

# Lenguaje PHP

---

DESARROLLO BACK-END

# Formularios

---

## 1. Acceso a formularios HTML desde PHP

# Formularios

---

- Fichero uno.php

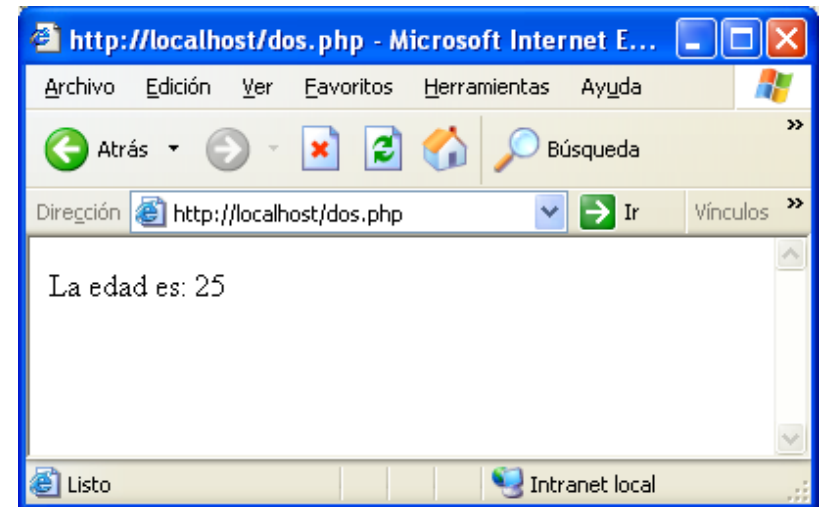
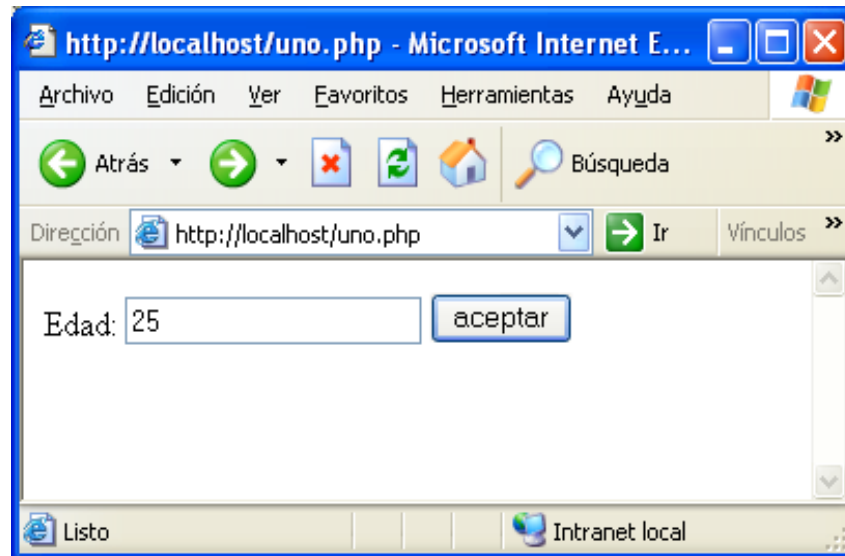
```
<HTML>
<BODY>
<FORM ACTION="dos.php" METHOD="POST">
    Edad: <INPUT TYPE="text" NAME="edad">
    <INPUT TYPE="submit" VALUE="aceptar">
</FORM>
</BODY>
</HTML>
```

- Fichero dos.php

```
<HTML>
<BODY>
<?PHP
    print ("La edad es: $edad");
?>
</BODY>
</HTML>
```

# Formularios

---



# Formularios

---

- A partir de PHP 4.2.0, el valor por defecto de la directiva de PHP **register\_globals** es off
- Esto tiene una gran importancia sobre los formularios, ya que no es posible acceder a las variables enviadas de la manera anterior (como variables globales). En su lugar hay que utilizar la variable predefinida de PHP **\$\_REQUEST**, escribiendo `$_REQUEST['edad']` en lugar de `$edad`
- Se puede poner `register_globals = on` en el fichero de configuración `php.ini`, pero no es recomendable por motivos de seguridad. Una alternativa que permite hacer mínimos cambios en el código ya existente es la siguiente:  
`$edad = $_REQUEST['edad'];`

