

# Capítulo 2

## La Web: Parte 2

Application
Transport
Network
Link
Physical

# Páginas estáticas

- Las **páginas estáticas** son documentos en algún formato.
  - P.ej. HTML, pdf, etc.
  - Suelen escribirse en **HTML** (ahora se usa HTML 5)

# HTML

- **HTML (lenguaje de marcado de hipertexto).**
  - ❑ HTML sirve para indicar cómo se van a formatear los documentos al mostrarlos en pantalla.
  - ❑ La **sintaxis** de HTML es muy *parecida* a la de ***XML***.
- **Cosas que se pueden representar con HTML:**
  - ❑ HTML permite producir páginas que incluyen texto, gráficos, hipervínculos, etc.
  - ❑ HTML también permite representar **tablas** y **formularios**

# HTML

HTML tiene **comandos de marcado**.

- Encerrados entre **etiquetas**.
- Una etiqueta está demarcada entre '`<`' y '`>`'.
- Etiquetas tienen un **nombre**.
- Etiquetas pueden tener o no **atributos**.
- Un atributo tiene un **nombre** y un **valor** (que es un **string**) separados por '='.
- **P.ej:** atributo de nombre *href* de un hiperenlace es un URL.

## Algunos de los comandos de marcado más usados

Tag	Description
<code>&lt;html&gt; ... &lt;/html&gt;</code>	Declares the Web page to be written in HTML
<code>&lt;head&gt; ... &lt;/head&gt;</code>	Delimits the page's head
<code>&lt;title&gt; ... &lt;/title&gt;</code>	Defines the title (not displayed on the page)
<code>&lt;body&gt; ... &lt;/body&gt;</code>	Delimits the page's body
<code>&lt;h n&gt; ... &lt;/h n&gt;</code>	Delimits a level <i>n</i> heading
<code>&lt;b&gt; ... &lt;/b&gt;</code>	Set ... in boldface
<code>&lt;i&gt; ... &lt;/i&gt;</code>	Set ... in italics
<code>&lt;center&gt; ... &lt;/center&gt;</code>	Center ... on the page horizontally
<code>&lt;ul&gt; ... &lt;/ul&gt;</code>	Brackets an unordered (bulleted) list
<code>&lt;ol&gt; ... &lt;/ol&gt;</code>	Brackets a numbered list
<code>&lt;li&gt;</code>	Starts a list item (there is no <code>&lt;/li&gt;</code> )
<code>&lt;br&gt;</code>	Forces a line break here
<code>&lt;p&gt;</code>	Starts a paragraph
<code>&lt;hr&gt;</code>	Inserts a Horizontal rule
<code>&lt;img src="..."&gt;</code>	Displays an image here
<code>&lt;a href="..."&gt; ... &lt;/a&gt;</code>	Defines a hyperlink

# HTML

```
<html>
<head><title> AMALGAMATED WIDGET, INC. </title> </head>
<body><h1> Welcome to AWI's Home Page</h1>
 <br>
We are so happy that you have chosen to visit <b> Amalgamated Widget's </b>
home page. We hope <i> you </i> will find all the information you need here.
<p>Below we have links to information about our many fine products.
You can order electronically (by WWW), by telephone, or by fax. </p>
<hr>
<h2> Product information </h2>
<ul>
  <li> <a href="http://widget.com/products/big"> Big widgets </a>
  <li> <a href="http://widget.com/products/little"> Little widgets </a>
</ul>
<h2> Telephone numbers</h2>
<ul>
  <li> By telephone: 1-800-WIDGETS
  <li> By fax: 1-415-765-4321
</ul>
</body>
</html>
```

(a)

## Welcome to AWI's Home Page



We are so happy that you have chosen to visit **Amalgamated Widget's** home page. We hope you will find all the information you need here.

Below we have links to information about our many fine products. You can order electronically (by WWW), by telephone, or by FAX.

---

### Product Information

- [Big widgets](#)
- [Little widgets](#)

### Telephone numbers

- 1-800-WIDGETS
- 1-415-765-4321

(b)

(a) HTML de una página de ejemplo. (b) La página formateada.

# Páginas dinámicas

- **Problema:** trabajar con páginas estáticas se torna muy ineficiente cuando la información cambia frecuentemente o cuando la información de la página varía de acuerdo a diferentes parámetros.
- Queremos evitar tener que modificar a mano las páginas estáticas a cada rato o tener que producir demasiadas páginas dinámicas.
- **Solución:** usar **páginas dinámicas**:
  - generadas por medio de scripts del lado del servidor.

