**HTML**  son las siglas de *HyperText Markup Language*. En esta guía veremos los elementos y formatos que deberíamos de tener en cuenta de cara a trabajar con un documento *HTML*.

1. **Estructura de un documento HTML.**

***<!DOCTYPE html>***

***<html>***

***<head>***

***<title>A basic HTML document</title>***

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

***</head>***

***<body>*** *<!-- This is a comment in HTML →*

***<h1>Web games with HTML</h1>***

***<p>We will need to learn:</p>***

***<ul>***

***<li>HTML5</li>***

***<li>CSS3</li>***

***<li>JavaScript</li>***

***</ul>***

***</body>***

***</html>***

Esto que vemos acá arriba es una estructura típica del documento *HTML* (por supuesto, está harto simplificada). Ésta tiene una serie de elementos que denominaremos **etiquetas**. Hablamos de *<html>, <li>,* etc. Las etiquetas agrupan información (¿se acuerdan de las llaves en C#? pues parecido) y les otorgan ciertas características en las que iremos indagando en un rato.

Por lo pronto, tenemos que decir que partimos de estas etiquetas:

* ***<html>***: se encarga de englobar todas las líneas de código que hemos escrito.
* ***<head>***: Aquí se especifica la información de todos los archivos que importemos, los *style*s (ya veremos qué es), etc. También contiene la información de la cabecera.
* ***<title>*** Ahí escribe el título de la página. ¿Saben las palabras que aparecen en las pestañas cuando tienes muchas abiertas? ¡Eso mismo!
* ***<body>***: Es el *cuerpo* de nuestro código. Básicamente lo que se ejecuta y se muestra por pantalla se pone acá.
* ***<h1>***: Básicamente es una forma de poner un trozo de texto en plan titulito (?). También existen *h2, h3,* etc. que irán disminuyendo el tamaño de la fuente empleada según se vaya incrementando el número de al lado de la *h*.
* ***<p>***: Permite escribir algo por pantalla de forma normal.
* ***<ul>***: Este permite mostrar una serie de puntos, siendo cada uno de ellos lo contenido en el ***<li>***.

Muy importante: Al igual que con las llaves en otros programas, las etiquetas necesitan (en su gran mayoría, pues hay excepciones) ser cerradas. Para ello se pone la misma etiqueta que la de apertura pero con un ***/*** en el inicio de la palabra: ***</ul>, </li>, …***

**2) Los *styles.***

Un *style* es como una mini función que permite alterar ciertos parámetros (color, fuente, …) de la información contenida en una etiqueta. Se coloca en la apertura del bloque, justo después de la palabra que define la etiqueta, como podemos ver en el siguiente ejemplo:

***<h1 style="color: #ff00cc">***

***Web games with HTML***

***</h1>***

Acá estamos cambiando el color de la frase de negro (el de defecto) a rosa. Ahora bien, tampoco es necesario que seas un sadomasoquista del carajo y pongas el número asociado al color. También puedes tirar por llamar a los tonos por medio de su respectivo nombre en inglés:

***<h1 style="color: pink">***

***Web games with HTML***

***</h1>***

Y sí, funcionaría igual que el de arriba.

Ahora imagínense que en verdad tenemos fleje etiquetas a las que queremos aplicarle *styles* diferentes. ¿No estaría bien estandarizar esta cosa para que así nos ahorremos tener que poner *style blablabla* una y otra vez? Pues se puede, añadiendo en el *<head>* una estructura de *style* como esta:

***<style type="text/css">***

***h1 {***

***font-size: 30pt;***

***color: #ff00cc; /\* RGB hex values \*/***

***}***

***</style>***

Ese *type* que vemos por ahí es lo que define el tipo de documento al que estamos haciendo referencia, y es que al fin y al cabo lo que estamos haciendo es crear un *minidocumento* dentro del nuestro que sea del formato *text/css* (ahora en un momento lo vemos de manera más detenida).

