

# Sistemas Web

Desarrollo básico

# Indice

- 1) HTML
- 2) CSS
- 3) JavaScript
- 4) PHP

# HTML

- HyperText Markup Language
- Lenguaje de etiquetado (No programación)
- Las etiquetas dan significado y aspecto al texto
- Permite escribir de manera estructurada
- La mayoría de etiquetas son dobles:
  - Apertura: `<html>`
  - Cierre: `</html>`

# HTML

Las etiquetas pueden tener atributos para aportar detalles (En las etiquetas dobles, sólo en la de apertura):

```
<body background="imagendefondo.jpg">
```

# HTML

- Las etiquetas deben estar correctamente anidadas
- Se cierran en orden inverso a su apertura
- Para abrir un archivo HTML sólo se necesita una navegador web que interprete la etiquetas (No todos se comportan igual)

# HTML

Contiene todo el código HTML

Cabecera con información  
que afecta a todo el documento

Contiene el cuerpo del documento:  
la información (con más etiquetas)

```
<html>
  <head>
    ...
    <title>
      Título de la página
    </title>
  </head>
  <body>
    ...
  </body>
</html>
```

# HTML

```
<html>
  <head>
    |   <title>
    |       Ejemplo página web SGSSI
    |   </title>
  </head>
  <body>
    |   Aquí iría toda la información de la página HTML.
  </body>
</html>
```

# HTML

Contenido etiqueta <title>

Ejemplo página web SGSSI - Brave

Ejemplo página web SGSSI x +



File | /home/mikel/EHU-LSI/Docencia/2021-2022/SGSSI/CAST/Temario/SistemaWeb/ejemplo.html

Aquí iría toda la información de la página HTML.

Contenido etiqueta <body>



# HTML

Posibles etiquetas dentro de `<head>`

- `<style>` definir estilos (aspectos) para los elementos del documento HTML
- `<meta>` mediante sus atributos aporta metadatos sobre el documento: autor, palabras clave, etc
- `<script>` permite incluir en el documento operaciones de un lenguaje tipo script

# HTML

Principales etiquetas dentro de `<body>`

- `<h1></h1>`
- `<h2></h2>`
- `<h3></h3>`
- `<h4></h4>`
- `<h5></h5>`
- `<h6></h6>`

Para indicar títulos. El 1 es el más importante y el 6 el menor

- `<p></p>` párrafo
- `<br>` salto de línea
- `<!-- -->` comentario (No se muestra en el navegador)

# HTML

## Principales etiquetas dentro de `<body>`

- `<b></b>` negrita
- `<i></i>` itálica
- `<u></u>` subrayado
- `<del></del>` tachado

# HTML

## Caracteres especiales:

- `&nbsp;`; espacio en blanco
- `&gt;`; >
- `&lt;`; <
- `&aacute;`; á
- `&euro;`; €
- `&amp;`; &
- `&ntilde;`; ñ
- `&quot;`; “

Si no se usa, todos los espacios en blanco se reducen a uno

# HTML

## Principales etiquetas dentro de `<body>`

- `<a></a>` para definir hiperenlaces
  - Atributo `href` para indicar el destino del enlace
  - Atributo `target` para indicar dónde se abre la nueva página
  - Atributo `name` para ir a secciones concretas del mismo o de otro documento

# HTML

## Principales etiquetas dentro de `<body>`

- `<img>` para insertar una imagen
  - Atributo `src` para indicar el path de la imagen
  - Atributo `alt` para indicar una descripción de la imagen
  - Atributos `height` y `weight` para indicar tamaño

# HTML

## Principales etiquetas dentro de `<body>`

- `<table>` para definir tablas
  - La tabla se divide en filas `<tr></tr>`
  - La fila se divide en columnas `<td></td>`

# HTML

Principales etiquetas dentro de `<body>`

...



