

HTML5



HTML + CSS + JAVASCRIPT





¿Cuándo y cómo comenzó todo?

Evolución Histórica

➡ **1989** – Tim Berners Lee (CERN), propone sistema para compartir documentos entre investigadores.

Berners Lee propone un sistema de hipertexto (texto que contiene enlaces a otros textos) basado en Internet.

➡ **1991** – “*HTML Tags*”, primer documento que describe 18 elementos iniciales y simples para construir un documento HTML



¿Cuándo y cómo comenzó todo?

Evolución Histórica

➡ **1994** – Primera conferencia de WWW en Ginebra, presentación de HTML+ . IETF (Internet Engineering Task Force) forma un grupo de trabajo HTML. Se libera la versión HTML 2. Se crea Netscape. Se forma el WWW Consortium

➡ **1995** – Se crean numerosas etiquetas nuevas para HTML y sale la versión HTML 3 (Tablas). Nace Internet Explorer. Netscape propone el uso de Frames. Se comienzan a usar las primeras hojas de estilos (CSS)



¿Cuándo y cómo comenzó todo?

Evolución Histórica

➔ **1997** – Versión HTML 3.2

➔ **2000** – XHTML 1.0, mayor estructura de los documentos.

➔ **2002 a 2009** – XHTML 2.0



➔ **2008** – Primera versión de HTML5. Etiquetas semánticas. (<nav>, <article>, ...)

➔ **1999** – Versión HTML 4.01

➔ **2001** – XHTML 1.1

➔ **2006** – Primeros pasos de HTML5 a partir de la colaboración entre WHATWG (Web Hypertext Application Tech Working Group) y W3C

➔ **2018** – HTML 5.2



GOBIERNO
DE ESPAÑA
MINISTERIO
DE INDUSTRIA, COMERCIO
Y TURISMO

EQI Escuela de
organización
industrial



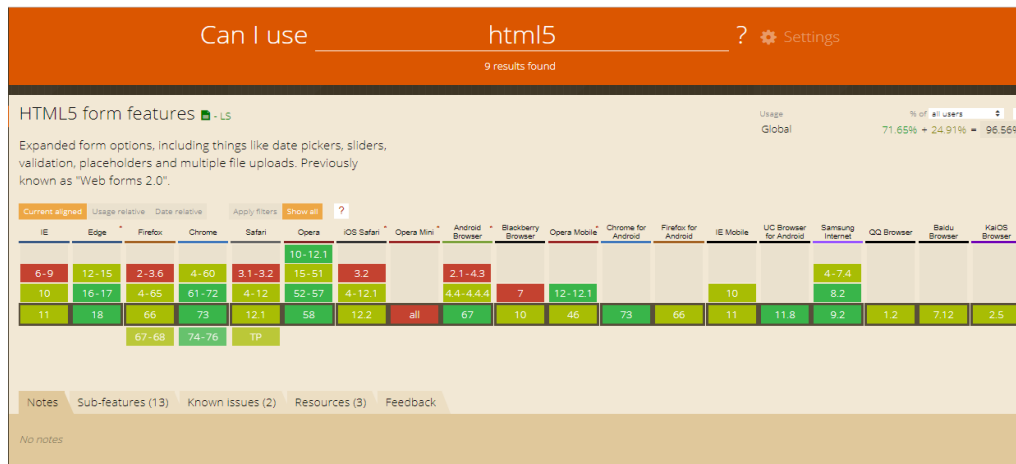
Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro





¿Soportan todos los navegadores HTML5?

Soporte de HTML5 en Navegadores



Nota : Los navegadores pueden leer cualquier tipo de documento de texto plano

<https://caniuse.com/#search=html5>



¿Qué nos ofrece HTML5?

Muchas mejoras

Entre las mejoras más significativas podemos destacar :

- Etiquetas descriptivas (semánticas)
- Mejora en el desarrollo de Aplicaciones Web
- Mejora en el manejo de animaciones (Animaciones con CSS)
- Mejora en el soporte multimedia (Vídeo, Audio, ...)
- Mejora en las comunicaciones (WebRtc, ...)
- Simplicidad en las declaraciones y creación de elementos (Nos olvidamos del atributo type en link, script y style)
- Uso 100% de Javascript en los <script>
- Interfaces de usuario mejorada , gracias a las nuevas funcionalidades de HTML5, CSS3 y Javascript
- Mejora en los formularios
- Selectores avanzados en CSS3
- Nos ayuda a prevenir la divitis





¿Qué nos ofrece HTML5?

Muchas mejoras

Entre las mejoras más significativas podemos destacar :

- Mejora en la accesibilidad (Lectores de pantalla y otros)

<https://www.w3.org/WAI/>

- Mejora en el rendimiento y carga

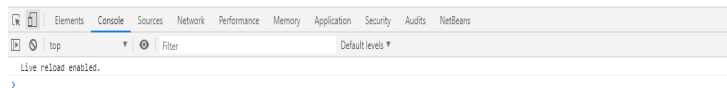
<https://developers.google.com/speed/?csw=1>



Herramientas necesarias

Google DevTools

- 1) Abrirmos el navegador Google Chrome
- 2) Pulsamos F12



- 3) Device ToolBar, Console, Sources y Network

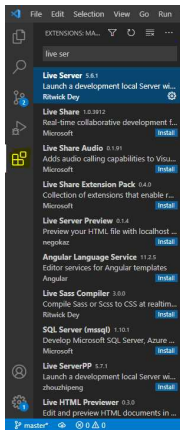


Herramientas necesarias

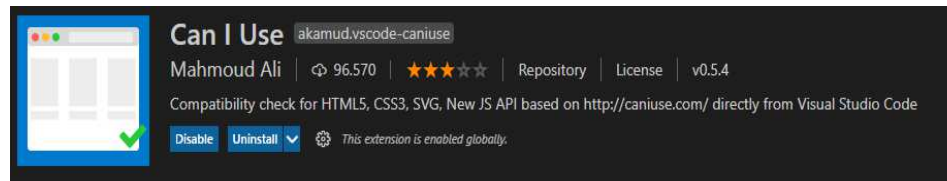
Visual Studio

1) Visual Studio (<https://code.visualstudio.com/>)

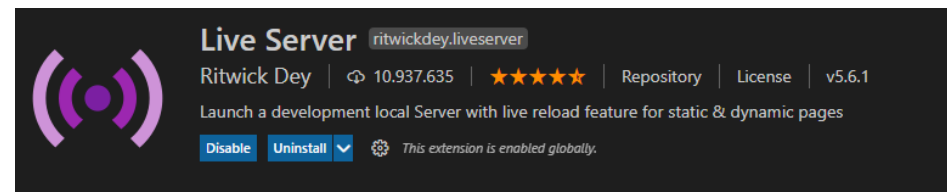
•Extensiones:



— Can I Use



— Live Server



GOBIERNO
DE ESPAÑA

MINISTERIO
DE INDUSTRIA, COMERCIO
Y TURISMO



Escuela de
organización
industrial



Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro

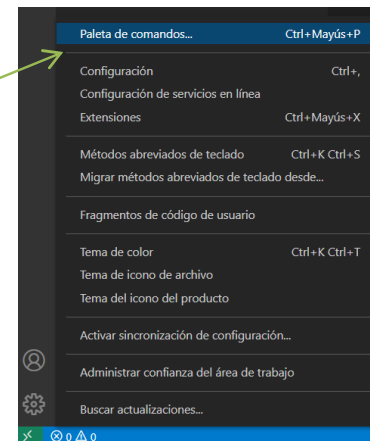
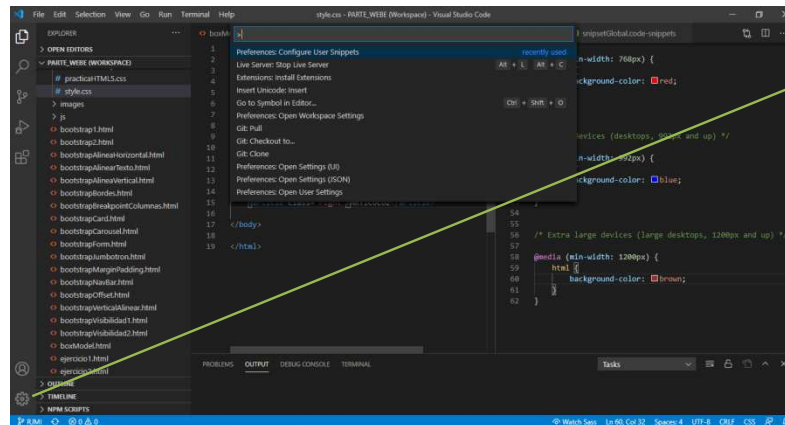




