



HTML: ESTRUCTURA Y SINTAXIS

DESARROLLO DE APLICACIONES WEB



PERO ANTES ... UN REPASO

- ¿Qué herramientas profesionales se usan para el desarrollo de sitios web?
- ¿Qué se necesita para configurar un servidor web local de pruebas?
- ¿Qué se necesita y cuáles son los costos para publicar en internet un sitio web? (Valores aproximados, debe incluir captura de precios de proveedores)

INTRODUCCIÓN

HTML



LANGUAGE HTML

- HTML es el lenguaje de marcado estándar para crear páginas web.



¿QUÉ ES EL HTML?



- HTML significa lenguaje de marcado de hipertexto
- HTML es el lenguaje de marcado estándar para crear páginas web
- HTML describe la estructura de una página web
- HTML consiste en una serie de elementos
- Los elementos HTML le dicen al navegador cómo mostrar el contenido
- Los elementos HTML etiquetan piezas de contenido como "este es un encabezado", "este es un párrafo", "este es un enlace", etc.

UN DOCUMENTO HTML SIMPLE

Ejemplo explicado:

- La declaración **<!DOCTYPE html>** define que este documento es un documento HTML5
- El elemento **<html>** es el elemento raíz de una página HTML.
- El elemento **<head>** contiene metainformación sobre la página HTML.
- El elemento **<title>** especifica un título para la página HTML (que se muestra en la barra de título del navegador o en la pestaña de la página)
- El elemento **<body>** define el cuerpo del documento y es un contenedor para todos los contenidos visibles, como encabezados, párrafos, imágenes, hipervínculos, tablas, listas, etc.
- El elemento **<h1>** define un encabezado grande
- El elemento **<p>** define un párrafo.

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

¿QUÉ ES UN ELEMENTO HTML?

Un elemento HTML se define mediante una etiqueta de inicio, algo de contenido y una etiqueta de finalización:

- `< tagname >` El contenido va aquí ... `< / tagname >`

El elemento HTML es todo, desde la etiqueta de inicio hasta la etiqueta de finalización:

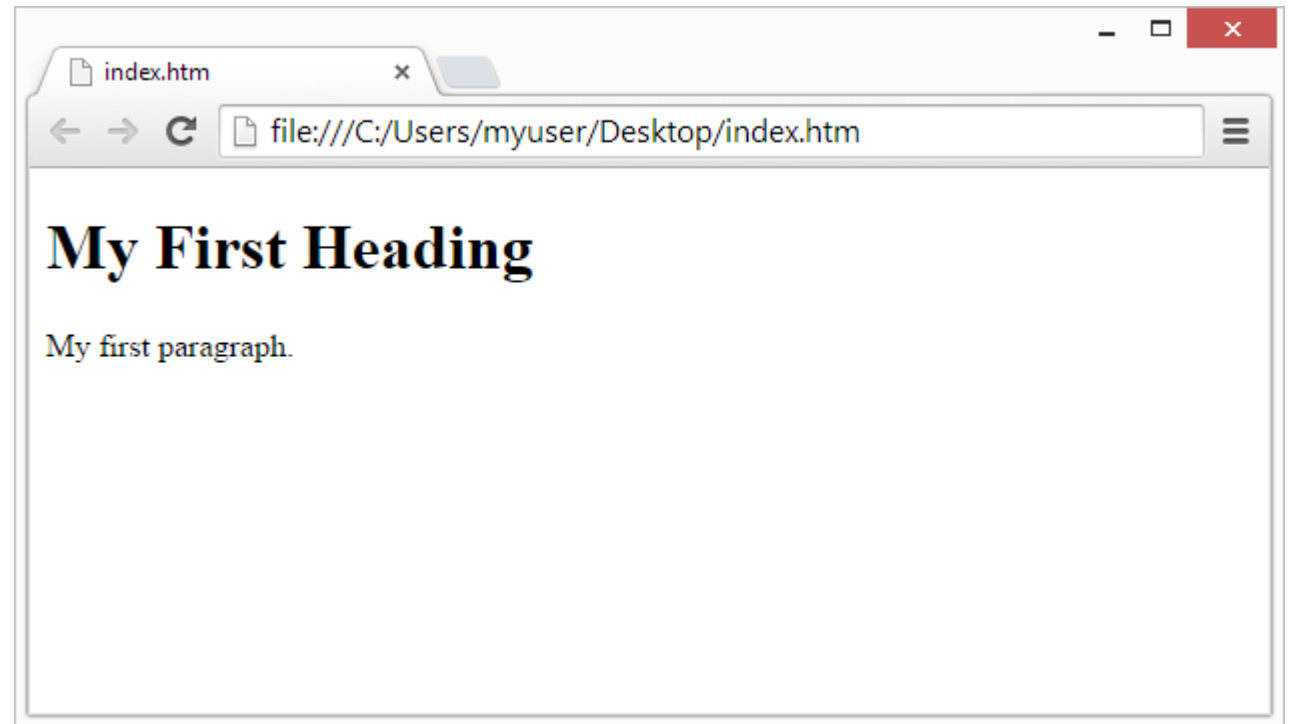
- `< h1 >` Mi primer encabezado `< / h1 >`
- `< p >` Mi primer párrafo. `< / p >`

Nota: Algunos elementos HTML no tienen contenido (como el elemento `
`). Estos elementos se llaman elementos vacíos. ¡Los elementos vacíos no tienen una etiqueta final!

Start tag	Element content	End tag
<code><h1></code>	My First Heading	<code></h1></code>
<code><p></code>	My first paragraph.	<code></p></code>
<code>
</code>	<i>none</i>	<i>none</i>

NAVEGADORES WEB

- El propósito de un navegador web (Chrome, Edge, Firefox, Safari) es leer documentos HTML y mostrarlos correctamente.
- Un navegador no muestra las etiquetas HTML, pero las usa para determinar cómo mostrar el documento:



ESTRUCTURA DE PÁGINA HTML



- **Nota:** Solo el contenido dentro de la sección **<body>** (el área blanca) se mostrará en un navegador.

EDITORES HTML

- Las páginas web se pueden crear y modificar utilizando editores HTML profesionales.
- Sin embargo, para aprender HTML, se recomienda un editor de texto simple como Notepad (PC) o TextEdit (Mac).
- Creemos que usar un editor de texto simple es una buena manera de aprender HTML.



PRIMERA PÁGINA WEB CON BLOC DE NOTAS / TEXTEDIT

PRÁCTICA HTML

DOCUMENTOS HTML

- Todos los documentos HTML deben comenzar con una declaración de tipo de documento: `<!DOCTYPE html>`.
- El documento HTML en sí comienza `<html>` y termina con `</html>`.
- La parte visible del documento HTML está entre `<body>` y `</body>`.

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

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

LA DECLARACIÓN <!DOCTYPE>

- La declaración <!DOCTYPE> representa el tipo de documento y ayuda a los navegadores a mostrar las páginas web correctamente.
- Solo debe aparecer una vez, en la parte superior de la página (antes de cualquier etiqueta HTML).
- La declaración <!DOCTYPE> no distingue entre mayúsculas y minúsculas.
- La declaración <!DOCTYPE> para HTML5 es:

```
<!DOCTYPE html>
```

ENCABEZADOS HTML

- Los encabezados HTML se definen con las etiquetas `<h1>` a `<h6>`.
- `<h1>` define el encabezado más importante.
- `<h6>` define el encabezado menos importante.

```
<h1>This is heading 1</h1>
```

```
<h2>This is heading 2</h2>
```

```
<h3>This is heading 3</h3>
```

PÁRRAFOS HTML

- Los párrafos HTML se definen con la etiqueta `<p>` :

```
<p>This is a paragraph.</p>
```

```
<p>This is another paragraph.</p>
```

ENLACES HTML

- Los enlaces HTML se definen con la etiqueta `<a>`:

```
<a href="http://www.ug.edu.ec">This is a link</a>
```


IMÁGENES HTML

- Las imágenes HTML se definen con la etiqueta .
- El archivo fuente (src), el texto alternativo (alt) width, y height se proporcionan como atributos:

```

```

ESTILOS HTML

- El atributo **style** HTML se usa para agregar estilos a un elemento, como color, fuente, tamaño y más.
- La configuración del estilo de un elemento HTML se puede hacer con el atributo style
- El atributo style HTML tiene la siguiente sintaxis:

```
<tagname style="property:value;">
```

- La propiedad es una propiedad CSS. El valor es un valor CSS.

Ejemplo:

- Establezca el color de fondo para una página en azul polvo:

```
<!DOCTYPE html>
<html>
<body style="background-color:powderblue;">

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

This is a heading

This is a paragraph.

ELEMENTOS DE FORMATO HTML

Los elementos de formato fueron diseñados para mostrar tipos especiales de texto:

- **** - Texto en negrita
- **** - Texto importante
- **<i>** - Texto en cursiva
- **** - Texto enfatizado
- **<mark>** - Texto marcado
- **<small>** - Texto más pequeño
- **** - Texto eliminado
- **<ins>** - Texto insertado
- **<sub>** - Texto del subíndice
- **<sup>** - Texto superíndice

COMENTARIOS HTML

- Los comentarios HTML no se muestran en el navegador, pero pueden ayudar a documentar su código fuente HTML.

```
<!-- Write your comments here -->
```

- Observe que hay un signo de exclamación (!) En la etiqueta de inicio, pero no en la etiqueta de finalización.
- El navegador no muestra comentarios, pero pueden ayudar a documentar su código fuente HTML.

- Con comentarios, puede colocar notificaciones y recordatorios en su código HTML:

```
<!-- This is a comment -->
```

```
<p>This is a paragraph.</p>
```

```
<!-- Remember to add more information here -->
```

COLORES HTML

Nombres de colores

- En HTML, se puede especificar un color utilizando un nombre de color:



COLOR DE FONDO

- Puede establecer el color de fondo para los elementos HTML:

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>  
<p style="background-color:Tomato;">Lorem ipsum...</p>
```

Hello World

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

COLOR DE TEXTO

- Puede configurar el color del texto:

```
<h1 style="color:Tomato;">Hello World</h1>  
<p style="color:DodgerBlue;">Lorem ipsum...</p>  
<p style="color:MediumSeaGreen;">Ut wisi enim...</p>
```

Hello World

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

COLOR DE BORDE

- Puede establecer el color de los bordes:

```
<h1 style="border:2px solid Tomato;">Hello World</h1>
```

```
<h1 style="border:2px solid DodgerBlue;">Hello World</h1>
```

```
<h1 style="border:2px solid Violet;">Hello World</h1>
```

Hello World

Hello World

Hello World

HTML Y ESTILOS CSS



¿Qué es CSS?

- Las **hojas de estilo en cascada** (CSS) se utilizan para formatear el diseño de una página web.
- Con CSS, puede **controlar** el color, la fuente, el tamaño del texto, el espacio entre los elementos, cómo se colocan y distribuyen los elementos, qué imágenes de fondo o colores de fondo se utilizarán, diferentes pantallas para diferentes dispositivos y tamaños de pantalla, y mucho ¡más!
- CSS ahorra mucho trabajo.
- Puede controlar el diseño de varias páginas web a la vez.

USANDO CSS

CSS se puede agregar a documentos HTML de 3 maneras:

- En línea : mediante el uso del atributo **style** dentro de elementos HTML
- Interno : mediante el uso de un elemento **<style>** en la sección **<head>**
- Externo : mediante el uso de un elemento **<link>** para vincular a un archivo CSS externo

La forma más común de agregar CSS es mantener los estilos en archivos CSS externos.



CSS EN LÍNEA

- Se utiliza un CSS en línea para aplicar un estilo único a un único elemento HTML.
- Un CSS en línea utiliza el atributo style de un elemento HTML.
- El siguiente ejemplo establece el color del texto del elemento <h1> en azul y el color del texto del elemento <p> en rojo:

```
<h1 style="color:blue;">A Blue Heading</h1>
```

```
<p style="color:red;">A red paragraph.</p>
```

A Blue Heading

A red paragraph.