# Páginas dinámicas

- **Problema:** Con un URL como los de antes no basta para especificar la página dinámica deseada.
  - Es necesario tener parámetros para la creación de páginas dinámicas.
  - Además hace falta poder ingresar los parámetros
- **Solución:** el URL contiene nombre de programa y parámetros. Los parámetros son ingresados por medio de formulario HTML.
  - P. ej: demo\_get2.asp?fname=Henry&name=Ford
  - **Este ejemplo nos dice:**
    - Los parámetros son pares: nombre=valor.
    - Se separa nombre de programa con '?'
    - Se separan parámetros con '&'.

# HTML

## Etiquetas para definir formularios HTML

Tag	Description
<code>&lt;form action="", method =""&gt; ... &lt;\form&gt;</code>	Declara un formulario. Action es URL de la página ejecutable que procesa formulario. Method especifica cómo los datos se mandan al servidor (p.ej. GET, POST)
<code>&lt;select&gt; ... &lt;\select&gt;</code>	Para especificar una lista de la que usuario elige un elemento.
<code>&lt;option value = ""&gt; ... &lt;\option&gt;</code>	Para indicar opción de <select>
<code>&lt;textarea rows = "" cols=""&gt; ... &lt;/textarea&gt;</code>	Control de ingreso de texto de varias líneas
<code>&lt;input name="" type="" value=""&gt;</code>	Permite definir campo de input donde type puede ser: button, radio, password, text, submit, checkbox, hidden, etc.



# Formularios HTML

(a) Formulario HTML para una orden de compra.

(b) La página formateada.

Una respuesta posible desde el navegador al servidor con la información llenada por el usuario.

Los parámetros ingresados por el usuario se envían del navegador al servidor de la siguiente forma:

customer=John+Doe&address=100+Main+St.&city=White+Plains&  
state=NY&country=USA&cardno=1234567890&expires=6/98&cc=mastercard&  
product=cheap&express=on

```
<html>
<head> <title> AWI CUSTOMER ORDERING FORM </title> </head>
<body>
<h1> Widget Order Form </h1>
<form ACTION="http://widget.com/cgi-bin/widgetorder" method=POST>
<p> Name <input name="customer" size=46> </p>
<p> Street Address <input name="address" size=40> </p>
<p> City <input name="city" size=20> State <input name="state" size =4>
Country <input name="country" size=10> </p>
<p> Credit card # <input name="cardno" size=10>
Expires <input name="expires" size=4>
M/C <input name="cc" type=radio value="mastercard">
VISA <input name="cc" type=radio value="visacard"> </p>
<p> Widget size Big <input name="product" type=radio value="expensive">
Little <input name="product" type=radio value="cheap">
Ship by express courier <input name="express" type=checkbox> </p>
<p><input type=submit value="submit order"> </p>
Thank you for ordering an AWI widget, the best widget money can buy!
</form>
</body>
</html>
```

(a)

## Widget Order Form

Name	<input type="text"/>		
Street address	<input type="text"/>		
City	<input type="text"/>	State	<input type="text"/>
		Country	<input type="text"/>
Credit card #	<input type="text"/>	Expires	<input type="text"/>
		M/C	<input type="radio"/>
		Visa	<input type="radio"/>
Widget size	Big	<input type="radio"/>	Little
		<input type="radio"/>	Ship by express courier
			<input type="radio"/>
<input type="button" value="Submit order"/>			
Thank you for ordering an AWI widget, the best widget money can buy!			

(b)

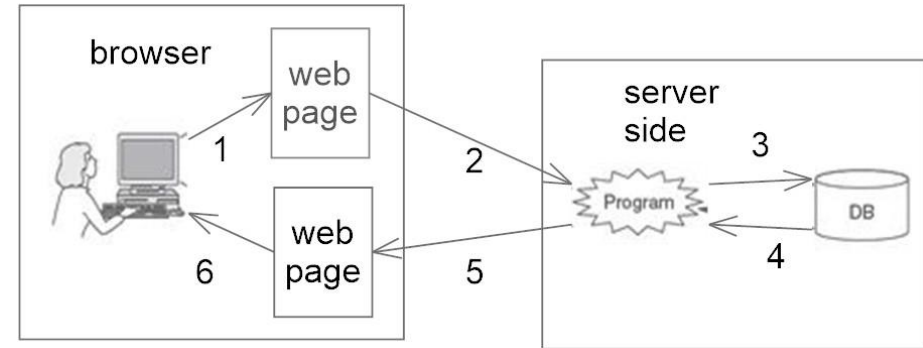
# Páginas dinámicas

- **Páginas dinámicas:** Son páginas web generadas por programas que se ejecutan en el servidor (posiblemente con una base de datos).
- **Tecnologías para producir páginas dinámicas**
- PHP, Java Server Pages , etc.
- Veremos un poquito de PHP.

