

HTML: HYPERTEXT MARKUP LANGUAGE

- Lenguaje de definición de **formato** de documentos
 - Define los elementos que pueden aparecer dentro del documento
 - Se especifican mediante etiquetas (*Tags*) de comienzo y final de documento
 - Texto ASCII encerrado entre < y >
 - Sólo se define el tipo de elemento y no la forma de representarlo
- Lenguaje estándar utilizado en la Web para el intercambio de documentos hipermedia que incluyen:
 - Texto
 - Imágenes (Fotos, Vídeo)
 - Audio
 - Vínculos (Links)
- Estándar especificado por el *World Wide Web Consortium*, **W3C**



ESTRUCTURA DE UN DOCUMENTO HTML

- Identificación SGML
 - Permite identificar la DTD adecuada para procesarlo

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
```

- Delimitación del documento HTML
 - Identificado con la etiqueta <HTML>
- Cabecera. Información asociada al documento
 - Identificado por la etiqueta <HEAD>
 - Contiene información descriptiva del documento y definiciones de formato para los elementos del resto del mismo
- Cuerpo del documento
 - Identificado con la etiqueta <BODY>
 - Contiene los elementos de presentación del documento



COMPONENTES DEL LENGUAJE HTML

- Etiquetas/Elementos
 - No vacías:
 - La mayoría de elementos comienzan por un tag, luego contenido, finalizando por un tag de terminación
`<TAG> contenido </TAG>`
 - Vacías: por ejemplo ``, `
`, etc.
- Atributos:
 - Específicos de cada etiqueta
 - Texto ASCII entre comillas (doble o simple) dentro de la etiqueta de comienzo
- HTML es insensible a Mayúsculas/Minúsculas:
 - `<TAG>` es equivalente a `<tag>` o `<TaG>`
- Espacios en blanco, tabulaciones y retornos de carros no son significativos



COMPONENTES DEL LENGUAJE HTML

Existen multitud de elementos HTML. A continuación se muestra un esqueleto de documento con los más representativos para las prácticas. Consultar la documentación de la W3 schools: <https://www.w3schools.com/html>

```
<!DOCTYPE ...>
<html>
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="...">
    <title>...</title>
  </head>

  <body>
    <h1>...</h1> <h2>...</h2> ...
    <ul> <li> ... </ul>
    <ol> <li> ... </ol>
    <img>
    <div> ... </div>
    <span> ... </span>

    <table>
      <tr> <th> ... </th></tr>
      <tr> <td> ... </td></tr>
    </table>
    <a href="..."></a>
    <form>
      <label>...</label>
      <input type="..." />
      <select> <option>...</option></select>
      <textarea>
      <button>
      <datalist>
    </form>
  </body>
</html>
```

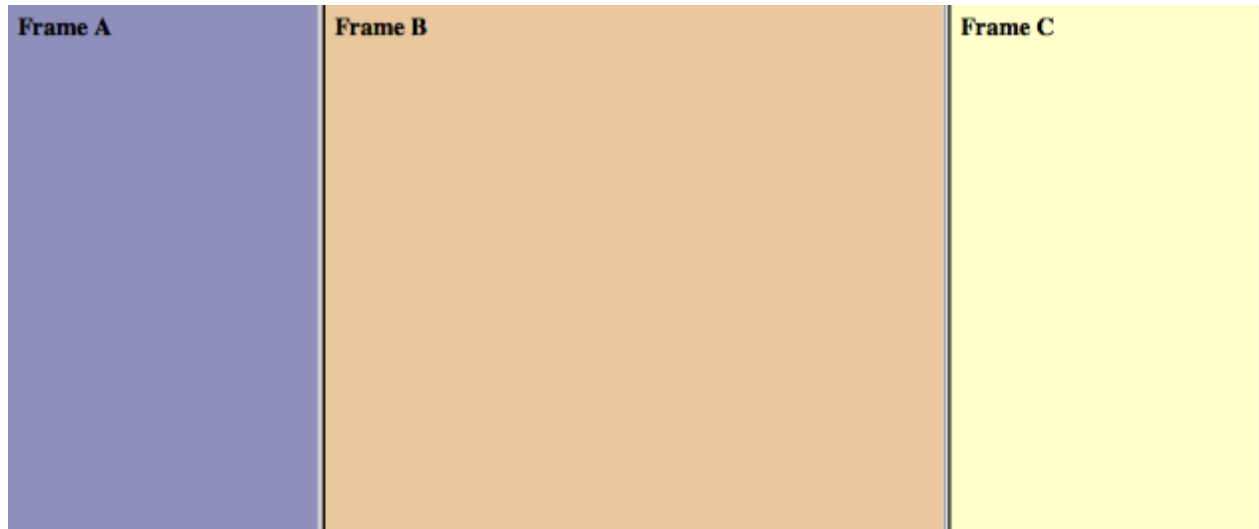


FRAMES (MARCOS)

- Con los frames (iframes en HTML5) varios documentos HTML pueden ser mostrados en la misma ventana del navegador
- Cada frame es independiente de los otros
- Han caído en desuso: seguridad y AJAX

EJEMPLO DE *FRAMES*