CSS INTERNO

- Se usa un CSS interno para definir un estilo para una sola página HTML.
- Un CSS interno se define en la sección <head> de una página HTML, dentro de un elemento <style>.
- El siguiente ejemplo establece el color del texto de TODOS los elementos <h1> (en esa página) en azul y el color del texto de TODOS los elementos <p> en rojo.
- Además, la página se mostrará con un color de fondo "azul polvo":

```
<!DOCTYPE html>
<html>
<head>
<style>
body {background-color: powderblue;}
h1   {color: blue;}
p    {color: red;}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

This is a heading

This is a paragraph.

CSS EXTERNO

- Se utiliza una hoja de estilo externa para definir el estilo de muchas páginas HTML.
- Para usar una hoja de estilo externa, agregue un enlace en la sección <head> de cada página HTML.
- La hoja de estilo externa se puede escribir en cualquier editor de texto.
- El archivo no debe contener ningún código HTML y debe guardarse con una extensión **.css**

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

```
body {
  background-color: powderblue;
}
h1 {
  color: blue;
}
p {
  color: red;
}
```

TABLAS HTML

- Las tablas HTML permiten a los desarrolladores web organizar los datos en filas y columnas.

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK
Laughing Bacchus Winecellars	Yoshi Tannamuri	Canada
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy

DEFINIR UNA TABLA HTML

- La etiqueta <table> define una tabla HTML.
- Cada fila de la tabla se define con una etiqueta <tr>.
- Cada encabezado de tabla se define con una etiqueta <th>.
- Cada tabla de datos / celda se define con una etiqueta <td>.
- Por defecto, el texto en los elementos <th> está en negrita y centrado.
- Por defecto, el texto en los elementos <td> es regular y está alineado a la izquierda.
- Los elementos <td> son los contenedores de datos de la tabla. Pueden contener todo tipo de elementos HTML; texto, imágenes, listas, otras tablas, etc.

```
<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
```

TABLA HTML: CELDA QUE ABARCA MUCHAS COLUMNAS

```
<table style="width:100%">
  <tr>
    <th>Name</th>
    <th colspan="2">Telephone</th>
  </tr>
  <tr>
    <td>Bill Gates</td>
    <td>55577854</td>
    <td>55577855</td>
  </tr>
</table>
```

- Para hacer que una celda abarque más de una columna, use el atributo **colspan**:

Name	Telephone	
Bill Gates	55577854	55577855

TABLA HTML: CELDA QUE ABARCA MUCHAS FILAS

- Para hacer que una celda abarque más de una fila, use el atributo **rowspan**:

Name:	Bill Gates
Telephone:	55577854
	55577855

```
<table style="width:100%">
  <tr>
    <th>Name:</th>
    <td>Bill Gates</td>
  </tr>
  <tr>
    <th rowspan="2">Telephone:</th>
    <td>55577854</td>
  </tr>
  <tr>
    <td>55577855</td>
  </tr>
</table>
```

TABLA HTML: AGREGAR UN TÍTULO

```
<table style="width:100%">
  <caption>Monthly savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>February</td>
    <td>$50</td>
  </tr>
</table>
```

- Para agregar un título a una tabla, use la etiqueta `<caption>`.
- La etiqueta `<caption>` debe insertarse inmediatamente después de la etiqueta `<table>`.

Month	Savings
January	\$100
February	\$50

LISTAS HTML

- Las listas HTML permiten a los desarrolladores web agrupar un conjunto de elementos relacionados en listas.

Una lista HTML desordenada:

- Artículo
- Artículo
- Artículo
- Artículo

Una lista HTML ordenada:

1. Primer elemento
2. Segundo elemento
3. Tercer elemento
4. Cuarto elemento

LISTA HTML DESORDENADA

- Una lista desordenada comienza con la etiqueta ``. Cada elemento de la lista comienza con la etiqueta ``.
- Los elementos de la lista se marcarán con viñetas (pequeños círculos negros) de forma predeterminada:

```
<ul>  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
</ul>
```

- Coffee
- Tea
- Milk

LISTA HTML ORDENADA

```
<ol>  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
</ol>
```

1. Coffee
2. Tea
3. Milk

- Una lista ordenada comienza con la etiqueta . Cada elemento de la lista comienza con la etiqueta .
- Los elementos de la lista se marcarán con números de forma predeterminada.

LISTAS DE DESCRIPCIÓN HTML

- HTML también admite listas de descripción.
- Una lista de descripción es una lista de términos, con una descripción de cada término.
- La etiqueta <dl> define la lista de descripción,
- la etiqueta <dt> define el término (nombre) y
- la etiqueta <dd> describe cada término:

```
<dl>  
  <dt>Coffee</dt>  
  <dd>- black hot drink</dd>  
  <dt>Milk</dt>  
  <dd>- white cold drink</dd>  
</dl>
```

Coffee
- black hot drink
Milk
- white cold drink



CREAR PÁGINA(S) WEB HTML5

CONSIDERANDO SU ESTRUCTURA ESENCIAL DE ACUERDO AL ENUNCIADO PROPUESTO.



HTML: ESTRUCTURA Y SINTAXIS

DESARROLLO DE APLICACIONES WEB





BLOQUES HTML Y ELEMENTOS EN LÍNEA

HTML

ELEMENTOS A NIVEL DE BLOQUE

- Un elemento de nivel de bloque siempre comienza en una nueva línea y ocupa todo el ancho disponible (se extiende hacia la izquierda y la derecha tanto como sea posible).
- El elemento `<div>` es un elemento de nivel de bloque.

```
<div>Hello World</div>
```

Hello World

El elemento DIV es un elemento de bloque, y siempre comenzará en una nueva línea y ocupará todo el ancho disponible (se extiende hacia la izquierda y la derecha tanto como sea posible).

ELEMENTOS A NIVEL DE BLOQUE EN HTML

<code><address></code>	<code><article></code>	<code><aside></code>	<code><blockquote></code>	<code><canvas></code>	<code><dd></code>	<code><div></code>
<code><dl></code>	<code><dt></code>	<code><fieldset></code>	<code><figcaption></code>	<code><figure></code>	<code><footer></code>	<code><form></code>
<code><h1>-<h6></code>	<code><header></code>	<code><hr></code>	<code></code>	<code><main></code>	<code><nav></code>	<code><noscript></code>
<code></code>	<code><p></code>	<code><pre></code>	<code><section></code>	<code><table></code>	<code><tfoot></code>	<code></code>
<code><video></code>						

ELEMENTOS EN LÍNEA

- Un elemento en línea no comienza en una nueva línea y solo ocupa el ancho que sea necesario.

Este es un elemento dentro de un párrafo.

- Un elemento en línea no puede contener un elemento de nivel de bloque.

```
<span>Hello World</span>
```

This is an inline span Hello World element inside a paragraph.

El elemento SPAN es un elemento en línea, y no comenzará en una nueva línea y solo ocupará todo el ancho que sea necesario.

ELEMENTOS EN LÍNEA EN HTML

<code><a></code>	<code><abbr></code>	<code><acronym></code>	<code></code>	<code><bdo></code>	<code><big></code>	<code>
</code>
<code><button></code>	<code><cite></code>	<code><code></code>	<code><dfn></code>	<code></code>	<code><i></code>	<code></code>
<code><input></code>	<code><kbd></code>	<code><label></code>	<code><map></code>	<code><object></code>	<code><output></code>	<code><q></code>
<code><samp></code>	<code><script></code>	<code><select></code>	<code><small></code>	<code></code>	<code></code>	<code><sub></code>
<code><sup></code>	<code><textarea></code>	<code><time></code>	<code><tt></code>	<code><var></code>		

EL ELEMENTO <DIV>

- El elemento <div> se usa a menudo como contenedor para otros elementos HTML.
- El elemento <div> tiene atributos no son necesarios, pero style, class y id son comunes.
- Cuando se usa junto con CSS, el elemento <div> se puede usar para diseñar bloques de contenido:

```
<div style="background-color:black;color:white;padding:20px;">  
  <h2>London</h2>  
  <p>London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.</p>  
</div>
```

London

London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.

Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.

EL ELEMENTO

- El elemento es un contenedor en línea que se usa para marcar una parte de un texto o una parte de un documento.
- El elemento tiene atributos no son necesarios, pero style, class y id son comunes.
- Cuando se usa junto con CSS, el elemento se puede usar para dar estilo a partes del texto:

The span element

My mother has **blue** eyes and my father has **dark green** eyes.

```
<p>My mother has <span style="color:blue;font-weight:bold">blue</span> eyes and my  
father has <span style="color:darkolivegreen;font-weight:bold">dark green</span>  
eyes.</p>
```

APUNTES

- Hay dos valores de visualización: bloque y en línea
- Un elemento de nivel de bloque siempre comienza en una nueva línea y ocupa todo el ancho disponible
- Un elemento en línea no comienza en una nueva línea y solo ocupa tanto ancho como sea necesario
- El elemento `<div>` es un nivel de bloque y a menudo se usa como contenedor para otros elementos HTML
- El elemento `` es un contenedor en línea utilizado para marcar una parte de un texto o una parte de un documento

EL ATRIBUTO CLASS

- El atributo HTML **class** se usa para especificar una clase para un elemento HTML.
- Varios elementos HTML pueden compartir la misma clase.
- Los atributos **class** se usa a menudo para señalar un nombre de clase en una hoja de estilo.
- También puede ser utilizado por un JavaScript para acceder y manipular elementos con el nombre de clase específico.
- El atributo **class** se puede usar en cualquier elemento HTML.
- El nombre de la clase distingue entre mayúsculas y minúsculas.

LSI. ANGEL VELOZ RODRIGUEZ, MGS.

```
<!DOCTYPE html>
<html>
<head>
<style>
.note {
  font-size: 120%;
  color: red;
}
</style>
</head>
<body>

<h1>My <span class="note">Important</span> Heading</h1>
<p>This is some <span class="note">important</span> text.</p>

</body>
</html>
```

My **Important** Heading

This is some **important** text.

LA SINTAXIS PARA CLASS

- Para crear una clase; escriba un carácter de punto (.), seguido de un nombre de clase. Luego, defina las propiedades CSS entre llaves {}

The class Attribute

Use CSS to style elements with the class name "city":

London

London is the capital of England.

Paris

Paris is the capital of France.

Tokyo

Tokyo is the capital of Japan.

```
<!DOCTYPE html>
<html>
<head>
<style>
.city {
    background-color: tomato;
    color: white;
    padding: 10px;
}
</style>
</head>
<body>

<h2 class="city">London</h2>
<p>London is the capital of England.</p>

<h2 class="city">Paris</h2>
<p>Paris is the capital of France.</p>

<h2 class="city">Tokyo</h2>
<p>Tokyo is the capital of Japan.</p>

</body>
</html>
```

CLASES MÚLTIPLES

- Los elementos HTML pueden pertenecer a más de una clase.
- Para definir varias clases, separe los nombres de las clases con un espacio, por ejemplo, `<div class = "city main">`.
- El elemento tendrá un estilo de acuerdo con todas las clases especificadas.
- En el siguiente ejemplo, el primer `<h2>` elemento pertenece tanto a la `city` clase como a la `main` clase, y obtendrá los estilos CSS de ambas clases:

```
<style>
.city {
  background-color: tomato;
  color: white;
  padding: 10px;
}

.main {
  text-align: center;
}
</style>
```

```
<h2 class="city main">London</h2>
<h2 class="city">Paris</h2>
<h2 class="city">Tokyo</h2>
```

Multiples Clases

London

Paris

Tokyo

DIFERENTES ELEMENTOS PUEDEN COMPARTIR LA MISMA CLASE

- Diferentes elementos HTML pueden apuntar al mismo nombre de clase.
- En el siguiente ejemplo, tanto `<h2>` y `<p>` puntos para la clase "ciudad" y compartirán el mismo estilo:

```
<h2 class="city">Paris</h2>
<p class="city">Paris is the capital of France</p>