# Formularios

---

- Acceso mediante la Super Variable (tiene el contenido de `$_POST`, `$_GET` y `$_COOKIE`) `$_REQUEST[ ]`
- Acceso mediante la Super Variable (Método GET)
  - `$_GET[ ]`
- Acceso mediante la Super Variable (Método POST)
  - `$_POST[ ]`

# Formularios

---

- Fichero uno.php

```
<HTML>
<BODY>
<FORM ACTION="dos.php" METHOD="POST">
    Edad: <INPUT TYPE="text" NAME="edad">
    <INPUT TYPE="submit" VALUE="aceptar">
</FORM>
</BODY>
</HTML>
```

- Fichero dos.php

```
<HTML>
<BODY>
<?PHP
    $edad = $_REQUEST['edad'];
    print ("La edad es: $edad");
?>
</BODY>
</HTML>
```

# Formularios

---

- Fichero uno.php

```
<HTML>
<BODY>
<FORM ACTION="dos.php" METHOD="POST">
    Edad: <INPUT TYPE="text" NAME="edad">
    <INPUT TYPE="submit" VALUE="aceptar">
</FORM>
</BODY>
</HTML>
```

- Fichero dos.php

```
<HTML>
<BODY>
<?PHP
    $edad = $_POST['edad'];
    print ("La edad es: $edad");
?>
</BODY>
</HTML>
```



# Formularios

---

- Fichero uno.php

```
<HTML>
<BODY>
<FORM ACTION="dos.php" METHOD="GET">
    Edad: <INPUT TYPE="text" NAME="edad">
    <INPUT TYPE="submit" VALUE="aceptar">
</FORM>
</BODY>
</HTML>
```

- Fichero dos.php

```
<HTML>
<BODY>
<?PHP
    $edad = $_GET['edad'];
    print ("La edad es: $edad");
?>
</BODY>
</HTML>
```

# Formularios

---

- Codificación de datos enviados por Formularios
- Modificador `enctype=`
- Posibilidades:
  - `application/x-www-form-urlencoded` : Todos los caracteres son codificados antes de ser enviados
  - `multipart/form-data` : Los caracteres no son codificados, normalmente lo usamos solo para realizar upload de datos.
  - `text/plain` : Los espacios son convertidos en caracteres "+", pero el resto de los caracteres especiales no son codificados.

# Formularios

---

- Acceso a los diferentes tipos de elementos de entrada de formulario
  - Elementos de tipo INPUT
    - TEXT
    - RADIO
    - CHECKBOX
    - BUTTON
    - FILE
    - HIDDEN
    - PASSWORD
    - SUBMIT
  - Elemento SELECT
    - Simple / múltiple
  - Elemento TEXTAREA

# Formulario

---

- TEXT

Introduzca la cadena a buscar:

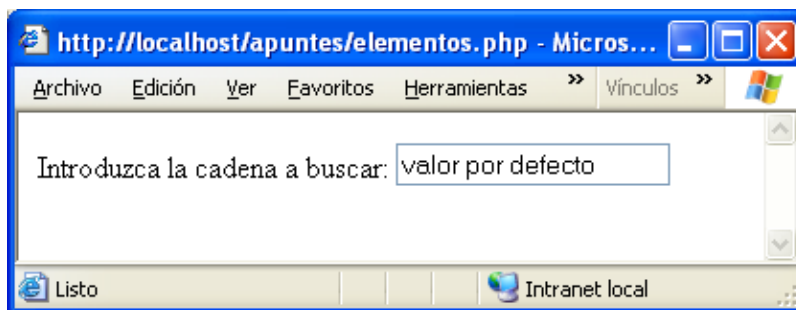
```
<INPUT TYPE="text" NAME="cadena" VALUE="valor por defecto" SIZE="20">
```

```
<?PHP
```

```
    $cadena = $_POST['cadena'];
```

```
    print ($cadena);
```

```
?>
```



