Páginas WEB Accesibles

Breve introducción a PHP

Luis Fernando Llana Díaz

Departamento de Sistemas Informáticos y Computación

13 de noviembre de 2008

Programación en el cliente

- JavaScript
- Applets de Java
- Flash de Macromedia.

Programación en el cliente

- JavaScript
- Applets de Java
- Flash de Macromedia.

Características

- Se ejecutan en el cliente: el navegador directamente o con plug-in.
- Son elementos externos al HTML. Marcas: `script`, `object`.
- JavaScript es interpretado en el ordenador, suele incluir errores (páginas sólo para Internet Explorer).
- Flash es tecnologías propietarias.... sólo los propietarios pueden desarrollar plug-ins.
- La accesibilidad en entredicho.

Programación en el servidor

- PHP,
- JSP, Java Servlets
- ASP,
- Muchos más: perl, python....

Programación en el servidor

- PHP,
- JSP, Java Servlets
- ASP,
- Muchos más: perl, python....

Características

- El cliente siempre recibe HTML, no necesita nada más que un navegador.
- Los cálculos se hacen en el servidor, en el lenguaje de programación elegido.
- Accesible, si el programador lo hace bien.

PHP: hola mundo I

- /usr/share/doc/php-doc/html/index.html
- http://www.php.net/manual/es/

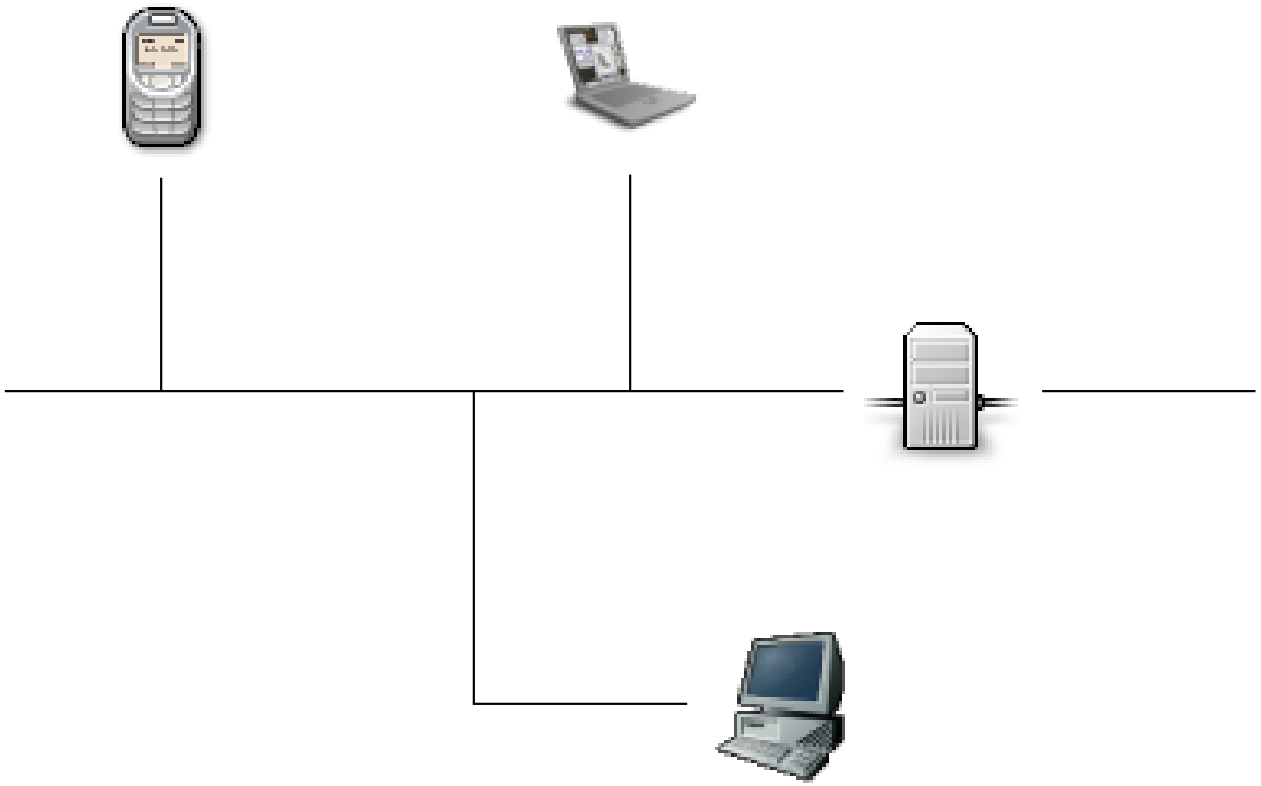
```
<html>
 <head>
  <title>Ejemplo de PHP</title>
 </head>
 <body>
  <?php echo " <p>Hola Mundo</p>"; ?>
 </body>
</html>
```