<style>
.city {
  background-color: tomato;
  color: white;
  padding: 10px;
}
</style>
```

Diferentes elementos pueden compartir la misma clase

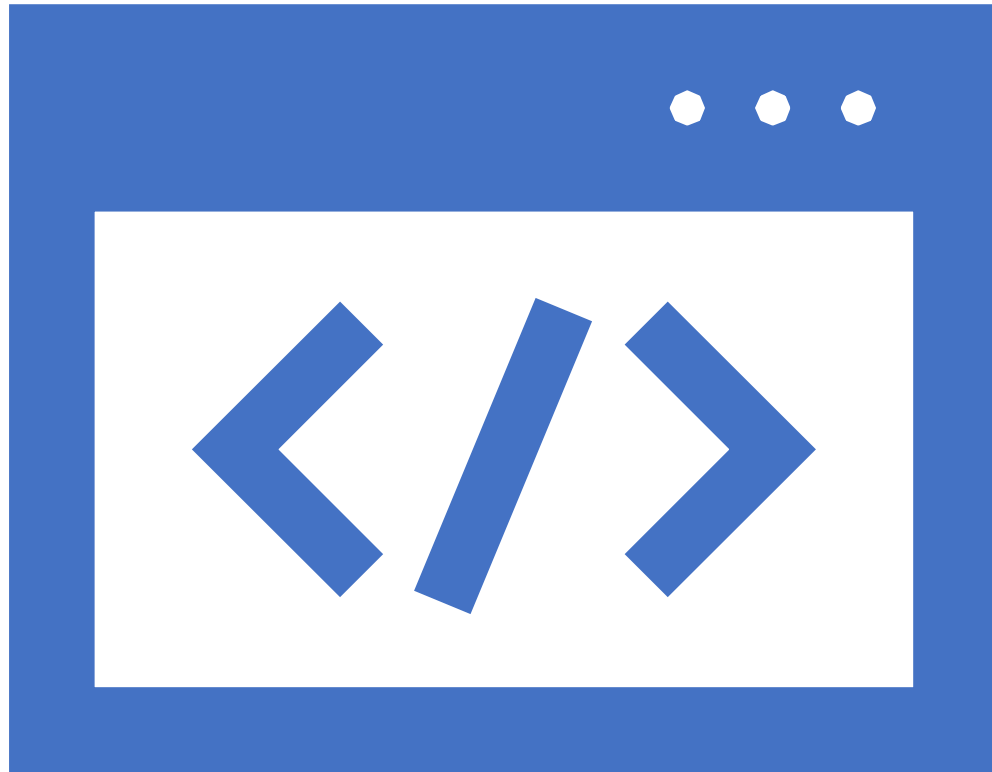
Paris

Paris is the capital of France.



CREAR PÁGINA(S) WEB HTML5

CONSIDERANDO SU ESTRUCTURA ESENCIAL DE ACUERDO AL ENUNCIADO PROPUESTO.



PARTES BÁSICAS DE UN DOCUMENTO HTML

DESARROLLO DE
APLICACIONES WEB

header (encabezado):

- Normalmente formado por una gran franja que cruza la parte superior de la página con un título de encabezamiento y/o un logo. Esta parte suele permanecer invariable mientras navegamos entre las páginas de un mismo sitio web.

navigation bar (barra/menú de navegación):

- Son los enlaces a las secciones principales del sitio web.
- Normalmente está formada por un menú con botones, enlaces o pestañas. Al igual que el encabezado, este contenido suele permanecer invariable en las diferentes páginas del sitio.

ELEMENTOS
COMUNES DE UN
DOCUMENTO
HTML

main content (contenido principal):

- Es la parte ancha central de nuestra página y contiene la mayor parte del contenido particular de una página web concreta; por ejemplo, el video que quieres ver, la narración que quieres leer, el mapa que quieres consultar, los titulares de las noticias, etc.
- Esta es la parte que puede variar mucho de una página a otra de nuestro sitio web.

sidebar (menú/barra lateral):

- Suele incluir algún tipo de información adicional, enlaces, citas, publicidad, etc.
- Normalmente está relacionado con el contenido principal de la página (por ejemplo, en una página de noticias, la barra lateral podría contener la biografía del autor o enlaces a artículos relacionados), pero en otras ocasiones encontraremos elementos recurrentes como un menú de navegación secundario.

footer (pie de página):

- Es la parte inferior de la página, que generalmente contiene la letra pequeña, el copyright o la información de contacto.
- Es el sitio donde se coloca la información común (al igual que en el encabezado), pero esta información no suele ser tan importante o es secundaria con respecto a la página en sí misma.
- El pie también se suele usar para propósitos SEO, e incluye enlaces de acceso rápido al contenido más popular.

...ELEMENTOS COMUNES DE UN DOCUMENTO HTML

Encabezado de Artículo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec a diam lectus. Set sit amet ipsum mauris. Maecenas congue ligula as quam viverra nec consectetur ant hendrerit. Donec et mollis dolor. Praesent et diam eget libero egestas mattis sit amet vitae augue. Nam tincidunt congue enim, ut porta lorem lacinia consectetur.

subsección

Donec ut librero sed accu vehicula ultricies a non tortor. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean ut gravida lorem. Ut turpis felis, pulvinar a semper sed, adipiscing id dolor.

Pelientesque auctor nisi id magna consequat sagittis. Curabitur dapibus, enim sit amet elit pharetra tincidunt feugiat nist imperdiet. Ut convallis libero in urna ultrices accumsan. Donec sed odio eros.

Otra subsección






Donec viverra mi quis quam pulvinar at malesuada arcu rhoncus. Cum sodis natoque penatibus et manis dis parturient montes, nascetur ridiculus mus. In rutrum accumsan ultricies. Mauris vitae nisi at sem facilis semper ac in est.

Vivamus fermentum semper porta. Nunc diam velit, adipiscing ut tristique vitae sagittis vel odio. Maecenas convallis ullamcorper ultricies. Curabitur ornare, ligula semper consectetur sagittis, nisi diam iaculis velit, is fringille sem nunc vet mi.

Relacionados

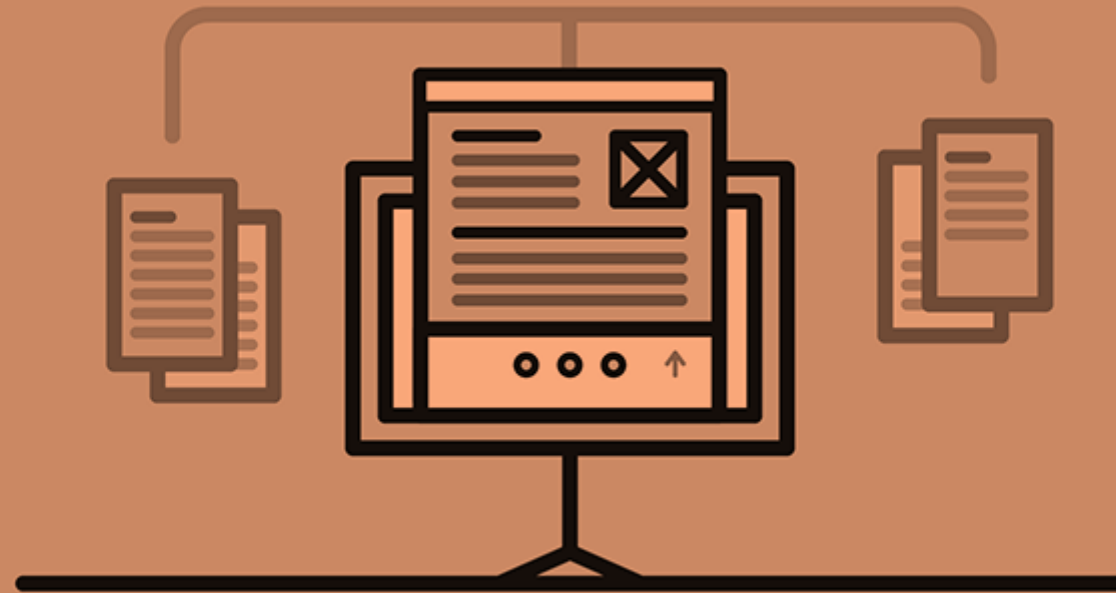
- [Me gusta estar al lado del mar.](#)
- [Aunque en el oriente de Ecuador.](#)
- [Nunca deja de llover](#)
- [Bueno...](#)

HTML PARA DAR ESTRUCTURA AL CONTENIDO

-  **header** (encabezado): `<header>`.
-  **navigation bar** (menú de navegación): `<nav>`.
-  **main content** (contenido principal): `<main>`, con varias subsecciones (además de la barra lateral) representadas por los elementos `<article>`, `<section>`, y `<div>`.
-  **sidebar** (barra lateral): `<aside>`.
-  **footer** (pie de página): `<footer>`.

PRÁCTICA

-
- Con base en la imagen de la diapositiva 4, cree una página web HTML5 que incluya hojas de estilo CSS y sus principales características propuestas en la imagen antes mencionada.



MAQUETACIÓN WEB

DESARROLLO DE
APLICACIONES WEB

DISEÑO WEB

La primera impresión es la que cuenta, es imprescindible contar con un sitio web con un diseño que impacte.

- Todo el mundo tiene al alcance de un clic a sus clientes.

El Diseño web es aquella actividad consistente en planificar, diseñar, mantener y crear páginas web

Un buen diseño web debe:

- Contar con una presentación clara y organizada
- Combinar muy bien los colores, utilizando en cada caso los más apropiados
- Tener un contenido ameno y agradable.
- Por otro lado también es importante que la página web cargue rápido, sea de fácil manejo y se adapte a las necesidades de los usuarios.

El diseño del sitio web debe hacer llegar a los clientes potenciales o usuarios el mensaje que una marca quiere transmitir.

TIPOS DE DISEÑO WEB



Diseño Web Fluido

■ Diseño web líquido o fluido

- Utiliza medidas relativas o porcentajes, en vez de píxeles, de esta forma los elementos se adaptan según el ancho de la pantalla.
- Esto pudiera provocar que las imágenes se estiren en pantallas grandes, que los textos sean difíciles de leer en dispositivos pequeños o que aparezcan incómodos espacios laterales rodeando toda la página del sitio.
- Este tipo de técnica es la menos efectiva y la menos recomendable.

TIPOS DE DISEÑO WEB



Diseño Web Adaptativo

- **Diseño web adaptativo, adaptable o adaptive**
 - Este tipo de diseño utiliza plantillas estáticas para tamaños de pantalla preestablecidos, basadas en puntos de ruptura, de forma que al alcanzar cierto límite de tamaño en la pantalla, el diseño cambia.
 - Esta técnica ofrece bajo nivel de usabilidad pues si el diseño cambia, es necesario actualizar todas las plantillas. Además, los contenidos se pueden seguir viendo reducidos si el acceso al sitio web es desde un móvil.

TIPOS DE DISEÑO WEB

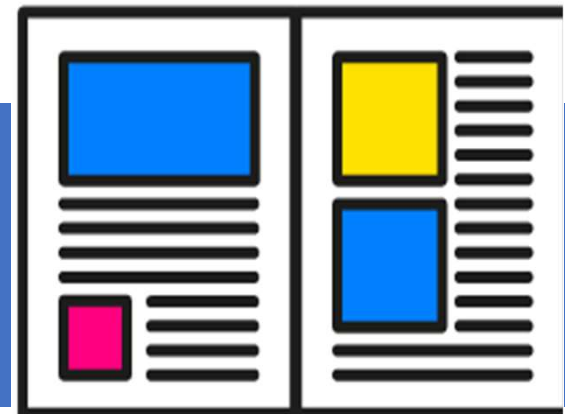


Diseño Web Responsivo

- **Diseño web responsivo o Responsive web design**
 - Con esta estrategia, tanto el diseño como el contenido se adaptan a cada pantalla, proporcionando una experiencia de usuario similar en distintos formatos o resoluciones.
 - Los contenidos se muestran en bloques que se reorganizan según las características del dispositivo y el navegador en uso. O sea, la información se mantiene igual pero la diagramación se optimiza.
 - Esta es la técnica para diseño web más óptima y eficiente.

- Se conoce como maquetación al oficio de diseño editorial que se encarga de organizar en un espacio, escrito, visual y en algunos casos contenido audiovisual en medios impresos y electrónicos, como libros, periódicos y revistas.
- La idea principal de maquetación de una web consiste en la distribución de los elementos de una página, es decir, textos, imágenes, enlaces y gráficos distribuidos de manera ordenada.
- Quien realiza esta actividad de manera profesional es un diseñador gráfico.
- Maquetar una web implica dar un formato determinado a todos los elementos de una página.

MAQUETACIÓN



■ Objetivo y Ventajas

- Mantener separado el contenido de la página de la presentación. De este modo haces más fácil el mantenimiento y cambios al contenido y diseño que tengas que hacer en el futuro.
- Reducir el tiempo de desarrollo y el tiempo que el usuario debe esperar a que se cargue completamente el sitio (Ventaja).

■ Proceso

- El proceso de maquetación web es la transformación del diseño de un producto web en un conjunto de archivos (html, css, js) capaces de ser reproducidos por los navegadores web.
- Es el proceso en el que el prototipo gráfico también denominado «layout» (con los requisitos estructurales y estéticos definidos y aprobados en un análisis inicial) pasa a transformarse en código html, css y js (estándares web) para que los navegadores puedan interpretarlo correctamente.

MAQUETACIÓN: PLANIFICACIÓN Y PROCESO



Estructurar una retícula (visible o invisible) capaz de mostrar los contenidos de la web:

A la hora de plantear una maquetación web debemos fijar unas zonas de trabajo que soporten la estructura de contenidos de nuestro sitio web y generar los recursos necesarios que faciliten la futura carga de contenidos.



Separar la presentación del contenido

Es muy importante, para facilitar tanto el proceso creativo como el futuro mantenimiento del sitio, separar el contenido y las funcionalidades de la web de los archivos que definen las propiedades gráficas (css).



Utilizar estándares en desarrollo de la maqueta

Para una correcta representación de un documento web se recomienda el uso de los estándares web (conjunto de recomendaciones lideradas por la W3C) que nos permiten mostrar la información de forma universal y robusta y poder acceder al mayor número de usuarios independientemente del dispositivo o la tecnología que usen.

- Del total de los encuestados, un 90% contestó que era muy importante tener esta aptitud para poder desempeñarse como desarrollador web.
- Esto puede deberse a que la maquetación de un sitio web se entiende como la forma de lograr que el diseño deseado para dicho sitio sea exitoso, cobre vida.
- Por ello, los diseñadores y los maquetadores suelen trabajar en conjunto (cuando una misma persona no realiza ambas funciones).

<https://blog.educacionit.com/2019/12/06/la-importancia-de-saber-maquetar-para-los-desarrolladores-web-conoce-los-resultados-de-nuestra-encuesta/>

LA IMPORTANCIA DE SABER MAQUETAR PARA
LOS DESARROLLADORES WEB



DESARROLLO DE APLICACIONES WEB

Programación del lado del cliente

JAVASCRIPT: Introducción

- ▶ JavaScript es el lenguaje de programación ligero más popular en Internet.
- ▶ JavaScript es un lenguaje interpretado (las secuencias de comandos se ejecutan sin compilación preliminar).
- ▶ Usualmente está embebido directamente en las páginas html a través de la etiqueta <script> de forma interna o externa.

- ▶ **Interna:**

