

# HTML

# HTML

profesor Vladimir Bataller

## <html>

Etiqueta que delimita todo el documento html. Va precedida de la declaración del tipo de documento.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="ISO-8859-1">
    <title>Primera aplicación html 5</title>
  </head>
  <body>
  </body>
</html>
```

.

# HTML

# HTML

profesor Vladimir Bataller

**<body>**

Delimita la parte visible del documento

**Tiene atributos de estilo, pero actualmente se recomienda emplear css para proporcionar el estilo.**

.

# HTML

# HTML

profesor Vladimir Bataller

## <head>

Contiene otras etiquetas destinadas a definiciones y títulos

**No tiene atributos**

# HTML

# HTML

profesor Vladimir Bataller

**<title>**

Permite poner un título al documento.  
Generalmente se muestra en la barra de título de la ventana del navegador.

```
<html>  
  <head>  
    <title> Página de entrada al catálogo de discos </title>  
  </head>  
  <body>  
  </body>  
</html>
```

# HTML

# HTML

profesor Vladimir Bataller

## <meta>

Se incluye dentro de la etiqueta HEAD y permite indicar información sobre la página para incluirla en buscadores.

**name** = "Nombre de la página en el buscador"

**content** = "descripción del contenido de la página"

También se puede emplear para recargar la página, en ese caso se emplean los siguientes atributos:

**http-equiv**= "refresh"

**content** = "intervalo de refresco en segundos;  
URL=http://dirección de la pagina a recargar"

```
<meta name = "Tutorial de HTML" content = "Completa guia sobre HTML">
```

```
<meta http-equiv= "refresh"
```

```
content ="5; URL=http://www.tutorhtml.es">
```

# HTML

# HTML

profesor Vladimir Bataller

## <meta>: no-cache

Para que no se guarde la página en el caché del navegador, se incluirán las siguientes etiquetas en el head.

```
<meta http-equiv="CACHE-CONTROL" content="NO-CACHE">  
<meta http-equiv="PRAGMA" content="NO-CACHE">
```

.

# HTML

# HTML

profesor Vladimir Bataller

**<p>**

Delimita un párrafo, se suele emplear sin la etiqueta de cierre.

## Propiedades css básicas para <p>

**text-align: center, left, right, justify**

**El texto, las imágenes, vídeos y controles de formulario se comportan del mismo modo. Estos no pueden estar directamente dentro del body. Tiene que estar dentro de una etiqueta que los contenga, por ejemplo la etiqueta "p" o la etiqueta "div".**

`<p style="text-align:justify;">`En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que vivía un hidalgo de los de lanza en astillero, adarga antigua, rocín flaco y galgo corredor.... `</p>`

# HTML

# HTML

profesor Vladimir Bataller

**<br/>**

Inserta un salto de línea, menor que el que introduce <P>. En xhtml se escribe como <br />

Primera linea <br> Segunda linea <br> tercera linea <P> cuarta linea  
(mas separada) </p> Quinta linea



# HTML

# HTML

profesor Vladimir Bataller

## Carateres especiales (entidades)

Para representar caracteres que no existen en inglés emplearemos las siguientes entidades.

ñ -> &ntilde;

Ñ -> &Ntilde;

á -> &aacute;

é -> &eacute;

í -> &iacute;

...

">" -> &gt; (mayor que)

"<" -> &lt; (menor que)

" -> &quot; (comillas dobles)

(espacio en blanco) -> &nbsp;

# HTML

# HTML

profesor Vladimir Bataller

**<h1,h2,...>**

Define un encabezado o titulo. Existen seis niveles desde <h1> para los títulos de mayor nivel hasta <h6> para los de menor

Nota: Estas etiquetas de cabecera introducen un salto de línea (sin necesidad de insertar la etiqueta <br/>). Si se desea eliminar este efecto se deberá asignar a la propiedad de estilo display el valor inline. <h2 style="display:inline;">

<h1> Noticias internacionales</h1>

<h2> España es el destino preferido por los alemanes</h2>

<h2> Francia aumentará el número de centrales nucleares </h2>

# HTML

# HTML

profesor Vladimir Bataller

**<pre>**

Obliga al navegador a mostrar el texto tal y como se ha escrito en el documento.

<pre> Primera línea

Segunda línea espacios....

..... Fin espacios

</pre>

# HTML

# HTML

profesor Vladimir Bataller

**<b,i, ...>**

Permiten mostrar el texto en negrita, cursiva, subrayado y otros formatos.

**<b> ... </b>** Negrita

**<strong> ... </strong>** Negrita con sentido de importancia

**<i> ... </i>** Cursiva o Itálica

**<em> ... </em>** Cursiva o Itálica con sentido de énfasis.

**<u> ... </u>** Subrayado -----> Desaconsejada, usar css

**<s> ... </s>** Tachado -----> Desaconsejada, usar css

**<del> ... </del>** Tachado

**<sup> ... </sup>** Superíndice

**<sub> ... </sub>** Subíndice

**<ins> ... </ins>** Texto insertado (subrallado).

**<mark>... </mark>** Texto destacado con rotulador marcador.

# HTML

# HTML

profesor Vladimir Bataller

## <img>

Inserta una imagen

**src=** "ruta y nombre de la imagen" (absoluta o relativa)

**alt=** "texto" (texto alternativo a la imagen, si esta no se puede cargar)

### Propiedades css

**vertical-align:** "top|middle|bottom" (indica cómo se alineará con el texto)

**border-width:** ancho del borde.

**height:** altura      **width:** ancho      **margin:** margen

**float:** "left|right" . La imagen flotará alineada al lado indicado. El resto del contenido bordeará a la imagen.