# HTML

Hasta ahora todas las etiquetas son para estructurar la información introducida cuando se creó la página

No permite interacción por parte del usuario

Para permitir que el usuario introduzca información e interaccione con la página hay que usar formularios

# HTML

- Un formulario se define mediante la etiqueta `<form></form>` en `<body>`
- Principales atributos:
  - `action` indica a qué URL se enviarán los datos
  - `method` qué método se usa para enviar los datos (GET, POST)

Lo veremos en PHP

# HTML

- Principales etiquetas dentro de `<form>`
- `<input>` para definir controles
  - `name` asignar un identificador para acceder a su valor
  - `value` define valore por defecto / texto a mostrar
  - `type` define los distintos tipos de campos

# HTML

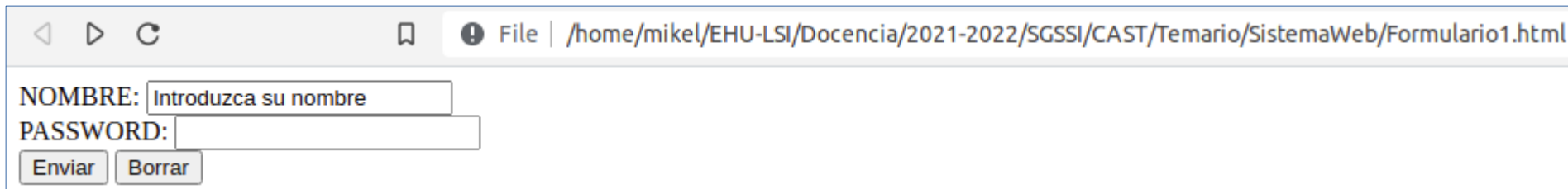
- Principales etiquetas dentro de `<form>`
- `<input>` a través de `type` se puede definir:
  - Botones: `button` (habría que definir qué acción ejecutan)
  - Botón de envío: `submit` (envía los datos al destino del action)
  - Botón de reseteo: `reset` (vacía los campos del formulario)
  - Casillas de verificación: `checkbox`
  - Radiobotones: `radiobutton`
  - Entrada de texto: `text`
  - Entrada de contraseñas: `password`
  - Selección de ficheros: `file`

Quando veamos JavaScript

# HTML

- Principales etiquetas dentro de `<form>`

```
<form action="destino.html">
  NOMBRE: <input type="text" name="nombre" value="Introduzca su nombre"><br>
  PASSWORD: <input type="password" name="contra"><br>
  <input type="submit" value="Enviar">
  <input type="reset" value="Borrar"><br><br>
</form>
```



A screenshot of a web browser window showing the rendered HTML form. The browser's address bar displays the file path: `/home/mikel/EHU-LSI/Docencia/2021-2022/SGSSI/CAST/Temario/SistemaWeb/Formulario1.html`. The form itself contains two input fields: "NOMBRE:" followed by a text input field with the placeholder text "Introduzca su nombre", and "PASSWORD:" followed by a password input field. Below these fields are two buttons: "Enviar" and "Borrar".

# HTML

- Principales etiquetas dentro de `<form>`

```
...
Valora la asignatura de SGSSI <br>
  <input type="radio" name="valor" value="poco">Poco interesante
  <input type="radio" name="valor" value="normal">Normal <br>
  <input type="radio" name="valor" value="muy">Muy interesante <br><br>

Indica los temas que m&aacute;s te interesan<br>
  <input type="checkbox" name="interes" value="tema5">Malware
  <input type="checkbox" name="interes" value="tema6">Seguridad F&iacute;sica
  <input type="checkbox" name="interes" value="tema7">Ingenier&iacute;a Social
  <input type="checkbox" name="interes" value="tema8">Redes
...
```

Al tener el mismo nombre,  
solo se puede elegir uno

Valor que se recoge

Texto que se muestra

# HTML

- Principales etiquetas dentro de `<form>`

```
Añade tu último informe de laboratorio  
<input type="file" name="informe"><br><br>
```

NOMBRE:

PASSWORD:

Valora la asignatura de SGSSI

☐ Poco interesante

☐ Normal

☐ Muy interesante

Indica los temas que más te interesan

☐ Malware ☐ Seguridad Física ☐ Ingeniería Social ☐ Redes

Añade tu último informe de laboratorio  No file chosen

# HTML

- Principales etiquetas dentro de `<form>`
- `<select>` para definir listas desplegables
  - Atributo `name` asignar un identificador para acceder a su valor
  - `<option>` para cada opción
  - Atributo `value` define el valor si se selecciona esa opción
  - Atributo `selected` para indicar seleccionada por defecto
- `<textarea>` para definir areas de texto
  - Tamaño mediante atributos



# HTML

```
Elige tu provincia
<select name="Provincia">
<option value="Ar">Araba</option>
<option value="Bi" selected>Bizkaia</option>
<option value="Gi">Gipuzkoa</option>

</select><br><br>

Introduce tu opini&oslash;n
<textarea name="opinion">

</textarea>
```

NOMBRE:

PASSWORD:

Valora la asignatura de SGSSI

- ☐ Poco interesante  
☐ Normal  
☐ Muy interesante

Indica los temas que m&as te interesan

- ☐ Malware ☐ Seguridad F&isica ☐ Ingenier&ia Social ☐ Redes

A&ntilde;a tu &lt;lt;imo informe de laboratorio  No file chosen

Introduce tu opini&on

Elige tu provincia

# CSS

- HTML puede dar aspecto y estructura a la información: Atributos `bghcolor`, `size`, `align`, ...
- No es recomendable hacerlo
- Es mejor separar
  - Información (Contenido)
  - Aspecto

# CSS

- Para dar aspecto usaremos hojas de estilo
- CSS: Cascading Style Sheets
- Permite cambiar completamente el aspecto de una página web sin tocar el código HTML

# CSS

- Los estilos se definen en un archivo aparte **.css**
- Se indica en la cabecera del fichero HTML con qué archivos CSS tiene que trabajar

`<link rel="stylesheet" type="text/css" href="ficherodeestilos.css">`



Una etiqueta link por cada .css

# CSS


- Funciona con reglas
- Cada regla tiene
  - Selector
  - Declaraciones
    - Propiedad
    - Valor

```
selector { propiedad:valor }  
  
/* Comentarios */
```

# CSS

- El selector puede ser una etiqueta HTML
- Las declaraciones afectan al contenido de dicha etiqueta

```
<link rel="stylesheet" type="text/css" href="basico.css">
```

```
p { color :  red }
```

```
<p> Esto es un párrafo de mi texto </p>
```

```
<p> Esto es otro párrafo de mi página web </p>
```

Esto es un párrafo de mi texto

Esto es otro párrafo de mi página web

# CSS

- Se pueden agrupar varios selectores separados por comas
- Las declaraciones afectan a todos

```
<h1> Este es el encabezado del texto </h1>  
<p> Esto es un párrafo de mi texto </p>  
<p> Esto es otro párrafo de mi página web </p>
```

```
p,h1 { color : red }
```

**Este es el encabezado del texto**

Esto es un párrafo de mi texto

Esto es otro párrafo de mi página web

# CSS

- Se pueden definir clases dentro de un mismo selector
- En el CSS se indica la clase mediante un punto
- En el HTML se indica la clase como un atributo de la etiqueta

```
<p class="uno"> Esto es un párrafo de mi texto </p>  
<p class="dos"> Esto es otro párrafo de mi página web </p>
```

Esto es un párrafo de mi texto

Esto es otro párrafo de mi página web

```
p.uno { color : red }  
  
p.dos {  
    color : purple ;  
    font-size: 50;  
}
```



# CSS

- También se pueden definir clases sin asociarlas a un selector concreto
- En el CSS se indican con un .
- En el HTML se usan donde se quiera

```
<h1 class="dos"> Este es el encabezado del texto </h1>  
<p class="uno"> Esto es un párrafo de mi texto </p>  
<p class="dos"> Esto es otro párrafo de mi página web </p>
```

**Este es el encabezado del texto**

Esto es un párrafo de mi texto

Esto es otro párrafo de mi página web

```
.uno { color : red }  
  
.dos {  
  color : purple ;  
  font-size: 50;  
}
```

# CSS

- Se pueden definir identificadores para elementos concretos
- En CSS se indican mediante #
- En HTML se indican mediante el atributo id

```
<h1 class="dos"> Este es el encabezado del texto </h1>  
<p id="primero" class="uno"> Esto es un párrafo de mi texto </p>  
<p id="segundo" class="dos"> Esto es otro párrafo de mi página web </p>
```

Este es el encabezado del texto

Esto es un párrafo de mi texto

Esto es otro párrafo de mi página web

```
.uno{color: red}  
  
.dos {  
  color: purple;  
  font-size: 50;  
}  
  
#primero {text-align: center}  
#segundo {text-align: right}
```

# CSS

- Ejemplos: <http://csszengarden.com/>
- El HTML es siempre el mismo
- Es importante estructurar bien el HTML
- La etiqueta HTML `<div>` sirve para crear “secciones”

# JavaScript

- JavaScript (JS) es un lenguaje de programación interpretado
  - No se compila
  - No se puede generar un ejecutable
- Se descarga con el fichero HTML
- Se ejecuta en el cliente (Navegador Web)
  - Puede estar desactivado

# JavaScript

- Permite unir el HTML con programación
  - Gestión de eventos
  - Comprobación de los datos del formulario
- Se puede definir en el propio HTML o en un fichero externo

