



<http://www.html6.es/modulos.html>

<https://developer.mozilla.org/es/>

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

<https://www.arkaitzgarro.com/html5/index.html>

<https://jml.neocities.org/>

HTML5 provee básicamente tres características

CSS



Estilo

HTML



Estructura

JS



Funcionalidad

Hypertext Markup Language)

1980, cuando el físico **Tim Berners-Lee**, trabajador del [CERN](#) (*Organización Europea para la Investigación Nuclear*) propuso un nuevo sistema de "hipertexto" para compartir documentos.

La primera propuesta oficial para convertir HTML en un estándar se realizó en 1993 por parte del organismo [IETF](#) (*Internet Engineering Task Force*)

En 1995, el organismo IETF organiza un grupo de trabajo de HTML y consigue publicar, el 22 de septiembre de ese mismo año, el estándar HTML 2.0.

A partir de 1996, los estándares de HTML los publica otro organismo de estandarización llamado [W3C](#) (*World Wide Web Consortium*)

La versión HTML 3.2 se publicó el 14 de Enero de 1997 y es la primera recomendación de HTML publicada por el W3C.

HTML 4.0 se publicó el 24 de Abril de 1998. Novedades: hojas de estilos CSS, la posibilidad de incluir pequeños programas o *scripts* en las páginas web, mejora de la accesibilidad de las páginas diseñadas, tablas complejas y mejoras en los formularios..

En 2006, el W3C se interesó en el desarrollo de HTML5, y en 2007 se unió al grupo de trabajo del WHATWG para unificar el proyecto.

Novedades

Incorpora etiquetas (canvas 2D y 3D, audio, vídeo) con codecs para mostrar los contenidos multimedia.

Etiquetas para manejar grandes conjuntos de datos: Datagrid, Details, Menu y Command. Permiten generar tablas dinámicas que pueden filtrar, ordenar y ocultar contenido en cliente.

Mejoras en los formularios. Nuevos tipos de datos (eMail, number, url, datetime ...) y facilidades para validar el contenido sin Javascript.

Visores: MathML (fórmulas matemáticas) y SVG (gráficos vectoriales). En general se deja abierto a poder interpretar otros lenguajes XML.

Drag & Drop. Nueva funcionalidad para arrastrar objetos como imágenes.

Web Semántica

Web Semántica (Web 3.0): header, footer, article, nav, time (fecha del contenido), link rel="" (tipo de contenido que se enlaza).

Estas etiquetas permiten describir cuál es el significado del contenido. Por ejemplo su importancia, su finalidad y las relaciones que existen.

Los buscadores podrán indexar e interpretar esta meta información para no buscar simplemente apariciones de palabras en el texto de la página.

Nuevas APIs(*Application Programming Interface*) y Javascript

API para hacer Drag & Drop. Mediante eventos.

API para trabajar Off-Line. Permite descargar todos los contenidos necesarios y trabajar en local.

API de Geolocalización para dispositivos que lo soporten.