Herramientas necesarias

Visual Studio

- Configuración de Snippets
- El uso de Snippets nos facilita el trabajo del día a día
- **CTRL + MAYÚSCULAS + P**: nos abre la consola de comandos sobre **vscode**, también lo podemos hacer clickando en el engranaje que hay en la parte inferior izquierda -> Paleta de comandos

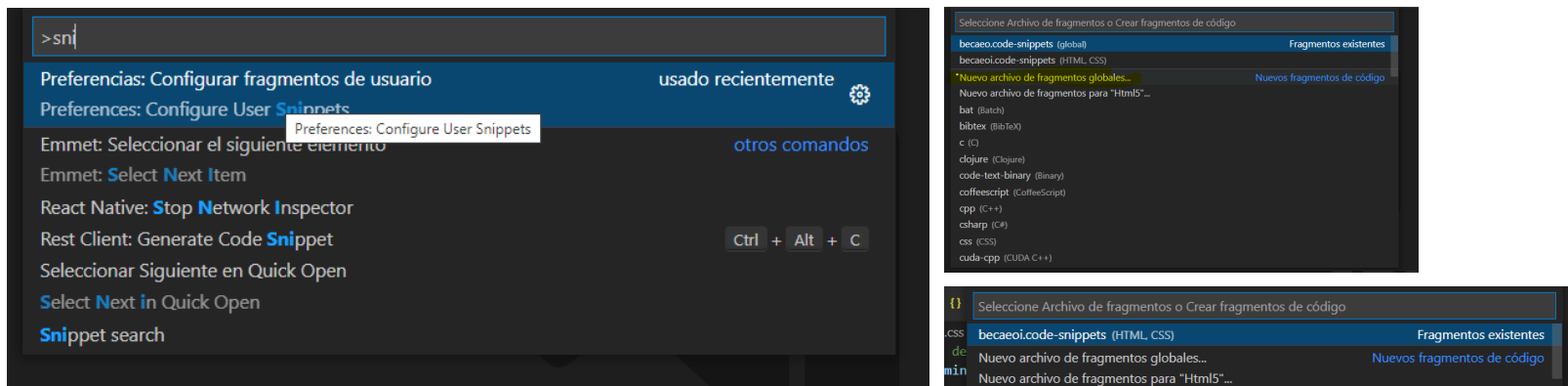




Herramientas necesarias

Visual Studio

- Tecleamos “snippet” y nos aparecerá la opción para configurarlos (Lo configuraremos de manera Global y le pondremos el nombre que aparece en la imagen de abajo, sólo pondremos el nombre **becaeoi**, vscode añadirá el resto de nombre)





Herramientas necesarias

Visual Studio

- Ejemplo configuración de Snippet

```
# style.css x {} snippetGlobal.code-snippets x
11 // "prefix": "log",
12 // "body": [
13 //   "console.log('$1');",
14 //   "$2"
15 // ],
16 // "description": "Log output to console"
17 // }
18 "til": {
19   "scope": "html,javascript,css",
20   "prefix": "til",
21   "body": "~"
22 },
23 "fe": {
24   "scope": "html,javascript",
25   "prefix": "fe",
26   "body": "<b>${CURRENT_DATE}/${CURRENT_MONTH}/${CURRENT_YEAR} ${CURRENT_HOUR}:${CURRENT_MINUTE}</b> : "
27 },
28 "res": {
29   "scope": "html,javascript",
30   "prefix": "res",
31   "body": "<br><b>RESPUESTA </b> : "
32 },
33 "dailytsb": {
34   "scope": "html,javascript",
35   "prefix": "dailytsb",
36   "body": " <details class='daily'><summary></summary><p><article><ul><li><b>Gestión</b></li>"
37 },
38 "detail": {
```



Herramientas necesarias

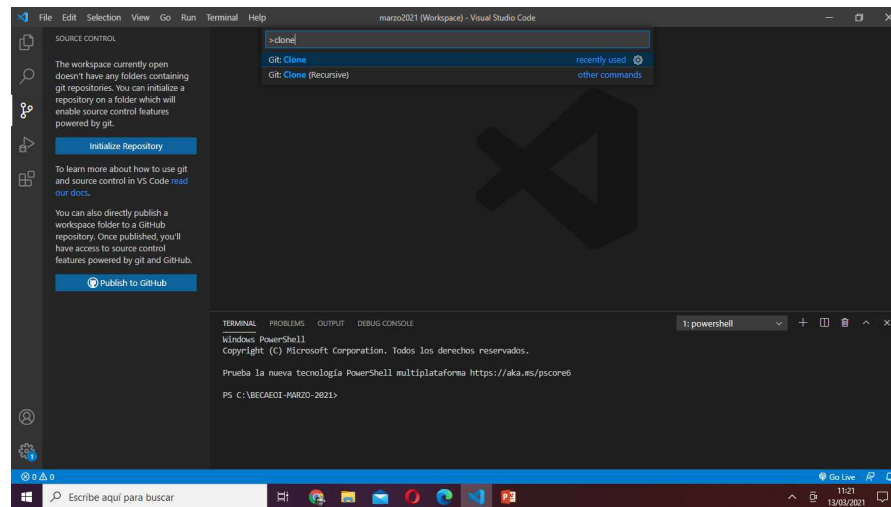
Git

1) Crear una carpeta en vuestro ordenador y clonar el siguiente repositorio, si ya la tenéis clonada en alguna carpeta no hace falta volver a clonarla, sólo tendremos que añadirla al área de trabajo.

[CTRL+MAYÚSCULAS+P](#)

[Git Clone](#)

<https://gitlab.com/e4320/becaeoi.git>



GOBIERNO
DE ESPAÑA

MINISTERIO
DE INDUSTRIA, COMERCIO
Y TURISMO



Escuela de
organización
industrial



Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro

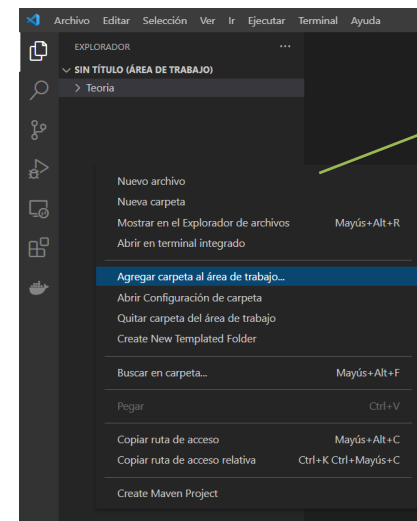
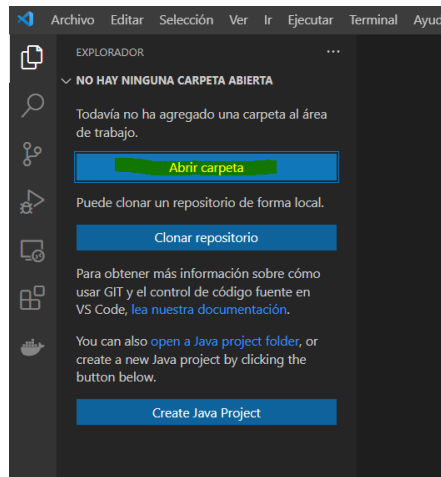




Herramientas necesarias

Git

Agregando una carpeta al área de trabajo : en la parte izquierda de **vscode** , según lo que nos aparezca lo podremos hacer de dos maneras.