# Páginas dinámicas

- **Pasos para generar páginas dinámicamente del lado del servidor:**

1. Un usuario llena un formulario y hace click en el botón de envío.
2. Se envía un mensaje al servidor web con el contenido del formulario.
  - Se proporciona el mensaje a un programa o una secuencia de comandos.
  - El programa procesa el mensaje.
3. El programa solicita información a un servidor de bases de datos.

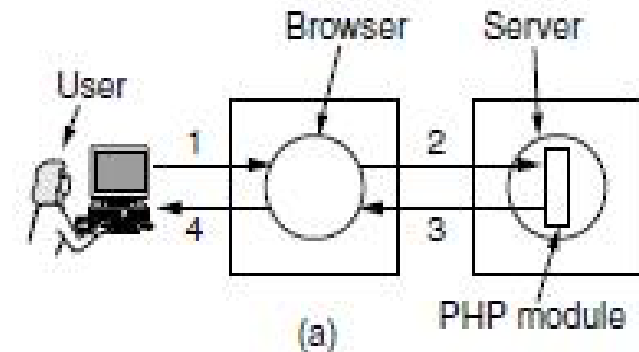


4. El servidor de bases de datos responde con la información requerida.
5. El programa genera una página HTML personalizada y la envía al cliente.
6. El browser muestra la página recibida al usuario.

# PHP

- **Enfoque PHP (Preprocesador de Hipertexto).**

- ❑ Se definen páginas dinámicas mediante la **inserción de comandos especiales dentro de páginas HTML**
- ❑ Para utilizar PHP el servidor web tiene que entender PHP.
  - El código PHP es interpretado por un servidor web.
  - PHP se diseñó para trabajar con el servidor web Apache.
  - PHP puede ser usado en la mayoría de los servidores web.



# PHP

- Algunas cosas que puede hacer PHP:
  - PHP puede generar contenido de página dinámica
  - PHP puede operar con archivos en el servidor.
  - PHP puede recolectar datos de formulario
  - PHP puede enviar y recibir receive cookies
  - PHP puede acceder a encabezados de pedido HTTP
  - PHP permite definir encabezados de respuesta HTTP
  - PHP permite acceder a base de datos
- PHP es gratuito y fácil de aprender y se ejecuta eficientemente.

# PHP

Un poquito de sintaxis fundamental de PHP

Construcción	Description
<?php ... ?>	PHP script Puede ir en cualquier lugar del documento
'\$' NAME	Declaración de variable Case sensitive
echo EXPR	Para mostrar datos en pantalla
//	comentarios
Define(name, value)	Definición de constantes
VARIABLE = EXPR	Igual que en C (+=, -=, *=, /=)
Include 'filename'	Toma el texto/código/markup en un archivo y lo copia en el archivo que usa la sentencia <i>include</i>

# PHP

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      $txt = "Hello world!";
      $x = 5;
      $y = 10.5;
      echo $txt;
      echo "<br>";
      echo $x;
      echo "<br>";
      echo $y;
    ?>
  </body>
</html>
```

```
Hello world!
5
10.5
```

# PHP

- **Acceso a campos de formularios:**

- `$_POST`: usado para recolectar datos de formulario luego de someter un formulario con método POST.
- P.ej: `$_POST['fname']` // recolecta valor del campo de nombre 'fname'
- `$_GET`: usado para recolectar datos de formulario luego de someter un formulario con método GET. Se usa como en el ítem anterior.



# PHP

Página web que  
obtiene input de  
formulario y llama un  
programa de  
servidor

```
<html><body>
  <form action="welcome.php" method="post">
    Name: <input type="text" name="name"><br>
    E-mail: <input type="text" name="email"><br>
    <input type="submit">
  </form>
</body> </html>
```

Programa PHP del  
servidor que crea  
una página web  
personalizada

```
<html><body>
  Welcome <php echo $_POST["name"]; ?><br>
  Your email address is: <?php echo $_POST["email"]; ?>
</body></html>
```

Página web resultante  
(para inputs "John" y  
"john.doe@example.com")