Básicamente, por cada etiqueta que declares dentro del *<style>* irás añadiendo una nueva estructura, de tal forma que ahora es más fácil reutilizar etiquetas (es decir, si vamos a usar *h1* más de una vez, por ejemplo, nos ahorramos el coñazo de escribir de nuevo lo de *style=blablabla*).

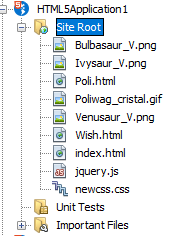
Pero aún podemos simplificar aún más todo este quilombo haciendo una llamada a un **documento *CSS style*.**

**3) Documento *CSS style*.**

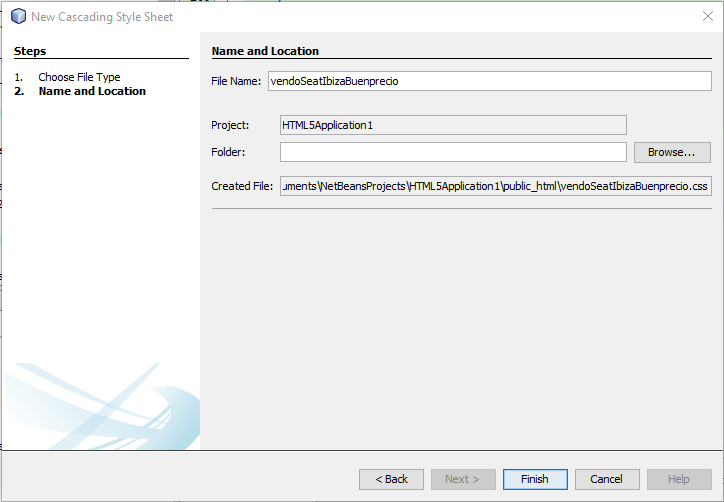
¿Se acuerdan del *from math import \** de *Python*? **\*Flashbacks from Vietnam\*** Básicamente estábamos importando todo el contenido de *math* es nuestro documento de tipo *.py*. Pues aquí tenemos, más o menos, lo mismo.

Imagínense que tenemos unos *styles* tó largos, que imposibilitan que se visione bien el resto del código. Eso puede llegar a suponer un problema, ya que te puedes liar. Es por eso que preferimos crear un *file* específico que se encargue de manejar todas estas cosas, de tal forma que nosotros solamente haremos una llamada a la localización de ese archivo y tan contentos. Primero veremos cómo hacer dicho documento y luego ya miraremos lo de la llamada :-)

Para crear el documento tenemos que ir a nuestro proyecto en la barra izquierda del *NetBeans*. Ahí apretamos en la carpeta *Site Root* (que es la que almacena todas nuestras cosas).



Hacemos *click* derecho y se desplegarán varias opciones. Apretamos la opción que pone *new* y de ésta saldrán los diferentes tipos de *file* que podemos crear. Aquí escogemos la opción que pone *Cascading Style Sheet*… que en efecto, son las siglas de *CSS* (plot twist). Y renombramos el archivo en la pantalla que se nos abre.



¡Le damos a *Finish* y ya estaría!

¿Qué metemos acá? Pues bien, volvamos al ejemplo que vimos antes:

***<style type="text/css">***

***h1 {***

***font-size: 30pt;***

***color: #ff00cc; /\* RGB hex values \*/***

***}***

***</style>***

El *style* que estamos definiendo está indicando que el contenido de dentro es de tipo *css*… así que sería harta tontería mantenerlo cuando queremos meter este en un archivo que ya es así por defecto. Así que quitamos el *style* y se queda esto:

***h1 {***

***font-size: 30pt;***

***color: #ff00cc; /\* RGB hex values \*/***

***}***

Así ya estaría en funcionamiento (y sin ningún error, claro) :-P

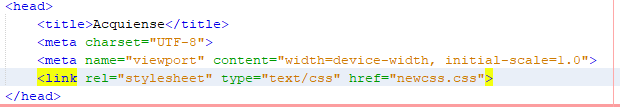
Vale, ¿y cómo hacemos la llamada desde el archivo *.html* (me refiero al principal, claro)?