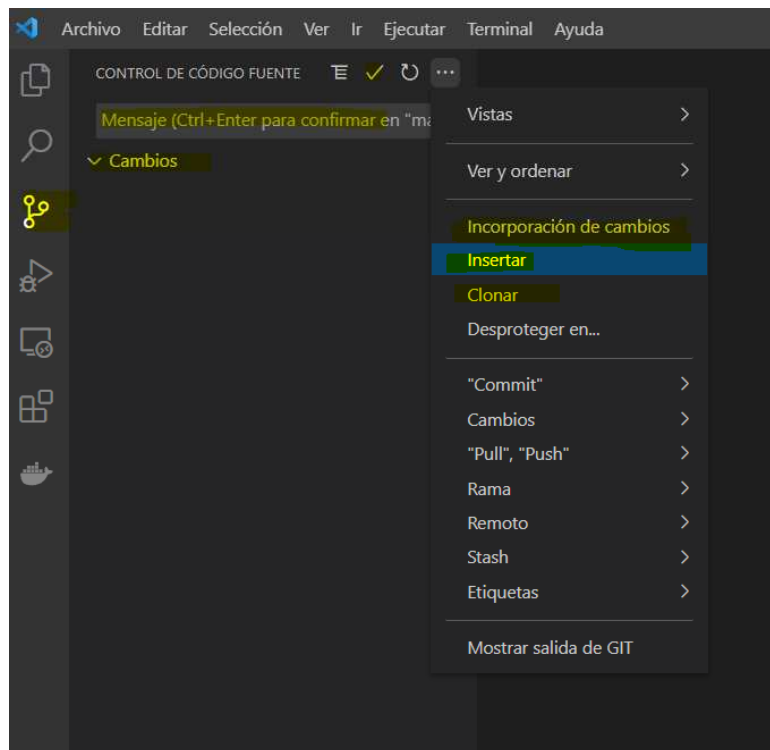


Este cuadro aparece
clickando en el botón
derecho sobre esta
sección



Herramientas necesarias

Git



3) Guardaremos todos los ejercicios y ejemplos en la rama con nuestras iniciales. Recordar al final del día subir todos los cambios al repositorio (commit y push, pero recordad siempre antes hacer pull)

Para tenerlo todo bien organizado ,crear dentro de vuestra rama una carpeta por cada tema (HTML, CSS, JAVASCRIPT, TYPESCRIPT) y guardad todos los ejercicios en la carpeta del tema correspondiente.



Herramientas necesarias

Acceso a MDN (Mozilla Developer Network)

¡¡ MUY IMPORTANTE Y VÁLIDO PARA TODA LA PARTE FRONT !!

Podemos buscar fácilmente información sobre cualquiera de los componentes que vamos a ver en la parte de HTML5, CSS y JAVASCRIPT.

Por ejemplo : Si quiero buscar información sobre el elemento `<select>` pondría esto en mi buscador Web : **mdn select** y el primer enlace que aparezca contendrá información sobre el elemento buscado

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/select>





HTML5

Estructura básica de documento Html5

Estructura básica

Documento Html5 básico

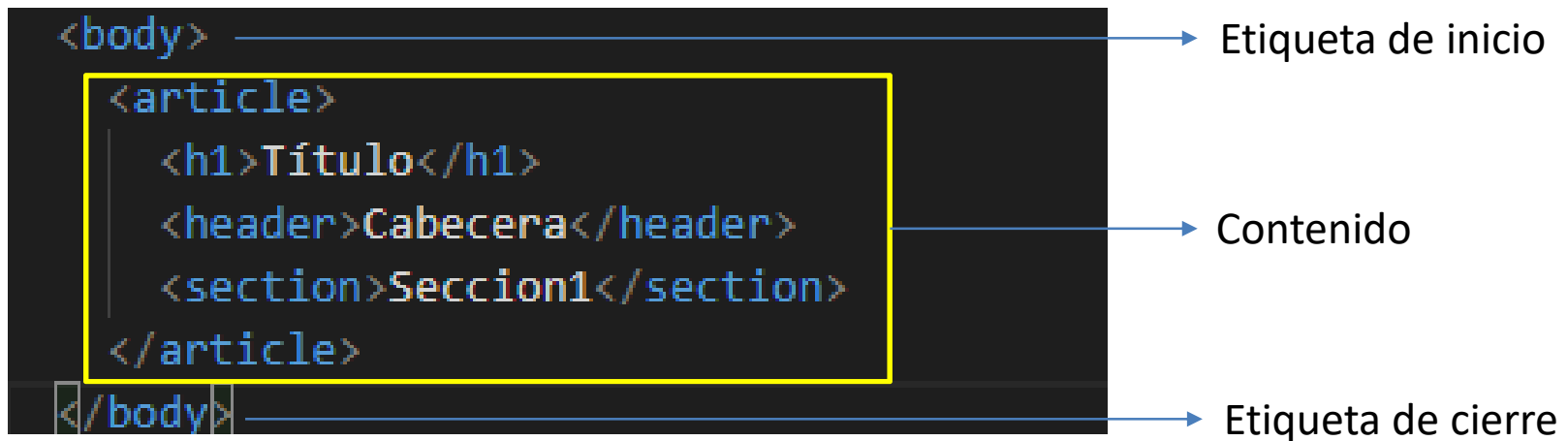
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8   <link rel="stylesheet" href="css/style.css">
9 </head>
10 <body>
11
12   <script src="js/fichero.js"></script>
13 </body>
14 </html>
```



Elementos HTML

Identificar los elementos HTML

- Una etiqueta de inicio y de cierre con el nombre del elemento y entre ellas el contenido (texto u otros elementos anidados)





Elementos HTML

Identificar los elementos HTML

- Los elementos HTML pueden contener atributos, los más importantes con : **id**, **class**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Document</title>
  <link rel="stylesheet" href="css/style.css">
</head>
<body>
  <header class="cabecera" id="cabecera1"></header>
  <script src="js/fichero.js"></script>
</body>
</html>
```



Identificador y clase

Atributo id

Id

`<section id="id1">.....</section>` : identificador del elemento, que debe ser único para todo el documento. Se suele usar para localizar el elemento y cambiar estilos, texto, ... sobre el mismo.

Ejemplo : Enlace a un elemento dentro del mismo documento

En el HTML :

```
<section id="id1">.....</section>
<a href="#id1">Posicionarnos en id1</a>
```

En el CSS :

```
#id{ color:red; }
```





Atributos Identificador y clase

Atributo class

class

`<section class="id1">...</section>` : hace referencia a una clase CSS de la cual se aplicarán los estilos a ese elemento.

Ejemplo : aplicar estilo colorRojo

En el HTML :

`<section class="colorRojo">...</section>`

En el CSS :

`.colorRojo { color:red; }`



Principales elementos HTML

doctype, html

<!DOCTYPE html>

Se coloca al inicio del documento. Un navegador puede leer cualquier tipo de documento de texto plano y tenemos que indicarle qué tipo de documento está leyendo para que sepa procesarlo y mostrarlo correctamente. Esta etiqueta debe ir al inicio del documento HTML y le indica al navegador que todo lo que viene a continuación lo tiene que tratar como código HTML5. A la hora de procesar el documento analizará que las etiquetas usadas para definir los elementos sean correctas

<html lang="es | en ..." dir="ltr | rtl">

Elemento padre del documento HTML.

El atributo **lang** indica el idioma del documento (Los motores de búsqueda y lectores de pantalla pueden acceder a este atributo) y el atributo **dir** indica la dirección que van a seguir el texto para su visualización (ltr -> left to right)

https://en.wikipedia.org/wiki/List_of_ISO_639-1_codes.