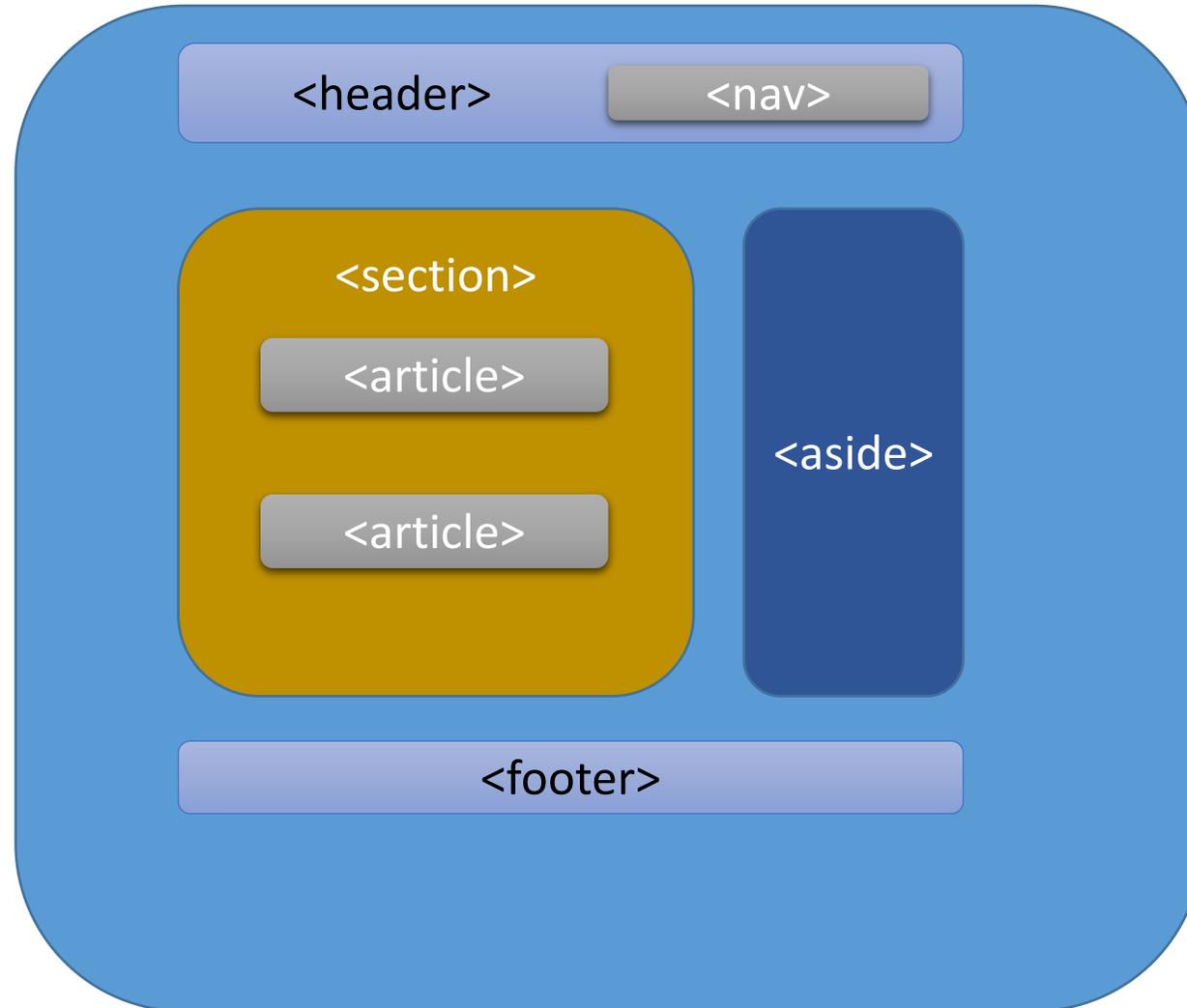
API Storage. Facilidad de almacenamiento persistente en local, con bases de datos (basadas en SQLite) o con almacenamiento de objetos por aplicación o por dominio Web (Local Storage y Global Storage). Se dispone de una Base de datos con la posibilidad de hacer consultas SQL.

WebSockets. API de comunicación bidireccional entre páginas. Similar a los Sockets de C.

WebWorkers. Hilos de ejecución en paralelo.

Estándar futuro. System Information API. Acceso al hardware a bajo nivel: red, ficheros, CPU, memoria, puertos USB, cámaras, micrófonos...

ESTRUCTURA BÁSICA HTML5



```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta content="text/html; charset=utf-8" http-equiv="content-type">
    <title>Titulo de la web</title>
  </head>
  <body>
    <header>
      <h1>Mi sitio web</h1>
      <p>Mi sitio web creado en html5</p>
    </header>
    <nav>Navegacion<br>
      <ul>
        <li>Inicio</li>
        <li>Programas</li>
        <li>Servicios</li>
        <li>Blog</li>
      </ul>
    </nav>
    <section>
      <article>
        <h2>Título de contenido
        <p> Contenido (además de imágenes, citas, videos etc.) </p></h2>
      </article>
    </section>
    <aside>
      <h3>Título de contenido</h3>
      <p>contenido</p>
    </aside>
    <footer> Creado por mi el 2016 </footer>
  </body>
</html>
```

Mi sitio web

Mi sitio web creado en html5

Navegacion

- Inicio
- Programas
- Servicios
- Blog

Título de contenido

Contenido (además de imágenes, citas, videos etc.)

Título de contenido

contenido

Creado por mi el 2016

Elemento raíz

<!doctype html> Define que el documento esta bajo el estándar de HTML 5

<html> Representa la raíz de un documento HTML o XHTML. Todos los demás elementos deben ser descendientes de este elemento.

https://developer.mozilla.org/es/docs/HTML/HTML5/HTML5_lista_elementos#Elemento_raíz

Metadatos del documento

<head> Representa una colección de metadatos acerca del documento, incluyendo enlaces a, o definiciones de, scripts y hojas de estilo.