Pues tendremos que hacer uso de una nueva etiqueta: **<link>**.

La veremos más en profundidad luego (pues tiene varias funciones muy interesantes que no está mal saber), pero por el momento nos vamos a quedar con que *<link>*, para este caso, permite enlazar el contenido del *.css* con el *.html*. La estructura a seguir sería, para el caso que hemos visto en la captura de pantalla de antes, esta:

**<link rel="stylesheet" type="text/css" href="vendoSeatIbizaBuenprecio.css">**

¿Qué cosas nos resultan interesantes de esta línea que tenemos?

1. *<link>* **NO** tiene etiqueta de cierre. Eso significa que no existe un *</link>*, ya que el contenido que estamos viendo va entre los *<>*.
2. Esta línea va dentro del ***<head>***, tal y como vemos en esta captura:  
   
3. Las dos primeras variables (*rel* y *type*) no las tocamos para nada. La que sí que toca cambiar es la de ***href***, pues es la que contendrá el nombre del documento (ojo, siempre con su extensión, que en este caso siempre será *.css*).

**4) Identificadores y clases.**

Ahora vamos a tocar dos elementos más (también referidos al cambio de las características del texto): los **identificadores** y las **clases**. Como antes, lo que haremos será colocar estas cosas en el archivo *.css* y luego hacer una llamada (que será la misma que tenemos arriba para todas y se debe hacer una vez, no se líen) para meterlas en el *.html*.

Un identificador funciona tal cual lo hace un *style*, con la diferencia de que él no modifica todas las etiquetas con un nombre específico. Es decir, nosotros teníamos antes un *style* llamado *h1*. Eso significa que cada vez que apliquemos la etiqueta con ese nombre las características del texto serán las que pone dicho *style*. Sin embargo, con el identificador lo que hacemos es modificar la apariencia de una etiqueta solamente en ese caso.

Un identificador se define en el *file* *.css* de la siguiente manera:

**#books {**

**font-size: 30pt;**

**color: #ff00cc;**

**}**

En el *.html* lo que tenemos que hacer para que se haga efectiva la llamada en una etiqueta es colocar el término clave **id**.

**<h1 id="books">**

**Useful books**

**</h1>**

Por otro lado, las clases en *HTML5* funcionan como una especie de recopilación de estilos aplicados a un grupo de elementos concreto (o no). A ver, de primeras tenemos dos formas de crear la clase. Si lo que queremos es aplicar de forma genérica unas mismas características cada vez que se emplee nuestra clase, entonces usaremos la siguiente (colocaremos esta estructura en el *.css*):

**.books{**

**color: pink;**

**}**

Como podemos observar, la clase se crea mediante la colocación de un punto delante del nombre que vamos a gastar (en este caso *books*).

Ahora bien… ¿y si queremos colocar excepciones para casos concretos? Imagínense que al final resulta que queremos aplicar la clase a 4 etiquetas pero 1 de ellas queremos que sea diferente. Lo suyo sería indicar una excepción para que se trate de otra forma. Volvemos al *.css* y probamos a seguir esta estructura (donde *h1* corresponde al nombre de la etiqueta:

**h1.books{**

**color: red;**

**}**

Lo único que nos faltaría por saber es cómo hacemos para realizar una llamada a la información del *.css*. Para ello nos vamos al documento principal y colocamos ***class=”[nombre de la clase]”*** justo después del nombre de la etiqueta (en nuestro ejemplo *h1*):

**<h1 class=”books”>**

**En verdad menudo mase de asignatura.**

**</h1>**

A continuación veremos una amalgama de elementos que nos resultarán interesantes de cara a representar diferentes elementos en nuestra página web.