Principales

head

<head>

Incluye la información necesaria para representar la página: título, codificación de caracteres y los archivos externos usados por el documento (CSS,Js).

<title>

Título de la página

<meta charset="utf8">

Codificación de caracteres de la página

<link href="css/style.css"> , <style>...</style>

Estilo CSS necesario para renderizar la página.

link se puede usar para establecer la relación la página con otros medios (Lectores del código de la página) (Impresora, pantalla, ...), es decir, cómo tratar la página al usar enviarla a estos medios.

<script src="js/javascript.js">

Código Javascript necesario para darle lógica a la página





HTML5

Elementos HTML5

Más sobre link

link

```
<link rel="icon" href="images/logoHTML5.png" type="image/png" sizes="16x16">
```

Con esto estamos definiendo el icono que va a parecer en la pestaña del navegador junto al Título de la página.

El valor del atributo **type** tiene que ser un valor **mime** válido y este indica el formato del contenido definido en **href**

https://developer.mozilla.org/es/docs/Web/HTTP/Basics_of_HTTP/MIME_types/Lista_completa_de_tipos_MIME





HTML5

Elementos HTML5

Más sobre head

base, meta

```
<base href="Dirección URL" target="_blank | _parent | _self | _top | nombreFrame" >
```

Define donde se deben abrir todos los enlaces definidos en la página

```
<meta charset="UTF-8">
```

```
<meta name="description" content="Curso HTML5 EOI">
```

```
<meta name="author" content="Mi nombre">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

- Representa los metadatos asociados al documento (descripción del documento, codificación de caracteres ...)
- Puede describir atributos con sus metadatos asociados. Esta etiqueta es muy importante en la aplicación de SEO y de motores de búsqueda.
- Una de las más importantes en HTML5 es la propiedad **viewport** que se usa para establecer el escalado de nuestra página dependiendo del dispositivo donde lo estemos visualizando. Establece el ancho con el ancho del dispositivo con el que se este visualizando la página y un Zoom inicial de 1 (Tamaño normal)





Nos ayudan a estructurar mejor nuestro documento

Elementos Html5 que van dentro del elemento **body**

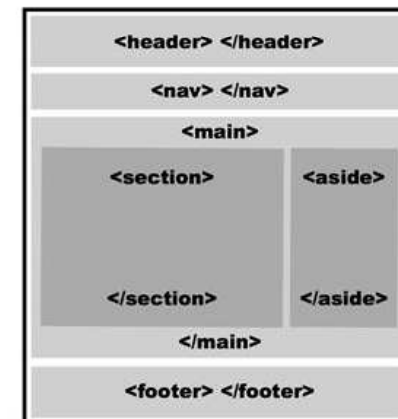
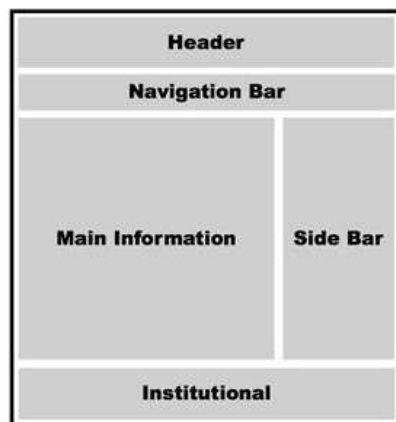
- **<header>** : define un encabezado para el cuerpo o las secciones dentro del cuerpo
- **<nav>** : define una sección que contiene la ayuda a la navegación (Menú principal del sitio Web o ayuda a la navegación)
- **<main>** : define una sección que representa el contenido principal de nuestra página / documento, el contenido que representa el tema principal del documento
- **<section>** : define una sección genérica. Se utiliza para organizar el contenido principal.
- **<aside>** : define una sección que contiene información relacionada con el contenido principal (referencias, vínculos, enlaces ...)
- **<article>** : define un elemento independiente del contenido (Una publicación, un artículo, entrada de un blog, un comentario ...)
- **<footer>** : define un pie para el cuerpo o las secciones dentro del cuerpo.
- **<div>** : define una sección genérica, cuando no aplica ninguno de los siguientes



Estructura

Elementos que van dentro del elemento **body**

- Los elementos anteriores se han creado con el fin de representar secciones específicas de una página Web. Se pueden usar con flexibilidad y se pueden aplicar a diferentes tipos de diseño.
- Siguen el patrón que mostramos en la siguiente imagen





Divitis

No abusar mucho de elementos **div**

- Antes de HTML5 `<div>` : se usaba (aún se sigue usando) el elemento `div` para crear una porción de información dentro de nuestra página, pero...

- El abuso de `<div>` es llamado divitis, no tengamos miedo, hay una vacuna para ello (HTML5)

```
<div>Esto  
  <div>es  
    <div>llamado  
      <div>Divitis  
    </div>  
  </div>  
</div>  
</div>
```



- Una página con muchos **<div>** tiene muy mala mantenibilidad
- Con los elementos semánticos es más fácil mantener y estructurar nuestra página

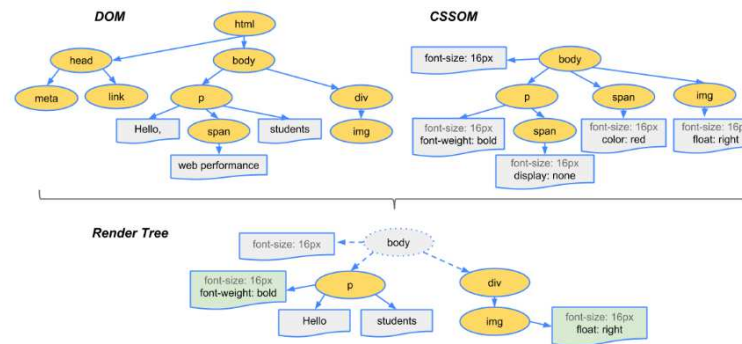


¿Qué hace el Navegador antes de mostrar una Web?

DOM y CSSOM

Estos son los pasos que sigue nuestro navegador para realizar la carga de una página Web (HTML + CSS) (Estructura de árbol, padres, hermanos, hijos ,) :

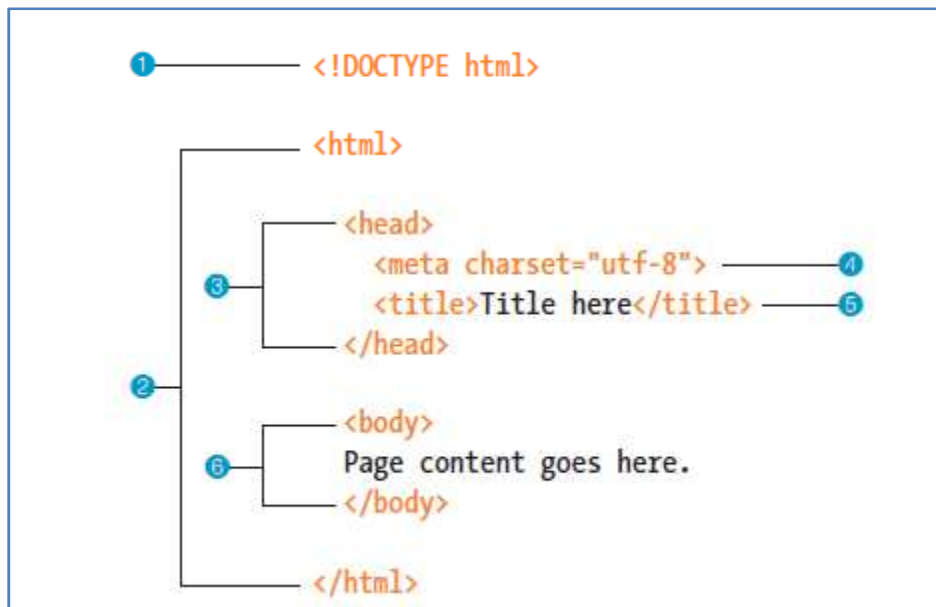
1. Se parsea / procesa las etiquetas HTML y se construye el árbol DOM a partir de las mismas
2. Se parsea / procesa las hojas de estilos (CSS) y se construye el árbol CSSOM
3. Se combinan el DOM y el CSSDOM en un árbol único
4. Se visualiza cada nodo del árbol final en la pantalla





Documento HTML5 básicos

Estructura documento HTML básico



1.- Tipo de documento (HTML5)

2.- Elemento raíz/padre <html>

3.- Metadatos, datos con información del documento (tipo de codificación usada, hojas de estilo CSS asociadas, aspecto de visualización por tipo de dispositivo ...)

6.- Cuerpo del documento. Básicamente, todo lo que pongamos aquí, es lo que podrá ver el usuario al cargar el documento Web en su navegador

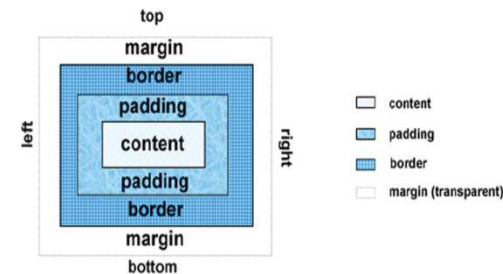


¿Cómo se ordenan los elementos en una Web?

Box Model, Flex Box , CSS Grid

=> **Box Model** : atributos **float** y **clear**

https://www.w3schools.com/cssref/pr_class_float.asp



- Pensemos en queda cada elemento es como una caja
- Las cajas ocupan todo el ancho del navegador si no le establecemos un tamaño (ancho x alto = width x height)
- Los elementos se posicionan siempre a la izquierda en una línea y siempre se produce un salto de línea al final del elemento si no le decimos lo contrario (Este comportamiento lo tienen los elementos bloque : div, article, section, nav,)