```
<html>
  <frameset cols="25%,50%,25%">
    <frame src="frame_a.htm" />
    <frame src="frame_b.htm" />
    <frame src="frame_c.htm" />
  </frameset>
</html>
```

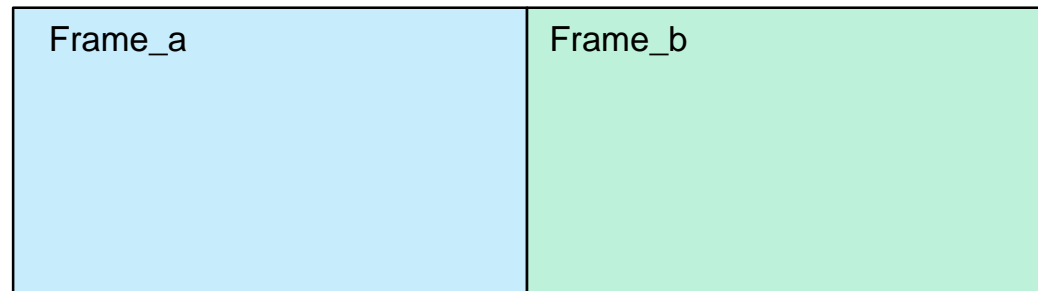


EJEMPLO DE *IFRAMES*

```
<html>
  <body>
    <p>Ejemplo de página con iframes</p>
    <table width="100%">
      <tr>
        <td>
          <iframe src="frame_a.htm" height="200px" width="100%"></iframe>
        </td>
        <td>
          <iframe src="frame_b.htm" height="200px" width="100%"></iframe>
        </td>
      </tr>
    </table>
  </body>
</html>
```



Ejemplo de página con iframes



LAYOUT EN HTML

- Tres formas de organizar dónde va cada cosa:
 - Tablas
 - Etiqueta `<div>`
 - CSS
 - Etiquetas HTML5

Main Title of Web Page

Menu

HTML

CSS

JavaScript

Content goes here



LAYOUT UTILIZANDO <TABLE>

- El único disponible en las primeras versiones de HTML
- Solía combinarse con el uso de frames, actualmente es más habitual el uso de AJAX
- Desventajas:
 - Obliga a crear ficheros muy grandes y complejos
 - Obliga a descargar información de presentación con cada página que se descarga
 - El rediseño de sitios y contenido existente es más laborioso (y costoso)
 - Es difícil (y costoso) mantener consistencia visual en un sitio
 - Las páginas basadas en tablas son menos accesibles a usuarios con necesidades especiales y a los que usan teléfonos móviles, PDAs o tabletas para acceder



LAYOUT UTILIZANDO <TABLE>

```
<html>
<body>
```

```
<table width="500" border="0">
```

```
<tr>
```

```
<td colspan="2" style="background-color:#FFA500; height: 40px; text-align: center; vertical-align: middle;">
<h1>Main Title of Web Page</h1>
```

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<tr valign="top">
```

```
<td style="background-color:#FFD700; width: 100px; text-align: center; vertical-align: top;">
<b>Menu</b><br />
HTML<br />
CSS<br />
JavaScript
```

```
</td>
```

```
</tr>
```

```
</table>
```

```
</body>
```

```
</html>
```

```
<td style="background-color:#EEEEEE; height:200px; width:400px; text-align:top;">
Content goes here</td>
```

```
</tr>
```

```
</table>
```

```
</body>
```

```
</html>
```

DON'T DO IT.

