "Viernes";
        $dia[6] = "Sábado”;
```

ej1.php

```
        // Impresión de los días de la semana
        echo ($dia[6] . "<Br>") ;
        echo ($dia[5] . "<Br>") ;
        echo ($dia[4] . "<Br>") ;
        echo ($dia[3] . "<Br>") ;
        echo ($dia[2] . "<Br>") ;
        echo ($dia[1] . "<Br>") ;
        echo ($dia[0] . "<Br>") ;
?>
</BODY>
</HTML>
```

Arreglos de diferentes tipos

En PHP, es posible crear arreglos de diferentes tipos de datos, es decir, un elemento puede ser un número entero, otro una cadena, otro un número real, etc.

ej2.php

```
<?PHP
    // Inicialización de un arreglo unidimensional
    // de varios tipos de datos.
    $Empleado[0] = 543;
    $Empleado[1] = "José Pérez";
    $Empleado[2] = "5675.5";
    $Empleado[3] = $Empleado[2]*1.5;
    $Empleado[4] = "Supervisor”;
```

ej2.php

```
// Impresión de los datos

    echo ("Número de trabajador: " . $Empleado[0] . "<Br>");
    echo ("Nombre: " . $Empleado[1] . "<Br>");
    echo ("Sueldo: " . $Empleado[2] . "<Br>");
    echo ("Sueldo con estímulos: " . $Empleado[3] . "<Br>");
    echo ("Puesto: " . $Empleado[4] . "<Br>");
?>
```

Inicialización de arreglos

Existen diversas maneras de inicializar arreglos en PHP. Es posible inicializar un arreglo sin especificar la posición del elemento, por defecto, se van agregando los elementos uno después del otro.

ej3.php

```
<?PHP
        // Inicializacion de un arreglo unidimensional
        $mes[] = "Enero";
        $mes[] = "Febrero";
        $mes[] = "Marzo";
        $mes[] = "Abril";
        $mes[] = "Mayo";
        $mes[] = "Junio";
        $mes[] = "Julio";
        $mes[] = "Agosto";
        $mes[] = "Septiembre";
        $mes[] = "Octubre";
        $mes[] = "Noviembre";
        $mes[] = "Diciembre”;
```

ej3.php

```
// Impresión de los datos

	echo ($mes[0] . "<Br>");
	echo ($mes[1] . "<Br>");
	echo ($mes[2] . "<Br>");
	echo ($mes[3] . "<Br>");
	echo ($mes[4] . "<Br>");
	echo ($mes[5] . "<Br>");
	echo ($mes[6] . "<Br>");
	echo ($mes[7] . "<Br>");
	echo ($mes[8] . "<Br>");
	echo ($mes[9] . "<Br>");
	echo ($mes[10] . "<Br>");
	echo ($mes[11] . "<Br>");
	echo ($mes[12] . "<Br>");
?>
```

Función array

En PHP existe la función array(). Esta función permite crear un arreglo de elementos, indicando los elementos del mismo dentro de los parámetros de la función.

ej4.php

```
<?PHP
        // Inicialización de un arreglo unidimensional
        $forma = array("Juan", "Contador", 5000);

        // Impresión de los datos

        echo ($forma[0] . "<Br>");
        echo ($forma[1] . "<Br>");
        echo ($forma[2] . "<Br>");
?>
```

Función array (cont.)

Dentro de la función array() también es posible indicar la posición del elemento.

ej5.php

```
<?PHP
        // Inicialización de un arreglo unidimensional
        $forma = array( 2=>"Juan",
                        1=>"Contador",
                        0=>5000);

        // Impresión de los datos

        echo ($forma[0] . "<Br>");
        echo ($forma[1] . "<Br>");
        echo ($forma[2] . "<Br>");
?>
```

Función count

La función count() devuelve el número de elementos que tiene el arreglo.

ej6.php

```
<?PHP

    // Inicializacion del arreglo
    $arreglo = array("uno","dos","tres",4);

    // Número de elementos del vector
    $elementos = count($arreglo);
    echo ("Los elementos del arreglo son: " . $elementos);

?>
```

Tipos de índices en arreglos

Es posible crear un arreglo que tenga como índices caracteres o, incluso, cadenas de texto.

Los índices no necesitan iniciar en 0 ni ser consecutivos.

ej7.php

```
<?Php
        $ciudad = array("Par" => "Paris",
                        "Lon" => "Londres",
                        "Ate" => "Atenas",
                        "Ber" => "Berlin",
                        "Lim" => "Lima");

        echo "<H2>"."Ciudades";
        echo "<H3>"."<Hr>";
        echo "Ciudad Lon: " . $ciudad[Lon];
        echo "<Br>";
        echo "Ciudad Ber: " . $ciudad[Ber];
        echo "<Br>";
?>
```

Arreglos multidimensionales

También es posible crear un arreglos multidimensionales.