```
# style.css X
1 article {
2   border: 1px solid blue;
3 }

boxModel.html X
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8   <link rel="stylesheet" href="style.css">
9 </head>
10
11 <body>
12
13   <article>
14     CONTENT
15   </article>
16
17 </body>
18
19 </html>
```

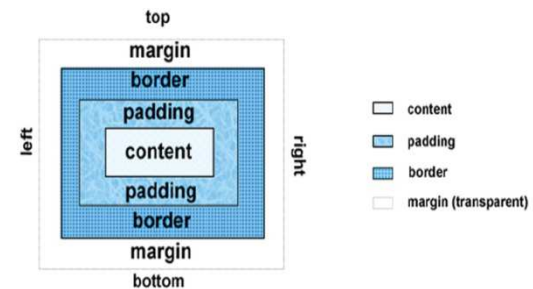




¿Cómo se ordenan los elementos en una Web?

Box Model, Flex Box, CSS Grid

=> **Box Model** : atributos **float** y **clear**



- Estableciendo **ancho x alto** en función del view width (vw) y del view height (vh)

```
# style.css X
1 article {
2   border: 1px solid blue;
3   width: 30vw;
4   height: 10vh;
5 }

boxModel.html X
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8   <link rel="stylesheet" href="/style.css">
9 </head>
10
11 <body>
12
13   <article>
14     CONTENT
15   </article>
16
17 </body>
18
19 </html>
```





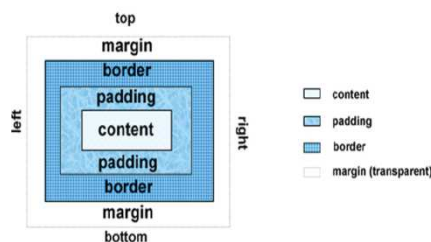
HTML5

Disposición de elementos

¿Cómo se ordenan los elementos en una Web?

Box Model, Flex Box , CSS Grid

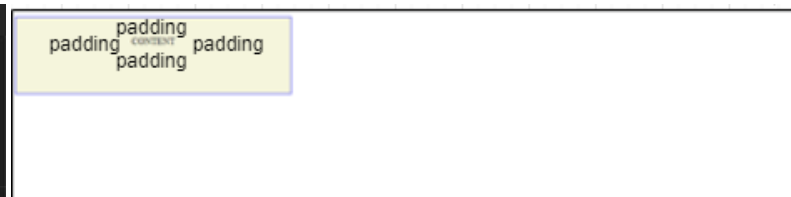
=> **Box Model** : atributos **float** y **clear**



- Estableciendo la propiedad **padding**

```
# style.css X
1 article {
2   border: 1px solid blue;
3   width: 30vw;
4   height: 10vh;
5   text-align: center;
6 }
7
8 .padding {
9   padding: 2em;
10  background-color: beige;
11 }

boxModel.html X
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8   <link rel="stylesheet" href="/style.css">
9 </head>
10
11 <body>
12
13   <article class="padding">
14     CONTENT
15   </article>
16
17 </body>
18
19 </html>
```



GOBIERNO
DE ESPAÑA

MINISTERIO
DE INDUSTRIA, COMERCIO
Y TURISMO



Escuela de
organización
industrial



Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro





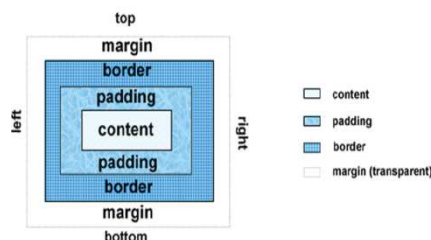
HTML5

Disposición de elementos

¿Cómo se ordenan los elementos en una Web?

Box Model, Flex Box , CSS Grid

=> **Box Model** : atributos **float** y **clear**



- Estableciendo la propiedad **border**

```
# style.css
1 article {
2   border: 1px solid blue;
3   width: 30vw;
4   height: 10vh;
5   text-align: center;
6 }
7
8 .padding {
9   padding: 2em;
10  background-color: beige;
11 }
12
13 .border {
14   border: 1px dotted red;
15 }

<? boxModel.html X
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8   <link rel="stylesheet" href="./style.css">
9 </head>
10
11 <body>
12
13   <article class="padding border">
14     CONTENT
15   </article>
16
17 </body>
18
19 </html>
```

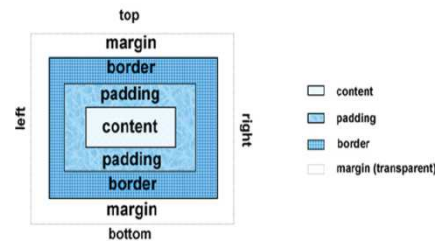




¿Cómo se ordenan los elementos en una Web?

Box Model, Flex Box , CSS Grid

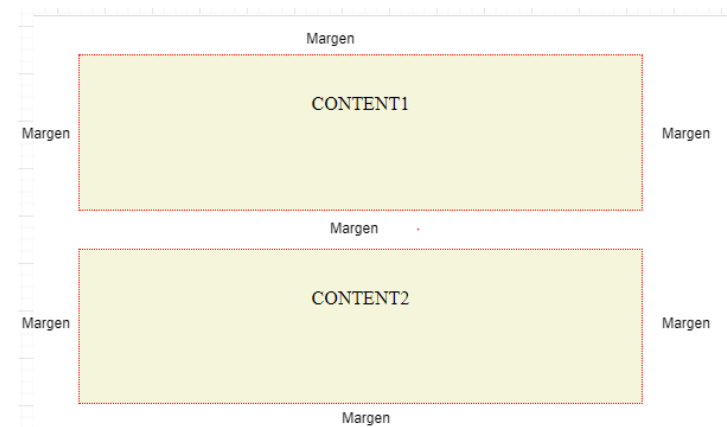
=> **Box Model** : atributos **float** y **clear**



- Estableciendo la propiedad **margin**