<title> Define el título del documento, el cual se muestra en la barra de título del navegador o en las pestañas de página. Solamente puede contener texto y cualquier otra etiqueta contenida no será interpretada.

<base> Define la URL base para las URLs relativas en la página.

<link> Usada para enlazar JavaScript y CSS externos con el documento HTML actual.

<meta> Define los metadatos que no pueden ser definidos usando otro elemento HTML.

<style> Etiqueta de estilo usada para escribir CSS en línea.

Scripting

<script> Define ya sea un script interno o un enlace hacia un script externo. El lenguaje de programación es JavaScript

<noscript> Define un contenido alternativo a mostrar cuando el navegador no soporta scripting.

Secciones

<body> Representa el contenido principal de un documento HTML. Solo hay un elemento <body> en un documento.

<section> Define una sección en un documento.



<nav> Define una sección que solamente contiene enlaces de navegación.



<article> Define contenido autónomo que podría existir independientemente del resto del contenido.



<aside> Define algunos contenidos vagamente relacionados con el resto del contenido de la página. Si es removido, el contenido restante seguirá teniendo sentido



<h1>,<h2>,<h3>,<h4>,<h5>,<h6> Los elementos de cabecera implementan seis niveles de cabeceras de documentos; <h1> es la de mayor y <h6> es la de menor importancia. Un elemento de cabecera describe brevemente el tema de la sección que introduce.

<header> Define la cabecera de una página o sección. Usualmente contiene un logotipo, el título del sitio Web y una tabla de navegación de contenidos.



<footer> Define el pie de una página o sección. Usualmente contiene un mensaje de derechos de autoría, algunos enlaces a información legal o direcciones para dar información de retroalimentación.



<address> Define una sección que contiene información de contacto.

<main> Define el contenido principal o importante en el documento. Solamente existe un elemento <main> en el documento.



Agrupación de Contenido

- <p>** Define una parte que debe mostrarse como un párrafo.
- <hr>** Representa un quiebre temático entre párrafos de una sección , artículo o cualquier contenido.
- <pre>** Indica que su contenido está preformateado y que este formato debe ser preservado.
- <blockquote>** Representa una contenido citado desde otra fuente.
- ** Define una lista ordenada de artículos.
- ** Define una lista de artículos sin orden.
- ** Define un artículo de una lista enumerada.
- <dl>** Define una lista de definiciones, es decir, una lista de términos y sus definiciones asociadas.
- <dt>** Representa un término definido por el siguiente <dd>.
- <dd>** Representa la definición de los términos listados antes que él.
- <figure>** Representa una figura ilustrada como parte del documento. 
- <figcaption>** Representa la leyenda de una figura. 
- <div>** Representa un contenedor genérico sin ningún significado especial.

Semántica a nivel de Texto

- <a>** Representa un hiperenlace.
- ** Representa un texto enfatizado , como un acento de intensidad.
- ** Representa un texto especialmente importante .
- <small>** Representa un comentario aparte. Letra pequeña
- <cite>** Representa el título de una obra .
- <time>** Representa un valor de fecha y hora.
- <code>** Representa un código de ordenador .
- <var>** Representa a una variable y resalta el texto.
- <sub>,<sup>** Representan un subíndice y un superíndice, respectivamente.
- <i>** Itálica o cursiva
- ** Negrita.
- <u>** Subrayado
- <mark>** Representa texto resaltado con propósitos de referencia, es decir por su relevancia en otro contexto.
- ** Sirve para aplicar estilo al texto o agrupar elementos en línea.
-
** Representa un salto de línea.
- <wbr>** Representa una oportunidad de salto de línea, es decir, un punto sugerido de envoltura donde el texto de múltiples líneas puede ser dividido para mejorar su legibilidad.

Contenido incrustado

**** Representa una imagen.

<iframe> Representa un contexto anidado de navegación, es decir, un documento HTML embebido.

<embed> Representa un punto de integración para una aplicación o contenido externo que por lo general no es HTML.

<object> Representa un recurso externo, que será tratado como una imagen, un sub-documento HTML o un recurso externo a ser procesado por un plugin.

<param> Define parámetros para el uso por los plugins invocados por los elementos <object>.

<video> Representa un video .

<audio> Representa un sonido o stream de audio.