PHP: hola mundo II

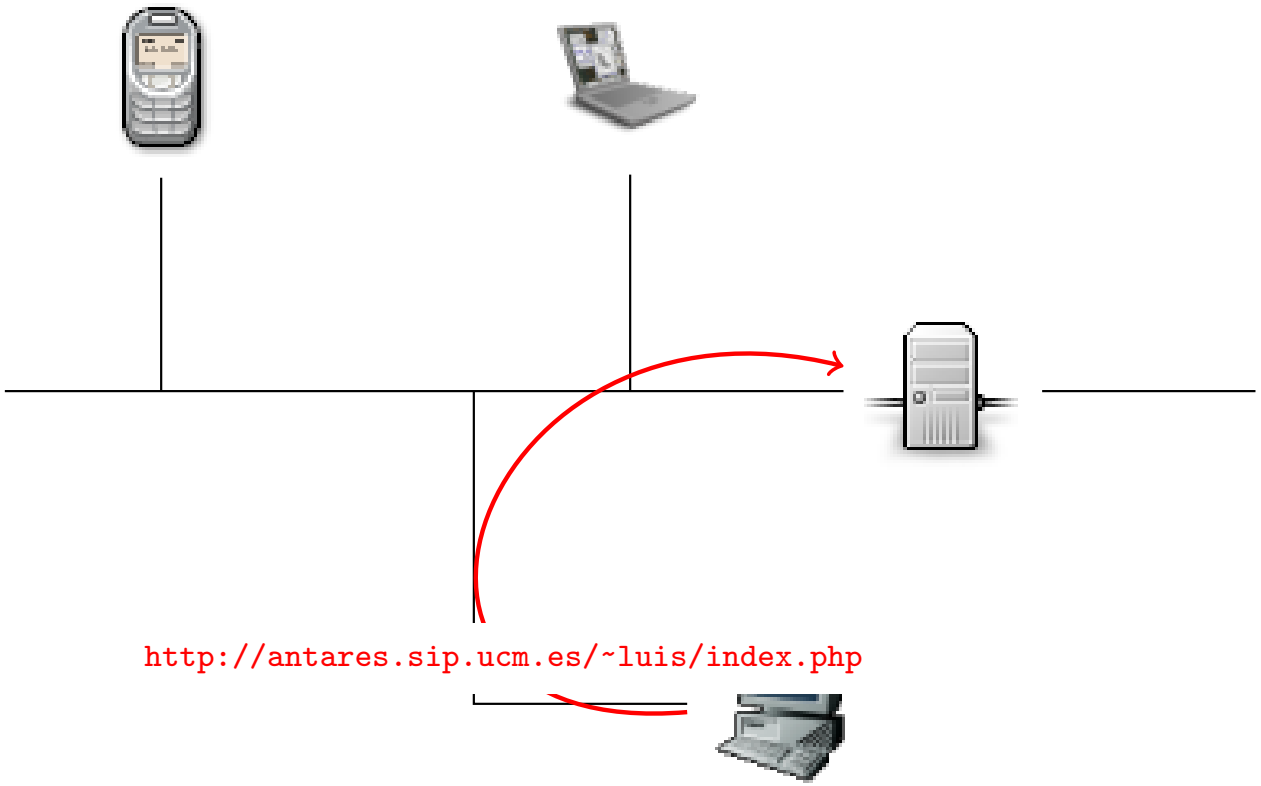
El cliente recibe

```
<html>
 <head>
  <title>Ejemplo de PHP</title>
 </head>
 <body>
   <p>Hola Mundo</p>
 </body>
</html>
```