# Formularios

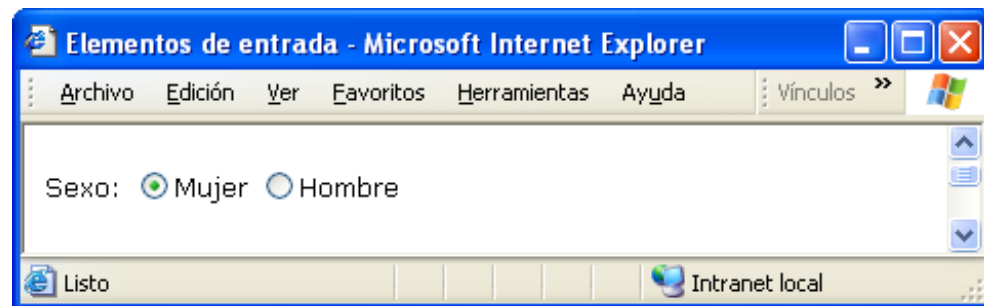
---

- RADIO

Sexo:

```
<INPUT TYPE="radio" NAME="sexo" VALUE="M" CHECKED>Mujer  
<INPUT TYPE="radio" NAME="sexo" VALUE="H">Hombre
```

```
<?PHP  
    $sexo = $_POST['sexo'];  
    print ($sexo);  
?>
```



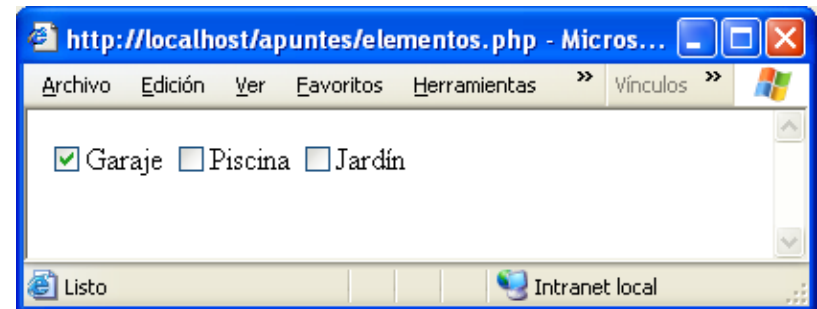
# Formularios

---

- CHECKBOX

```
<INPUT TYPE="checkbox" NAME="extras[]" VALUE="garaje" CHECKED>Garaje  
<INPUT TYPE="checkbox" NAME="extras[]" VALUE="piscina">Piscina  
<INPUT TYPE="checkbox" NAME="extras[]" VALUE="jardin">Jardín
```

```
<?PHP  
    $extras = $_POST['extras'];  
  
    foreach ($extras as $extra)  
        print ("{$extra<BR>\n"});  
?>
```



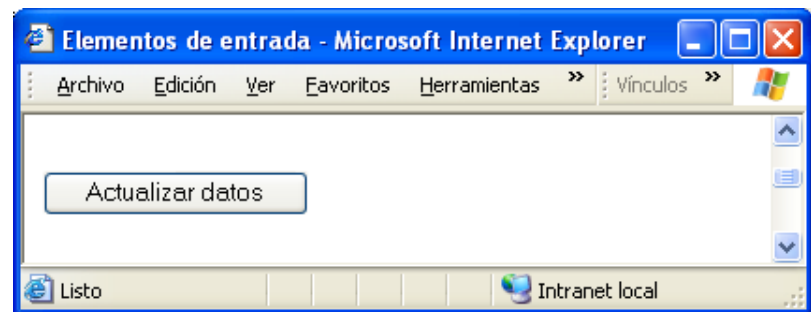
# Formularios

---

- **BUTTON**

```
<INPUT TYPE="button" NAME="actualizar" VALUE="Actualizar datos">
```

```
<?PHP
    $actualizar = $_POST['actualizar'];
    if ($actualizar)
        print ("Se han actualizado los datos");
?>
```