```
# style.css X
1 article {
2   border: 1px solid blue;
3   width: 30vw;
4   height: 10vh;
5   text-align: center;
6 }
7
8 .padding {
9   padding: 2em;
10  background-color: beige;
11 }
12
13 .border {
14   border: 1px dotted red;
15 }
16
17 .margin {
18   margin: 2em;
19 }

< boxModel.html X
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8   <link rel="stylesheet" href="./style.css">
9 </head>
10
11 <body>
12
13   <article class="padding border margen">
14     CONTENT1
15   </article>
16   <article class="padding border margen">
17     CONTENT2
18   </article>
19
20 </body>
21
22 </html>
```





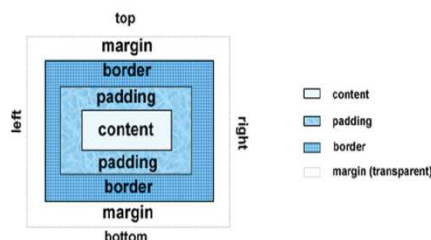
HTML5

Disposición de elementos

¿Cómo se ordenan los elementos en una Web?

Box Model, Flex Box, CSS Grid

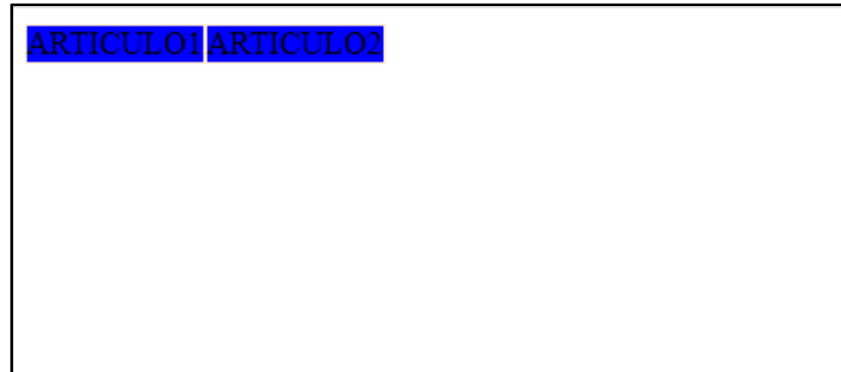
=> **Box Model** : atributos **float** y **clear**



- Alinear cajas una a continuación de otra en la misma línea con **float:left**

```
# style.css X
1 article {
2   border: 1px solid #blanchedalmond;
3   background-color: #blue;
4   text-align: center;
5 }
6
7 .left {
8   float: left;
9 }

boxModel.html X
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7   <title>Document</title>
8   <link rel="stylesheet" href="style.css">
9 </head>
10
11 <body>
12
13
14   <article class="left">ARTICULO1</article>
15   <article class="left">ARTICULO2</article>
16
17 </body>
18
19 </html>
```





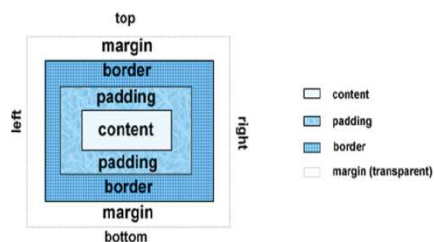
HTML5

Disposición de elementos

¿Cómo se ordenan los elementos en una Web?

Box Model, Flex Box, CSS Grid

=> **Box Model** : atributos **float** y **clear**



- Alinear cajas una a continuación de otra en la misma línea con **float:right**

```
# style.css X
1 article {
2   border: 1px solid #blanchedalmond;
3   background-color: #blue;
4   text-align: center;
5 }
6
7 .left {
8   float: left;
9 }
10
11 .right {
12   float: right;
13 }

boxModel.html X
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial
7   <title>Document</title>
8   <link rel="stylesheet" href="/style.css">
9 </head>
10
11 <body>
12
13   <article class="right">ARTICULO1</article>
14   <article class="right">ARTICULO2</article>
15
16
17 </body>
18
19 </html>
```





HTML5

Disposición de elementos

¿Cómo se ordenan los elementos en una Web?

Box Model, Flex Box , CSS Grid

⇒ **FlexBox (CSS3)** : Permite la alineación de elementos en una dirección (filas o columnas)

Se usan **flex containers** y dentro de ellos **flex ítems**. Los estilos se aplican de manera independiente sobre los contenedores y sus elementos. La disposición, comportamiento (crecer, encoger, apilar, envolver, justificar, alinear, ...)

https://www.w3schools.com/css/css3_flexbox.asp

⇒ **CSS Grid** : Permite la alineación de los elementos en varias direcciones (filas o columnas)

Define una página como un **grid** con **grid rows** , **grid columns** y **grid areas**

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





Texto

Elementos de texto

- **<h1>** : Elemento que representa un título principal o de cabecera
- **<h2>** , **<h3>**, **<h4>**, **<h5>**, **<h6>** : subtítulos
- **<p>** : representa un párrafo , por defecto todos los navegadores incluyen un margen en la parte superior para separar un margen de otro
- **<pre>** : representa un bloque de texto con formato previo (Código fuente , citas,)
- **** : elemento que puede contener un párrafo , una frase o una sola palabra
- **
** : insertar un salto de línea
- **<wbr>** : sugiere un salto del línea al navegador, para cortar el texto, cuando no hay suficiente espacio para mostrarlo todo
- **** : para mostrar el texto en Itálica



Texto

Elementos de texto

- **** : para mostrar el texto en negrita
- **<i>** : para mostrar el texto en cursiva
- **<u>** : para mostrar el texto en subrayado
- **** : para mostrar el texto en negrita
- **<mark>** : remarca el texto que es relevante acorde al contexto (términos buscados , ...)
- **<small>** : para mostrar un texto en letra pequeña
- **<cite>** : para mostrar una cita (Autor, avisos legales ...)
- **<address>** : para mostrar datos de contacto



Texto

Elementos de texto

- **<time>**: para mostrar el tiempo en formato legible (Atributos : datetime, pubdate)
- **<code>** : para mostrar código
- **<data>** : para incluir datos genéricos (Atributos : value)



Enlaces

Enlaces / vínculos (links)

- La **conexión de documentos a otros documentos a través de enlaces** es lo que hace posible la Web
- Un **vínculo** es un contenido asociado con una dirección URL que indica la ubicación del mismo
- Se pueden realizar vínculos a direcciones de correo con **mailto** o a números de teléfono con **tel**
- Se pueden crear enlaces a porciones del mismo documento usando el atributo **id** (Se posiciona sobre esa porción del documento).
- `Texto del enlace`

href = URL del recurso al que hacemos referencia

Enlace de un documento a otro documento

`Ir a índice`





Enlaces

Enlaces / vínculos (links)

target = dónde queremos abrir el enlace (en la misma ventana o frame, en el frame padre o en uno nuevo y blanco)

Enlace de un documento a otro , abriéndolo en una nueva página

`Ir a índice`

Enlace a una parte de un documento

`<article id="parte1">Esta es la parte1</article>`

`Ir a la parte1`



Enlaces

Enlaces / vínculos (links)

mailto y *tel*, los enlaces a **direcciones de correo** y a **números de teléfono** se pueden usar como buena práctica En el pie de nuestra página para completar la información de contacto.

Enlace para enviar un correo

Enviar correo a pepe

Enlace para llamar a un número de teléfono

Llamar a pepe

ping = url's separadas por coma a ejecutar en el servidor al hacer click en el enlace

download = booleano que si esá presente indica que el documento tiene que ser descargado en lugar de Abrirlo

Descargar index.html en lugar de abrirlo en el navegador

Guardar index.html

Ejecutar operacion1.js,operacion2.js en el servidor

Ir a índice





HTML5

Imágenes

Uso de imágenes

Imágenes en la Web

- Las imágenes son la segunda parte más importante en la Web después del texto

- :**

inserta en el documento la imagen indicada en *src*

width = Ancho de la imagen

height = Alto de la imagen

alt = texto que aparece si la imagen no se puede cargar

srcset = lista de imágenes con diferentes resoluciones

sizes = lista de Media Queries y tamaños para imágenes

crossorigin = establece credenciales para imágenes con CORS