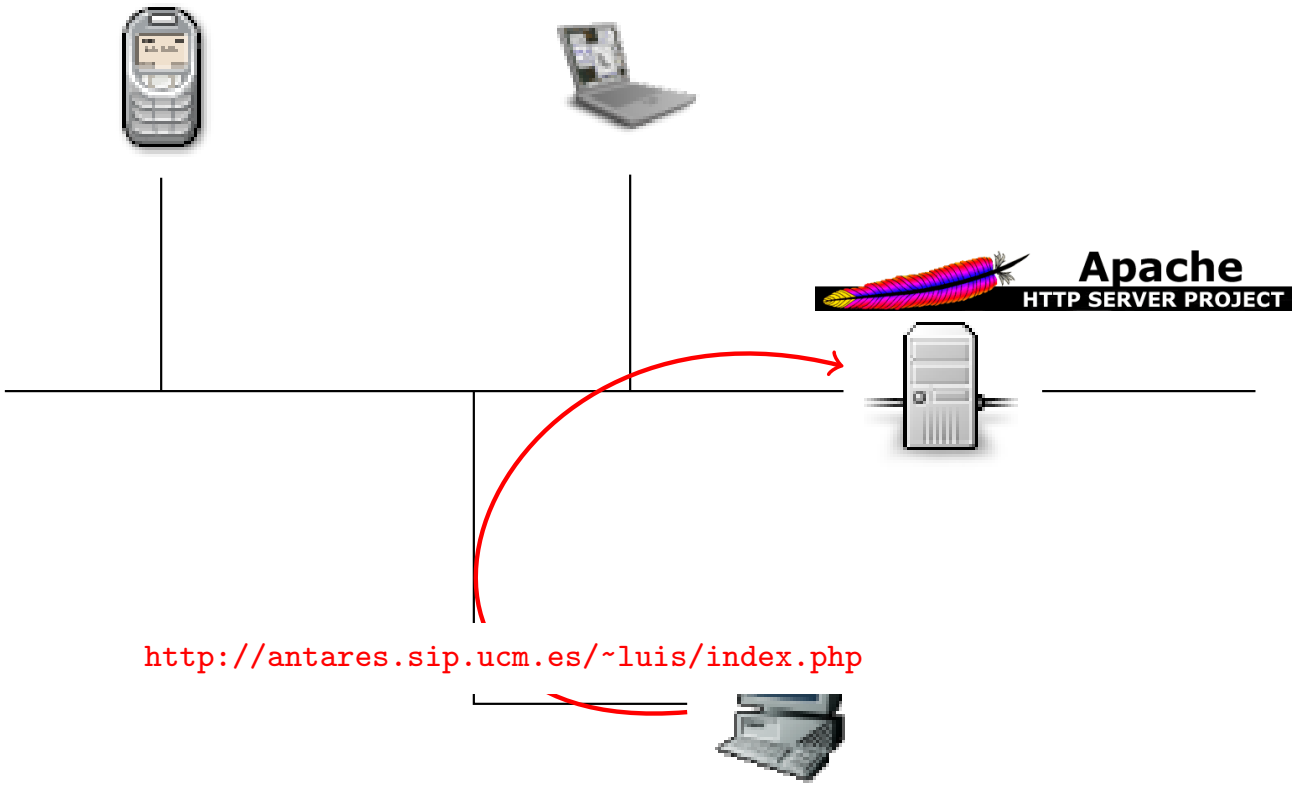
Cómo funciona



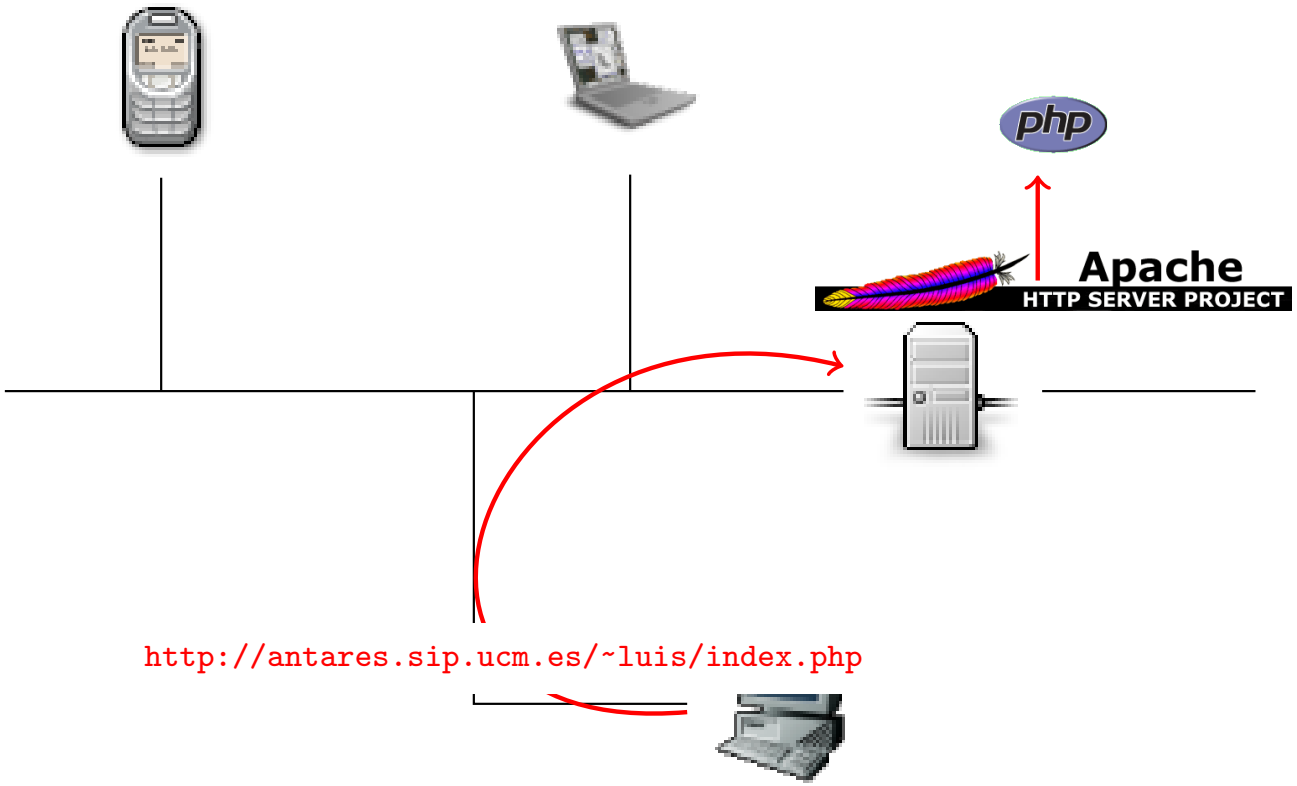
Cómo funciona



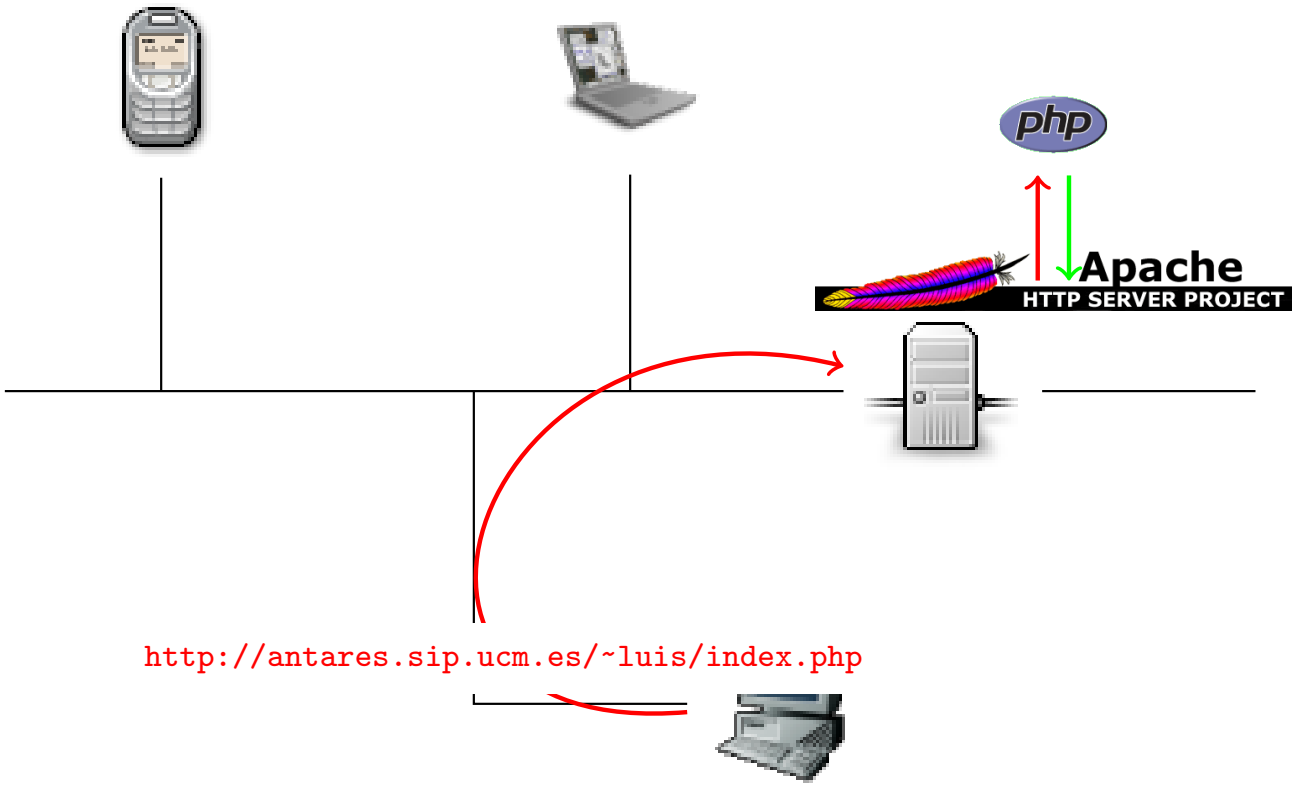