```
<?php

        $supermercado = array(
                “electrodomesticos” => array(“Televisor”,  “Heladera”),
                “alimentos” => array(“Carne”, “Leche”, “Verduras”)
                );

        echo ($supermercado[“Electrodomesticos”][1]);
?>
```

Ejercicio

Crear un arreglo para almacenar el nombre de una materia. Los índices que se ocupan para obtener el nombre de la materia está dado por la clave de la misma.

Ejercicio

Describir los artículos de una tienda de autoservicios separados por secciones (5 secciones con 5 artículos cada una).

Cadenas

Cadenas

PHP posee varias herramientas para procesar y/o dar formato a las cadenas. El formato de cadenas es muy utilizado al mostrar datos directamente en una página web o para almacenar información en una base de datos.

echo y print()

La instrucción echo y la función print se utilizan para mostrar una o varias cadenas en la pantalla.

Para imprimir una cadena de texto mediante la función print(), el texto se debe ingresar dentro de los paréntesis de la función.

ej8.php

```
<?php
	$i=5;
	echo "Texto impreso con echo: ".$i;
	echo "<br>";
	print(”Texto impreso con print: ".$i);
?>
```

Función printf()

Además de permitir imprimir texto, printf permite imponer un cierto formato a la cadena en cuestión. Su sintaxis es:

printf(“Formato”, $cadena);

Donde formato es una letra que indica el tipo de formato que se desea imponer a la cadena.

Función printf()

Letra de Formato	Especificación de lo que realiza
%b	Expresa a la variable como número binario.
%c	Expresa a la variable como carácter ASCII.
%d	Expresa a la variable como número decimal.
%f	Expresa a la variable como número de punto flotante.
%o	Expresa a la variable como número octal.
%s	Expresa a la variable como un string (cadena).
%x	Expresa a la variable como número hexadecimal en minúsculas.
%X	Expresa a la variable como número hexadecimal en Mayúsculas.

ej9.php

```
<?Php
	$num = 2947.3262;
	printf("Número sin formato: ".$num."<Br>");
	printf("Número con formato: %f <br>", $num);
	printf("Dos decimales: %7.2f<br>",$num);
	printf("Octal: %o<br>",$num);
	printf("Hexadecimal %x<br>",$num);
	printf("Hexadecimal %X<br>",$num);
	printf("Binario %b<br>",$num);
?>
```

Función sprintf()

Esta función permite dar formato a una cadena y almacenar el resultado del formato en otra variable. Su sintaxis es:

```
$var = sprintf ("Formato", $cadena);
```

Función ucwords()

Esta función transforma el primer carácter de cada palabra en letra mayúsculas.

ej10.php

```
<?php
	// sprintf
	$cadena = "hola!!";
	$variable = sprintf("%s<BR>", $cadena);
	echo $variable;
	echo "<BR>";

	// ucword
	 $nombre = "pedro correa";
	echo "Cadena original: " . $nombre;
	echo "<Br><Br>";
	echo "Cadena con formato ucword: " . ucwords($nombre);
?>
```

Función strtoupper()

Esta función transforma todos los caracteres de una cadena a letras mayúsculas.

Función strtolower()

Esta función transforma todos los caracteres de una cadena a letras minúsculas.

ej11.php

```
<?php
	// strtoupper
	$frase = "funcion strtoupper.";
	echo "Frase inicial: " . $frase;
	echo "<BR>";
	echo "Frase en Mayúsculas: " . strtoupper($frase);
	echo "<BR><BR>";

	// strtolower
	$frase = "FUNCION STRTOLOWER.";
	echo "Frase inicial: " . $frase;
	echo "<BR>";
	echo "Frase en Mayúsculas: " . strtoupper($frase);
	echo "<BR>";
?>
```

Función trim()

Elimina los espacios en blanco al inicio y al final de la cadena. Su sintaxis es:

```
trim($cadena)
```

Función ltrim()

Elimina los espacios en blanco al principio de la cadena. Su sintaxis es:

ltrim($cadena)

Función chop()

Elimina los espacios en blanco al final de la cadena. Su sintaxis es:

```
chop($cadena)
```

ej12.php

```
<?PHP
        $cadena = " buenas tardes ";
        echo "<H1>trim</h1>”;
        echo “Cadena” . $cadena . “con espacios.”;
        echo “<BR><BR>”;
        echo “Cadena” . Trim($cadena) . “sin espacios.”;

        echo “<H1>ltrim</h1>”;
        echo “Cadena” . $cadena . “con espacios.”;
        echo “<BR><BR>”;
        echo “Cadena” . Ltrim($cadena) . “sin espacio al inicio.”;

        echo “<H1>chop</h1>”;
        echo “Cadena” . $cadena . “con espacios.”;
        echo “<BR><BR>”;
        echo “Cadena” . Chop($cadena) . “sin espacio al final.”;
?>
```

Función strlen()

Función que permite obtener la longitud de la cadena de caracteres enviada como parámetro.

strlen($cadena)

Función split()

Función que permite separar la cadena de caracteres enviada como parámetro, según el separar indicado.

split(separador, $cadena)

Función substr()

Función que permite dividir la cadena de caracteres enviada como parámetro, desde la posición inicial indicada hasta la longitud indicada.

substr($cadena, inicio, longitud)

ej13.php