**5) Elementos totalmente aleatorios y sin mucha conexión entre sí (?)**

**<div>**

La etiqueta *<div>* nos puede resultar especialmente útil cuando queremos agrupar bloques con características específicas. Pongámonos en la situación de que queremos, por ejemplo, poner dos líneas en azul pero la primera queremos que corresponda a un título y la segunda a texto corriente. Si no tuviéramos el *<div>* tendríamos que aplicar dos *styles* diferentes. Sin embargo eso no es necesario porque agrupamos con esta etiqueta las dos líneas y le aplicamos el *style* para que sean azules, mientras que dentro ponemos lo que nos dé la gana. Quedaría algo así:

**<div style="color:#0000FF">**

**<h3>El título de la primera línea.</h3>**

**<p>La línea de texto normalucha.</p>**

**</div>**

Atributos.

1. ***align.*** El único del que dispone *<div>*. Tiene tres posibilidades (right, left y center) y básicamente alinea el texto.  
    **<div align="right">**

**Hola (?)**

**</div>**

**<span>**

*<span>* hace justo lo contrario que *<div>*. Mientras que este último agrupaba, el primero la acción que realiza es separar un cachito de alguna línea de texto o de lo que sea con tal de cambiarle algún aspecto. Por ejemplo, si queremos colocar de azul la palabra *coche*  de la frase *Me he comprado un coche nuevo* entonces tocará hacer lo siguiente:

**<p>Me he comprado un <span style="color:blue">coche</span> nuevo.</p>**

**Tablas.**

Realizar una tabla requiere de que trabajemos en un archivo *.css* o que coloquemos la tabla dentro de una etiqueta *<style>*. A continuación pondré un ejemplo y lo comentaremos, para indicar sus respectivas partes y puntos a tener en cuenta:

**<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>**

**<tr>**

**<td>John</td>**

**<td>Doe</td>**

**<td>80</td>**

**</tr>**

**</table>**

Vale, aquí vemos cinco etiquetas, las cuales corresponden a cada uno de los elementos de la tabla.

1. ***<table>*** es la que agrupa todos los elementos que conforman la tabla.
2. ***<tr>*** es la que indica cuál es la fila que actúa a modo de cabecera (es decir, la de los títulos).
3. ***<th>*** se encarga de definir cada uno de las columnas de la cabecera.
4. ***<tr>*** agrupa todas las columnas (es decir, monta la fila) ordinarias.
5. ***<td>*** nos indica cada columna ordinaria.

*Nota: La tabla también tiene algunos atributos interesantes, nombrados*[***aquí***](https://www.w3schools.com/html/html_tables.asp)***.***

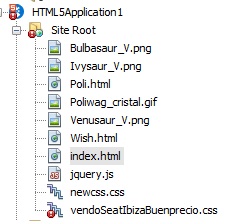
**Imágenes.**

Para colocar fotos en *HTML5* toca hacer uso de una nueva etiqueta llamada ***<img>***.

Aquí vemos un ejemplo base:

**<img src="VendoSeatIbiza.jpg">**

***src*** es la que define qué estamos buscando. Lo normal es añadir la imagen al proyecto que estemos haciendo (como aparece en la imagen de abajo), de tal manera que solo tengamos que colocar el título de la foto y ya.



Además, podemos añadirle tres atributos más: ***alt***, ***width*** y ***height***.

**<img src="TalkingHeads.jpg" alt="Banda muy tópica, no?" width="500" height="600">**

***alt*** lo que hace es colocar una nota al pie de foto. ***width*** y ***height*** se encargan de redimensionar la imagen (si no las pones tranqui, se pone en la que tiene de normal).

Cabe destacar que ***src=””*** admite también el formato *.png* y foto alojadas en páginas web (para que funcione eso debes meter en las comillas la *URL*).