Cómo funciona



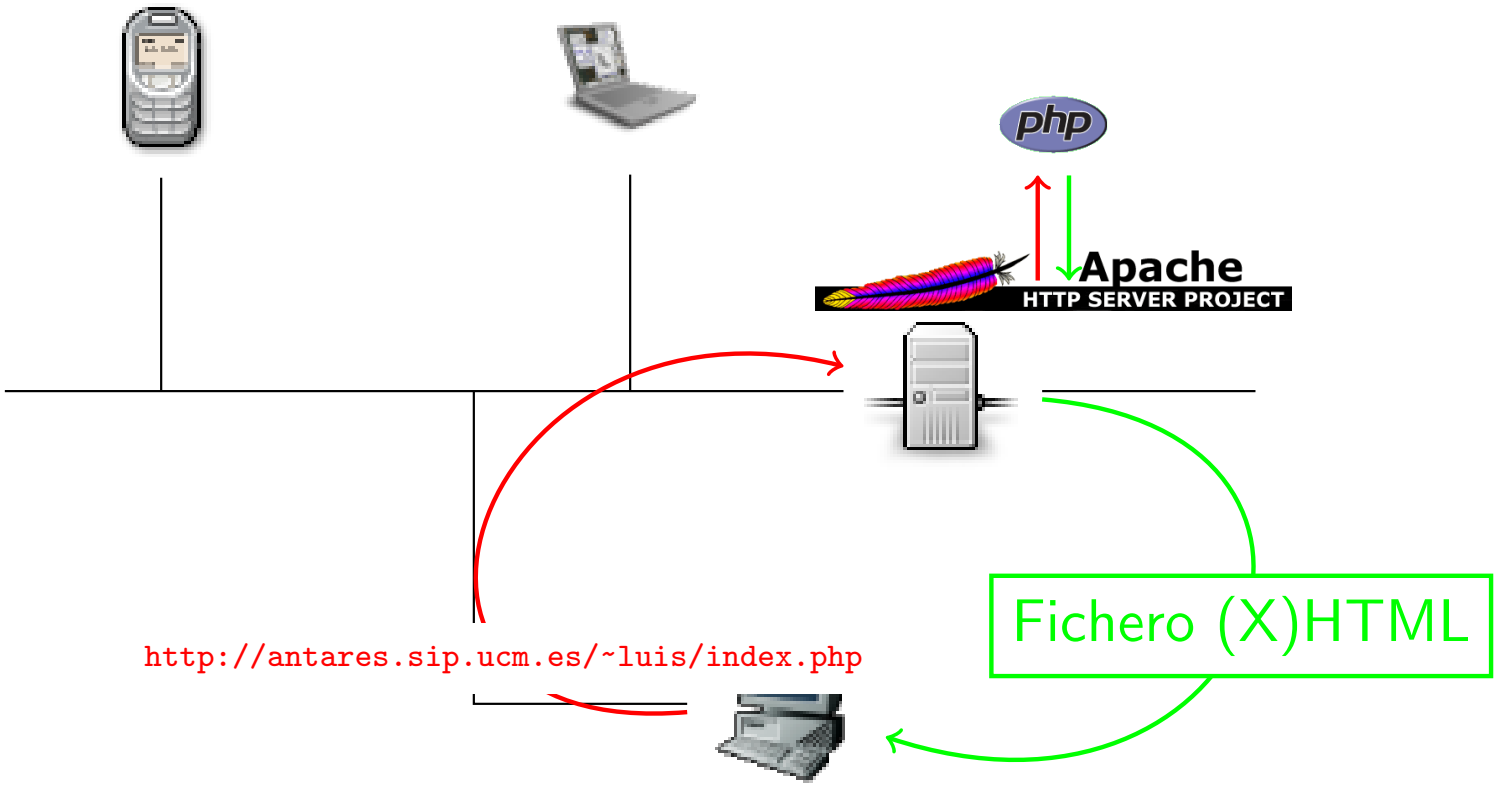
Cómo funciona



Cómo funciona



Cómo funciona



Saliendo de HTML

Un programa, la salida del programa se inserta en su lugar

```
<?php echo("si quieres servir documentos XHTML".
           " o XML, haz como aqu&iacute;\n"); ?>
```

