

# Más sobre PHP



**Área de Ingeniería Telemática**



## Destino (info2.php)

```
<?php
session_start();
include("inicio.php");
?>
<pre>
<?php
print "Get: "; print_r($_GET);
print "Post: "; print_r($_POST);
print "Request: "; print_r($_REQUEST);
print "Files: "; print_r($_FILES);
print "Session: "; print_r($_SESSION);
print "Server: "; print_r($_SERVER);
print "Env: "; print_r($_ENV);
?>
</pre>
<?php
include("fin.php");
?>
```

Los nombres que son clave en el array asociativo `$_SESSION`, deben tener un nombre válido de variable. Es decir, no pueden empezar por número

# **`$_GET`, `$_POST`, `$_REQUEST`**

Get: Array

```
([parametro] => valor )
```

Post: Array

```
(  
    [nombre] => Plano del departamento  
    [MAX_FILE_SIZE] => 300000  
)
```

Request: Array

```
(  
    [parametro] => valor  
    [nombre] => Plano del departamento  
    [MAX_FILE_SIZE] => 300000  
    [PHPSESSID] => 26465da5723cccfacc1869072db5a8f7  
)
```

## **\$FILES, \$\_SESSION**

Files: Array

```
(  
[imagen] => Array  
    (  
        [name] => plano_departamento.gif  
        [type] => image/gif  
        [tmp_name] => /tmp/phpx1UEna  
        [error] => 0  
        [size] => 192884  
    )  
)
```

Session: Array

```
(  
[MiDato] => Esto quiero guardar  
)
```

# **\$\_SERVER**

```
Server: Array
(
    [HTTP_HOST] => localhost
    [HTTP_USER_AGENT] => Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.7.12) Gecko/20050915 Firefox/1.0.7
    [HTTP_ACCEPT] => text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
    [HTTP_ACCEPT_LANGUAGE] => en-us,en;q=0.5
    [HTTP_ACCEPT_ENCODING] => gzip,deflate
    [HTTP_ACCEPT_CHARSET] => ISO-8859-1,utf-8;q=0.7,*;q=0.7
    [HTTP_KEEP_ALIVE] => 300
    [HTTP_CONNECTION] => keep-alive
    [HTTP_REFERER] => http://localhost/info.php
    [HTTP_COOKIE] => PHPSESSID=26465da5723cccfac1869072db5a8f7
    [CONTENT_TYPE] => multipart/form-data; boundary=-----8767276471703
    [CONTENT_LENGTH] => 193310
    [PATH] => /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/bin/X11
    [SERVER_SIGNATURE] => Apache/2.0.54 (Unix) DAV/2 PHP/4.4.0 Server at localhost Port 80
    [SERVER_SOFTWARE] => Apache/2.0.54 (Unix) DAV/2 PHP/4.4.0
    [SERVER_NAME] => localhost
    [SERVER_ADDR] => 192.168.2.40
    [SERVER_PORT] => 80
    [REMOTE_ADDR] => 192.168.2.1
    [DOCUMENT_ROOT] => /www
    [SERVER_ADMIN] => fran@navarparty.org
    [SCRIPT_FILENAME] => /www/info2.php
    [REMOTE_PORT] => 4079
    [GATEWAY_INTERFACE] => CGI/1.1
    [SERVER_PROTOCOL] => HTTP/1.1
    [REQUEST_METHOD] => POST
    [QUERY_STRING] => parametro=valor
    [REQUEST_URI] => /info2.php?parametro=valor
    [SCRIPT_NAME] => /info2.php
    [PHP_SELF] => /info2.php
    [PATH_TRANSLATED] => /www/info2.php
    [argv] => Array
        (
            [0] => parametro=valor
        )
    [argc] => 1
)
```

## **\$\_ENV**

Env: Array

```
(  
  [HZ] => 100  
  [TERM] => linux  
  [SHELL] => /bin/bash  
  [HUSHLOGIN] => FALSE  
  [USER] => root  
  [LD_LIBRARY_PATH] => /programas/apache2/lib:  
  [PATH] =>  
    /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/bin/X11  
  [MAIL] => /var/mail/root  
  [PWD] => /programas/apache2  
  [LANG] => es_ES@euro  
  [HOME] => /root  
  [SHLVL] => 2  
  [LOGNAME] => root  
  [_] => /programas/apache2/bin/httpd  
)
```

# Cómo NO hay que completar valores

```
<?php session_start(); include("inicio.php"); ?>
<html>
<html>
<head>
  <title>Ejemplo con selects</title>
</head>
<body>
<form method="post" action="info.php">
  <select name="day">
    <option value="1">1</option>
    <option value="2">2</option>
    <option value="3">3</option>
    ... etc ...
    <option value="31">31</option>
  </select>
  <select name="month">
    <option value="1">Enero</option>
    <option value="2">Febrero</option>
    <option value="3">Marzo</option>
    ... etc ...
    <option value="12">Diciembre</option>
  </select>
  <select name="year">
    <option value="2005"> 2005 </option>
    <option value="2006"> 2006 </option>
    <option value="2007"> 2007 </option>
    ... etc ...
    <option value="2014"> 2014 </option>
  </select>
</form>
</body>
</html>
<?php include("fin.php"); ?>
```