<source> Permite a autores especificar recursos multimedia alternativos para <video> o <audio>.

<canvas> Representa un área de mapa de bits en el que se pueden utilizar scripts para renderizar gráficos como gráficas, gráficas de juegos o cualquier imagen visual al vuelo.

<map> En conjunto con <area>, define un mapa de imagen.

<area> En conjunto con <map>, define un mapa de imagen.

<svg> Define una imagen vectorial embebida.

<math> Define una fórmula matemática.

Elementos interactivos

- <details>** Representa un widget desde el que un usuario puede obtener información o controles adicionales.
- <summary>** Representa un resumen, título o leyenda para un elemento `<details>` dado.
- <command>** Representa un comando que un usuario puede invocar.
- <menu>** Representa una lista de comandos .

Datos tabulares

<table>	Representa datos con más de una dimensión.
<caption>	Representa el título de una tabla.
<colgroup>	Representa un conjunto de una o más columnas de una tabla.
<col>	Representa una columna de una tabla.
<tbody>	Representa el bloque de filas que describen los datos concretos de una tabla.
<thead>	Representa el bloque de filas que describen las etiquetas de columna de una tabla.
<tfoot>	Representa los bloques de filas que describen los resúmenes de columna de una tabla.
<tr>	Representa una fila de celdas en una tabla.
<td>	Representa una celda de datos en una tabla.
<th>	Representa una celda encabezado en una tabla.

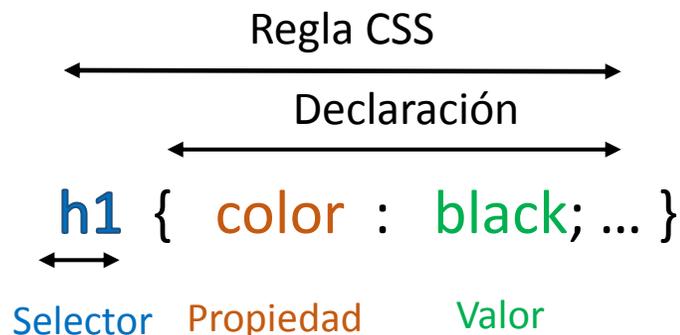
Formularios

- <form>** Representa un formulario, controles que puede ser enviado a un servidor para procesamiento.
- <fieldset>** Representa un conjunto de controles.
- <legend>** Representa el título de un <fieldset>.
- <label>** Representa el título de un control de formulario.
- <input>** Representa un campo de datos escrito que permite al usuario editar los datos.
- <button>** Representa un botón .
- <select>** Representa un control que permite la selección entre un conjunto de opciones.
- <datalist>** Representa un conjunto de opciones predefinidas para otros controles.
- <optgroup>** Representa un conjunto de opciones, agrupadas lógicamente.
- <option>** Representa una opción en un elemento <select>, o una sugerencia de un elemento <datalist>.
- <textarea>** Representa un control de edición de texto multi-línea.
- <keygen>** Representa un control de par generador de llaves.
- <output>** Representa el resultado de un cálculo.
- <progress>** Representa el progreso de finalización de una tarea.
- <meter>** Representa la medida escalar (o el valor fraccionario) dentro de un rango conocido.

CSS HOJAS DE ESTILO

Las hojas de estilo sirven para separar completamente el contenido del diseño de una página web, de tal forma que, si lo estructuramos correctamente, es posible cambiar totalmente el aspecto de nuestra web haciendo modificaciones en las hojas de estilo (css) sin tocar ni una sola línea de su contenido (html).

Componentes de un estilo CSS básico



- **Regla:** cada uno de los estilos que componen una hoja de estilos CSS.
 - **Selector:** indica el elemento o elementos HTML a los que se aplica la regla CSS.
 - **Declaración:** especifica los estilos que se aplican a los elementos. Está compuesta por una o más propiedades CSS.
 - **Propiedad:** característica que se modifica en el elemento seleccionado, como por ejemplo su tamaño de letra, su color de fondo, etc.
 - **Valor:** establece el nuevo valor de la característica modificada en el elemento.

Existen tres formas diferentes de integrar las hojas de estilo con HTML.