El valor de una expresión:

```
<?= expression ?> Esto es una abreviatura
de " <? echo expression ?>"
```

```
<?php
if ($expression) {
   ?>
    <strong>This is true.</strong>
   <?php
} else {
   ?>
    <strong>This is false.$</strong>
   <?php
}
?>
```

Formularios I

```
<form action="accion.php" method="POST">
 Su nombre: <input type="text" name="nombre" />
 Su edad: <input type="text" name="edad" />
 <input type="submit">
</form>
```

Fichero accion.php

```
Hola <?=$_POST["nombre"]; ?>.
Tiene <?=$_POST["edad"]; ?> a&ntilde;os
```

El cliente finalmente recibe

```
Hola José.
Tiene 22 años
```

Formularios II

```
<form action="accion.php" method="GET">
 Su nombre: <input type="text" name="nombre" />
 Su edad: <input type="text" name="edad" />
 <input type="submit">
</form>
```

Fichero accion.php

```
Hola <?=$_GET["nombre"]; ?>.
Tiene <?php=$_GET["edad"]; ?> a&ntilde;os
```

El cliente finalmente recibe

```
Hola José.
Tiene 22 años
```

- GET para datos que *ocupan poco*.
  Los datos van en la URL, se puede acceder con un enlace:

```
<a href="accion.php?nombre=patata&amp;edad=23">
```

- POST para datos que *ocupan mucho*
  Se pueden mandar ficheros

```
<form id="formulario" action="procesaSugerencia.php"
        method="post"
        enctype="multipart/form-data">
.............
.............
        <input id="fichero" type="file"
               name="fichero"  tabindex="7">
.............
</form>
```

Guardando ficheros

```
<?php
function guardaFichero($directorio,$fichero) {
  $fp = fopen($directorio."/lock.txt", "w+");
  while (! flock($fp, LOCK_EX)) { sleep(1); }
  $timeStamp=getmicrotime();
  $ficheroDestino = $timeStamp . $_FILES[$fichero]["name"];
  move_uploaded_file($_FILES[$fichero]["tmp_name"],
                     $directorio."/".$ficheroDestino);
  flock($fp, LOCK_UN); fclose($fp);
  return $ficheroDestino;
}
while (list($clave,$valor) = each($_FILES)) {
  guardaFichero(directorioDestino,$clave);
?>
<tr>
  <td><?= $valor["name"]?></td>
  <td><?= $valor["size"]?></td>
  <td><?= $valor["type"]?></td>
</tr>
<?php } ?>
```

Datos permanentes: sesiones I

```
session_start();
define(accesos,"accesos");
define(continuar,"continuar");
define(borrar,"borrar");
$accesos=$_SESSION[accesos];
if ( isset($_GET[borrar]) ) {
  unset($_SESSION[accesos]);
}
if ( isset($_SESSION[accesos]) ) {
  $_SESSION[accesos]++;
} else {
  $_SESSION[accesos]=1;
}
$accesos=$_SESSION[accesos];
```

Datos permanentes: sesiones II

```
<?php
  session_start();
  ..............
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=ut
    <title>Ejemplo de sesiones</title>
  </head>
  <body>
    <p>Has accedido <?=$accesos ?> veces</p>
    <form action="session.php">
      <p>
        <button name="<?=concontinuar?>">Continuar</button>
        <button name="<?=borrar?>">Borrar sesion</button>
      </p>
    </form>
  </body>
</html>
```

Tipos I

PHP no es lenguaje tipado

- una variable puede cambiar de tipo cuando se hace una asignación.
- no se declaran las variables
- se puede usar una variable sin darle un valor antes.

Tipos II

Los tipos se corresponden con los valores, no con las variables.

Cuatro tipos escalares

- boolean
- integer
- float (número de punto-flotante, también conocido como 'double')
- string

Tipos III

Dos tipos compuestos

- array, indexados con cualquier tipo: tabla hash.
- object

Dos tipos especiales

- resource
- NULL

Variables I

```php
<?php
$var = "Bob";
$Var = "Joe";
echo "$var, $Var";      // outputs "Bob, Joe"

$4site = 'not yet';     // invalid; starts with a number
$_4site = 'not yet';    // valid; starts with an underscore
?>
```

PRECAUCIÓN

se puede usar una variable sin dar un valor inicial, PHP no se va a quejar.

