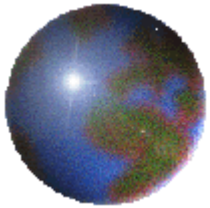
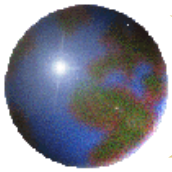


Desarrollo de Portales WEB Lenguaje PHP

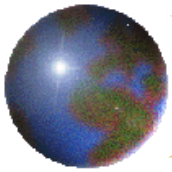
Ing. Alberto Moreno

Funciones de Usuario



Funciones

- La función podría ser definida como un conjunto de instrucciones que permiten procesar las variables para obtener un resultado. Puede que esta definición resulte un poco vaga si no nos servimos de un ejemplo para ilustrarla.



Funciones

- Sintaxis:

```
Function nombreFuncion([argumentos])
{
  sentencias;
}

nombreFuncion();
```

Funciones

```php
<?php


function hacer_encabezado($titulo)
{
$encabezado="<html>\n<head>\n\t<title>$titulo</title>\n</head>\n";
echo $encabezado;
}

hacer_encabezado("ESTAMOS EN CLASE");


?>
```

Ejercicio de Funciones

- Crear funciones que permitan realizar lo siguiente:
  - hacer_encabezado("ESTAMOS EN CLASE");=> Colocar de color rojo el fondo de la pagina web. Titulo head: estamos en clase.
  - Tabladatos(fila;celdas;dato); => función que genere una tabla de N filas y N celdas con el texto: $dato.

Ejercicio de Funciones

- Solución:

```php
<?php
function hacer_encabezado($titulo)
{
$encabezado="<html><head><title>$titulo</title></head><body bgcolor=red>";
echo $encabezado;
}
function tabladatos($fila,$celdas,$dato)
{
echo "<table border=2>";
for ($i=1;$i<=$fila;$i++){
    echo "<tr>";
    for ($j=1;$j<=$celdas;$j++){
        echo "<th> $dato </th>";
    }
}

echo "</tr>";
echo "</table>";
}
hacer_encabezado("ESTAMOS EN CLASE");
tabladatos(3,5,"idat y php");

?>
```

Ejercicio de Funciones

- Se puede generar un archivo php como si fuera una librería de funciones y luego referenciarla con la palabra reservada "include(librería.php);
- Evaluaremos el ejemplo: ***libreria.php***

```
<?php
function fondotitulo($titulo)
{
$encabezado="<html>\n<head>\n\t<title>$titulo</title>
</head><body bgcolor=orange>\n";
echo $encabezado;
echo "<h1>$titulo</h1>";
}

?>
```

Ejercicio de Funciones

- Luego en el archivo “prueba.php”

```
<html>
<head>

<title>prueba</title>
</head>

<body>
<?php
include("libreria.php");
fondotitulo("ESTAMOS APRENDIENDO");

?>
</body>
</html>
```

Funciones

```php
<?php

function muestranombre ($titulo = "Sr.")
{
    print "Estimado $titulo:<br>";
}
muestranombre ();
muestranombre ("Prof.");

?>
```

- Salida:

```
Estimado Sr.:
Estimado Prof.:
```

Funciones

- Los argumentos con valores por defecto deben ser siempre los últimos:

```
function muestranombre ($nombre, $titulo= "Sr.")
{
    print "Estimado $titulo $nombre:\n";
}
muestranombre (“Fernández”);
muestranombre (“Fernández”, "Prof.");
```

- Salida:

```
Estimado Sr. Fernández:
Estimado Prof. Fernández:
```

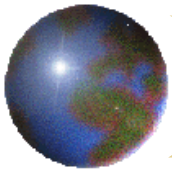
Funciones

- Las **funciones** son un ejemplo algo más complejo de expresiones. Son expresiones que valen el valor que retornan (una función retorna un valor mediante el comando **return**)

```php
<?php
function prueba () {
return 5;
}// La función prueba devuelve el valor 5

$a=prueba(); //El valor de $a es 5
?>
```

- Como la función prueba() devuelve el valor 5, el valor de la expresión 'prueba()' es 5. Por lo tanto, teclear $a=prueba() es esencialmente lo mismo que escribir $a=5. Normalmente las funciones no devuelven un valor fijo, sino que suele ser calculado.



Tablas o Arreglos

- Sintaxis:

```
array ([clave =>] valor, ...)
```

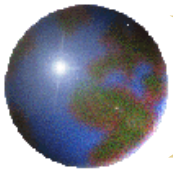
- La clave es una cadena o un entero no negativo. El valor puede ser de cualquier tipo válido en PHP, incluyendo otro array
- Ejemplos:

```
$color = array ('rojo'=>101, 'verde'=>51, 'azul'=>255);
$medidas = array (10, 25, 15);
```

- Acceso:

```
$color['rojo'] // No olvidar las comillas
$medidas[0]
```

- El primer elemento es el 0



Tablas

- La estructura de control **foreach** permite iterar sobre arrays
- Sintaxis:

```
foreach (expresión_array as $valor)
   sentencia
foreach (expresión_array as $clave => $valor)
   sentencia
```

- Ejemplos:

```
foreach ($color as $valor)
   print "Valor: $valor<BR>\n";
foreach ($color as $clave => $valor)
   print "Clave: $clave; Valor: $valor<BR>\n";
```

- Salida:

```
Valor: 101
Valor: 51
Valor: 255
Clave: rojo; Valor: 101
Clave: verde; Valor: 51
Clave: azul; Valor: 255
```

Arreglos

- Un array es un tipo de datos con una importante peculiaridad: **no** contiene un único valor, sino un **conjunto de valores** referenciados con un **índice**. Aunque al principio puedan parecer poco amistosos, su uso es utilísimo e imprescindible en programación.

```php
<?php
$semana = array(
0=>"lunes",
1=>"martes",
2=>"miércoles");
?>
```

Arreglos

- Ahora podemos **recuperar cualquier valor** específico **refiriéndonos a su índice**.
  Por ejemplo, para presentar el nombre del tercer día de la semana podríamos escribir simplemente

```php
<?php
$semana = array(
0=>"lunes",
1=>"martes",
2=>"miércoles");
echo $semana[2]; //Acceso al tercer elemento del vector $semana
?>
```

Arreglos

```php
<?PHP

$a = array(
    "uno" => 1,
    "dos" => 2,
    "tres" => 3,
    "diecisiete" => 17
);

foreach($a as $k => $v) {
    print "\$a[$k] => $v.<br>";
}
?>
```

$a[uno] => 1.
$a[dos] => 2.
$a[tres] => 3.
$a[diecisiete] => 17.