```
<img src= "./images/buzon.gif" align= "top" alt="enviar email">
```

# HTML

# HTML

profesor Vladimir Bataller

**<a>**

Determina un hipervínculo

**href=** "vinculo destino" (url relativa o absoluta del destino del enlace)

**target=**"\_blank|nombreVentana" -> Indica la ventana en la cual se quiere abrir el hipervínculo.

**<a href=** "indice.html"> Pulse aquí para ir al índice</a>

# HTML

# HTML

profesor Vladimir Bataller

## <a target>

El atributo target indica en que ventana o marco se debe cargar el archivo html al que apunta el hipervínculo

**target= "nombre\_ventana"**(nombre del marco o ventana donde se cargará el archivo html al que apunta el hipervínculo)

**target= "\_blank"** Abre el documento html apuntado en una ventana nueva.

**target= "\_self"** (Se muestra el archivo enlazado en el marco en el que se encuentra el enlace)

**target= "\_parent"** (El archivo enlazado se muestra sustituyendo al conjunto de marcos dentro del cual se encuentra el marco que contiene el enlace.

**target= "\_top"** (Suprime todos los marcos y muestra el hipervínculo en pantalla completa.

### MenuIzauiero.htm:

```
<a href="lunes.htm" target="principal">lunes</a>
```

```
<a href="martes.htm" target="principal">martes</a>
```

```
<a href="anuncio.htm" target="_blank">mostrar anuncio</a>
```

# HTML

# HTML

profesor Vladimir Bataller

**<a> +  
<img>**

El uso combinado de estas dos etiquetas nos permite dotar a una imagen de funcionalidad de hipervínculo

```
<a href= "vinculo destino">  
    <img src= "imagen a mostrar">  
</a>
```

```
<a href= "indice.html">  
    <img src= "atras.gif">  
</a>
```



# HTML

# HTML

profesor Vladimir Bataller

## <map>

Delimita un conjunto de etiquetas <area> que definen un mapa sensible (a cada area le corresponde un enlace)

**name** = "Nombre que asignamos al mapa para emplearlo posteriormente sobre una imagen"