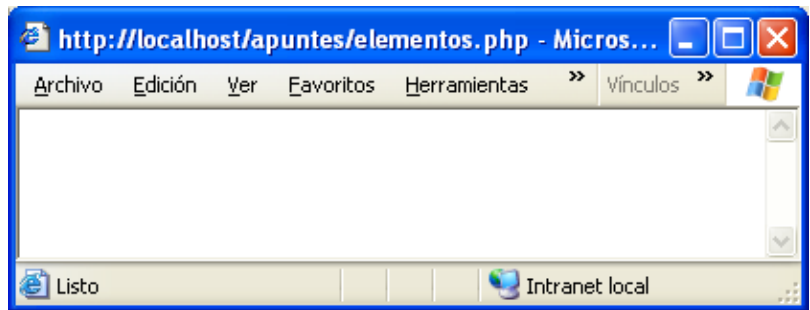
# Formularios

---

- HIDDEN

```
<?PHP
    print("<INPUT TYPE='hidden' NAME='username' VALUE=' $usuario'>\n");
?>
```

```
<?PHP
    $username = $_POST['username'];
    print ($username);
?>
```





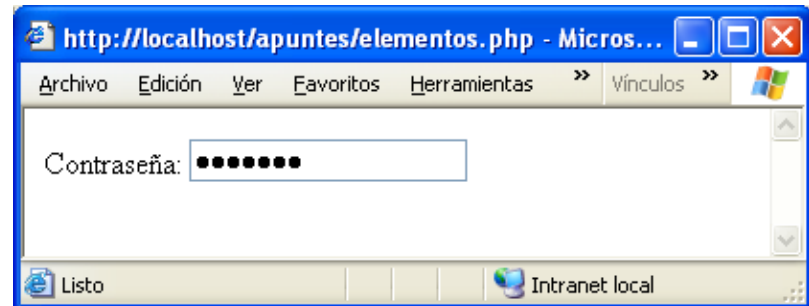
# Formularios

---

- PASSWORD

Contraseña: <INPUT TYPE="password" NAME="clave">

```
<?PHP
    $clave = $_POST['clave'];
    print ($clave);
?>
```



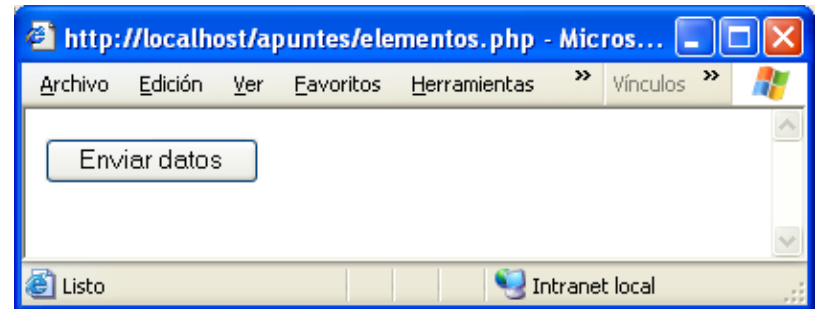
# Formularios

---

- SUBMIT

```
<INPUT TYPE="submit" NAME="enviar" VALUE="Enviar datos">
```

```
<?PHP
    $enviar = $_POST['enviar'];
    if ($enviar)
        print ("Se ha pulsado el botón de enviar");
?>
```