```
<script type="text/javascript">  
    document.write("<p>Mi primer JavaScript</p>");  
</script>
```

- ▶ **Externa:**

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

Archivo con extensión .js



▶ http://www.w3schools.com/js/js_intro.asp

JAVASCRIPT: Qué puede hacer?

- ▶ **JavaScript puede reaccionar a eventos** - Un JavaScript puede ser configurado para ejecutarse cuando algo sucede, como cuando una página ha terminado de cargar o cuando un usuario da click en un elemento html.
- ▶ **JavaScript puede manipular elementos HTML** - JavaScript puede leer y modificar el contenido de un elemento HTML a través del uso de DOM.
- ▶ **JavaScript se puede utilizar para validar datos** - un JavaScript se puede utilizar para validar formularios de entrada de datos.
- ▶ **JavaScript puede ser utilizado para detectar el browser del visitante** - y dependiendo del browser cargar otra página diseñada específicamente para ese navegador.
- ▶ **JavaScript puede ser usado para crear cookies** - JavaScript puede ser usado para almacenar y recuperar información en la computadora del visitante.



JAVASCRIPT: Sintaxis

- ▶ JavaScript puede ser colocado dentro de las secciones `<head>` o `<body>` de una página html.

```
<html>
  <body><h1>My Web Page</h1>
    <p id="demo">Un párrafo</p>
    <script type="text/javascript">
      document.getElementById("demo").innerHTML="Mi primer
      JavaScript";
    </script>
  </body>
</html>
```

- ▶ **Nota:** Colocar el código javascript debajo de la página asegura que éste no se ejecute antes de que el elemento `<p>` sea creado.



JAVASCRIPT: Sintaxis (Funciones y eventos)

- ▶ El ejemplo anterior de JavaScript es ejecutado cuando la página carga.
- ▶ Sin embargo, algunas veces nosotros necesitamos ejecutar código JavaScript cuando ocurre un evento, por ejemplo cuando un usuario hace click sobre un botón.
- ▶ Entonces colocamos el script dentro de una **function**.
- ▶ Normalmente las funciones son usados en combinación con **eventos**.

```
<html>
  <head>
    <script type="text/javascript">
      function myFunction() {
        document.getElementById("demo").innerHTML="Mi primera función Javascript";
      }
    </script>
  </head>
  <body>
    <h1>Mi página web</h1>
    <p id="demo">Un párrafo</p>
    <button type="button" onclick="myFunction()">Try it</button>
  </body>
</html>
```

Llama a la función cuando ocurre el evento click



▶ Revisar lista de eventos en: https://www.w3schools.com/jsref/dom_obj_event.asp

JavaScript Display Possibilities

JavaScript can mostrar datos en diferentes maneras:

Escribiendo dentro de un elemento HTML, utilizando `innerHTML`.

Ej. `document.getElementById("demo").innerHTML = 5 + 6;`

Escribiendo como salida del documento HTML, utilizando `document.write()`.

Ej. `document.write(5 + 6);`

Escribiendo una caja de alerta, utilizando `window.alert()` o `alert()`.

Ej. `alert(5+ 6);`

Escribiendo en la consola del navegador, utilizando `console.log()`.

`console.log(5 + 6);`

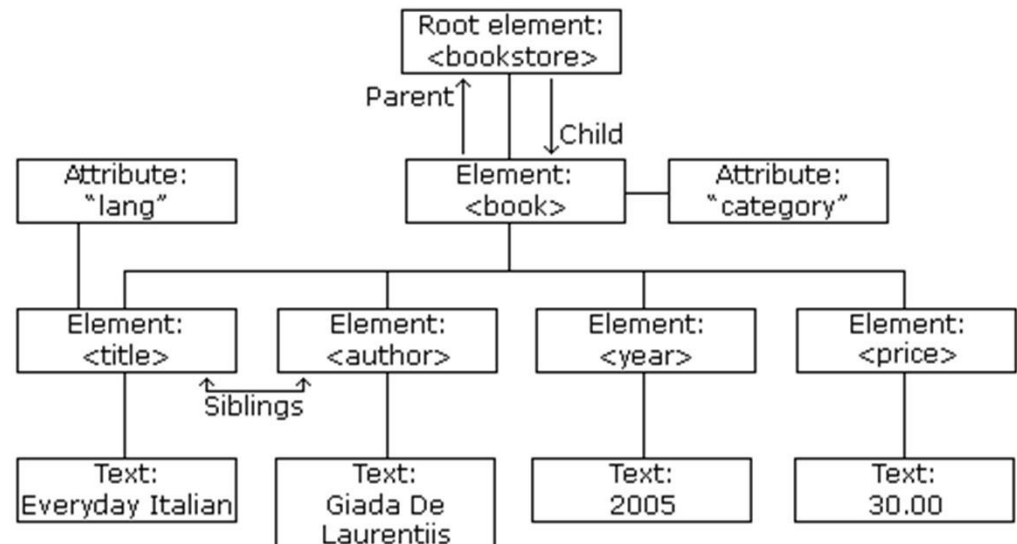


DOM : DOCUMENT OBJECT MODEL

- ❑ Estándar que define un API para la representación de documentos HTML y XML.
- ❑ Representa estos documentos como árboles de nodos. En DOM todo es un nodo.
- ❖ Es una interfaz de programación de aplicaciones para acceder y manipular dinámicamente contenido, estructura y estilo en documentos utilizando lenguajes como ECMAScript (JavaScript).

Ejemplo XML DOM

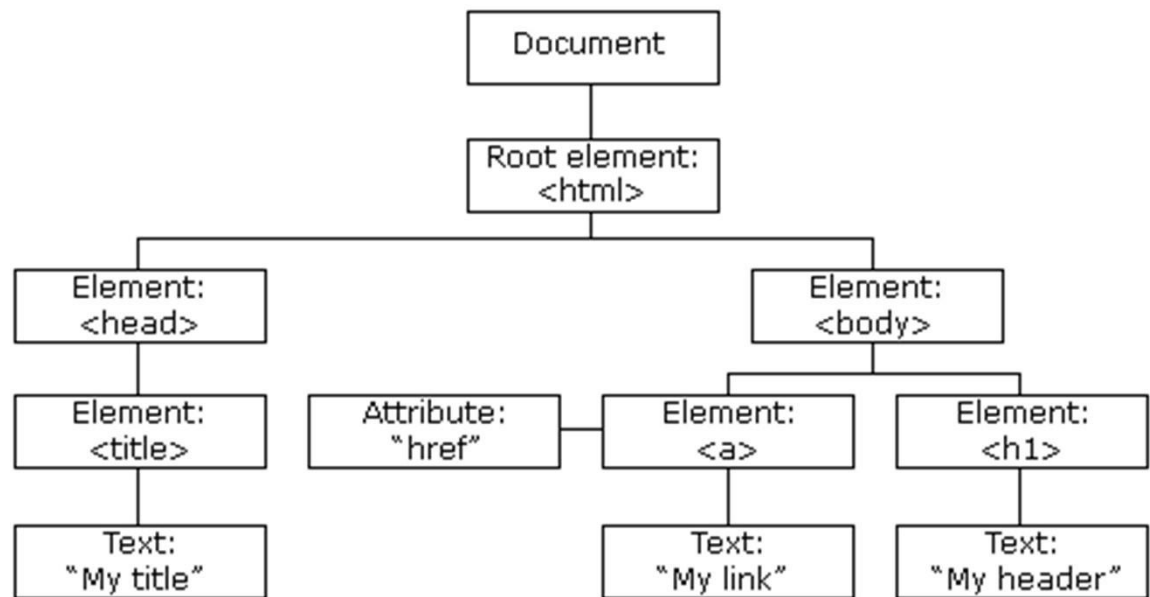
<http://www.w3.org/DOM/>



DOM : DOCUMENT OBJECT MODEL

- ❖ El **HTML DOM** es un estandar para obtener, cambiar, añadir o eliminar elementos HTML.

Ejemplo HTML DOM



<http://www.w3schools.com/html/dom/default.asp>



JavaScript: Métodos principales

- ▶ **getElementById(id)**
- ▶ Permite obtener un elemento del documento html por su ID. Retorna un elemento.
- ▶ Ejemplo:
 - ▶ `document.getElementById("demo").innerHTML = "Hello JavaScript";`
 - ▶ `document.getElementById('myImage').src='pic_bulbon.gif'`



JavaScript: Métodos principales

- ▶ **getElementsByTagName(tagname)**
- ▶ Permite obtener elementos por su nombre de etiqueta.
Retorna un arreglo de elementos
- ▶

```
var x = document.getElementsByTagName("p");  
var i;  
for (i = 0; i < x.length; i++) {  
    x[i].style.backgroundColor = "red";  
}
```



JavaScript: Métodos principales

- ▶ **getElementsByTagName(name)**
- ▶ Permite obtener elementos por el valor de su atributo name. Retorna un arreglo de elementos
- ▶

```
var x = document.getElementsByTagName("animal");  
var i;  
for (i = 0; i < x.length; i++) {  
    if (x[i].type == "checkbox") {  
        x[i].checked = true;  
    }  
}
```



JavaScript: Métodos principales

- ▶ **querySelector(selector)**
- ▶ Retorna el primer elemento cuyo selector CSS coincida con el especificado.
- ▶ `document.querySelector("#demo").innerHTML = "Hello World!";`
- ▶ `var y= document.querySelector("a[target]");`
- ▶ `y.style.color= "red";`



JavaScript: Métodos principales

- ▶ **querySelectorAll(selector)**
- ▶ Retorna todos los elementos cuyo selector CSS coincida con el especificado.
- ▶

