

Definición de Funciones

Antonio Espín Herranz

Contenidos

- Funciones definidas por el programador
- Ficheros de librerías (con funciones)

Funciones

- ```
<?php
function foo ($arg_1, $arg_2, ..., $arg_n)
{
 echo "Función de ejemplo.\n";
 return $retval;
}
foo($arg_1, ... $arg_n);
?>
```
- La llamada a la función será mediante su nombre seguido de ().
- Si tuviera parámetros se encerrarán entre los paréntesis.
- Podemos llamar a una función antes de declararla.
- Las funciones no se pueden redefinir.

# Funciones condicionales

```
<?php
```

```
 $makefoo = true;
```

```
 bar();
```

```
 if ($makefoo) {
 function foo ()
```

```
 {
 echo "I don't exist until program execution reaches me.\n";
 }
 }
```

```
 if ($makefoo) foo();
```

```
 function bar()
```

```
 {
 echo "I exist immediately upon program start.\n";
 }
```

```
?>
```

# La instrucción Return

- Con la instrucción return la ejecución de la función termina y se devuelve un valor al exterior.

```
• <?php
function mayor ($x, $y){
 $msg = "";
 if ($x > $y) {
 $msg = $x." es mayor que".$y;
 } else {
 $msg = $y." es mayor que".$x;
 }
 return $msg;
}
?>
```

# Parámetros de una función

- Dos formas de pasar los parámetros:
  - Por valor: Cualquier cambio de la variable dentro de la función no altera el contenido de la misma.
  - Por referencia: Si realizamos algún cambio de la variable dentro de la función, el cambio se verá reflejado dentro de la función. La variable se tiene que preceder de un &.

# Ejemplo

```
function porvalor ($parametro1){
 $parametro1="hola";
 echo "
" . $parametro1; //imprime "hola"
}

$mivariable = "esto no cambia";
porvalor ($mivariable);
echo "
" . $mivariable; //imprime "esto no cambia"
```

```
function porreferencia(&$cadena) {
 $cadena = 'Si cambia';
}

$str = 'Esto es una cadena';
porreferencia ($str);
echo $str; // Imprime 'Si cambia'
```

# Parámetros opcionales

- Si una función tiene parámetros y parámetros opcionales, estos tendrán que ir al final.

```
function miFuncion($color, $title, $header='Welcome'){
 // Cuerpo de la función ...
}
```



# Archivos de librerías

- Las funciones las podemos almacenar en otro fichero aparte.
- Nos facilita la reutilización de código.
- Con una instrucción `require` o `include` en nuestro fichero haciendo referencia al fichero de funciones ya las podemos utilizar.

# Ejemplo

```
// Fichero: libreria.php
<? //función de encabezado y colocación del titulo
 function hacer_encabezado($titulo)
 {
 $encabezado="<html>\n<head>\n\t<title>$titulo</title>\n</head>\n";
 echo $encabezado;
 }
?>
```

```
// Fichero pagina.php
<?
 include("libreria.php");
 $titulo="Mi Web";
 hacer_encabezado($titulo);
?>
<body>El cuerpo de la página</body></html>
```

# Archivos de Librerías

- **include\_once** y **require\_once**: Nos aseguran que el fichero sólo se incluirá una vez en el proyecto.
- Una clase, puede que sea necesaria incluirla en mas de un fichero PHP, con estas funciones sólo será incluida una vez.