```
<?PHP
	echo "<H1> strlen </H1>";
	$cadena = "La insoportable levedad del ser";
	echo "La longitud de la cadena ".$cadena." es: " . strlen($cadena);
	echo "<BR>";

	echo "<H1>split</H1>";
	$frase = "Bonjour Monde";
	$fraseSeparada = split(" ", $frase);
	echo $fraseSeparada[0] ."<br>";
	print($fraseSeparada[1]."<br>");

	echo "<H1>substr</H1>";
	$subFrase = substr($cadena,10,5);
	echo "Cadena original: ".$cadena."<br>";
	echo "Subcadena: ".$subFrase."<br>";
?>
```

Función strpos()

Función que permite buscar la cadena2 dentro de la cadena1 indicando la posición en la que se encuentra la coincidencia.

strpos($cadena1, $cadena2)

Función ereg()

Función que permite buscar la cadena1 dentro de la cadena2, devuelve verdadero si encuentra la coincidencia (1).

ereg($cadena1, $cadena2)

Función ereg()

Función que permite buscar la cadena1 dentro de la cadena2, devuelve verdadero si encuentra la coincidencia (1).

ereg($cadena1, $cadena2)

ej14.php

```
<?PHP
	echo "<H1> strpos </H1>";
	$cadena = "La insoportable levedad del ser";
	echo "Cadena: ".$cadena."<br>";
	$buscar="leve";
	echo "Busca " .$buscar. " y la encuentra en posición: ".strpos($cadena,$buscar);
	echo "<BR>";

	echo "<H1>ereg</H1>";
	echo "Cadena: ".$cadena."<br>";
	echo $buscar. " se encuentra en la cadena?: ". ereg($buscar,$cadena);
	$buscar = Leve;
	echo $buscar. " se encuentra en la cadena?: ". ereg($buscar,$cadena);

	echo "<H1>eregi</H1>";
	echo "Cadena: ".$cadena."<br>";
	echo $buscar. " se encuentra en la cadena?: ". eregi($buscar,$cadena);
?>
```

Función ereg_replace()

Función que permite buscar una cadena en un texto y reemplazar dicha cadena con otra.

ereg_replace($cadBus, $cadRem,texto)

Función eregi_replace()

Función que permite buscar una cadena en un texto y reemplazar dicha cadena con otra, sin discriminar entre mayúsculas y minúsculas.

eregi_replace($cadBus, $cadRem,texto)

ej15.php

```
<?PHP
	// ereg_replace
	echo "<H1> ereg_replace </H1>";
	$cadena = "Anita ensucia la tina";
	echo "Cadena: ".$cadena."<br>";
	$buscar="ensucia";
	$reemplazar = "lava";
	echo "Reemplaza " .$buscar. " por ".$reemplazar.": ".ereg_replace($buscar,$reemplazar,$cadena);
	echo "<BR>";

	// eregi_replace
	echo "<H1>eregi_replace</H1>";
	echo "Cadena: ".$cadena."<br>";
	$buscar="Ensucia";
	echo "Reemplaza " .$buscar. " por ".$reemplazar.": ".eregi_replace($buscar,$reemplazar,$cadena);
?>
```

Función strcmp()

Función que permite comparar dos cadenas. Regresa 0 en caso de que las cadenas sean iguales, mayor a cero en caso de que la cadena1 sea más grande y menor a cero en caso de que la cadena1 sea la más grande.

strcmp($cadena1, $cadena2)

Función strpos()

Función que permite obtener la posición donde se encuentra la cadena buscada dentro del texto deseado.

strpos($texto, $cadena)

ej16.php

```
<?PHP
	// strcmp
	echo "<H1> strcmp </H1>";
	$c1 = "Juan";
	$c2 = "juan";
	echo $c1." y " .$c2." son: ".strcmp($c1,$c2);
	echo "<BR>";
	$c2 = "Juan";
	echo $c1." y " .$c2." son: ".strcmp($c1,$c2);
	echo "<BR>";

	// strpos
	echo "<H1>eregi_replace</H1>";
	$cadena = "Anita lava la tina";
	$buscar = "lava";
	echo "La cadena buscada se encuentra en la posición: " . strpos($cadena,$buscar);
?>
```

Función sort()

Función que permite ordenar un arreglo por abecedario.

sort($nombres)

Función sort()

Función que permite ordenar un arreglo por abecedario.

sort($nombres)

Función shuffle()

Función que mezcla los elementos del arreglo de manera aleatoria.

shuffle($nombres)

Ejercicio

Implementar las funciones sort y shuffle