```
var x = document.querySelectorAll(".example");  
x[0].style.backgroundColor = "red";
```
- ▶

```
var x = document.querySelectorAll("a[target]");  
var i;  
for (i = 0; i < x.length; i++) {  
    x[i].style.border = "10px solid red";  
}
```



JavaScript: Métodos principales

- ▶ **createElement(tagname)**

- ▶ Crea un nodo de tipo elemento, según el nombre de etiqueta especificado.

- ▶

```
var para = document.createElement("P");           // Create a
<p> element
para.innerHTML = "This is a paragraph.";          // Insert
text
document.getElementById("myDIV").appendChild(para); //
Append <p> to <div> with id="myDIV"
```



JavaScript: Métodos principales

- ▶ **appendChild(child)**

- ▶ Agrega el nodo especificado, como último hijo.

- ▶ var item;

- ▶ var lista = document.querySelector("ul");//selecciona elemento ul

- for (i = 0;i < 10;i++) {

- ▶ item = document.createElement("li");//crea elemento li

- ▶ item.innerHTML = " Item" + i; // escribe contenido en elemento item

- ▶ lista.appendChild(item); // agrega el item a la lista
- }



JavaScript: Métodos principales

- ▶ **removeChild(child)**
- ▶ Elimina el nodo especificado.
 - ▶ `var lista = document.querySelector("ul");`//selecciona la lista ul
 - ▶ `var items = document.querySelector("ul li");`//selecciona elementos li de la lista ul
 - ▶

```
for (var i = 0; i < lista.length; i++) {  
    if (lista.hasChildNodes()) {  
        lista.removeChild(items[i]); // elimina cada item de la lista  
    }  
}
```



JavaScript: objeto Style

- ▶ El objeto style de cada elemento html representa su estilo individual.
- ▶ `document.getElementById("demo").style.display = "none";`
- ▶ `document.getElementById("demo").style.fontSize = "35px";`

```
<head>
<script type="text/javascript">
  function cambiarEstilo(color)
  {
    document.body.style.backgroundColor="lavender";
    document.getElementById("p1").style.color=color;
    document.getElementById("p1").style.fontFamily="Arial";
  }
</script>
</head>

<body>
<p id="p1">Hello world!</p>
<input type="button" onclick="cambiarEstilo('blue')" value="Cambiar estilo" />
</body>
</html>
```



JavaScript: Validación de Formularios

- ▶ El elemento `<form>` de html define un formulario que es utilizado para recolectar datos de entrada del usuario.

Ejemplo:

```
<form name="form_datos" action="/action_page.php" method="get">  
  <label>Nombre:</label>  
  <input type="text" name="nombre"/></br>  
  <label>Apellidos:</label>  
  <input type="text" name="apellidos"/></br>  
  <input type="submit" value="Enviar"/>  
</form>
```



JavaScript: Validación de Formularios

- ▶ El elemento `<form>` de html define un formulario que es utilizado para recolectar datos de entrada del usuario.

Propiedades de los elementos de formulario Html

propiedad	descripción
name	es el nombre único del formulario.
action	es el lugar al cual se envía el formulario para ser procesado. El <i>action</i> define la URL a la cual se envía dicho formulario.
method	método de envío de los datos insertados en un formulario. El <i>method</i> puede ser: GET = envía los datos en una cadena "visible". Conveniente para enviar pocos datos. POST = envía los datos en forma "invisible". Conveniente para enviar una gran cantidad de datos.
target	define la ventana o marco (frame) en la que se mostrarán o procesarán los resultados del formulario. El valor es el mismo que el utilizado en HTML (blank, self, top, nombre_marco, etc..)



JavaScript: Validación de Formularios

Un formulario html contiene elementos de formaulario. Los elementos de formulario más importantes son los `<input>`.

Los elementos input pueden ser mostrados de manera diferente, dependiendo del valor de su atributo type.

Ejemplos:

método	descripción
submit	envía el formulario.
reset	restablece el formulario a los valores por defecto.

Botones: `<input type="button">`, `<input type="reset">`, `<input type="submit">`

Cajas de texto: `<input type="text">`, `<input type="password">`, `<input type="number">`

Casillas de verificación: `<input type="checkbox">`

Botones de opcion: `<input type="radio">`

Campos ocultos: `<input type="hidden">`

Cajas de fecha o tiempo: `<input type="date">`, `<input type="time">`

Cajas para selección de archivos: `<input type="file">`

Etc.



JavaScript: Validación de Formularios

Adicionalmente existen otros elementos de formulario como:

El elemento `<select>` que define una lista drop-down (combo).Ejemplo:

```
<select name="cars">  
  <option value="volvo">Volvo</option>  
  <option value="saab">Saab</option>  
  <option value="fiat">Fiat</option>  
  <option value="audi">Audi</option>  
</select>
```

El elemento `<textArea>` que define una caja de texto de varias líneas

```
<textarea name="message" rows="10" cols="30">  
  The cat was playing in the garden.  
</textarea>
```



Validación de Formularios

Ejemplo validación campo vacío.

```
function validar_datos() {  
    //valido el nombre  
    if (document.frm_datos.nombre.value.length==0) {  
        alert("Por favor ingrese su nombre")  
        document.frm_datos.nombre.focus()  
        return 0;  
    }  
}
```

```
<form name="frm_datos" action="" method="GET" target="_self">  
Escribe tu nombre: <input type="text" name="nombre"><br>  
<input type="submit" value="Enviar datos"><br>  
<input type="reset" value="Borrar">  
</form>
```



Validación de Formularios

Uso de expresiones regulares para validación

Las expresiones regulares (regex) son modelos que describen las combinaciones de caracteres en el texto. Se podrían definir como una serie de caracteres que forman un patrón, de tal forma que podemos comparar el patrón con otros conjuntos de caracteres para ver las coincidencias.

Manual para expresiones regulares

<http://www.desarrolloweb.com/articulos/2033.php>

http://www.w3schools.com/js/js_obj_regexp.asp



Validación de Formularios

Carácter	Texto buscado
^	Principio de entrada o línea.
\$	Fin de entrada o línea.
*	El carácter anterior 0 o más veces.
+	El carácter anterior 1 o más veces.
?	El carácter anterior una vez como máximo (es decir, indica que el carácter anterior es opcional).
.	Cualquier carácter individual, salvo el de salto de línea.
x y	x o y.
{n}	Exactamente n apariciones del carácter anterior.
{n,m}	Como mínimo n y como máximo m apariciones del carácter anterior.
[abc]	Cualquiera de los caracteres entre corchetes. Especifique un rango de caracteres con un guión (por ejemplo, [a-f] es equivalente a [abcdef]).
[^abc]	Cualquier carácter que no esté entre corchetes. Especifique un rango de caracteres con un guión (por ejemplo, [^a-f] es equivalente a [^abcdef]).
\b	Límite de palabra (como un espacio o un retorno de carro).

\B	Cualquiera que no sea un límite de palabra.
\d	Cualquier carácter de dígito. Equivalente a [0-9].
\D	Cualquier carácter que no sea de dígito. Equivalente a [^0-9].
\f	Salto de página.
\n	Salto de línea.
\r	Retorno de carro.
\s	Cualquier carácter individual de espacio en blanco (saltos de página o saltos de línea).
\S	Cualquier carácter individual que no sea un espacio.
\t	Tabulación.
\w	Cualquier carácter alfanumérico, incluido el de subíndice (Za-z0-9_).
\W	Cualquier carácter que no sea alfanumérico. Equivalente a [^a-zA-Z0-9_].



Validación de Formularios

Ejemplo validación campo fecha.

```
var regexFecha = /^(0[1-9]|1\d|2\d|3[1-2])\/(0[1-9]|1[0-2])\/\d{4}$/;
var valueFecha = document.frm_datos.fechaNac.value;
if( valueFecha =='' ){
    alert("Por favor ingrese su fecha de nacimiento");
    document.frm_datos.fechaNac.focus();
    return 0;
}else if(!valueFecha.match(regexFecha)){
    alert("El formato de la fecha ingresada no es correcto. Por favor verifique.");
    document.frm_datos.fechaNac.focus();
    return 0;
}
```

```
var regexFecha = /^(0[1-9]|1\d|2\d|3[1-2])\/(0[1-9]|1[0-2])\/\d{4}$/;
```



Validación de Formularios

Ejemplo validación grupos de opciones.

```
var grupo = document.frm_datos.tipoSubscripcion;
if (!validar_Un_Check(grupo)) {
    alert("Por favor seleccione al menos un tipo de suscripción");
    return 0;
}
alert("validación correcta");
}

function validar_Un_Check(checks) {
    for (i = 0; lcheck = checks[i]; i++) {
        if (lcheck.checked) {
            return true;
        }
    }
    return false;
}
```