# Formularios

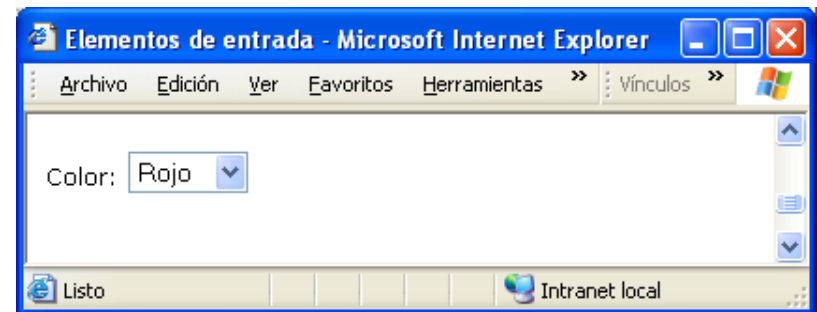
---

- SELECT simple

Color:

```
<SELECT NAME="color">
  <OPTION VALUE="rojo" SELECTED>Rojo
  <OPTION VALUE="verde">Verde
  <OPTION VALUE="azul">Azul
</SELECT>
```

```
<?PHP
  $color = $_POST['color'];
  print ($color);
?>
```



# Formularios

- SELECT múltiple

Idiomas:

```
<SELECT MULTIPLE SIZE="3" NAME="idiomas[]">
  <OPTION VALUE="ingles" SELECTED>Inglés
  <OPTION VALUE="frances">Francés
  <OPTION VALUE="aleman">Alemán
  <OPTION VALUE="holandes">Holandés
</SELECT>
```

```
<?PHP
  $idiomas = $_POST['idiomas'];
  foreach($idiomas as $idioma)
    print ("{$idioma}<BR>\n");
?>
```



# Formularios

---

- TEXTAREA

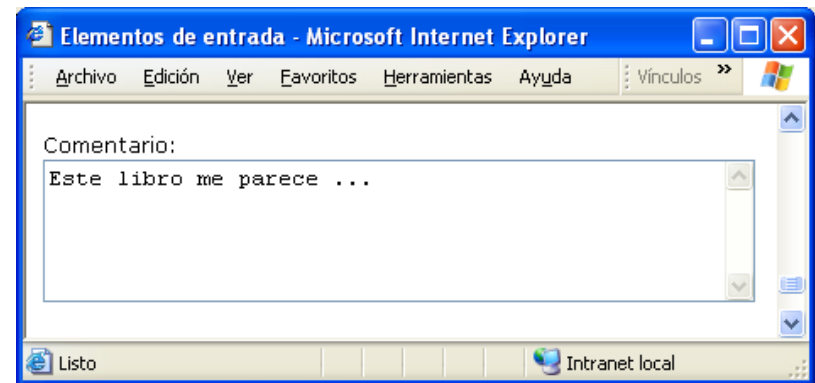
Comentario:

```
<TEXTAREA COLS="50" ROWS="4" NAME="comentario">
Este libro me parece ...
</TEXTAREA>
```

```
<?PHP
```

```
    $comentario = $_POST['comentario'];
    print ($comentario);
```

```
?>
```

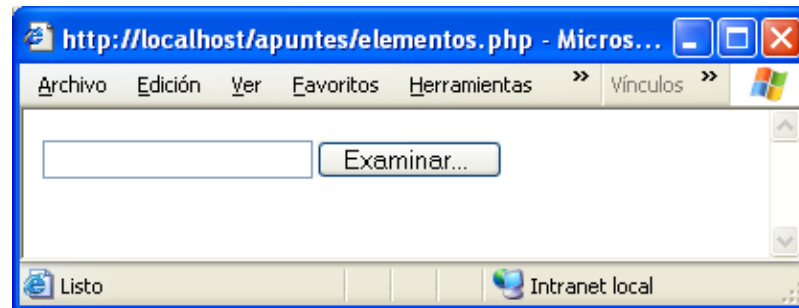


# Formularios

---

- Upload de Archivos

```
<FORM ACTION="procesa.php" METHOD="post"  
  ENCTYPE="multipart/form-data">  
  <INPUT TYPE="file" NAME="file">  
</FORM>
```



