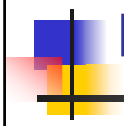


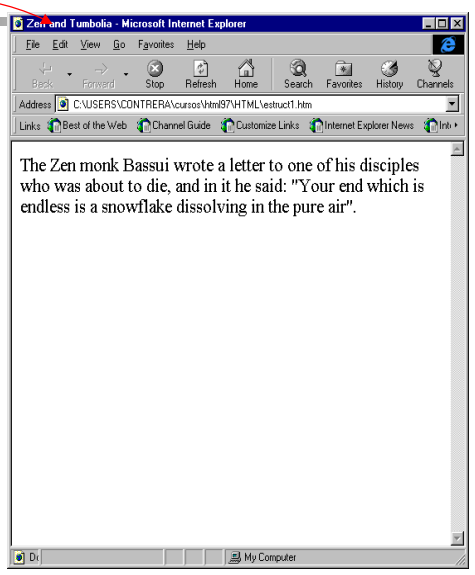
HTML, PHP y bases de datos

Estrella Pulido Cañabate

HTML

Estructura de un documento HTML

```
<HTML>
<HEAD>
<TITLE>Zen and Tumbolia</TITLE>
</HEAD>
<BODY>
The Zen monk Bassui wrote a letter to
    one of his disciples who was about
    to die, and...
</BODY>
</HTML>
```



Formularios HTML

- Formulario empieza con <FORM> y termina con </FORM>
- Dos atributos obligatorios
  - METHOD
    - especifica cómo se pasan los datos al programa que los procesará
      - GET – añadidos al final de la URL
      - POST – como un paquete aparte ⇐
  - ACTION
    - URL de la página que se cargará cuando el usuario pulse el botón ENVIAR

Campos INPUT

<INPUT TYPE="tipo" NAME="nombre" VALUE="valor_defecto>

- text - una línea de texto
- password - aparecen *
- checkbox - opciones no excluyentes
- radio - opciones excluyentes
- reset - limpia el contenido del formulario
- submit - envía datos a la aplicación
- image - submit con imagen en vez de botón
- hidden – son invisibles y sirven para enviar información al programa que procesa la información del formulario

Campo SELECT

```
<SELECT NAME="nombre" SIZE="N" MULTIPLE>
<OPTION> alternativa 1
<OPTION> alternativa 2
....
<OPTION> alternativa N
</SELECT>
```

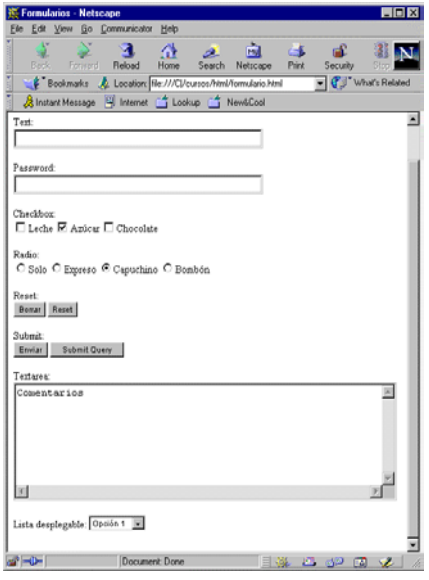
- lista desplegable
- SIZE - número de elementos visibles
- MULTIPLE - se pueden elegir más de un elemento

Campo TEXTAREA

<TEXTAREA NAME="nombre" ROWS="filas visibles"
COLS="columnas visibles">
Texto por defecto
...
</TEXTAREA>

- Campos de entrada de más de una línea
- Se puede hacer "scrolling"