Variables II

```
<ul>
  <li>:<?= $i ?>:</li>
  <?php $i = true; ?>
  <li>:<?= $i ?>:</li>
  <?php $i = 1; ?>
  <li>:<?= $i ?>:</li>
  <?php $i = 2.3; ?>
  <li>:<?= $i ?>:</li>
  <?php $i = 'Hola'; ?>
  <li>:<?= $i ?>:</li>
</ul>
```

```
<ul>
  <li>:1:</li>
  <li>:1:</li>
  <li>:2.3:</li>
  <li>:Hola:</li>
</ul>
```

Constantes

```php
<?php
define(alumnosDB,"alumnos");
define(dirFotos,"/home/luis/docencia/alumnos/fotos/");
define(urlFotos,"/~luis/fotosAlumnos/");
?>
```

```php
    $nombreFichero=dirFotos.$dni.$ext;
    move_uploaded_file($foto['tmp_name'],$nombreFichero);
    $ficheroFoto=urlFotos.$dni.$ext;
```

Arrays I

```
$correo=array("para"=>"alonso@sip.ucm.es",
                    "de" => "FROM: FORTE registration",
                    "subject" => "Registration FORTE 2004");
define(early,"early");
define(late,"late");
define(forte2004,"forte2004");
define(tutorial,"tutorial");
define(workshop,"workshop");
define(forte2004_workshop,"forte2004_workshop");
define(regular,"regular");
define(student,"student");

$tarifa[forte2004][student][late]=300;
$tarifa[forte2004][student][early]=250;
$tarifa[forte2004][regular][late]=500;
$tarifa[forte2004][regular][early]=450;
```

Arrays II

```
$tarifa[tutorial][student][late]=25;
$tarifa[tutorial][student][early]=25;
$tarifa[tutorial][regular][late]=25;
$tarifa[tutorial][regular][early]=25;

$tarifa[workshop][student][late]=300;
$tarifa[workshop][student][early]=250;
$tarifa[workshop][regular][late]=300;
$tarifa[workshop][regular][early]=250;

$tarifa[forte2004_workshop][student][late]=550;
$tarifa[forte2004_workshop][student][early]=400;
$tarifa[forte2004_workshop][regular][late]=750;
$tarifa[forte2004_workshop][regular][early]=600;
```

Arrays III

```
  <?php
$a[]=7;
$a[]=6;
$a[]=5;
$a[]=4;
$a[]=3;
$a[]=2;
$a[]=1;
$a[]=0;
?>
<ul>
<?php foreach($a as $v) { ?>
<li><?=$v?></li>
<?php } ?>
```

Arrays IV

```
  <ul>
<li>7</li>
<li>6</li>
<li>5</li>
<li>4</li>
<li>3</li>
<li>2</li>
<li>1</li>
<li>0</li>
  </ul>
```

Operadores aritméticos

`-$a`	Negación
`$a + $b`	Adición
`$a - $b`	Substracción
`$a * $b`	Multiplicación
`$a / $b`	División
`$a % $b`	Módulo
`$a++`	Post-incremento
`$a--`	Post-decremento

Operadores de asignación

```
<?php

$a = 3;
$a += 5; // define $a como 8, como si hubiesemos dicho: $a = $a +
$b = "&iexcl;Hola ";
$b .= "a todos!"; // define $b como "&iexcl;Hola a todos!", tal co

?>
```

Operadores de comparación

`$a == $b`	Igual
`$a != $b`	Diferente
`$a <> $b`	Diferente
`$a !== $b`	No idénticos
`$a < $b`	Menor que
`$a > $b`	Mayor que
`$a <= $b`	Menor o igual que
`$a >= $b`	Mayor o igual que

Operadores lógicos

`$a and $b`	Y
`$a or $b`	O
`$a xor $b`	O exclusivo (Xor)
`! $a`	No
`$a && $b`	Y
`$a || $b`	O

Operadores de bit

`$a & $b`	Y
`$a | $b`	O
`$a ^ $b`	O exclusivo
`~ $a`	No
`$a << $b`	Desplazamiento a izquierda
`$a >> $b`	Desplazamiento a derecha

Concatenación de cadenas

```php
<?php
$a = "&iexcl;Hola ";
$b = $a . "Mundo!"; // ahora $b contiene "&iexcl;Hola Mundo!"

$a = "&iexcl;Hola ";
$a .= "Mundo!";    // ahora $a contiene "&iexcl;Hola Mundo!"
?>
```