**Hipervínculos.**

Un hipervínculo es, básicamente, una *URL* asociada a una palabra (lo típico de “Pulsa aquí” y todas esas movidas de croatas y serbios). Su forma base es la siguiente:

**<a href="**[**https://www.youtube.com/watch?v=bJ9r8LMU9bQ**](https://www.youtube.com/watch?v=bJ9r8LMU9bQ)**">Rock the Casbah.</a>**

*Tiene un porrón de atributos, que dejo* [***acá***](https://www.w3schools.com/tags/tag_a.asp)*.*

También se pueden referenciar páginas locales (es decir, que tienes tú) mediante la ***<a>***:

**<p><a href="html\_images.asp">HTML Images</a> is a link to a page on this website.</p>**

**Botones**

La implementación de botones en *HTML5* (al menos a un nivel más rudimentario) no es muy compleja. La estructura que habitualmente sigue es la que vemos a continuación:

**<button type="button" onclick="alert('Buen precio.')">Vendo Seat Ibiza</button>**

Vemos una serie de partes muy importantes:

1. *type* lo que está haciendo es dándole a entender al navegador que el elemento es un botón
2. *onclick* es quizás lo más importante que veremos por acá. Básicamente nos está indicando qué acciones realizar cuando se haga click en el botón. En este caso mostrará un mensaje (con el *alert()* lo sacamos por pantalla en una ventana chiquita).
3. Finalmente, lo que hay entre *<button>* y *</button>* es el texto contenido en el botón, el que aparece a modo de indicativo de lo que hace.



[*Atributos.*](https://www.w3schools.com/tags/tag_button.asp)

**Scripts.**

El delicado momento ha llegado. Un script es, básicamente, un trozo de código que se introduce en la página para que haga una acción determinada. Es necesario un lenguaje ajeno al de *HTML5* (pues éste no soporta bucles, condicionales, ni nada de eso). El que nosotros usaremos será *Javascript*, del cual hablaré en otro documento, de tal forma que acá solamente voy a mostrar la estructura para insertar un *script* en el *.html*.

Básicamente utilizaremos la etiqueta ***<script*>** para que se encargue de todo el trabajo sucio. Otra opción sería crear un fichero de tipo *.js*, mediante el cual no haría falta hacer uso de la etiqueta *<script>*.

**<script>**

**document.getElementById("demo").innerHTML = "Hello JavaScript!";**

**</script>**

**Barra de navegación.**

A nivel más básico, una barra de navegación se compone de una serie de términos que son una puerta a otras páginas. ¿Se acuerdan al inicio de todo cuando hablábamos de ***<ul>***? Pues vuelve a aparecer aquí. Esta etiqueta lo que hacía era agrupar como si fuera un listado una serie de elementos. La forma que toma sería esta:

**<ul>**

**<li><a href="#home">Home</a></li>**

**<li><a href="#news">News</a></li>**

**<li><a href="#contact">Contact</a></li>**

**<li><a href="#about">About</a></li>**

**</ul>**

el hipervínculo de la etiqueta *<a>* funciona como siempre: nos lleva a una dirección que nosotros indicamos. En este caso, donde pone *#* nosotros debería ir la *URL*.

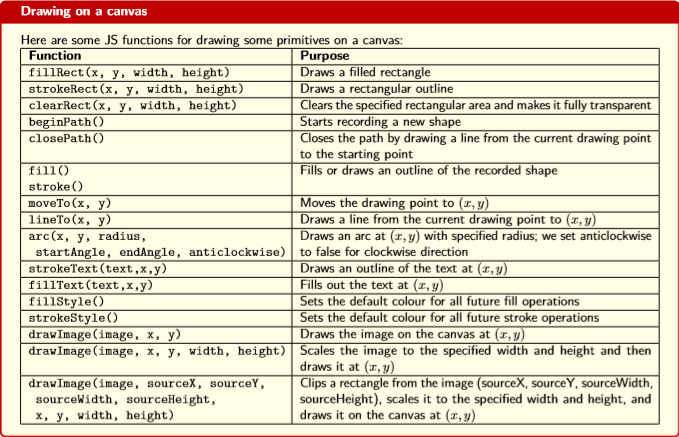
[*Otras cosas relativas a las barras.*](https://www.w3schools.com/css/css_navbar.asp)