```
<map name="casa">
```

```
  <area shape= "poly" coords="2,23,57,83,1" href="tejado.html">
```

```
  <area shape= "rect" coords="21,101,35,138" href="puerta.html">
```

```
</map>
```

# HTML

# HTML

profesor Vladimir Bataller

## <area>

Establece un área sensible dentro de un mapa.

**shape**= "tipo de area a definir" (Forma)

hay 3 tipos:

**rect** (Área rectangular) coords="X1,Y1,X2,Y2".

**poly** (Polígono) coords="X1,Y1,X2,Y2,X3,Y3, ...".

**circle** (Círculo) coords="X1,Y1, R".

**coords**= "coordenadas" (Lista de valores separada por comas)

**href**= "URL" (Dirección con la que enlaza ese área)

```
<map name="casa">
```

```
  <area shape= "poly" coords="2,23,57,83,1" href="tejado.html">
```

```
  <area shape= "rect" coords="21,101,35,138" href="puerta.html">
```

```
</map>
```

# HTML

# HTML

profesor Vladimir Bataller

**<img  
usemap>**

El atributo usemap nos permite asignar un mapa sensible a un imagen

**Usemap**="#nombre\_del\_mapa" (asignamos a la imagen el nombre del mapa sensible a emplear, precedido del signo almohadilla)

```
<map name="casa"> .... </map>
```

```
<img src= "./images/edificio.gif" usemap="#casa">
```

# HTML

# HTML

profesor Vladimir Bataller

**<ul> +  
<li>**

<ul> Contiene a las etiquetas <li>, que son los elementos de una lista no ordenada.

**type**= tipo (Tipo de marca que queremos utilizar, por defecto es un punto o un asterisco, según el tipo de visualizador).

Puede tomar tres valores:

**disk** = punto relleno

**circle** = circunferencia (hueca)

**square** = cuadrado

```
<ul type="disk">  
  <li>Lunes</li>    <li>Martes</li>  
  <li>Miércoles</li>  
</ul>
```

# HTML

# HTML

profesor Vladimir Bataller

**<ol> +  
<li>**

<ol> Contiene a las etiquetas <li>, que son los elementos de una lista ordenada.

**start**= número (número por el que debe empezar la lista)

**type**= tipo (Tipo de numeración a utilizar, por defecto es una lista numérica)

Hay 5 tipos de numeración a emplear:

**1** =numéricamente ( 1, 2, 3, 4.. . )

**a** =letras minúsculas (a, b, c, d...)

**A** =letras mayúsculas (A, B, C, D... )

**i** =números romanos en minúsculas (i, ii, iii, iv...)

**I** =números romanos en mayúsculas (I, II, III, IV...)

```
<ol type="I">
```

```
  <li>Lunes</li>    <li>Martes</li>
```

```
  <li>Miércoles</li>
```

```
</li>
```

# HTML

# HTML

profesor Vladimir Bataller

**<dl><dt>  
<dd>**

**<dl>** Contiene pares de etiquetas (Termino **<dt>**, Definición **<dd>**), que son los elementos de una lista de definiciones.

Ejemplo:

```
<dl>  
  <dt>Lunes</dt>  
  <dd>D&iacute;a de la luna</dd>  
  <dt>Martes</dt>  
  <dd>D&iacute;a de Marte</dd>  
  <dt>Mi&eacute;rcoles</dt>  
  <dd>D&iacute;a de mercurio</dd>  
</dl>
```

Lunes

Día de la luna

Martes

Día de Marte

# HTML

# HTML

profesor Vladimir Bataller

**`<hr/>`**

Dibuja una línea horizontal, sirve para separar bloques de texto.

Párrafo primero

`<hr/>`

Párrafo segundo

# HTML

# HTML

profesor Vladimir Bataller

## **<table>**

Contiene las etiquetas que describen la estructura de una tabla

**<thead></thead>** agrupa las filas cabecera de la tabla.

**<tbody></tbody>** agrupa las filas de datos de la tabla.

**<tfoot></tfoot>** agrupa las filas del pie de tabla.

### **Propiedades css para <table>**

border: 1px solid black;

border-collapse: collapse;



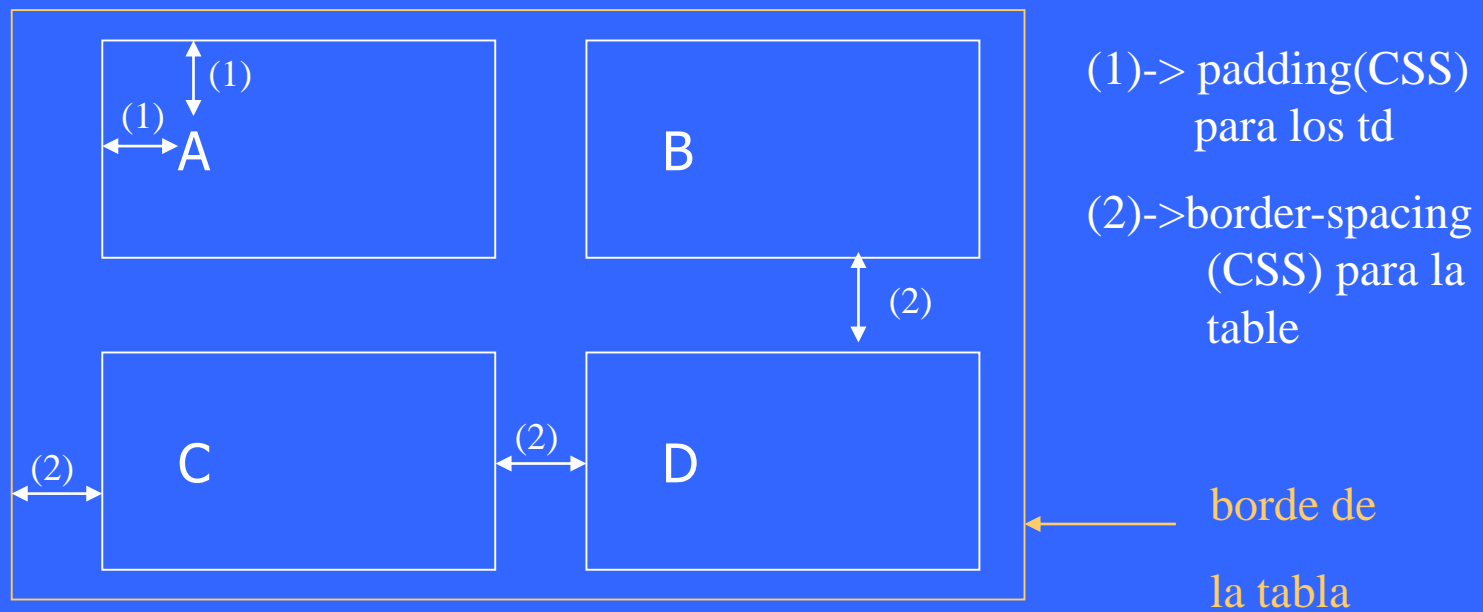
# HTML

# HTML

profesor Vladimir Bataller

## <table>

Contiene las etiquetas que describen la estructura de una tabla



```
<table><tbody>
  <tr>    <td>A</td>    <td>B</td>    </tr>
  <tr>    <td>C</td>    <td>D</td>    </tr>
</tbody></table>
```

# HTML

# HTML

profesor Vladimir Bataller

**<tr>**

Crea una fila en una tabla. Contiene a las etiquetas <td> que representan a cada una de las celdas de esa fila.