1.- Atributo style junto con cualquier etiqueta HTML

```
<h1 style="color: red; font-family: Verdana, Arial;" >Título</h1>
```

2.- Dentro de la cabecera (head) donde se insertarían los diferentes estilos

```
<style>  
  h1 { color: red; font-family: Verdana, Arial; }  
  p { color: gray; font-family: Times; }  
</style>
```

3.- Sección head indicar la ruta a un archivo externo donde se incluirían los diferentes estilos.

```
<link rel="stylesheet" type="text/css" href="estilos.css" />
```

Dentro del fichero incluir `p { color: gray; font-family: Times; }`

Selectores básicos

Selector universal

```
* { margin: 0; padding: 0; }
```

Selector de tipo o etiqueta

```
p { ... }      h1, h2, h3 { color: #8A8E27; font-family: Arial, Helvetica, sans-serif; }
```

Selector descendente

```
p a { color: red; }      Muestra en rojo todos los elementos tipo <a> que a su vez estén dentro de la etiqueta <p>
```

Selector de clase

Usar el atributo **class** de HTML para indicar directamente la regla CSS que se aplica

```
<p class="destacado">IES MANJON..</p>
```

Regla **.destacado { color: red; }**

Selectores de ID

```
<p>Primer párrafo</p>
```

```
<p id="destacado">Segundo párrafo</p>
```

```
<p>Tercer párrafo</p>
```

Regla **#destacado { color: red; }**

COLORES

Los colores en CSS se pueden indicar de cinco formas diferentes: palabras clave, colores del sistema, RGB hexadecimal, RGB numérico y RGB porcentual.

Palabras clave:

aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, orange, purple, red, silver, teal, white, yellow

RGB decimal

```
p { color: rgb(71, 98, 176); }
```

RGB porcentual

```
p { color: rgb(27%, 38%, 69%); }
```

RGB hexadecimal

```
p { color: #4762B0; }
```

```
body { background-color: #FFFFFF; color: #000000; }  
h1, h2, h3, h4, h5, h6 { color: #CC0000; }
```

Fondo blanco, letra negra, y titulares en rojo

Tipografía

color Color del texto

Valores <color> | inherit

font-family Tipo de letra

Valores ((<nombre_familia> | <familia_generica>) (, <nombre_familia> | <familia_generica>)*) | inherit

font-size Tamaño de letra

Valores <tamaño_absoluto> | <tamaño_relativo> | <medida> | <porcentaje> | inherit

font-weight Anchura de la letra

Valores normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | inherit

font-style Estilo de la letra

Valores normal | italic | oblique | inherit

font-style Estilo de la letra

Valores normal | italic | oblique | inherit

PROPIEDADES DEL TEXTO

text-align Alineación del texto

Valores left | right | center | justify | inherit

line-height Interlineado

Valores normal | <numero> | <medida> | <porcentaje> | inherit

text-decoration Decoración del texto

Valores none | (underline || overline || line-through || blink) | inherit

text-transform Transformación del texto

Valores capitalize | uppercase | lowercase | none | inherit

vertical-align Alineación vertical

Valores baseline | sub | super | top | text-top | middle | bottom | text-bottom | <porcentaje> | <medida> | inherit