```

```

- <picture> :** inserta en el documento la imagen indicada en *source* o *img*, se utiliza *source* para proporcionar varias imágenes para diferentes resoluciones, útil para crear sitios

Web responsivos

```
<picture>
  <source srcset="/media/examples/surfer-240-200.jpg"
        media="(max-width: 200px)">
  
</picture>
```



GOBIERNO
DE ESPAÑA

MINISTERIO
DE INDUSTRIA, COMERCIO
Y TURISMO

EQI Escuela de
organización
industrial



Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro



SISTEMA NACIONAL DE
**GARANTÍA
JUVENIL**



Uso de imágenes

Imágenes en la Web

- **<figure>** : representa contenido independiente del contenido de la página, normalmente es una imagen (Vídeo, ilustraciones, imágenes ...)
- **<figcaption>** : este elemento presenta un título para el contenido de un elemento

```
<figure>
  
  <figcaption>An elephant at sunset</figcaption>
</figure>
```



HTML5

Listas

Uso de listas

Usando listas

- A veces la información la tenemos que presentar como listas de elementos

- **** Listas desordenadas

- **** Listas ordenadas

```
<ul>
  <li>Milk</li>
  <li>Cheese
    <ul>
      <li>Blue cheese</li>
      <li>Feta</li>
    </ul>
  </li>
</ul>
```

- **<li reversed start type="">** : elemento de una lista , tanto ordenada como desordenada

reversed = orden inverso

start = indica el elemento por el cual se comienza a contar en las listas ordenadas

type = tipo de indicador de la lista (1 = números, a = letras minúsculas, A = letras mayúsculas

i = letras romanas minúsculas, I = letras romanas mayúsculas)

- **<dl>** : lista de términos y descripciones

- **<dt>** : define los términos

```
<p>Lista de términos:</p>
<dl>
  <dt>Calor</dt>
  <dd>Esto es lo que significa calor ...</dd>
  <dt>Calima</dt>
  <dd>Esto es lo que significa Calima</dd>
  <dt>Frío</dt>
  <dd>Esto es l que significa frío</dd>
</dl>
```



GOBIERNO
DE ESPAÑA

MINISTERIO
DE INDUSTRIA, COMERCIO
Y TURISMO

EQI Escuela de
organización
industrial



Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro



SISTEMA NACIONAL DE
**GARANTÍA
JUVENIL**



Uso de listas

Usando listas

- **<dd>** : define las descripciones
- **<blockquote>** : este elemento representa un bloque de texto que se cita desde otra parte del documento (Normalmente se usa **<p>** en lugar de este elemento)
- **<details>** : crea un control que se expande cuando se hace click sobre él, el contenido se engloba en el elemento **<p>**
- **<summary>** : texto desde donde se dispara **<details>**
- Ejemplo de uso de **<details>** y **<summary>**

```
<details>
  <summary>Details</summary>
  Este es un ejemplo de Details
</details>
```

▼ Details

Este es un ejemplo de Details



GOBIERNO
DE ESPAÑA

MINISTERIO
DE INDUSTRIA, COMERCIO
Y TURISMO

EQI Escuela de
organización
industrial



Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro





Uso de Tabla

Tablas de datos

- Las tablas disponen la información en filas y columnas. Las tablas son muy útiles para visualizar información de reportes y resúmenes con estilos muy diversos gracias al uso de CSS
- **<table>** : define una tabla
- **<tr>** : define una fila de celdas
- **<th colspan="número" rowspan="número">** : define una celda de cabecera de tabla
colspan = número de columnas que abarca
rowspan = número de filas que abarca
- **<td colspan="número" rowspan="número">** : define una celda de cuerpo de tabla



Uso de Tabla

Tablas de datos

- Ejemplo de tabla básica:

```
<table>
  <thead>
    <tr>
      <th colspan="2">Cabecera de la tabla</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Texto columna 1</td>
      <td>Texto columna 2</td>
    </tr>
  </tbody>
</table>
```

Cabecera de la tabla	
Texto columna 1	Texto columna 2



Uso de Tabla

Tablas de datos

- **<thead>** : para agrupar la cabecera de la tabla (Etiquetas de filas, columnas ...)
- **<tbody>** : para agrupar el cuerpo de la tabla
- **<tfoot>** : para agrupar el pie de la tabla.
- Ejemplo de tabla con agrupación:

```
<table>
  <caption>Resumen 2018</caption>
  <thead>
    <tr>
      <th scope="col">Producto</th>
      <th scope="col">Unidades vendidas</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">Donuts</th>
      <td>3,000</td>
    </tr>
    <tr>
      <th scope="row">Stationery</th>
      <td>18,000</td>
    </tr>
  </tbody>
</table>
```

Producto	Unidades vendidas
Donuts	3,000
Stationery	18,000

Resumen 2018



Uso de Formularios

Organizar la información

- El propósito principal de los formularios es permitir a cualquier usuario , poder seleccionar o Insertar y modificar información y poder enviar esta a un servidor para procesarla
- `<form name="" method="GET | POST" action="" target="" enctype="" accept-charset="">`

name = nombre del formulario. Se usa para acceder al mismo y a todo los elementos que el contiene el formulario.

method = indica el método usado para enviar la información contenida en el formulario al servidor.

GET -> enviar información de manera pública dentro de la URL y limitado a 255 caracteres.

POST -> enviar información de manera privada y sin restricción de tamaño.

action = indica la URL del fichero en el servidor donde se va a procesar la información.

target = indica donde será mostrada la respuesta enviada por el servidor al procesar la petición.

Es similar a los *links* (_blank, _self, _parent, _top)



HTML5

Formularios

Uso de Formularios

Organizar la información

enctype = indica la codificación que se va a usar para enviar la información al servidor.

application/x-www-form-urlencoded -> los datos están codificados

multipart/form-data -> los datos no están codificados

text/plain -> solo los espacios están codificados

accept-charset = indica el tipo de codificación que se va a aplicar al formulario. EL valor por defecto se coge del **<meta charset="utf8">** definido al inicio del documento Html

UTF-8, ISO-8859-1

```
<form action="" method="get" class="form-example">
  <div class="form-example">
    <label for="name">Nombre: </label>
    <input type="text" name="name" id="name" required>
  </div>
  <div class="form-example">
    <label for="email">Email: </label>
    <input type="email" name="email" id="email" required>
  </div>
  <div class="form-example">
    <input type="submit" value="Dar de alta">
  </div>
</form>
```

Nombre:

Email:



GOBIERNO
DE ESPAÑA

MINISTERIO
DE INDUSTRIA, COMERCIO
Y TURISMO

EQI Escuela de
organización
industrial



Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro





Uso de Formularios

Elementos de un formulario

- `<input type="text | email | search | url | tel | number | range | date | datetime-local | week | month | time | hidden | password | color | checkbox | radio | file | button | submit | reset | image" maxlength="" minlength="" min="" max="" step="">`

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input>

Define un campo para entrada de datos .

maxlength : máximo número de caracteres permitidos

minlength : mínimo número de caracteres permitidos

min : valor mínimo permitido para el campo

max : valor máximo permitido para el campo

step : valor por el que el campo aumentará/decrementará



Uso de Formularios

Elementos de un formulario

Dependiendo del tipo declarado en el atributo **type** representará un tipo de dato concreto, estos son los tipos posibles :

text : texto genérico

email : correo electrónico

search : texto para realizar búsquedas

url : para URLs

tel : para un número de teléfono

number : para un valor numérico

range : para un rango de números



Uso de Formularios

Elementos de un formulario

date : para una fecha

datetime-local : para una fecha y una hora

week : para el valor numérico de una semana

month : para el valor numérico de un mes

time : para una hora (horas y minutos)

hidden : para un campo de texto oculto

password : para una contraseña. Oculta los caracteres escritos con asteriscos o puntos

color : para un color

checkbox : para un checkbox (Activar o desactivar una opción)

radio : para seleccionar una opción entre varias

file : para seleccionar un fichero del equipo

button : para crear un botón

submit : un botón para enviar los datos del formulario

reset : para resetear los valores del formulario a sus valores iniciales

image : usa una imagen como botón submit. Requiere el elemento *input* para indicar con *src* la URL de la imagen





Uso de Formularios

Elementos de un formulario

Ejemplos :