```
<p><input name="tipoSubscripcion" type="checkbox" value="mensual" >Mensual</input></p>
<p><input name="tipoSubscripcion" type="checkbox" value="semestral" >Semestral</input></p>
<p><input name="tipoSubscripcion" type="checkbox" value="anual" >Anual</input></p>
```



Conociendo PHP



*Introducción a uno de
los lenguajes de
programación web
más usados en
Internet*

*LSI. Angel Veloz Rodríguez, Mgs.
angel.velozr@ug.edu.ec*

Audiencia

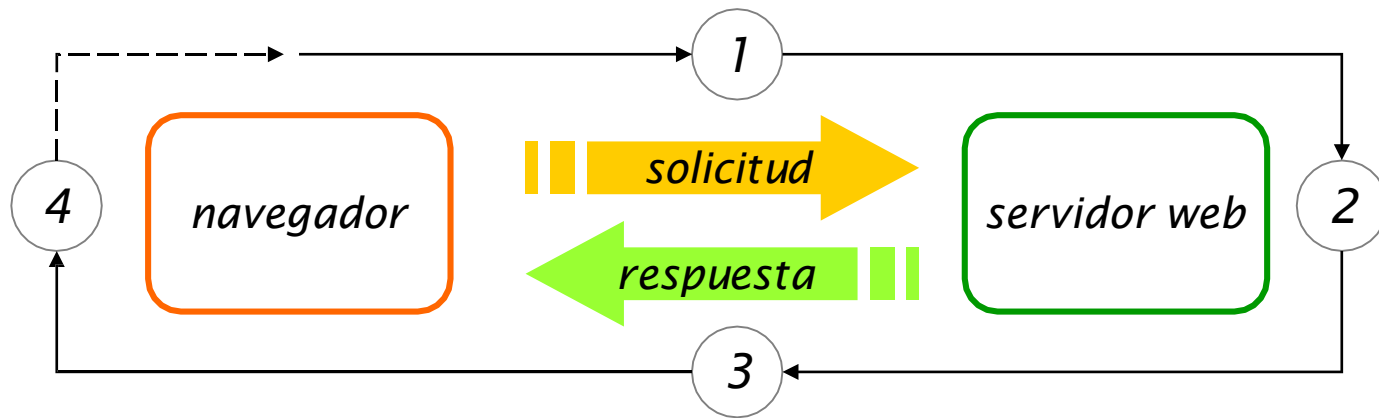
- *Web*
- *Programación*



¿Qué es PHP?

*Lenguaje de programación
para generar páginas web*

El ciclo web



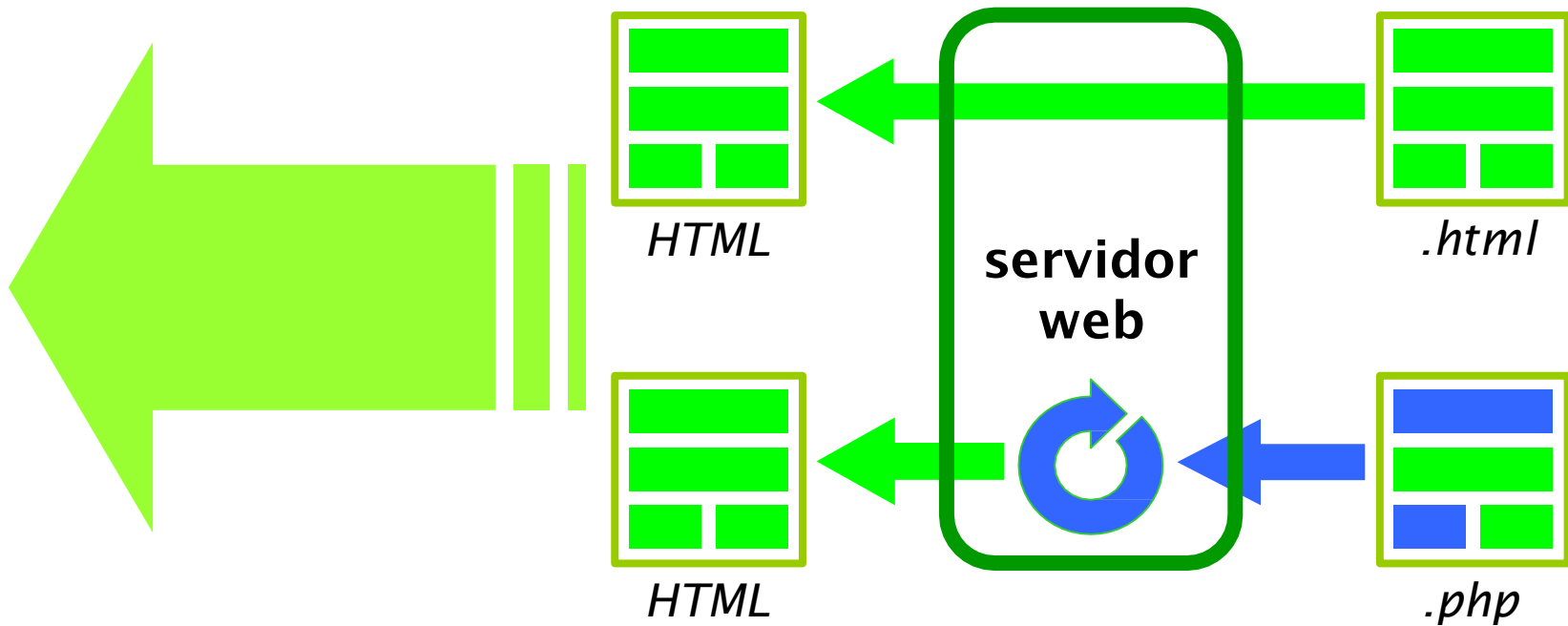
presentación



proceso

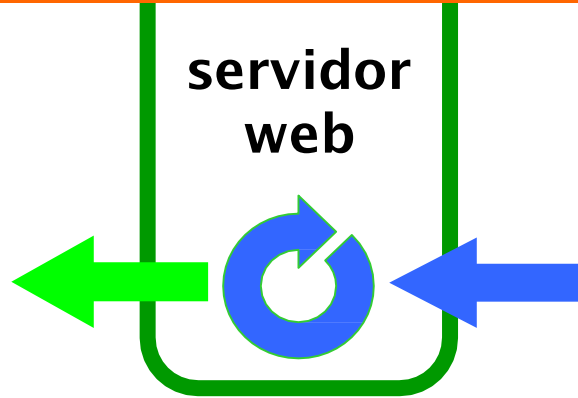
respuesta web

El contenido del documento .html es pasado tal cual a la respuesta HTML



Un documento dinámico (como .php) es procesado para obtener HTML

respuesta web



en el proceso, un documento dinámico puede involucrar diversos recursos



base de datos



email



archivos



gráficos



etc

¿Por qué PHP?

- *Es ampliamente soportado por los servicios de hosting*
- *Es libre*
- *Un número significativo de sites se ha escrito en PHP (Wikipedia , YouTube, Yahoo, Facebook, WordPress...)*
- *Muchos paquetes importantes están escritos en PHP (WordPress, OS Commerce, Joomla, Drupal...)*
- *Es simple, directo, flexible...*
- *Ha evolucionado para soportar programación orientada a objetos*



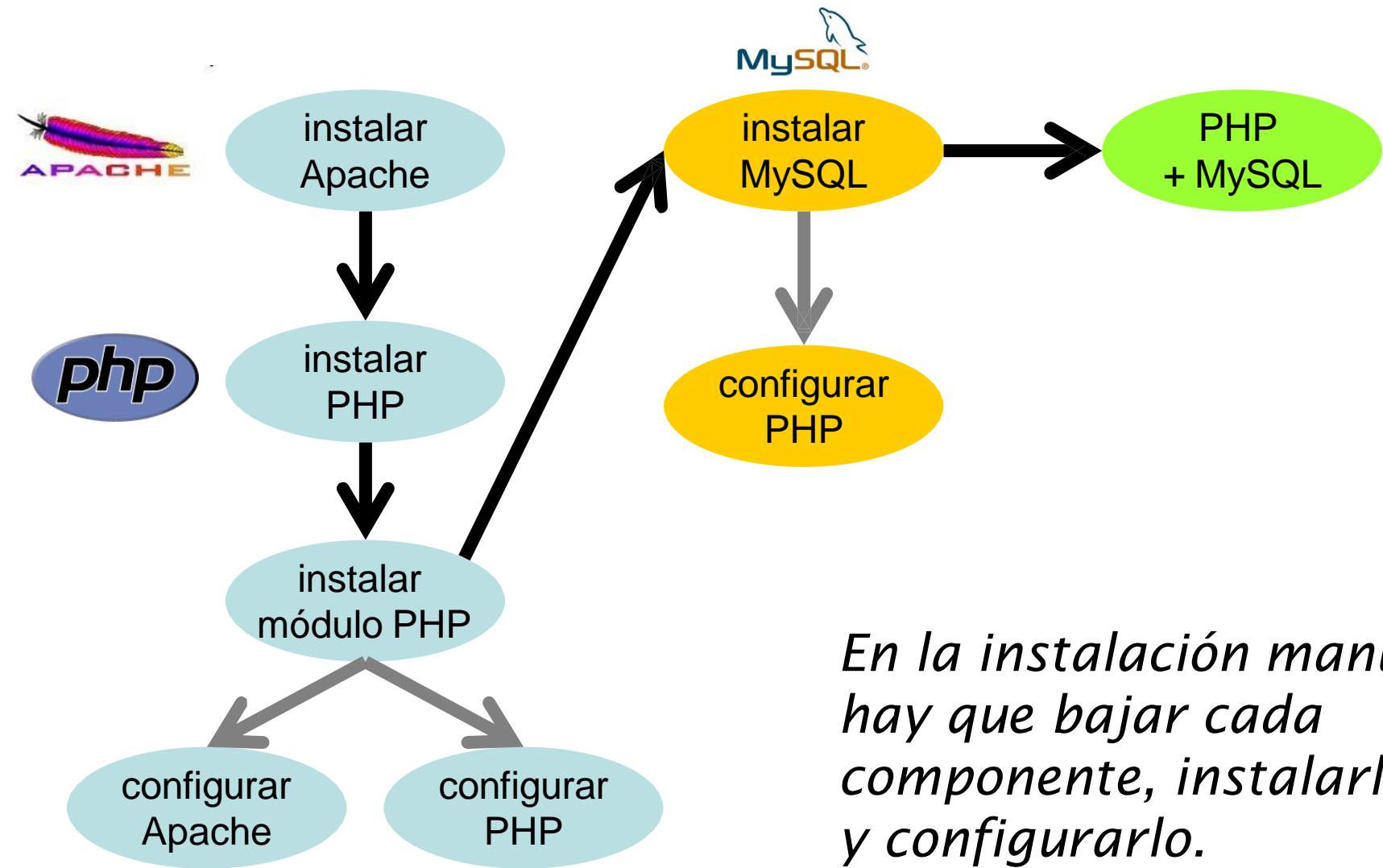
WIKIPEDIA
The Free Encyclopedia



Conociendo PHP

para empezar

instalando PHP



En la instalación manual hay que bajar cada componente, instalarlo y configurarlo.

LAMP, WAMP, ...



Linux



APACHE

A



MySQL

M



php

P



Windows



APACHE

A



MySQL

M



php

P



Mac



APACHE

A



MySQL

M



php

P

- Típicamente, PHP se instala para trabajar junto con Apache y MySQL
- Cuando la instalación aparece en Linux, se la llama LAMP
- Cuando la instalación aparece en Windows, se la llama WAMP
- Y así por el estilo...

XAMPP



- Hay varios proyectos que difunden paquetes que contienen esas instalaciones típicas.
- Usarlos puede facilitar la instalación, y el día a día con el entorno de desarrollo.

- XAMPP es uno de tales paquetes. No sólo provee PHP, sino también Perl y Python y está disponible tanto para Linux, como para Windows, Mac OS X y Solaris



XAMPP

Editores, IDEs



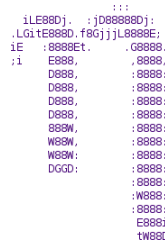
```
...
iLE88Dj. :jD88888Dj:
.LG1tE888D.f8GjjL8888E:
iE :8888Et. :88888.
:1 E888. :8888.
D888. :8888.
D888. :8888.
D888. :8888.
888W. :8888.
W88W. :8888.
DGGD: :8888.
:8888.
W8881
tw88D
```



aptana

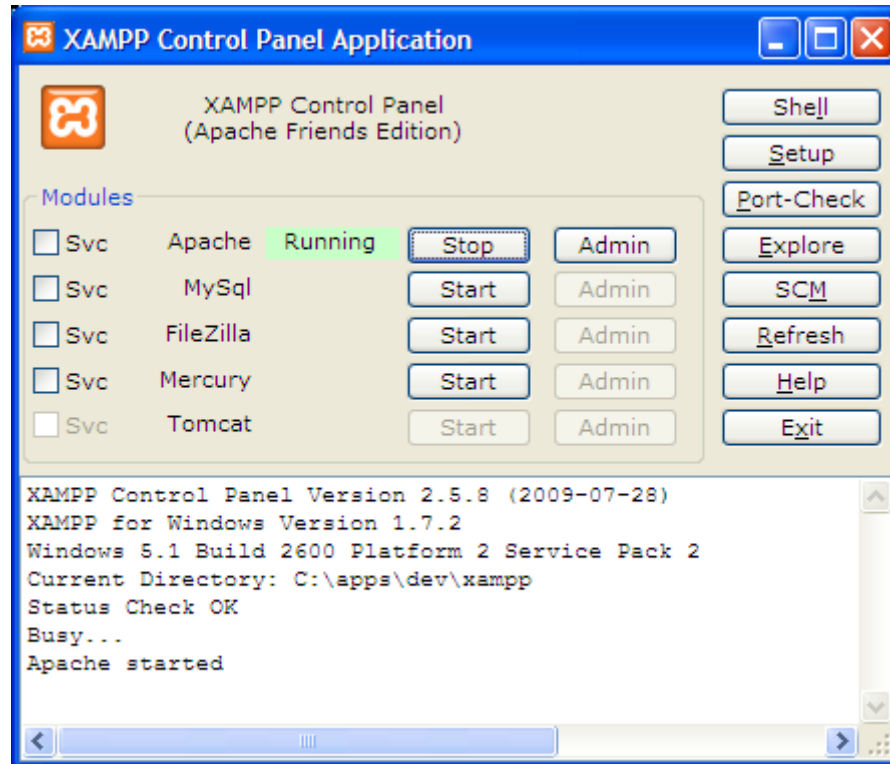
- Igual que con HTML, CSS, Javascript y otros lenguajes web, es posible trabajar usando un simple editor de texto, como Notepad, aunque contar con un entorno de desarrollo puede facilitar mucho las cosas.
- Hay editores de texto muy útiles para programación, como Notepad++, jEdit, ...
- Y entornos de desarrollo muy buenos, como Dreamweaver, Aptana, Eclipse, Zend Studio...

Editores, IDEs



- *Estos son los que suelo usar*
- **Notepad** *cuando no hay nada más a la mano*
- **Notepad++** *es excelente, ligero, con FTP y una buena utilidad para convertir Unicode*
- **Dreamweaver** *es excelente para el diseño HTML, tiene buen soporte para PHP*
- **Aptana** *esta basada en Eclipse, pero tiene mucho mejor soporte para PHP, HTML, Javascript...*
- **Nano**, *cuando estoy Linux y no hay nada más a la mano*

Empezando



El panel de control de XAMPP, desde donde podemos iniciar y detener sus servicios

Conociendo PHP

lo básico

¡Hola Mundo!

`hola_mundo.php`