text-indent Tabulación del texto

Valores <medida> | <porcentaje> | inherit

letter-spacing Espaciado entre letras

Valores normal | <medida> | inherit

word-spacing Espaciado entre palabras

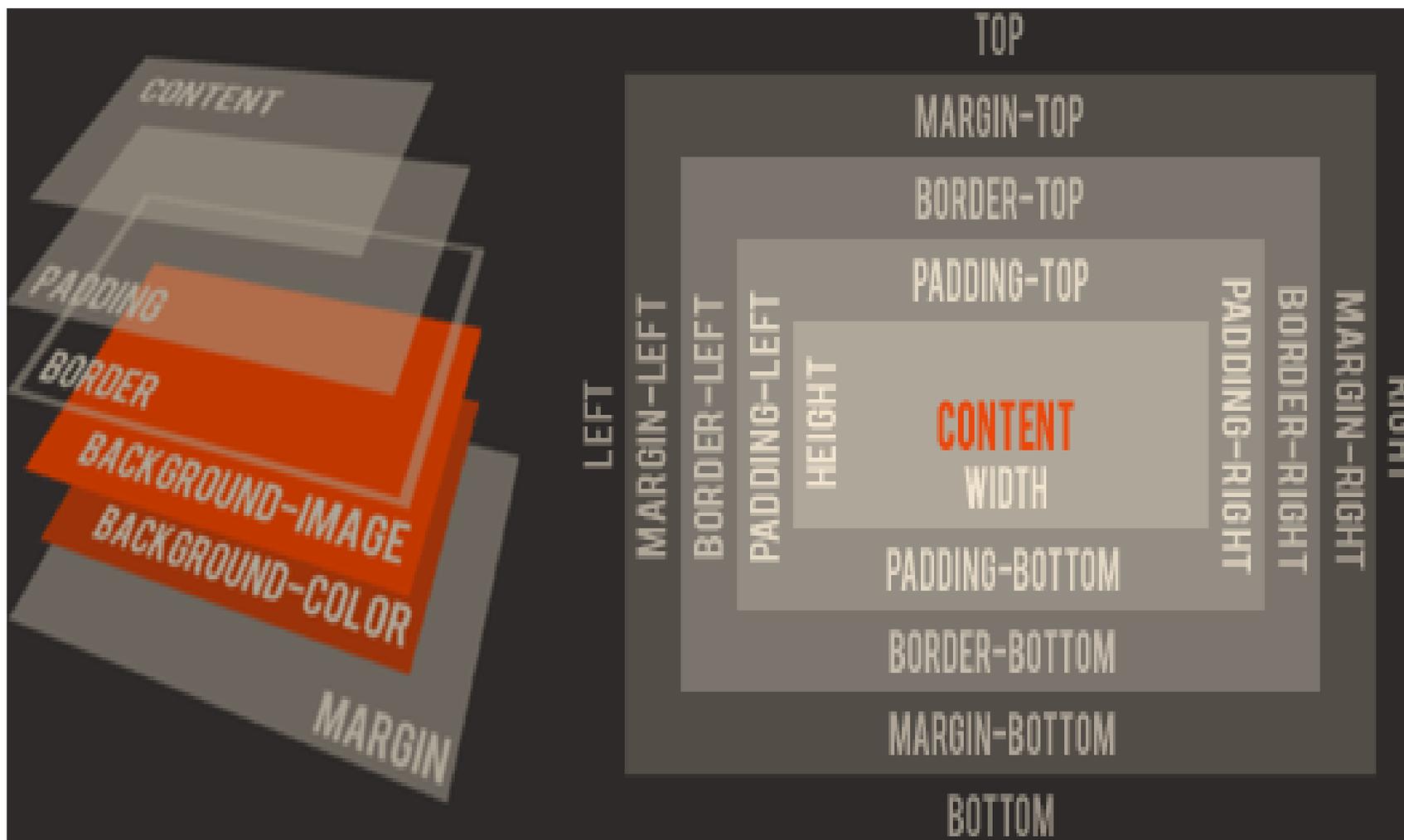
Valores normal | <medida> | inherit

white-space Tratamiento de los espacios en blanco

Valores normal | pre | nowrap | pre-wrap | pre-line | inherit

Modelo de cajas (box model)

Las cajas de una página se crean automáticamente. Cada vez que se inserta una etiqueta o elemento en la página, se crea una nueva caja rectangular que encierra los contenidos del elemento.



Características de JavaScript

¿Qué es JavaScript?

Lenguaje de programación interpretado utilizado fundamentalmente para dotar de comportamiento dinámico a las páginas web.

Cualquier navegador web actual incorpora un intérprete para código JavaScript (se puede desactivar).

- Su sintaxis se asemeja a la de C++ y Java.
- Sus objetos utilizan herencia basada en prototipos.
- Es un lenguaje débilmente tipado.

“Hola mundo” con JavaScript

Hay diferentes formas de embeber el código JavaScript en una página HTML:

1. Incluirlo directamente en la página HTML mediante la etiqueta `<script>`.

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv= "content-type"
  content="text/html;charset=utf-8">
  <title>HolaMundo</title>
</head>
<body>
  <script>
    alert("Hola mundo en JavaScript")
  </script>
</body>
</html>
```

2. Utilizar el atributo `src` de la etiqueta `<script>` para especificar el fichero que contiene el código JavaScript.

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type" content="text/html;
  charset=utf-8">
  <title>HolaMundo</title>
  <script type="text/javascript" src="HolaMundo.js"></script>
</head>
<body></body>
</html>
```

3. Incluir JavaScript en los elementos XHTML

```
<!DOCTYPE html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8">
  <title>Ejemplo de código JavaScript en el propio documento</title>
</head>
<body>
  <p onclick="alert('Hola mundo')">Un párrafo de texto.</p>
</body>
</html>
```