```
<input type="text" name="nombre">
```

```
<input type="image" src="buttonsubmit.png" width="100">
```

```
<textarea rows="12" cols="12">
```

Área de texto con varias filas *rows* y columnas *cols*. Para introducir varias líneas de texto

```
<label for="comentarios">Comentarios:</label>
```

```
<textarea id="comentarios" name="comentarios"
  rows="5" cols="33">
```

Pues hace calor

```
</textarea>
```

Comentarios:

Pues hace calor



Uso de Formularios

Elementos de un formulario

Crea una lista desplegable con varias opciones `<option value="..."></option>`

Se usa `<optgroup>` para agrupar los valores

```
<select>
  <optgroup label="panes">
    <option value="1">Barra1</option>
    <option value="2">Barra2</option>
  </optgroup>
  <optgroup label="lacteos">
    <option value="1">Leche1</option>
    <option value="2">Leche2</option>
  </optgroup>
</select>
```



Uso de Formularios

Elementos de un formulario

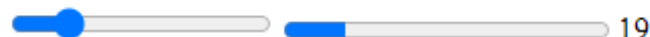
- **`<button type="button" | submit">`**

Crea un botón. El atributo *type* sirve para describir el propósito del botón

- **`<output>`**

Representa la salida generada por la acción de un formulario

```
<form oninput="resultado.value=rango.value;pro.value=rango.value;">
  <input type="range" id="rango" value="0">
  <progress id="pro" value="0" max="100">0%</progress>
  <output id="resultado">Sin valor todavía</output>
</form>
```



- **`<meter>`**

Representa una medida o el valor actual de un intervalo

```
<meter value="60" min="0" max="100" low="40" high="80" optimum="100">60</meter>
```



MINISTERIO
DE INDUSTRIA, COMERCIO
Y TURISMO

EQI Escuela de
organización
industrial



Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro





Uso de Formularios

Elementos de un formulario

- **<button type="button | submit">**

Crea un botón. El atributo *type* sirve para describir el propósito del botón

- **<output>**

Representa la salida generada por la acción de un formulario

```
<form oninput="resultado.value=rango.value;pro.value=rango.value;">
  <input type="range" id="rango" value="0">
  <progress id="pro" value="0" max="100">0%</progress>
  <output id="resultado">Sin valor todavía</output>
</form>
```



- **<meter>**

Representa una medida o el valor actual de un intervalo

```
<meter value="60" min="0" max="100" low="40" high="80" optimum="100">60</meter>
```



- **<progress>**

Representa una medida de progreso

```
<progress value="30" max="100">30%</progress>
```



GOBIERNO
DE ESPAÑA
MINISTERIO
DE INDUSTRIA, COMERCIO
Y TURISMO

EQI Escuela de
organización
industrial



Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro





Uso de Formularios

Elementos de un formulario

- **<datalist>**

Crea un lista de valores disponibles para otros controles. Funciona junto con el elemento *option* para definir las opciones

Ejemplo :

```
<datalist id="listaTels">  
  <option value="123123123" label="Tlf1">  
  <option value="456456456" label="Tlf2">  
</datalist>  
<p><label>Teléfono : <input type="tel" name="telefono" list="listaTels"></label></p>
```

Un formulario web que muestra un campo de entrada de tipo 'tel' con un 'datalist' asociado. El campo está etiquetado como 'Teléfono :'. Al hacer clic en el campo, se abre un menú desplegable que muestra dos opciones: '123123123 Tlf1' y '456456456 Tlf2'.



HTML5

Formularios

Uso de Formularios

Elementos de un formulario

- **<label>**

Crea una etiqueta para identificar un elemento de formulario

```
<label for="nombre">Nombre: </label>
<input type="text" name="nombre" id="nombre"></label>
```

Nombre:

- **<fieldset>**

Agrupar los elementos de forma conjunta. El elemento puede contener un elemento **<legend>** para definir el título de la sección

```
<fieldset>
  <legend>Leyenda Fieldset</legend>
  <section>
    Contenido
  </section>
</fieldset>
```

Leyenda Fieldset
Contenido



GOBIERNO
DE ESPAÑA
MINISTERIO
DE INDUSTRIA, COMERCIO
Y TURISMO

EQI Escuela de
organización
industrial



Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro





Uso de Formularios

Elementos de un formulario

- Dentro de los elementos que componen los formularios hay unos atributos comunes:

disabled : activa o desactiva el elemento (`<input disabled="true">`)

readonly : indica que el valor del elemento es de solo lectura (`<input readonly>`)

placeholder : muestra un texto en el fondo del campo como sugerencia de lo que se puede escribir en el mismo. (`<input placeholder="Teléfono">`)

autocomplete : activa o desactiva el autocompletado *on* / *off* (`<input autocomplete="on">`)

required : indica que este campo es obligatorio (`<input required="true">`)

multiple : indica que se pueden introducir varios valores, aplica a `<input type="email | file">`

autofocus : indica al navegador que este elemento se seleccione nada más cargar el documento



HTML5

Formularios

Uso de Formularios

Elementos de un formulario

pattern : expresión regular para validar el valor introducido en el campo

novalidate : atributo para **<form novalidate>** que indica que el formulario no debería ser validado

formnovalidate : atributo para **<input>** o **<button>** que indica que el valor del elemento no tiene que ser validado





HTML5

Media

Uso de Audio y Vídeo

Elementos multimedia

- object : para embeber cualquier tipo de elemento multimedia (imagen , vídeo , ...)

```
<object data="picture.svg" type="image/svg+xml">
  
</object>
```

- video : para embeber cualquier tipo de elemento vídeo

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/video>

```
<video src="highlight_reel.mp4" width="640" height="480"
poster="highlight_still.jpg" controls autoplay>
  Your browser does not support HTML5 video. Get the <a
href="highlight_reel.mp4">MP4 video</a>
</video>
```



GOBIERNO
DE ESPAÑA

MINISTERIO
DE INDUSTRIA, COMERCIO
Y TURISMO

EQI Escuela de
organización
industrial



Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro





Uso de Audio y Vídeo

Elementos multimedia

width = ancho en pixeles

height = alto en pixeles

Poster = url de imagen a visualizar mientras se carga el vídeo

controls = controles para gestionar la reproducción del vídeo

autoplay = hace que el vídeo se reproduzca automáticamente una vez se haya descargado

- audio : para embeber cualquier tipo de elemento audio

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/audio>

```
<audio id="whiterabbits" controls preload="auto">
  <source src="percussiongun.mp3" type="audio/mp3">
  <source src="percussiongun.ogg" type="audio/ogg">
  <source src="percussiongun.webm" type="audio/webm">
  <p>Download "Percussion Gun":</p>
  <ul>
    <li><a href="percussiongun.mp3">MP3</a></li>
    <li><a href="percussiongun.ogg">Ogg Vorbis</a></li>
  </ul>
</audio>
```



GOBIERNO
DE ESPAÑA

MINISTERIO
DE INDUSTRIA, COMERCIO
Y TURISMO

EQI Escuela de
organización
industrial



Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro





⇒ Enlaces de interés

- <https://developer.mozilla.org/es/docs/HTML/HTML5>
- <https://www.w3c.es/>
- <https://developers.google.com/web/>
- <https://www.mozilla.org/es-ES/firefox/developer/>
- <http://webdevfoundations.net/color>
- <http://atom.io>
- <http://www.robotstxt.org>
- <https://www.ietf.org>