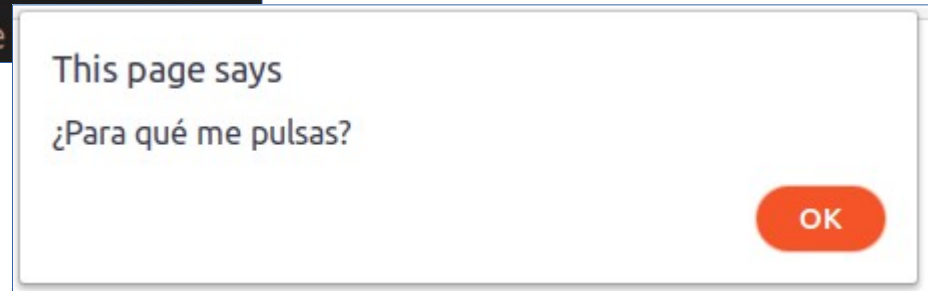
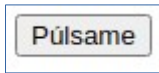
`<script></script>`

`<script type="text/javascript" src="fichero.js"></script>`

# JavaScript

- Gestión de eventos
  - Algunas etiquetas HTML soportan eventos
    - Onload
    - Onclick
    - Onmouseover
    - Onselect
    - ...

```
<input type="button" value="Púlsame"  
onclick="window.alert('¿Para qué me pulsas? me')"/>
```



# JavaScript

- Gestión de eventos
  - Se puede llamar a funciones definidas en JS
  - Desde JS se puede acceder a los campos del formulario
  - Usado para comprobaciones de datos en local

```
<form name="formulario">
  ...
  Nombre: <input type="text" name="nombre">
  <input type="button" value="P&uacute;lame" onclick="comprobardatos()">
  ...
```

```
function comprobardatos() {
  window.alert( document.formulario.nombre.value );
}
```

Nombre del formulario

Nombre del campo

Valor del campo

# JavaScript

- Funcionamiento “habitual” de un formulario
  - El usuario rellena el formulario
  - El botón “Enviar” NO hace Submit (Es de tipo button, no Submit)
  - El botón “Enviar” llama a una función JS que comprueba que los datos del formulario son correctos (En función de lo que se necesite)
  - La función JS, tras comprobar que todo es correcto, hace submit (A donde corresponda)



# JavaScript

```
function comprobardatos() {  
    var edad= document.formulario.edad.value;  
    if (edad < 18) {  
        window.alert ("Lo siento. Debes ser mayor de edad");  
    }  
    else{  
        document.formulario.submit();  
    }  
}
```

Introduce tu edad (debes ser mayor de 18)

Campo1

Campo2

Campo3

Campo4

Campo5

Campo6

This page says

Lo siento. Debes ser mayor de edad

# PHP

- HTML, CSS y JavaScript son útiles para hacer páginas estáticas
- ¿Cómo hacemos que la página pueda variar?
- Una página que muestre un listado de alumnos
  - Si hoy hay 8 alumnos, la tabla debe tener 8 filas
  - Si otro día hay 30 alumnos, la tabla debe tener 30 filas
- Necesitamos una forma de generar el código HTML de forma dinámica

# PHP

- PHP es un lenguaje de programación que se ejecuta en el servidor
  - Permite consultas a una base de datos
  - Es un lenguaje de programación
    - Se pueden poner instrucciones condicionales
    - Se pueden usar bucles
    - ...
  - Genera HTML justo antes de enviarlo al cliente