# Formularios

---

- Upload de Archivos

```
<?php
if($_FILES["file"]["error"] > 0)
{
    echo "Error: " . $_FILES["file"]["error"] . "<br />";
}
else
{
    echo "Upload: " . $_FILES["file"]["name"] . "<br />";
    echo "Type: " . $_FILES["file"]["type"] . "<br />";
    echo "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";
    echo "Stored in: " . $_FILES["file"]["tmp_name"];
    if(file_exists("upload/" . $_FILES["file"]["name"]))
    {
        echo $_FILES["file"]["name"] . " ya existe. ";
    }
    else
    {
        move_uploaded_file($_FILES["file"]["tmp_name"],
        "upload/" . $_FILES["file"]["name"]);
        echo "Almacenado en: " . "upload/" . $_FILES["file"]["name"];
    }
}
?>
```

# Sesiones

---



# Sesiones

---

1. Introducción
2. Manejo de sesiones
3. Autenticación de usuarios

# Sesiones

---

- A veces es necesario mantener el estado de una conexión entre distintas páginas o entre distintas visitas a un mismo sitio
  - Ejemplos: aplicaciones personalizadas, carrito de la compra, control de acceso
- HTTP es un **protocolo sin estado**: cada conexión entre el cliente y el servidor es independiente de las demás
- Para mantener el estado entre diferentes conexiones hay que establecer lo que se conoce como una **sesión**
- Las sesiones permiten disponer de unas variables con valores persistentes durante toda la conexión del usuario. Estas variables pueden almacenarse en el cliente mediante *cookies* o en el servidor
- PHP dispone de una biblioteca de funciones para la gestión de sesiones

# Sesiones

---

- Funciones de PHP para el manejo de sesiones

`session_start()`

Inicializa una sesión y le asigna un identificador de sesión único. Si la sesión ya está iniciada, carga todas las variables de sesión

`session_register(variable)`

Registra una variable de sesión, equivalente a `$_SESSION[]`

`session_unregister(variable)`

Elimina una variable de sesión, equivalente a `unset($_SESSION[])`

# Sesiones

---

- Funciones de PHP para el manejo de sesiones

`session_is_registered(variable)`

Comprueba si una variable está registrada. Devuelve *true* en caso afirmativo y *false* en caso contrario

`session_destroy()`

Cierra una sesión

`session_id()`

Devuelve o setea el id de la sesión actual

# Sesiones

---

- Funciones de PHP para el manejo de sesiones

- `session_start ()`
  - inicializa una sesión y le asigna un identificador de sesión único. Si la sesión ya está iniciada, carga todas las variables de sesión
- `$_SESSION[ 'nombre' ] = valor;`
  - registra una variable de sesión
- `unset ( $_SESSION[ 'nombre' ] );`
  - elimina una variable de sesión
- `if (isset ( $_SESSION[ 'nombre' ] ))`
  - comprueba si una variable está registrada. Devuelve *true* en caso afirmativo y *false* en caso contrario
- `session_destroy ()`
  - cierra una sesión

# Sesiones

---

- El manejo de las sesiones se realiza de la siguiente forma:
  - Todas las páginas deben realizar una llamada a `session_start()` para cargar las variables de la sesión
  - Esta llamada debe estar colocada antes de cualquier código HTML
  - Conviene llamar a `session_destroy()` para cerrar la sesión

# Cookies

---

- Son los archivos alojados en el cliente que nos permiten identificar un petición determinada.
- Cada vez que el navegador realiza una petición a un dominio, si hay una *cookie* asociada, esta se envía al servidor junto con la petición.
- PHP nos entrega una serie de Funciones y variables predefinidas para crear y acceder a *cookies*

# Cookies

---

- Funciones

```
setcookie(nombre, valor, expiracion, path, dominio,  
secure, httponly (5.2.0));
```

Crea o elimina una *cookie*

```
setrawcookie(nombre, valor, expiracion, path,  
dominio, secure, httponly);
```



# Cookies

---

- Recuperación de una *cookie*

Se utiliza la variable predefinida `$_COOKIE[]` o `$HTTP_COOKIE_VARS` (deprecated)

Limpiar una *cookie*

```
setcookie("user", "", time()-3600);
```