```
<table><tbody>
  <tr>    <td>A</td>    <td>B</td>    </tr>
  <tr>    <td>C</td>    <td>D</td>    </tr>
</tbody></table>
```

# HTML

# HTML

profesor Vladimir Bataller

## <td>

Determina una celda de la tabla

**rowspan**= número (Número número de filas que ocupará la celda, por defecto sólo toma una)

**colspan**= número (Número de columnas que ocupará la celda, por defecto sólo ocupa una)

### Propiedades css para <td>

border: 1px solid black;

padding: 5px 10px;

```
<table><tbody>
```

```
<tr>    <td>A</td>    <td>B</td>    </tr>
```

```
<tr>    <td>C</td>    <td>D</td>    </tr>
```

```
</tbody></table>
```

# HTML

# HTML

profesor Vladimir Bataller

## <th>

Determina una celda de cabecera de la tabla. Suelen estar en el <thead>

**rowspan**= número (Número número de filas que ocupará la celda, por defecto sólo toma una)

**colspan**= número (Número de columnas que ocupará la celda, por defecto sólo ocupa una)

### Propiedades css para <th>

border: 1px solid black;

padding: 5px 10px;

```
<table >
```

```
<thead><tr>
```

```
<th>Día</th>
```

```
<th>Hora</th>
```

```
</tr></thead>
```

```
<tbody><tr>
```

```
<td>Lunes</td>
```

```
<td>9:00</td>
```

```
</tr></tbody>
```

```
</table>
```

# HTML

# HTML

profesor Vladimir Bataller

## rowspan y colspan

Permiten que una celda ocupe varias filas y/o varias columnas.

```
<table width="100">
<thead>
  <tr>
    <th rowspan="2">&nbsp;</th>
    <th colspan="2">dias</th>
  </tr>
  <tr> <th>Lunes</th><th>Martes</th>  </tr>
</thead>
<tbody>
  <tr><th>Mañana</th><td>&nbsp;</td><td>&nbsp;</td></tr>
  <tr><th>Tarde</th><td>&nbsp;</td><td>&nbsp;</td></tr>
</tbody>
</table>
```

	dias	
	Lunes	Martes
Mañana		
Tarde		

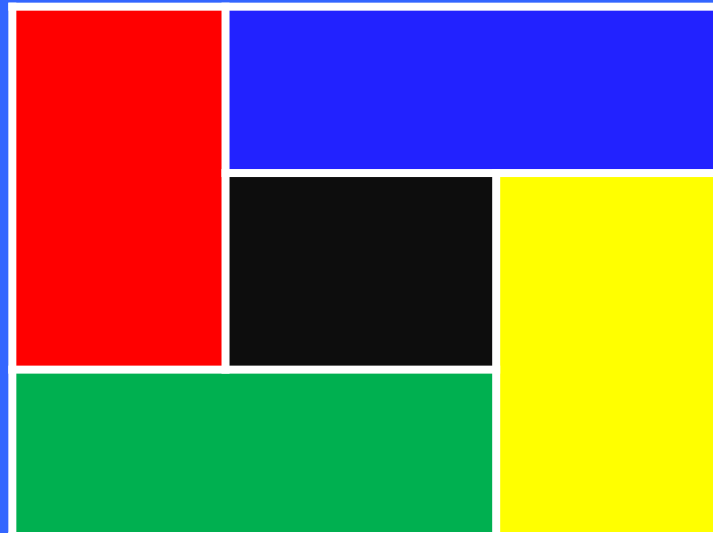
# HTML

# HTML

profesor Vladimir Bataller

## rowspan y colspan

## Ejercicio colspan y rowspan



.

# HTML

# HTML

profesor Vladimir Bataller

**<iframe>**

Muestra un documento html dentro de un área de otro documento.

.

```
<iframe src="anuncio.html"  
style="border:none;height:200px;width:100px"></iframe>
```

# HTML

# HTML

profesor Vladimir Bataller

## <form>

Contiene al resto de etiquetas de un formulario. Los formularios permiten al usuario insertar información y enviarla al servidor

**action**= "url" (Indica pagina activa a solicitar cuando se pulsa el botón de envío del formulario. Esa página será la que reciba los datos insertados en el formulario)

**method**= "POST / GET" (Get= máximo 240 caracteres, Post=sin limite)

Nota: Se podría decir que un formulario equivale a un enlace <A>, ya que solicita la página indicada en el atributo *action*, pero con la diferencia de que además le envía información introducida por el usuario.

```
<form action="InsertarUsuario.asp" method="post">
```

...