```
<?php  
echo '¡Hola Mundo!';  
?>
```

- *El código php se escribe entre etiquetas <?php y ?>*
- *Para escribir algo en la página, podemos usar echo*
- *Las cadenas se escriben entre apóstrofes 'así', ó entre comillas "así"*
- *El nombre del archivo tiene la extensión php*

¡Hola Mundo!

hola_mundo.php

```
<?php  
echo '¡Hola Mundo!';  
?>
```

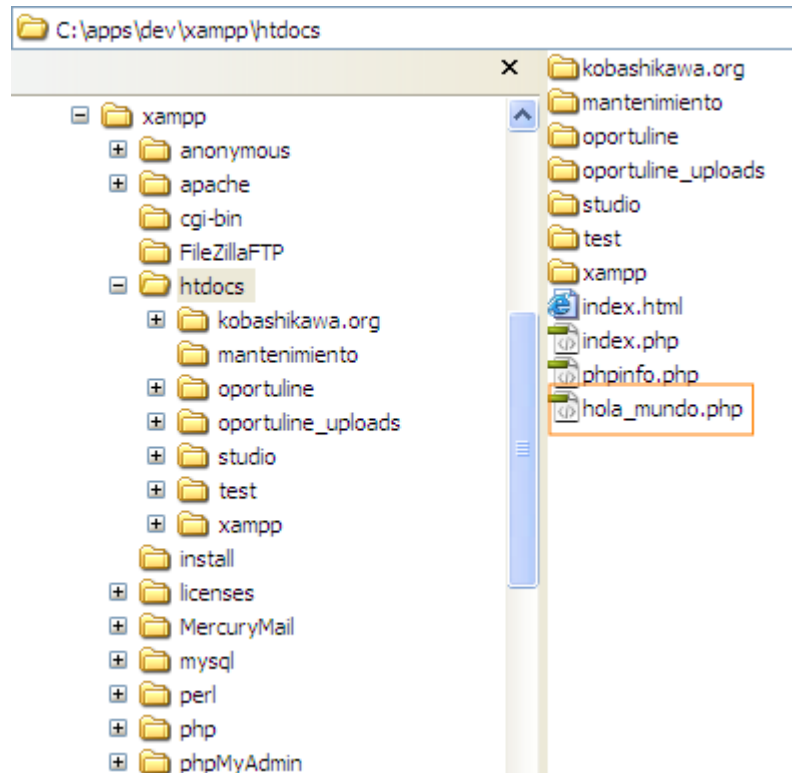


htdocs

...¡Hola Mundo!



htdocs



htdocs/index.php



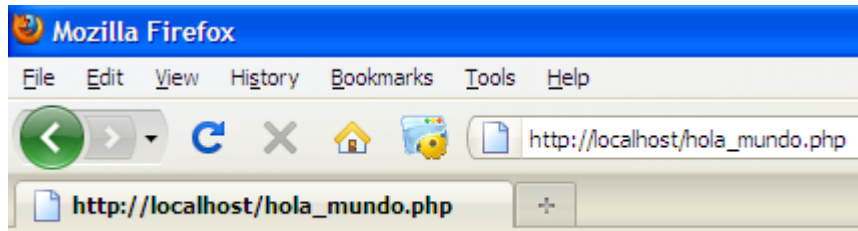
<http://localhost/index.php>

htdocs/hola_mundo.php



http://localhost/hola_mundo.php

...¡Hola Mundo!



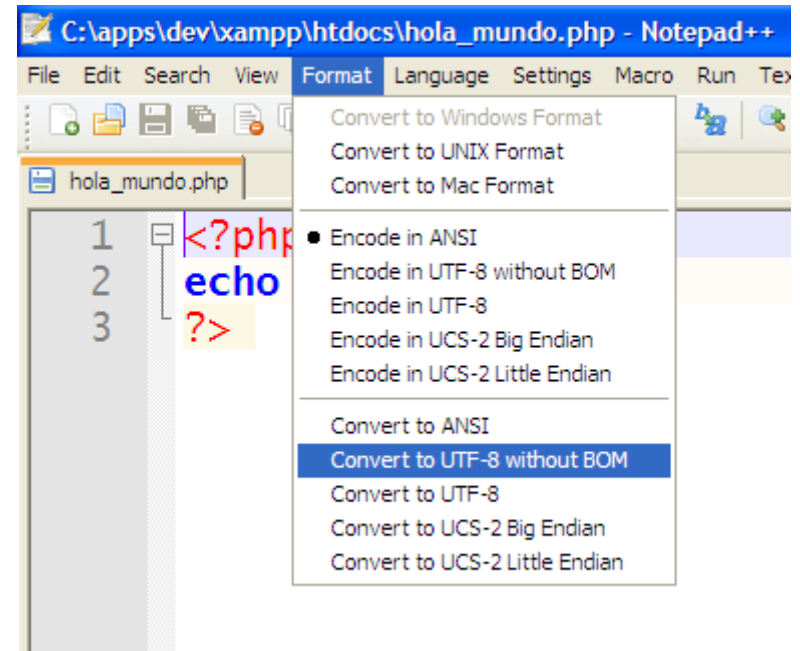
¡Hola Mundo!

unicode guardado como ASCII



¡Hola Mundo!

unicode guardado como UTF-8

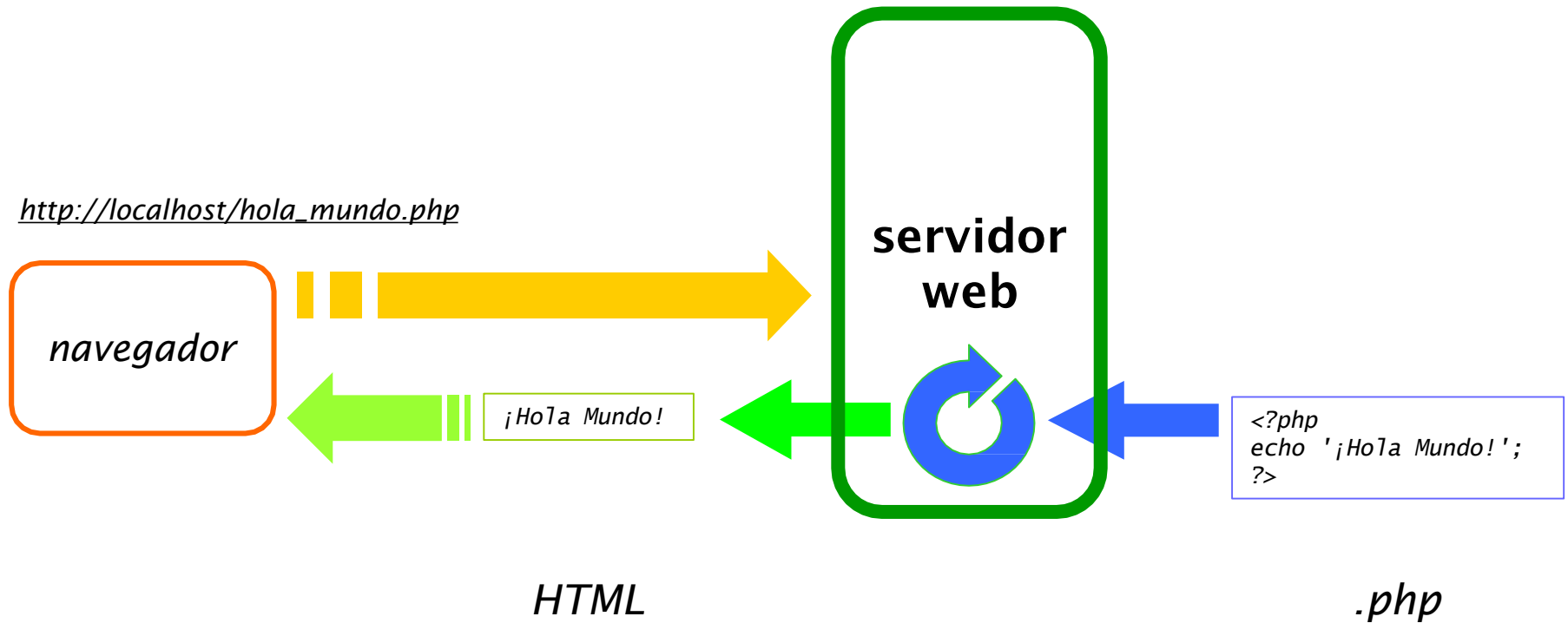


*En Notepad++ es fácil
comprobar y cambiar el encode*



*Si usamos unicode, tener cuidado de
guardar el archivo en UTF-8 (sin BOM)*

...¡Hola Mundo!



...¡Hola Mundo!

`hola_mundo.php`

```
<html>
<body>
  <h1>
    <?php echo '¡Hola Mundo!';?>
  </h1>
</body>
</html>
```

- *Un documento php es básicamente html con código inserto*
- *Los bloques php se reemplazan por su resultado*

...¡Hola Mundo!

```
<html>  
<body>  
  <h1>  
    ¡Hola Mundo!  
  </h1>  
</body>  
</html>
```


Cuadrados

cuadrados.php

```
<html>
<body>
  <h1>Cuadrados</h1>
  <table border="1">
    <?php for ($i=1; $i<=10; $i++) { ?>
      <tr>
        <td><?php echo $i;?></td>
        <td><?php echo $i*$i;?></td>
      </tr>
    <?php } ?>
  </table>
</body>
</html>
```

...Cuadrados

```
<html>
<body>
  <h1>Cuadrados</h1>
  <table border="1">
    <tr>
      <td>1</td>
      <td>1</td>
    </tr>
    <tr>
      <td>2</td>
      <td>4</td>
    </tr>
    ...
    <tr>
      <td>10</td>
      <td>100</td>
    </tr>
  </table>
</body>
</html>
```

 <http://localhost/cuadrados.php>

Cuadrados

1	1
2	4
3	9
4	16
5	25
6	36
7	49
8	64
9	81
10	100

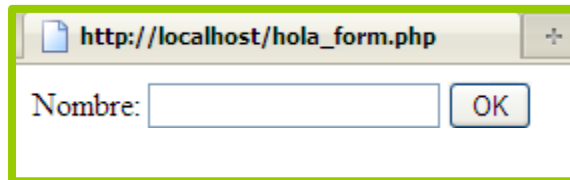
Conociendo PHP

formularios

Hola Tú

hola_form.php