Quando el navegador no soporta JavaScript

La etiqueta `<noscript>` permite definir qué texto proporcionar al usuario cuando el navegador no soporta o no tiene activado JavaScript

Esta etiqueta tiene que ir dentro del `<body>`

```
<body>
  <noscript>
    <p>
      Esta página requiere JavaScript para su correcto funcionamiento.
      Compruebe si JavaScript está deshabilitado en el navegador.
    </p>
  </noscript>
  <p>Texto del documento</p>
</body>
```

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta content="text/html; charset=utf-8" http-equiv="content-type">
    <title>script Hola Mundo</title>
  </head>
  <body>
    <script >
      alert("Hola Mundo!");
    </script>

    <noscript>
      <p>
        Esta página requiere JavaScript para su correcto funcionamiento.
        Compruebe si JavaScript está deshabilitado en el navegador.
      </p>
    </noscript>

  </body>
</html>
```

JavaScript puede cambiar el contenido HTML

Uno de los muchos métodos JavaScript HTML es **getElementById ()** .

Busca un elemento HTML en el documento (con id = "prueba") y cambia el contenido del elemento (**innerHTML**) por "IES PADRE MANJÓN"

```
<!DOCTYPE html>
```

```
<!--ejercicio06-->
```

```
<html>
```

```
<body>
```

```
<h1>Javascript</h1>
```

```
<p id="prueba">JavaScript cambia el contenido HTML </p>
```

```
<button type="button" onclick='document.getElementById("prueba").innerHTML = "IES PADRE MANJON"'>Pulsa!</button>
```

```
</body>
```

```
</html>
```

Variables y expresiones

```
<!DOCTYPE html>
<!--ejercicio07-->
<html>
<body>
  <h1>Javascript</h1>

  <p id="suma">Resultado suma </p>
  <p id="texto">Resultado texto </p>

  <script>
    var x, y, nombre;
    x = 5;
    y = 10;
    nombre="PEPE";

    document.getElementById("suma").innerHTML = "La suma es:" + (x+y);
    document.getElementById("texto").innerHTML = "Tu nombre=" + nombre;
  </script>
</body>
</html>
```

Funciones en Javascript

```
<html>
<!--ejercicio05-->

<head>
  <script language=javascript>
    function openwindow()
    {
      m = window.open("http://www.ideal.es");
    }
    function closewindow()
    {
      m.close()
    }
  </script>
</head>
<body>
  <button type="button" onclick="openwindow()">Abrir!</button>
  <button type="button" onclick="closewindow()">Cerrar!</button>
</body>
</html>
```

Mostrar en una ventana el valor introducido en una caja de texto

```
<html>
<!--ejercicio03-->

<head>
  <script type="text/javascript">
    function myfunction(txt)
    {
        alert(txt)
    }
  </script>
</head>

<body>
  <form name="form1">
    <input name="entrada">
    <input type="button" onclick="myfunction(form1.entrada.value)" value="Llamada a función">
  </form>
  <p>Pulsa el botón y verás el texto del formulario</p>
</body>
</html>
```

Bucles Javascript

```
<!DOCTYPE html>
<!--ejercicio04-->

<html>
  <body>
    <script >
      for (i = 1; i <= 6; i++){
        document.write("<h" + i + ">Cabecera de nivel " + i)
        document.write("</h" + i + ">")
      }
    </script>
  </body>
</html>
```

```
<!DOCTYPE html>
<!--ejercicio08-->
<html>
  <body>

  <h1>JavaScript Loops</h1>

  <p id="demo"></p>

  <script>
    var cars = ["BMW", "Volvo", "Saab", "Ford", "Fiat", "Audi"];
    var text = "";
    var i=0;
    while (i < cars.length) {
      text += cars[i] + "<br>";
      i++;
    } document.getElementById("demo").innerHTML = text;
  </script>

  </body>
</html>
```

CANVAS

```
<html>
<head>
  <script>
    function draw() {
      var canvas = document.getElementById("canvas");
      if (canvas.getContext) {
        var ctx = canvas.getContext("2d");

        ctx.fillStyle = "rgb(200,0,0)";
        ctx.fillRect (10, 10, 100, 100);

        ctx.fillStyle = "rgba(0, 0, 200, 0.5)";
        ctx.fillRect (30, 30, 100, 100);
      }
    }
  </script>
</head>
<body onload="draw();" >
  <canvas id="canvas" width="150" height="150"></canvas>
</body>
</html>
```