Main Title of Web Page

Menu
HTML
CSS
JavaScript

Content goes here

LAYOUT UTILIZANDO <DIV>

```
<html>
<body>

<div id="container" style="width:500px">

<div id="header"
style="background-color:#FFA500;">
<h1 style="margin-bottom:0;">
Main Title of Web Page</h1></div>

<div id="menu"
style="background-color:#FFD700;
height:200px;width:100px;float:left;">

<b>Menu</b><br />
HTML<br />
CSS<br />
JavaScript</div>
```

```
<div id="content" style="background-
color:#EEEEEE;height:200px;width:400px;
float:left;">
Content goes here</div>

<div id="footer" style="background-
color:#FFA500;clear:both;text-
align:center;">
Copyright © 2011 W3Schools.com</div>

</div>

</body>
</html>
```



http://www.w3schools.com/tags/tag_div.asp

LAYOUT EN HTML5

```
<header></header>
```

```
<nav></nav>
```

```
<article></article>
```

```
<section></section>
```

```
<aside></aside>
```

```
<footer></footer>
```

CSS: CASCADE STYLE SHEETS

- Usado para describir la semántica de presentación de un documento escrito con un lenguaje de marcado
- El objetivo es separar el contenido del documento (HTML o similar) de cómo se presenta (layout, colores, fuentes, etc.)
- Ventajas:
 - Forma rápida y sencilla de cambiar el aspecto del documento
 - Ofrecer accesibilidad
 - Más control y flexibilidad de presentación
 - Compartir formato entre múltiples páginas
 - Distinta presentación para distinto tipo de dispositivo (ordenador, móvil, impreso...) o entre distinto tipo de usuarios de la aplicación (p.ej., usuarios de distintas compañías)

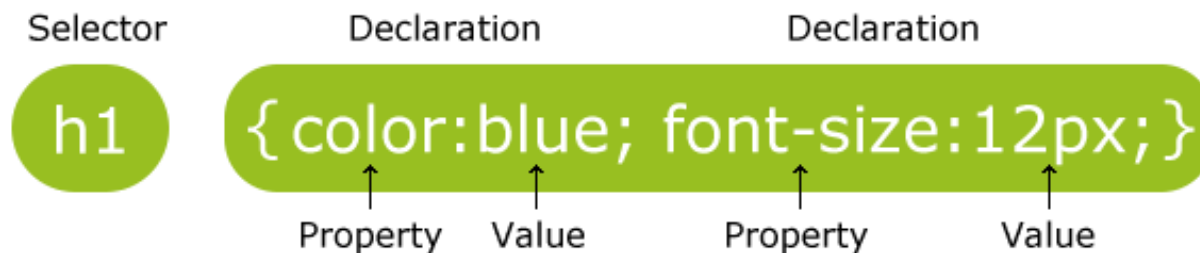


SINTAXIS CSS

- Un estilo CSS es una combinación de propiedades predefinidas a las que se asigna un valor (*propiedad:valor*). Cada declaración separada por “;”
- *Estilos inline* (mala práctica, solo en casos muy específicos): aplican directamente sobre un elemento del documento

```
<p style="color:sienna;margin-left:20px">This is a paragraph.</p>
```

- Lo más habitual es definir hojas de estilos compuestas por reglas de aplicación
- Las reglas constan de un selector y las declaraciones del estilo entre llaves



- Se pueden definir:
 - *Hojas de estilo internas*: en el propio documento como contenido del tag <style>
 - *Hojas de estilo externas*: en ficheros externos vinculados al documento



EJEMPLO CSS

```
<html>
<head>
<link rel="stylesheet" type="text/css"
href="ex2.css" />
</head>

<body>

<h1>This is a header 1</h1>
<hr />

<p>You can see that the style
sheet formats the text</p>

<p><a href="http://www.w3schools.com"
target="_blank">This is a link</a></p>

</body>
</html>
```

```
body {background-color:tan;}
h1 {color:maroon;font-size:20pt;}
hr {color:navy;}
p {font-size:11pt;margin-left:15px;}
a:link {color:green;}
a:visited {color:yellow;}
a:hover {color:black;}
a:active {color:blue;}
```

This is a header 1

You can see that the style sheet formats the text

[This is a link](http://www.w3schools.com)