Ejemplo



Ejemplo de uso

```
<form name="inscripcion" method="post" action="mailto:direccion@correo.es">
Nombre:
<br>
<input type="text" name="nombre" size="20" maxlength="20">
<br><br>
Apellidos:
<br>
<input type="text" name="apellidos" maxlength="60" size="60">
<br><br>
Facultad:
<br>
<input type="text" name="facultad" size="60" maxlength="60">
<br><br>
Estudios:
<br>
<select name="estudios">
<option>Primer Ciclo</option>
<option>Segundo Ciclo</option>
<option>Doctorado</option>
</select>
<br><br>
<input type="submit" name="Submit" value="Enviar">
</form>
```

Definición de tablas

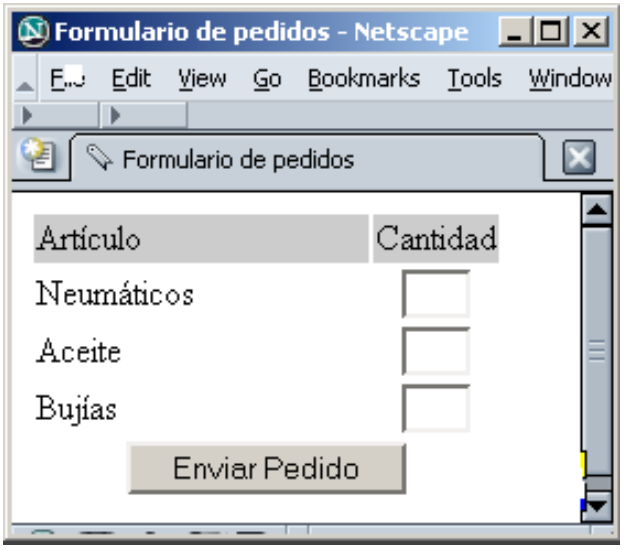
- Inicio y fín de tabla

  `<TABLE> </TABLE>`
- Definición de filas

  `<TR> </TR>`
- Definición de celdas
  - celdas de datos `<TD> </TD>`
  - celdas cabecera `<TH> </TH>`

formulario.html

```
<html>
<head> <title>Formulario de pedidos</title> </head>
<form action="procespedido.php" method=post>
<table border=0>
<tr bgcolor=#cccccc>
  <td width=150>Artículo</td>  <td width=15>Cantidad</td>
</tr>
<tr>
  <td>Neumáticos</td>  <td align="center"><input type="text" name="neumcant" size="3"
     maxlength="3"></td>
</tr>
<tr>
  <td>Aceite</td> <td align="center"><input type="text" name="aceitecant" size="3"
     maxlength="3"></td>
</tr>
<tr>
  <td>Bujías</td> <td align="center"><input type="text" name="bujcant" size="3"
     maxlength="3"></td>
</tr>
<tr>
  <td colspan="2" align="center"><input type="submit" value="Enviar Pedido"></td>
</tr>
</table>
</form>
</html>
```



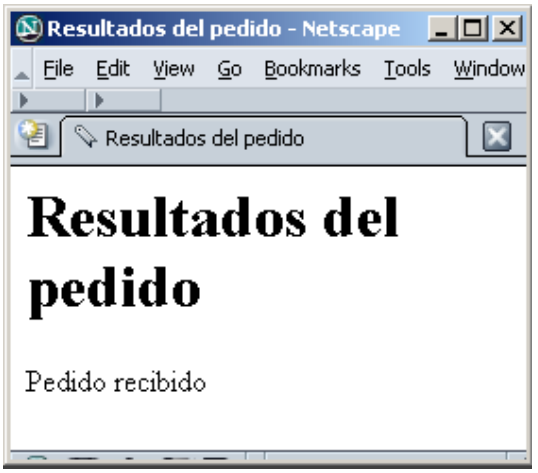
PHP

procespedido.php

```
<html>
<head>
  <title>Resultados del pedido</title>
</head>
<body>
<h1>Resultados del pedido</h1>

<?php
  echo '<p>Pedido recibido</p>';
?>

</body>
</html>
```



Notas sobre el código

- Código PHP

  <?php

  ...