```
<html>
<body>
  <form action="hola.php" method="post">
    Nombre:
    <input type="text" name="nombre"/>
    <input type="submit" value="OK"/>
  </form>
</body>
</html>
```



A screenshot of a web browser window. The address bar shows the URL `http://localhost/hola_form.php`. The page content displays the text "Nombre:" followed by a text input field and an "OK" button. The entire browser window is outlined with a green border.

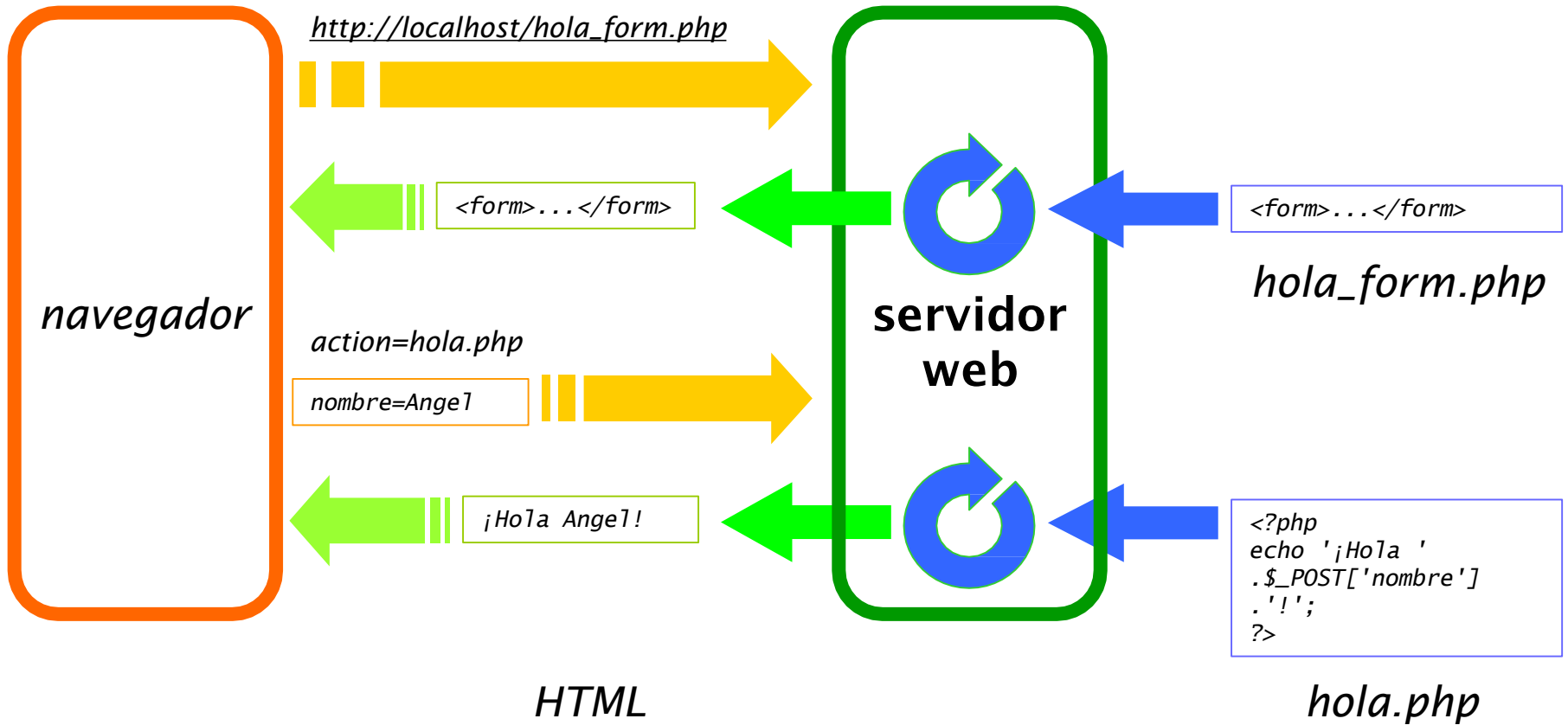
...Hola Tú

hola.php

```
<html>
<body>
<h1>
  <?php echo '¡Hola ' . $_POST['nombre'] . '!' ;?>
</h1>
</body>
</html>
```



...Hola Tú

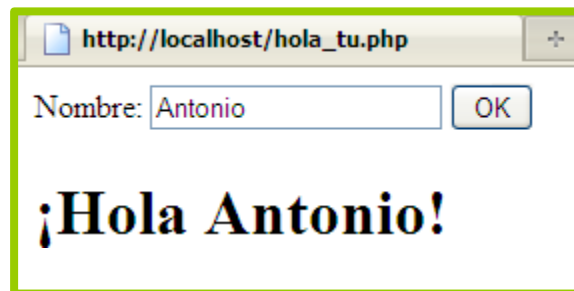


...Hola Tú

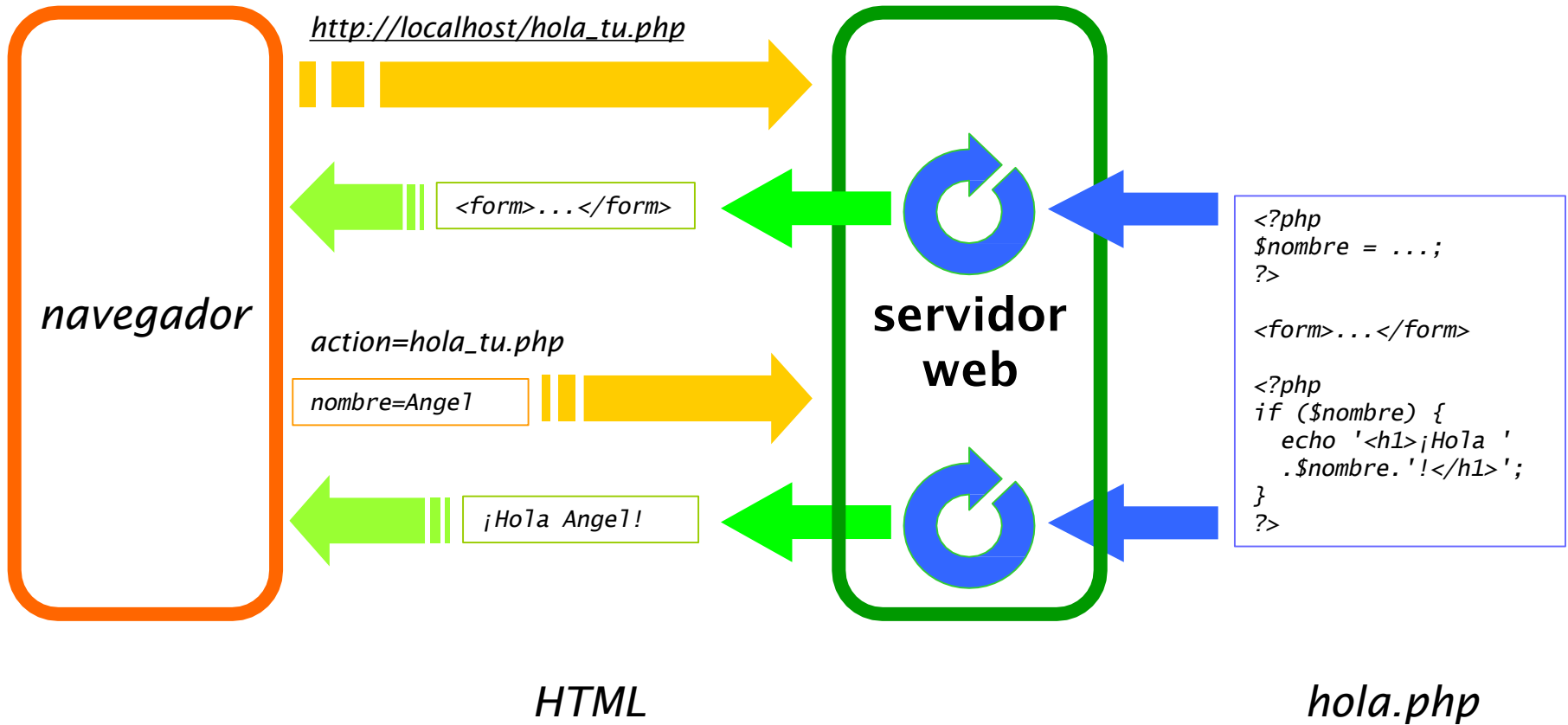
hola_tu.php

versión de una página

```
<html>
<body>
  <?php
    $nombre = isset($_POST['nombre'])?$_POST['nombre']:'';
  ?>
  <form action="hola_tu.php" method="post">
    Nombre:
    <input type="text" name="nombre"
      value="<?php echo $nombre;?>" />
    <input type="submit" value="OK">
  </form>
  <?php if ($nombre) {
    echo '<h1>¡Hola ' . $nombre . '!</h1>';
  }?>
</body>
</html>
```



...Hola Tú



Conociendo PHP

un poco más

Bloques básicos

```
$x = 1+1;
```

```
if ($x>0) {  
  ...  
}
```

```
if ($x>0) {  
  ...  
} else {  
  ...  
}
```

```
switch ($x) {  
  case 1:  
    ...  
    break;  
  ...  
  default  
    ...  
    break;  
}
```

```
for ($i=0; $i<10; $i++) {  
  ...  
}
```

```
while ($i<10) {  
  ...  
}
```

```
do {  
  ...  
} while ($i<10);
```

asignación

decisión

- *if*
- *if/else*
- *switch*

repetición

- *for*
- *while*
- *do-while*

Arrays

```
$x = 1;  
$a = array(2, 3, $x, 'Hola');
```

```
(  
    [0]=>2  
    [1]=>3  
    [2]=>1  
    [3]=>'Hola'  
)
```

a[3]

'Hola'

```
foreach ($a as $item) {  
    echo $item.'  
}
```

```
2  
3  
1  
'Hola'
```

```
foreach ($a as $indice=>$item) {  
    echo $indice.': '.$item.'  
}
```

```
0: 2  
1: 3  
2: 1  
3: 'Hola'
```

Funciones

```
function hola_mundo() {  
    echo '¡Hola Mundo!';  
}
```

```
hola_mundo();
```

```
function hola($x) {  
    echo '¡Hola '.$x.'!';  
}
```

```
hola('Mundo');
```

```
function saludo($y) {  
    return '¡Hola '.$y.'!';  
}  
function hola($x) {  
    echo saludo($x);  
}
```

```
hola('Mundo');
```

```
¡Hola Mundo!
```


Clases

```
class Hola() {  
    function saludo($y) {  
        return '¡Hola '.$y.'!';  
    }  
    function hola($x) {  
        echo $this->saludo($x);  
    }  
}
```

```
$h = new Hola();  
$h->hola('Mundo');
```

¡Hola Mundo!

include

cabecera.php

```
echo '<h1>Conociendo PHP</h1>';
```



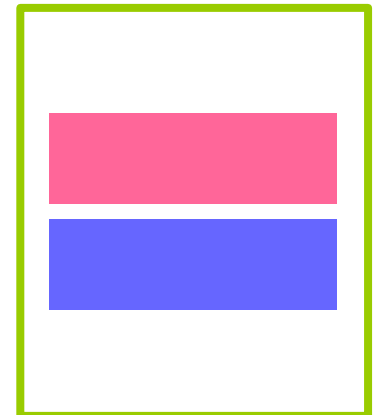
_hola_mundo.php

```
echo 'Hola Mundo';
```



hola_mundo.php

```
<html>
<body>
  <?php include 'cabecera.php';?>
  <?php include '_hola_mundo.php';?>
</body>
</html>
```



require

funciones.php

```
function hola() {  
    return '¡Hola Mundo!';  
}
```



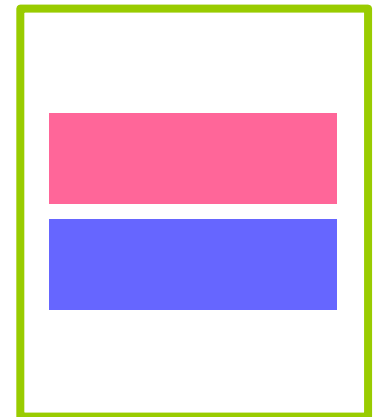
_hola_mundo.php

```
echo hola();
```



hola_mundo.php

```
<html>  
<body>  
    <?php require('funciones.php');?>  
    <?php include '_hola_mundo.php';?>  
</body>  
</html>
```



include & require

- *include() y require() son similares, la diferencia es que si no existe el archivo a incluir, include() emitirá una alarma, mientras que require() terminará el programa.*
- *Un mismo archivo se puede incluir más de una vez.*
- *Para prevenir duplicaciones en las inclusiones existen **include_once()** y **require_once()**.*

Frameworks

- *Un framework es una forma de organizar el trabajo.*
- *Luego de un tiempo desarrollando, uno empieza a notar ciertos patrones de organización. Aún sin proponérselo, va surgiendo un framework.*
- *Algunos frameworks muy difundidos en la comunidad PHP son CakePHP, CodeIgniter, Zend, Symfony...*
- *Un framework estandarizado obliga a seguir ciertas convenciones de organización, pero pone a nuestra disposición código ampliamente probado y optimizado.*



Referencias

- *PHP site*
 - <http://php.net>
- *W3Schools – PHP*
 - <http://www.w3schools.com/PHP/default.asp>
- *CakePHP*
 - <http://cakephp.org>
- *CodeIgniter*
 - <http://codeigniter.com>
- *Zend Framework*
 - <http://framework.zend.com/>
- *Symphony*
 - <http://www.symfony-project.org/>
- *Coquette Icons*
 - <http://dryicons.com/free-icons/preview/coquette-icons-set/>



Conociendo PHP



*Introducción a uno de
los lenguajes de
programación web
más usados en
Internet*

*LSI. Angel Veloz Rodríguez, Mgs.
angel.velozr@ug.edu.ec*



*preguntas,
comentarios*