Condicional

```
if ($a > $b) {
    print "a es mayor que b";
} elseif ($a == $b) {
    print "a es igual que b";
} else {
    print "a es mayor que b";
}
```

Bucles I

```
$i = 1;
while ($i <= 10) {
    print $i;
    $i++;
}
```

```
<?php
for ($i = 1; $i <= 10; $i++) {
    print $i;
}
?>
```

Bucles II

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

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

Inclusión de ficheros I

- `require`, `require_once` (error).
- `include`, `include_once` (warning).

vars.php

```
$color = 'green';
$fruit = 'apple';
```

test.php

```
echo "A $color $fruit"; // A
require 'vars.php';
echo "A $color $fruit"; // A green apple
```

Inclusión de ficheros II

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
<!-- $Id: transparencias.tex,v 1.2 2008-11-13 09:46:42 luis Exp $

<?php require "definiciones.php";?>
<html lang="<?php echo $idioma; ?>">
<?php
  $tituloPagina="Luis Fernando Llana Díaz - Docencia del curso 200
  $basePagina="/~luis/docencia2004-2005.php";
  $i=0;
  $menu[$i][enlace]='#tutorias';
  $menu[$i][texto]='Tutorías';  $i++;

  $menu[$i][enlace]='#concurrente';
  $menu[$i][texto]='Programación Concurrente';
  $i++;

  $menu[$i][enlace]='#labProg';
  $menu[$i][texto]='Laboratorio de Programación';
  $i++;
```

Inclusión de ficheros III

```
  $menu[$i][enlace]='#accesibilidadWEB';
  $menu[$i][texto]='Aplicaciones WEB accesibles';
  $i++;

  $menu[$i][enlace]='principal.php#ghostview';
  $menu[$i][texto]='Visor de PostScript (ficheros .ps)';
  $i++;

  $menu[$i][enlace]='principal.php#knoppix';
  $menu[$i][texto]='GNU/Linux Knoppix';
  $i++;

  require "cabecera.php";
?>
```

Inclusión de ficheros IV

```
<body id="inicio">
<?php echo creaMenu($menu)?>
    <div class="contenido">
    ........................
    ........................
    ........................
    <?php
require "pie.php";
    ?>
    </div>
  </body>
</html>
```

Funciones I

```
function foo ($arg_1, $arg_2, ..., $arg_n) {
    echo "Funci&oacute;n de ejemplo.\n";
    return $retval;
}
```

Funciones II

Parámetros por valor

```
function sum(sum1,sum2) {
  return sum1+sum2;
}
```

Parámetros por referencia

```
function add_some_extra(&$string) {
   $string .= ' y algo m&aacute;s.';
}
$str = 'Esto es una cadena, ';
add_some_extra($str);
echo $str;    // Saca 'Esto es una cadena, y algo m&aacute;s.'
```

Funciones III

Valores por defecto en funciones

```
function makecoffee($type = "cappuccino")
{
    return "Making a cup of $type.\n";
}
echo makecoffee();
echo makecoffee(null);
echo makecoffee("espresso");
```

ERROR

```
function makeyogurt($type = "acidophilus", $flavour)
{
    return "Making a bowl of $type $flavour.\n";
}

echo makeyogurt("raspberry");   // won't work as expected
```

Funciones IV

Funciones con tamaño variable de argumentos

```
function foo()
{
    $numargs = func_num_args();
    echo "Number of arguments: $numargs<br />\n";
    if ($numargs >= 2) {
        echo "Second argument is: " . func_get_arg(1) . "<br />\n
    }
}

foo (1, 2, 3);
```

Scope

```php
$a = 1; /* global scope */
function Test(){
    echo $a; /* reference to local scope variable */
}
Test();
```

```php
$a = 1;
$b = 2;
function Sum() {
    global $a, $b;

    $b = $a + $b;
}
Sum();
echo $b;
```

Variables predefinidas

- `$GLOBALS` References all variables available in global scope
- `$_SERVER` Server and execution environment information
- `$_GET` HTTP GET variables
- `$_POST` HTTP POST variables
- `$_FILES` HTTP File Upload variables
- `$_REQUEST` HTTP Request variables
- `$_SESSION` Session variables
- `$_ENV` Environment variables
- `$_COOKIE` HTTP Cookies
- `$php_errormsg` The previous error message
- `$HTTP_RAW_POST_DATA` Raw POST data
- `$http_response_header` HTTP response headers
- `$argc` The number of arguments passed to script
- `$argv` Array of arguments passed to script

Y mucho más....

....en la documentación de php:
`http://es2.php.net/manual/es/refs.basic.text.php`