# Cómo SÍ hay que completar valores

```
<?php session_start(); include("inicio.php"); ?>
<html>
<html>
<head>
  <title>Ejemplo con selects</title>
</head>
<body>
<form method="post" action="info.php">
  <select name="day">
<?php
for ($i=1;$i<=31;$i++)
  print "<option value=\"\$i\">$i</option>\n";
?>
  </select>
  <select name="month">
<?php
$meses = array("Enero", "Febrero", "Marzo", ..., "Diciembre");
foreach($meses AS $clave => $actual){
?>
  <option value="<?php echo ($clave+1);?>"><?php echo $actual;?></option>
<?php
?>
  </select>
```

```
<select name="year">
<?php
$init_year = strftime("%Y");
$end_year = $init_year+10;
for($i=$init_year;$i<$end_year;$i++)
  print "<option value=\"\$i\">$i</option>\n";
?>
  </select>
</form>
</body>
</html>
<?php include("fin.php"); ?>
```

## Mandar mails

- PHP puede utilizar el Mail Transport Agent (MTA) del sistema para enviar e-mails
- Función **mail()**

```
bool mail ( string to, string subject, string message [,  
string additional_headers [, string  
additional_parameters]] )
```

- Ejemplo:

```
mail("alumno@unavarra.es", "Datos del programa", "Los  
datos son:\n\n1\n2\n3\n\n y ya está",  
"From:felix.espina@unavarra.es\r\n"."CC:otro@unavarra.es  
");
```

# Variables

- Las variables son locales a su "scope"
- Si queremos que "traspasen" estos límites, usar global
- Este comportamiento no se cumple con las "superglobals" puesto que existen en todo el script: `$_GET`, `$_POST`, `$_SERVER`, `$_SESSION`, etc
- Ejemplo:

```
$a = "DatoA";  
$b = "DatoB";  
function hacer_algo($a)  
{  
    global $b;  
    global $_GET['c'];  
  
    return $a.$b.$_GET['c'];  
}
```

## Includes

- Cuando se realiza un `include()` de un fichero:
- El código en él se ejecuta. Por ejemplo: primer ejemplo de hoy
- Las funciones en él, no se ejecutan, sólo “se definen”. Permiten su uso, pero no son ejecutadas hasta que se llaman
- Con las conexiones y demás elementos de patrón “Singleton”, con `require_once()` o `include_once`

# Acceso al sistema y ficheros

- Podemos ejecutar comandos del sistema

```
system("ls -al /etc/init.d")
```

- Trabajar con ficheros y directorios

```
$dir = "/etc/php5/";

// Abrir directorio conocido y leer sus contenidos
if (is_dir($dir)) {
    if ($dh = opendir($dir)) {
        while (($file = readdir($dh)) !== false) {
            echo "filename:$file:filetype:".filetype($dir.$file)."\n";
        }
        closedir($dh);
    }
}
```

## Clases y objetos

- Siempre que sea posible, usar clases y objetos
- Pese a que PHP4 realiza una implementación “particular”, funciona y da buenos resultados
- La orientación a objetos siempre permite escalabilidad y control sobre el programa
- Si no se usan objetos, por lo menos dividir el programa en partes y trabajar con funciones e includes

# Clases y objetos PHP 4

```
<?php
class Carrito {
    var $items; // Items en nuestro carrito de compras

    function agregar_item($artnr, $num) {
        $this->items[$artnr] += $num;
    }

    function retirar_item($artnr, $num) {
        if ($this->items[$artnr] > $num) {
            $this->items[$artnr] -= $num;
            return true;
        } elseif ($this->items[$artnr] == $num) {
            unset($this->items[$artnr]);
            return true;
        } else {
            return false;
        }
    }
}

}
?>
```

```
<?php
class Carrito_Con_Nombre extends Carrito {
    var $duenyo;

    function definir_duenyo ($nombre) {
        $this->duenyo = $nombre;
    }
}
?>
```

## Clases y objetos PHP 4

```
<?php
    $carrito = new Carrito;
    $carrito->agregar_item("10", 1);

    $otro_carrito = new Carrito;
    $otro_carrito->agregar_item("0815", 3);
?>
```

```
<?php
    $carrito_n = new Carrito_Con_Nombre; // Crear un carrito con nombre
    $carrito_n->definir_duenyo("kris"); // Nombrar el carrito
    print $carrito_n->duenyo; // imprimir el nombre del dueño
    $carrito_n->agregar_item("10", 1); // (funcionalidad heredada de
                                        carrito)
?>
```

## Clases y objetos PHP 4

- **NO** es posible separar la definición de una clase en varios archivos.
- **NO** es posible separar la definición de una clase en bloques PHP diferentes
- Las clases pueden ser extensiones de otras clases
  - La clase extendida o derivada tiene todas las variables y funciones de la clase base
  - **No** es posible substraer de una clase, es decir, remover la definición de cualquier función o variable existente
  - Una clase extendida siempre depende de una clase base única, lo que quiere decir que no se soporta herencia múltiple
  - Las clases son extendidas usando la palabra clave 'extends'
- ¡Las clases deben ser definidas antes de ser usadas! => el orden en que se definen las clases es importante.

## Clases y objetos PHP 4

- En PHP 4, sólo se permiten inicializadores constantes para variables *var*
- Para inicializar variables con valores no-constantes => función de inicialización (constructora) que sea llamada automáticamente cuando un objeto es construido a partir de la clase

```
<?php
class Carrito {
    /* Ninguna de estas expresiones funciona en PHP 4. */
    var $fecha_hoy = date("Y-m-d");
    var $nombre = $primer_nombre;
    var $duenyo = 'Fred ' . 'Jones';
    /* Aunque, las matrices que contienen valores constantes funcionan */
    var $items = array("VCR", "TV");
}
/* Asi es como debe declararse. */
class Carrito {
    var $fecha_hoy;
    var $nombre;
    var $duenyo;
    var $items = array("VCR", "TV");

    function Carrito() {
        $this->fecha_hoy = date("Y-m-d");
        $this->nombre = $GLOBALS['primer_nombre'];
        /* etc. . . */
    }
}
?>
```