TIPOS DE SELECTORES

- Selector de elemento
 - Usado para especificar el estilo de los elementos de un tipo determinado
 - Se define con el nombre del elemento
- Selector *id*
 - Usado para especificar el estilo de un elemento con un *id* determinado
 - Usa el atributo *id* de los elementos HTML y se define con “#”
- Selector *class*
 - Permite especificar un estilo para todos los elementos de una determinada “clase”
 - Utiliza el atributo *class* de HTML y se define con “.”
- Combinación de selectores:
 - Reglas de selección en función de la relación entre selectores:
 - Espacio: selector de descendientes
 - >: selector de hijos
 - ~: selector de hermanos
 - +: selector de hermanos adyacentes

https://www.w3schools.com/css/css_combinators.asp



EJEMPLO DE SELECTOR ID

```
<html>
<head>
<style type="text/css">
#para1 {
text-align:center;
color:red;
}
</style>
</head>

<body>
<p id="para1">Hello World!</p>
<p>This paragraph is not affected by the style.</p>
</body>
</html>
```



Hello World!

This paragraph is not affected by the style.



EJEMPLO SELECTOR CLASS

```
<html>
<head>
<style type="text/css">
.center {
text-align:center;
}
</style>
</head>

<body>
<h1 class="center">Center-aligned heading</h1>
<p class="center">Center-aligned paragraph.</p>
</body>
</html>
```



Center-aligned heading

Center-aligned paragraph.



CASCADA...

- ... lo que por fin explica el nombre
- Sobre un elemento determinado pueden aplicar distintos estilos definidos en distintos “sitios”
- Se impone un orden de precedencia que determina qué selector prevalece según tres criterios (de mayor a menor):
 - Importancia (!important)
 - Especificidad:
 - Estilo *inline*
 - Selector id
 - Selector class
 - Selector de elemento
 - Orden



PROPIEDADES CSS RELEVANTES

Existen multitud de propiedades CSS. A continuación se listan algunas de las más relevantes para la presentación y el layout del documento, pero se recomienda consultar la documentación de W3 schools en

<https://www.w3schools.com/css>

- display:
 - flex: flex-direction, flex-flow, ...
 - block
 - inline
- position
- z-index
- height/width max-width
- overflow
- float
- font-family/style/weight
- background-color
- border
- color
- text-align:



JSON: JAVASCRIPT OBJECT NOTATION

- Según el W3C:
 - JSON is a syntax for storing and exchanging data
 - JSON is text, written with *JavaScript* object notation
- Muy popular: a día de hoy es seguramente el formato más utilizado para intercambiar datos entre aplicaciones web
- Es legible por personas e interpretable por algoritmos
- Almacena los datos siguiendo la notación de **objetos** y **arrays JS**

```
{
  "artists" : [
    {
      "artistname" : "Deep Purple",
      "albums" : [
        {
          "albumname" : "Machine Head",
          "year" : "1972",
          "genre" : "Rock"
        },
        {
          "albumname" : "Stormbringer",
          "year" : "1974",
          "genre" : "Rock"
        }
      ]
    }
  ]
}
```



SINTAXIS JSON

- Un objeto *JS* es un conjunto desordenado de pares nombre/valor entre llaves ({})
 - El nombre es una cadena de caracteres que identifica el atributo
 - El valor puede ser un objeto, un array, un número, cadena de caracteres, true, false o null

```
{ }
```

```
{ "artistname" : "Pink Floyd" }
```

```
{  
  "artistname" : "Pink Floyd",  
  "formed" : "1964"  
}
```

```
{  
  "artistname" : "Pink Floyd",  
  "formed" : "1964",  
  "origin" : "Cambridge, United Kingdom"  
}
```



SINTAXIS JSON

- Un array es una colección ordenada de valores
- Permite construir listas de valores

```
{
  "artists" : [
    {
      "artistname" : "Pink Floyd",
      "formed" : "1964"
    },
    {
      "artistname" : "Bruce Springsteen",
      "born" : "1949"
    },
    {
      "artistname" : "U2",
      "formed" : "1976"
    }
  ]
}
```



DATOS ANIDADOS

```
{
  "artists" : [
    {
      "artistname" : "Pink Floyd",
      "formed" : "1964",
      "albums" : [
        {
          "albumname" : "The Dark Side of the Moon",
          "year" : "1973",
          "genre" : "Rock"
        },
        {
          "albumname" : "Wish You Were Here",
          "year" : "1975",
          "genre" : "Rock"
        }
      ]
    }
  ]
}
```