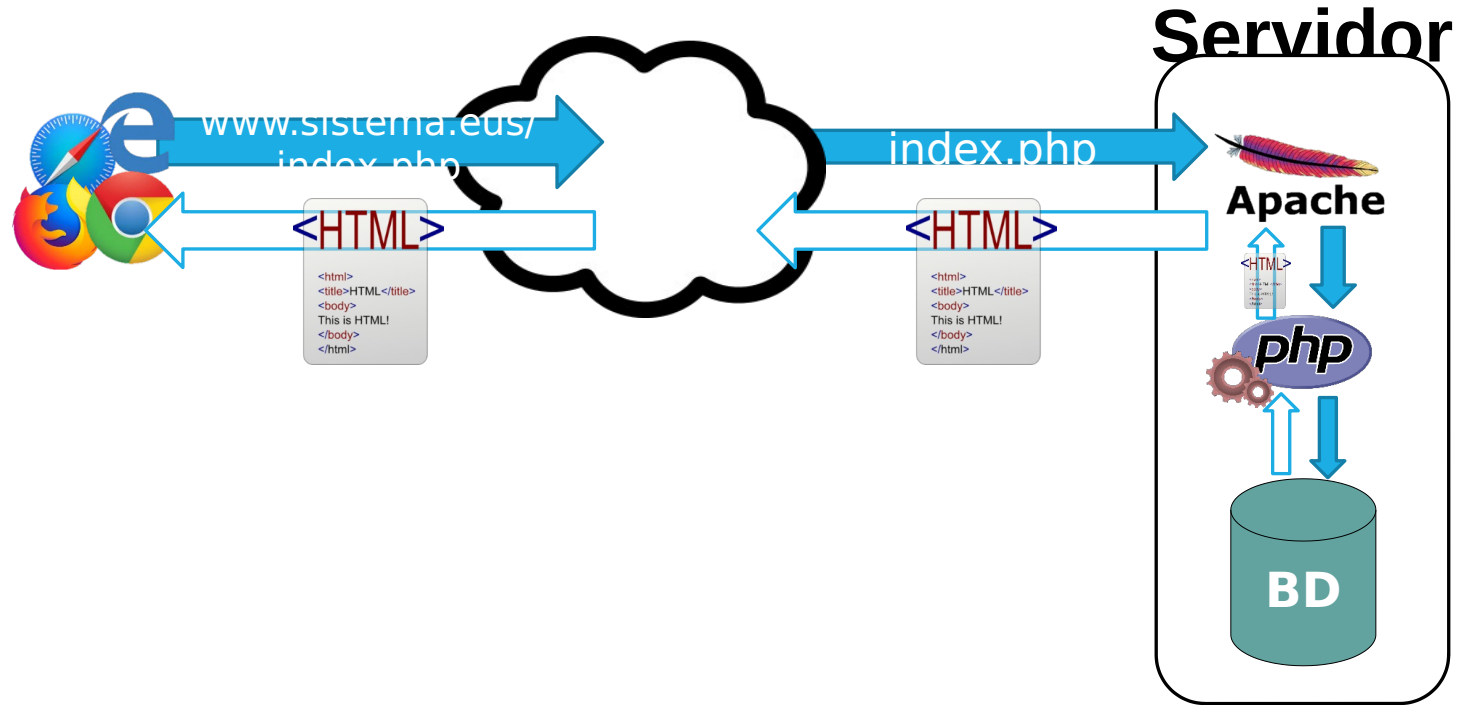
# PHP

- Para ejecutar PHP es necesario un servidor web
  - Apache
  - Microsoft IIS
  - NGINX
  - Tomcat
  - ...
- Los archivos PHP se ejecutan a través del servidor web

# PHP

- El navegador web le “pide” al servidor web que ejecute el archivo PHP
- El servidor web le devuelve al navegador el archivo HTML generado
  - Con el CSS y JavaScript necesario

# PHP



# PHP

- El código PHP debe ir entre etiquetas `<?php` y `?>`
- Se puede poner código PHP entre el HTML
- **NO** se puede poner código HTML entre PHP (directamente)

```
<body>
```

```
<?php
```

```
    echo ("Hola mundo");
```

```
?>
```

```
</body>
```



```
<?php
```

```
<body>
```

```
    echo ("Hola mundo");
```

```
</body>
```

```
?>
```



# PHP

- Para generar HTML desde PHP, hay que “escribirlo por pantalla”
- Al ejecutar un archivo PHP, lo que no esté dentro de las etiquetas de PHP, se queda tal cual en el resultado

```
<body>
```

```
<?php
```

```
    echo ("Hola mundo");
```

```
?>
```

```
</body>
```

```
<?php
```

```
    echo ("<body>");
```

```
    echo ("Hola mundo");
```

```
    echo ("</body>");
```

```
?>
```



# PHP

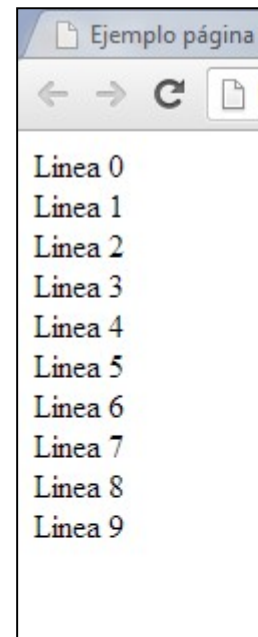
- Con PHP podemos generar HTML de manera dinámica

```
<body>

<?php
    for( $i=0 ; $i<10 ; $i++) {
        echo "linea ".$i."<br>";
    }
?>

</body>
```