```
<html><body>
  Welcome John
  Your email address is john.doe@example.com
</body></html>
```

# PHP

- Tipos de datos de PHP
  - String: secuencia de caracteres entre comillas.
  - Integer: número entero entre - 2,147,483,648 y 2,147,483,647
  - Float: número con punto decimal o número en forma exponencial.
  - Boolean: valores booleanos TRUE y FALSE
  - Array: un arreglo almacena varios valores en una variable.
  - Object: PHP permite definir clases y objetos
- Operadores
  - de comparación (como en C), de asignación (como en C)
  - Para los distintos tipos de datos

# PHP

- **Acceso a información de encabezados HTTP:**

- ☐ `$_SERVER`: contiene información de encabezados, caminos y localización de scripts.
- ☐ Para acceder a encabezados poner como argumento alguna de las siguientes:
  - `HTTP_USER_AGENT`, `SERVER_ADDR`, `SERVER_NAME`, `SERVER_SOFTWARE`, `SERVER_PROTOCOL`, `REQUEST_METHOD`, `REQUEST_TIME`, `QUERY_STRING`, `HTTP_ACCEPT`, `HTTP_ACCEPT_CHARSET`, `HTTP_HOST`, etc.
- ☐ P.ej.: Para acceder al encabezado User-Agent: `$_SERVER['HTTP_USER_AGENT']`
- ☐ p.ej.: nombre del archivo del script ejecutándose: `$_SERVER['PHP_SELF']`
- ☐ P.ej.: URL completo de la página corriente: `$_SERVER['HTTP_REFERER']`
- ☐ P.ej.: el camino del script corriente: `$_SERVER['SCRIPT_NAME']`
- ☐ P.ej.: el nombre de dominio del host servidor: `$_SERVER['SCRIPT_NAME']`
- ☐ P.ej.: el encabezado Host del pedido actual: `$_SERVER['HTTP_HOST']`

# PHP

- ```
<!DOCTYPE html>
<html>
<body>

<?php
echo $_SERVER['PHP_SELF'];
echo "<br>";
echo $_SERVER['SERVER_NAME'];
echo "<br>";
echo $_SERVER['HTTP_HOST'];
echo "<br>";
echo $_SERVER['HTTP_REFERER'];
echo "<br>";
echo $_SERVER['HTTP_USER_AGENT'];
echo "<br>";
echo $_SERVER['SCRIPT_NAME'];
?>

</body>
</html>
```

```
/demo/demo_global_server.php
35.194.26.41
35.194.26.41
https://tryphp.w3schools.com/showphp.php?filename=demo
_global_server
Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/89.0.4389.82 Safari/537.36
/demo/demo_global_server.php
```

# PHP

- **Definición de encabezados de respuesta HTTP:**

- Hay que usar la función header()
- Se deben fijar encabezados antes de la etiqueta <html> aparezca.
- P.ej:

```
<?php header('Content-Type: text/plain');  
        header('Expires: Fri, 18 Jan 2002 05:30:00 GMT'); ?>  
<html>  
<body>  
...
```

# PHP

- **Definición de cookies:**

- ☐ Setcookie() define cookie para ser enviada junto con el resto de los encabezados HTTP.
- ☐ Esta función debe usarse antes de generar cualquier salida, o sea antes que la etiqueta <html>
- ☐ Un cookie se crea con la función  
setcookie(name, value, expire, path, domain, secure, httponly)
- ☐ <?php  
\$cookie\_name = "user";  
\$cookie\_value = "Alex Porter";  
setcookie(\$cookie\_name, \$cookie\_value, time() + (86400 \* 30), "/");  
?>

# PHP

- **Acceso al valor de una cookie:**

- ☐ \$\_COOKIE se usa para retornar el valor de una cookie.

- ☐ P.ej:

```
<?php
```

```
    echo '¡Hola ' . htmlspecialchars($_COOKIE["nombre"]). '!';
```

```
?>
```

- ☐ Asumiendo que la cookie "nombre" ha sido definida anteriormente, si el valor de esa cookie es “Juan” el resultado del ejemplo será :

- ☐ ¡Hola Juan!

- ☐ htmlspecialchars — Convierte caracteres especiales en entidades HTML

# PHP: uso de include

- En “footer.php” tenemos:
- ```
<?php
echo "<p>Copyright &copy; 1999-" .
date("Y") . " W3Schools.com</p>";
?>
```
- Tenemos el siguiente script:
- ```
<!DOCTYPE html>
<html>
<body>

<h1>Welcome to my home page!</h1>
<p>Some text.</p>
<p>Some more text.</p>
<?php include 'footer.php';?>

</body>
</html>
```

**Welcome to my home page!**

Some text.

Some more text.

Copyright © 1999-2021 W3Schools.com