**Animaciones.**

A ver, aquí yo no voy a decir que esto no es necesario, pero hay muchas formas de animar mediante *CSS* sin emplear un Canvas… pero claro, es un poco absurdo pararse a explicar cada forma cuando lo que vamos a utilizar de normal es el canvas. De todas formas dejo el enlace donde hay info acerca de las animaciones en CSS, para quien le interese (pincha [**aquí**](https://www.w3schools.com/css/css3_animations.asp)).

**Canvas.**

Esto sí que es importante. El canvas es la estructura que más gastaremos a la hora de animar o mostrar cualquier elemento en el navegador. Su funcionamiento tira de un porrón de funciones, que son mayoritariamente las de esta bonita tabla para nada sacada de los apuntes de Amengual (**?**):



¿Cómo se declara el canvas? Pues mediante la siguiente sentencia:

**<canvas id="myCanvas" width="200" height="100" style="border:1px solid #000000;">**

**</canvas>**

El **id** actúa (como era de esperar) como identificador del canvas (pues puedes tener varios). **Width** y **height** corresponden al alto y ancho del recuadro del que va a disponer el canvas (que será donde hagamos todos los dibujos), mientras que **style** se refiere al relleno del dibujo, así como el borde que lo delimita.

Por supuesto, todo lo que suponga dibujar requiere de hacer un ***<script>***. Por ejemplo, para dibujar una línea:

**<script>**

**var c = document.getElementById("myCanvas");**

**var ctx = c.getContext("2d");**

**ctx.moveTo(0,0);**

**ctx.lineTo(200,100);**

**ctx.stroke();**

**</script>**

Atentos a que el canvas que hemos declarado fuera del script hay que “importarlo” por así decirlo al script de JS introduciéndolo en una nueva variable.

**ANEXO**

Código para una tabla (rollo pokédex)

**HTML5**

<html>

<head>

<title>TODO supply a title</title>

<meta charset="UTF-8">

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

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

</head>

<body>

<table style="width:100%">

<tr>

<th>Primera</th>

<th>Segunda</th>

<th>Tercera</th>

</tr>

<tr>

<td><img src="pokms/Bulbasaur\_V.png"><div>Bulbasaur</div></td>

<td><img src="pokms/Ivysaur\_V.png"><div>Ivysaur</div></td>

<td><img src="pokms/Venusaur\_V.png"><div>Venusaur</div></td>

</tr>

<tr>

<td><img src="pokms/Charmander\_V.png"><div>Charmander</div></td>

<td><img src="pokms/Charmeleon\_V.png"><div>Charmeleon</div></td>

<td><img src="pokms/Charizard\_V.png"><div>Charizard</div></td>

</tr>

<tr>

<td><img src="pokms/Squirtle\_V.png"><div>Squirtle</div></td>

<td><img src="pokms/Wartortle\_V.png"><div>Wartortle</div></td>

<td><img src="pokms/Blastoise\_V.png"><div>Blastoise</div></td>

</tr>

</table>

</body>

</html>

**CSS**

table {

font-family: arial, sans-serif;

border-collapse: collapse;

width: 100%;

}

td, th {

border: 1px solid #dddddd;

text-align: left;

padding: 8px;

}

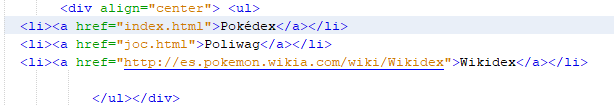
tr:nth-child(even) {

background-color: #dddddd;

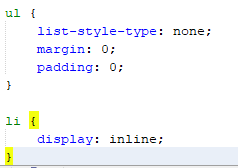
}

Barra de navegación:

**HTML**



**CSS**



a