Variables con \$



# PHP

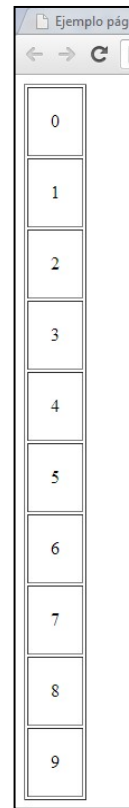
- Con PHP podemos generar HTML de manera dinámica

```
<body>

<?php
    printf("<table border=1 cellpadding=20>");
    for ( $i=0 ; $i<10 ; $i++) {
        printf("<tr><td>".$i."</td></tr>");
    }
?>
</table>

</body>
```

No es muy coherente abrir la tabla en PHP  
y cerrarla fuera, pero se puede



Ejemplo página

0
1
2
3
4
5
6
7
8
9

# PHP

- PHP permite conexiones a Bases de Datos

```
<?php
$conexion = new mysqli("NombreHost", "Usuario", "Contraseña", "nombreBD");

if ($conexion->connect_errno) {
    echo "ERROR: (" . $conexion->connect_errno . ") " . $conexion->connect_error;
}

$resultado = $conexion->query($sql);

mysqli_close($conexion);
?>
```

Para generar una conexión con el SGBD (MySQL)

Trazas (Debugger!)

Código error

Texto error

Ejecutar una sentencia SQL y recoger el resultado (SELECT)

Cerrar la conexión

# PHP

- PHP permite conexiones a Bases de Datos

```
<?php
```

```
$fila= mysqli_fetch_array ($resultado);
```

```
$valor = $fila [ 'nombreatributo'];
```

```
?>
```

Obtiene una fila (tupla) del resultado

Existen varias formas de navegar por el resultado

Extrae de la tupla el valor de la columna "Nombre atributo"

También se puede usar un índice numérico correspondiente al orden de los campos en la SQL

# PHP

- Ejemplo listar nombres en una BD

```
<?php
...
$resul= $conexion->query("SELECT * FROM Usuarios");

while ($fila = mysqli_fetch_array($resul)) {
    echo ($fila ['nombre']);
}
...
?>
```

# PHP

- Ejemplo crear desplegable con los nombres de usuarios

```
<select name="usuarios">
<option selected disabled>-- Elige un usuario --</option>
<?php
    $con = mysqli_query("SELECT * FROM Usuarios ORDER BY Apellido ASC");
    while ($row = mysqli_fetch_array ($con)) {
        $dni = $row['dni'];
        $nom=$row['nombre'];
        $ap=$row['Apellido'];
    ?>
    <option value="<?php $dni ?>"><?php $ap.' '.$nom ?></option>
    <?php
    }
    ?>
</select>
```

# PHP

- Una página PHP puede recibir los datos de un formulario HTML

```
<form action="fichero.php" method="POST">
```

- Para acceder a los datos del formulario, lo hace mediante su nombre

```
$_POST ['nombre del campo del formulario']
```

```
$_GET ['nombre del campo del formulario']
```

# PHP

- PHP permite el uso de sesiones
- Una sesión es un espacio de memoria accesible desde cualquier archivo PHP que pida acceso

Tiene que ser la primera línea del archivo

```
<?php session_start();
```

- Para acceder a la variable de sesión

```
$_SESSION['nombrevariable'] = valor;
```

Creación

```
If $_SESSION ['nombrevariable'] > x
```

Acceso



# PHP

- Las variables de sesión caducan (desaparecen) pasado un tiempo
- Se puede comprobar si existen
- Se puede cerrar una sesión en cualquier momento
- Usadas para “pasar” valores entre páginas sin usar formularios

# PHP

- Funcionamiento “habitual” de un formulario con PHP
  - El usuario rellena el formulario
  - El botón “Enviar” NO hace submit (Es de tipo button)
  - El botón “Enviar” llama a una función JavaScript que comprueba que los datos del formulario son correctos
  - La función JavaScript tras comprobar que todo es correcto, hace submit a un archivo PHP
  - El archivo PHP trata los datos recibidos y se comunica con la BD
  - El archivo PHP genera el código HTML correspondiente (CSS, JS), y responde al usuario