```
</form>
```



# HTML

# HTML

profesor Vladimir Bataller

## <input>

Permite al usuario insertar información para el posterior envío al servidor. El atributo type permite seleccionar el control a visualizar

**type="text | password | checkbox | radio | hidden | submit | reset | button"** (Indica el tipo de control que se presentará)

text -> Campo de texto de una sola línea

password -> Campo de texto, muestra asteriscos para ocultar la pw

checkbox -> Campo de marcado de una opción

radio -> Campo de selección de una opción entre varias posibles

hidden -> Campo de texto oculto para enviar determinados datos

submit -> Botón que al ser pulsado envía el formulario

reset -> Borra los datos insertados en todos los campos del formulario

```
<form action="InsertarUsuario.asp" method="post">
```

```
    <input type="...">
```

```
</form>
```

# HTML

# HTML

profesor Vladimir Bataller

## <input> (text)

type = "text" -> Permite al usuario insertar información en un Campo de texto de una sola línea.

**name**= "Nombre\_del\_campo" (Nombre con el que se identificará el campo dentro del formulario)

**maxlength**= número (Número máximo de caracteres a introducir en el campo.)

**size**= número (La anchura visible del campo en caracteres)

**value**= "texto" (Valor inicial del campo)

```
<form action="InsertarUsuario.asp" method="post"  
name="formInsertar">
```

```
    Apellidos: <input type="text" name="apellidos">
```

```
</form>
```

# HTML

# HTML

profesor Vladimir Bataller

## <input> (password)

Type = "password" -> Permite al usuario insertar una contraseña sin que sea visible (para ello se mostrarán asteriscos en lugar de los caracteres realmente insertados).

**name**= "Nombre\_del\_campo" (Nombre con el que se identificará el campo dentro del formulario)

**maxlength**= número (Número máximo de caracteres a introducir en el campo.)

**size**= número (La anchura visible del campo en caracteres)

**value**= "texto" (Valor inicial del campo)

```
<form action="InsertarUsuario.asp" method="post"  
name="formInsertar">
```

```
    Clave: <input type="password" name="clave">
```

```
</form>
```

# HTML

# HTML

profesor Vladimir Bataller

## <input> (checkbox)

Type = "checkbox" -> Este campo aparece como una casilla que puede estar marcada o no

**name**= "Nombre\_del\_campo" (Nombre con el que se identificará el campo dentro del formulario)

**checked**=**"checked"** (Indica si el estado del control esta inicialmente activado)

**value**= "texto" (que tomará el campo si la casilla está seleccionada)

```
<form action="InsertarUsuario.asp" method="post"  
name="formInsertar">
```

```
    <input type="checkbox" name="carnet" value="si">  
carnet de conducir. </form>
```

# HTML

# HTML

profesor Vladimir Bataller

## <input> (radio)

Type = "radio" -> El uso de varios controles radio con el mismo nombre permite al usuario elegir entre varias opciones mutuamente exclusivas.

**name**= "Nombre\_del\_campo" (Nombre con el que se identificará el campo dentro del formulario)

**checked** = "**checked**" (Indica si el estado del control esta inicialmente activado)

**value**= "texto" (que tomará el campo si la casilla está seleccionada)

```
<input type="radio" name="estado" value="so"  
        checked="checked"/> soltero
```

```
<input type="radio" name="estado" value="ca" />casado
```

```
<input type="radio" name="estado" value="ph" />pareja hecho
```

# HTML

# HTML

profesor Vladimir Bataller

## <input> (hidden)

Type = "hidden" -> Permite enviar información interna al servidor sin que sea visible para el usuario.

**name**= "Nombre\_del\_campo" (Nombre con el que se identificará el campo dentro del formulario)

**value**= "texto" (Valor inicial del campo, dado que el usuario no puede modificarlo, solamente se podría modificar por código javascript)