  ?>
- Instrucciones terminan en ;
- **echo** – imprime cadena que aparece en argumento

Comentarios

- En HTML

  <!– El comentario va aquí. -->

- En PHP

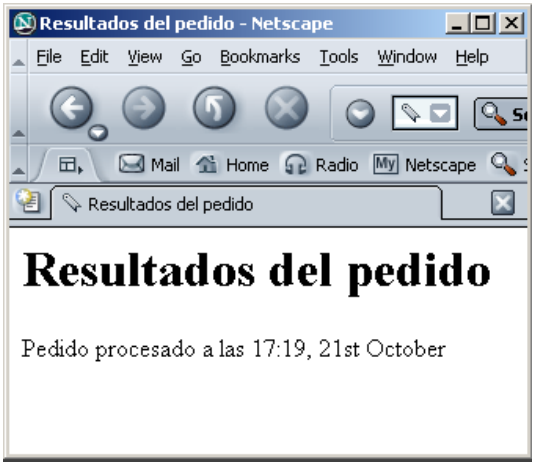
  # Esto es un comentario.

  // Esto también es un comentario.

  /* Esto es un comentario más largo que ocupa dos líneas */

procespedido.php

```
<html>
<head>
  <title>Resultados del pedido</title>
</head>
<body>
<h1>Resultados del pedido</h1>
<?php
  echo '<p>Pedido procesado a las ';
  echo date('H:i, jS F');
  echo '</p>';
?>
</body>
</html>
```



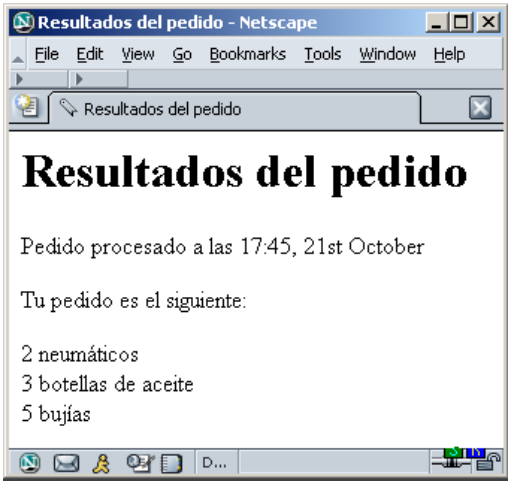
La función date()

date('H:i, jS F')

- Recibe como argumento una cadena de formato
- Cada letra de la cadena representa una parte de la fecha y hora
- H es la hora
- i son los minutos
- j es el día del mes
- S representa el sufijo "th‟
- F es el mes

procespedido.php

```
<html>
<head><title>Resultados del pedido</title></head>
<body>
<h1>Resultados del pedido</h1>
<?php
  echo '<p>Pedido procesado a las ';
  echo date('H:i, jS F');
  echo '</p>';
//crear nombres de variable cortos
  $neumcant = $_REQUEST['neumcant'];
  $aceitecant = $_REQUEST['aceitecant'];
  $bujcant = $_REQUEST['bujcant'];
echo '<p>Tu pedido es el siguiente: </p>';
echo $neumcant. ' neumáticos<br />';
echo $aceitecant. ' botellas de aceite<br />';
echo $bujcant. ' bujías<br />';
?>
</body>
</html>
```



Concatenación de cadenas

- Operador de concatenación .

```
echo $neumcant. ' neumáticos<br />';
```

Variables

- Posiciones de memoria donde se almacenan valores
- Ocho tipos
  - Escalar: booleana, entera, punto flotante, cadenas
  - No escalar: vectores, objetos
  - Recursos
  - NULL
- Se les puede asignar valor con =
- No hay que declararlas

Reglas sintácticas para nombres de variables

- Deben empezar con $
- Pueden contener caracteres, números y _
- El primer carácter después de $ no puede ser un número
- Sensibles a mayúsculas y minúsculas

Paso de valores

- HTML
  - Formulario

    <input type="text" name="**dni**">
  - Desde enlace

    <a href="procesa.php"?**dni**=<? =$dni ?>
    &nombre=...
- En fichero php

  $dni = $_REQUEST[`dni´]

Visualización de valores

- En general
  echo $dni
- Dentro de código HTML
  <? =$dni ?>
- Ejemplo
  <tr><td>DNI
  <tr><td><? =$dni ?>

Instrucciones condicionales

```
if( $canttotal == 0 )
{
echo 'El pedido está vacío<br />';
}
else
{
echo $neumcant.' neumáticos<br />';
echo $aceitecant.' botellas de aceite<br />';
echo $bujcant.' bujías<br />';
}
```

Bucles while

while(*condición*) *expresión*;

```
$num = 1;
while ($num <= 5 )
{
   echo $num."<br />";
   $num++;
}
```

Bucles for

for(*expresión1*; *condición*; *expresión2*)
   expresión3;

```
<?
for ($distancia = 50; $distancia <= 250; $distancia +=
   50)
{
echo "<tr>\n <td align='right'>$distancia</td>\n";
echo " <td align=right>". $distancia / 10 ."</td>\n</tr>\n";
}
?>
```

Acceso a BD con PHP

Acceso a BD con PHP

- Los pasos básicos para consultar una base de datos en la web son:
  - Establecer una conexión con la base de datos apropiada.
  - Consultar la base de datos.
  - Extraer los resultados.
  - Mostrar los resultados al usuario.
  - Cerrar la conexión

Establecer una conexión con la BD

- Función pg_connect(): Establece una conexión con la BD.
- Normalmente recibe como argumentos el nombre de la máquina donde se ejecuta PostgreSQL, el usuario y la contraseña con los que conectarse.

```
$con = pg_connect("dbname=DBTEST user=alumnodb
    password=eps1");
if (!$con)
{
    echo 'Error: No se pudo conectar a la base de datos';
    exit;
}
```

Consultar la BD

- Se realiza la consulta mediante la función **pg_query()**

```
$resultado = pg_query($con, "SELECT autor, email
FROM autores");
if (!$resultado) {
  echo "Ocurrió un error.\n";
  exit;
}
```

La función devuelve un identificador del resultado (que permite extraer los resultados de la consulta) en caso de éxito y *false* en el caso de que se haya producido algún fallo.

Extraer los resultados

- Cada fila de resultado se procesa con la función **pg_fetch_row()**
  - Devuelve la fila como un vector
  - Cada columna es un elemento del vector (empezando a partir de la posición 0)

```
while ($fila = pg_fetch_row($resultado)) {
    echo "Autor: $fila[0]  E-mail: $fila[1]";
    echo "<br />\n";
}

?>
```

Extraer los resultados (II)

- La función **pg_num_rows()** devuelve el número filas obtenidas en la consulta.

```
$num_filas = pg_num_rows($resultado);
```

Introducir nueva información en la BD

- Se realizan los mismos pasos que empleamos para consultar la BD.
- Como diferencia, la consulta se compondrá de un INSERT, UPDATE o DELETE en lugar de SELECT
- Tras ejecutar pg_query() se puede llamar a la función pg_num_rows() para conocer cuántas filas se han introducido, actualizado o borrado.

Cerrar la conexión

pg_free_result($con)

Secuencias en postgreSQL

- Para crear una columna cuyo valor se autoincremente se utiliza el tipo de datos **serial**

```
CREATE TABLE curso (
  numero serial,
  ....)
```

- Para insertar

```
INSERT INTO curso(numero, ....)
  VALUES(NEXTVAL('curso_numero_seq'),...)
```

tabla columna

Ventanas de aviso

```
echo "
<SCRIPT language='JavaScript'>
<!--
alert('¡Registro borrado!');
-->
</script>
";
```

En el laboratorio

- Todos los ficheros php y html en carpeta **public_html**
- En navegador

  http://localhost/~usuario/fichero.php