```
<input type="hidden" name="numero_usuario"  
value="333792000AA3">
```

# HTML

# HTML

profesor Vladimir Bataller

**<input>  
(submit)**

Type = "submit" -> al pulsar este botón se efectúa la petición de la página indicada en el atributo acción del formulario y así mismo enviar los datos recogidos por este.

**src = "archivo\_de\_imagen"** (permite indicar la imagen que queremos que muestre el botón)

**value= "texto\_del\_botón"** (El texto que muestra el botón)

```
<input type="submit" value="enviar datos">
```

# HTML

# HTML

profesor Vladimir Bataller

**<input>  
(reset)**

Type = "reset" -> al pulsar este botón se borrarán los datos insertados por el usuario en el formulario.

**src = "archivo\_de\_imagen"** (permite indicar la imagen que queremos que muestre el botón)

**value= "texto\_del\_botón"** (El texto que muestra el botón)

```
<input type="reset" value="borrar todo">
```



# HTML

# HTML

profesor Vladimir Bataller

## <select>

Control para seleccionar un elemento de una lista. Delimita un conjunto de etiquetas <option> que representan las opciones disponibles.

**name**= "Nombre\_del\_campo" (Nombre con el que se identificará el campo dentro del formulario)

**size**= número (Número de opciones visibles. Si se indica uno se presenta como un menú desplegable (combobox), si se indica más de uno se presenta como una lista con barra de desplazamiento)

**multiple**= "**multiple**" (Permite seleccionar más de un valor para el campo)

```
<select name="provincia" size="1">  
  <option value="28" selected="selected">madrid</option>  
  <option value="08">barcelona </option>  
</select>
```

# HTML

# HTML

profesor Vladimir Bataller

## <option>

Esta etiqueta representa cada una de las opciones disponibles en una lista <select>.

**value= "texto"** (valor que tomará el campo si esta es la opción seleccionada)

**selected="selected"** (Indica que esta es la opción por defecto. si no se especifica, por defecto sale la primera opción de la lista)

```
<select name="provincia" size="1">  
  <option value="28" selected="selected">madrid</option>  
  <option value="08">barcelona </option>  
</select>
```

# HTML

# HTML

profesor Vladimir Bataller

**<textarea>**

Campo de texto de múltiples líneas.

**name**= "Nombre\_del\_campo" (Nombre con el que se identificará el campo dentro del formulario)

**cols**= número (Número de columnas de texto)

**rows**= número (Número de filas de texto visibles)

**readonly**="readonly" (el usuario no puede modificar el contenido)

**El texto inicial que muestra el textarea se indica dentro de la etiqueta.**

**<textarea>En un lugar de la mancha... </textarea>**

# HTML

# CSS

profesor Vladimir Bataller

## Qué son las CSS

Una hoja de estilo nos permite indicar por ejemplo el tamaño de la fuente, color y estilo de cierto párrafo, mediante la definición de estilos y aplicación de los mismos.

Es posible crear una sola hoja de estilo que compartan muchos documentos, y al hacer un cambio en la hoja de estilo todos los documentos que la usan tendrán la apariencia deseada.

Ventajas:

- Separación del formato y el contenido

- Fácil mantenimiento

- Mayor precisión en la aplicación del formato.

**CSS: Cascading Style Sheet (Hojas de estilo en cascada)**

# HTML

# CSS

profesor Vladimir Bataller

## Más ventajas

Las hojas de estilo nos permiten controlar de forma más precisa el aspecto de nuestros documentos html

- El posible uso de diversas unidades de medida: pixeles, puntos, picas, mm, en los principales elementos del HTML.
- El posicionamiento de bloques en cualquier parte del documento HTML (Capas o CSS-P)
- Mejor control sobre las fuentes que es necesario para otras tecnologías relacionadas como la fuentes dinámicas.
- Es posible cambiar las características de una hoja de estilo mediante el uso de lenguajes de programación web como "Javascript".

**También se emplea para especificar el formato en otros lenguajes como XML**

# HTML

# CSS

profesor Vladimir Bataller

## Etiquetas de estilos

CSS introduce dos nuevas etiquetas `<style>` y `<span>`.

**`<style>`** : Se emplea para definir una hoja de estilo dentro de un documento html, normalmente se pone en la cabecera (head)

**`<span>`** : Sirve para delimitar un bloque de texto y por tanto poderle aplicar un determinado estilo. No inserta salto de línea.

**`<div>`** : Aunque existe en html, esta etiqueta se emplea con mucha frecuencia para aplicar estilos a bloques de texto de forma similar a la etiqueta `<span>` pero introduce un salto de línea. Así mismo se emplea para crear capas (bloques de texto e imágenes posicionables)

# HTML

# CSS

profesor Vladimir Bataller

## atributos de estilos

CSS introduce tres nuevos atributos (style, class e id)

**style:** Permite aplicar características CSS directamente a un elemento (a una etiqueta).

**class:** permite marcar un bloque de texto u otros elementos como miembro de una clase de estilo definida en una hoja de estilo.

**id:** Permite identificar inequívocamente un bloque de texto u otros elementos. Así mismo, se podrá definir un estilo en una hoja de estilo que se aplique a este bloque.

# HTML

# CSS

profesor Vladimir Bataller

## El atributo style

Permite asignar estilos CSS in-situ, es decir se especifican las características CSS junto al elemento sobre el que se aplica

El atributo style de la siguiente forma:

```
<ETIQUETA style="propiedad:valor; propiedad:valor; ..... ;">
```

Elementos sobre los que se aplica el estilo

```
</ETIQUETA>
```

```
<h1 style="color:red; font-weight:bold;">
```

Heading 1 color rojo y letra tipo negrita

```
</h1>
```



# HTML

# CSS

profesor Vladimir Bataller

## **<style>**

Permite crear una hoja de estilo dentro del propio documento html. Se especifica en la sección de cabecera <head>

**type** = "Tipo\_MIME" (Especifica el tipo de sintaxis que se va a emplear para definir el bloque de estilo)

En el caso de las hojas de estilo siempre tomará el valor "text/css".

**<head>**

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

...

**</style>**

....

**</head>**

# HTML

# CSS

profesor Vladimir Bataller

## <style> (sintaxis)

El contenido de la etiqueta <style> consiste en una serie de "selectores" para los cuales se definen las características de estilo.

**La sintaxis será:**

**Selector**{propiedad:valor; propiedad:valor; ..... ;}

**Ejemplo:**

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

```
  h1 {color:#101010; text-align:left;}
```

```
  h2 {color:#303030; font-style:italic; text-align:center;}
```

```
  p {color:#202020; font-style:italic; text-align:justify;}
```

```
</style>
```

**En este ejemplo hemos empleado como selectores determinadas etiquetas html. Más adelante veremos otros tipos de selectores.**

# HTML

# CSS

profesor Vladimir Bataller

## Hojas externas

Son archivos de texto que contienen definiciones de estilos con la misma sintaxis que en la etiqueta `<style>`, pero sin escribir dicha etiqueta.

Para hacer uso de una hoja externa desde una pagina html se incluirá la etiqueta `<link>` en las sección de cabecera de la página:

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

### Ejemplo:

Definimos el archivo **estilos.css** cuyo contenido podría ser el siguiente:

```
h1 {color:#101010; text-align:left;}
```

```
h2 {color:#303030; font-style:italic; text-align:center;}
```

```
p {color:#202020; font-style:italic; text-align:justify;}
```

Ahora en el documento html incluiremos:

```
<head>
```

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

```
</head>
```

# HTML

# CSS

profesor Vladimir Bataller

## Selectores de etiqueta

Permiten establecer un estilo para cada etiqueta html

### Sintaxis:

**Etiqueta**{propiedad:valor; propiedad:valor; ..... ;}

### Ejemplo:

**h1** {color:#101010; text-align:left;}

**h2** {color:#303030; font-style:italic; text-align:center;}

**p** {color:#202020; font-style:italic; text-align:justify;}

# HTML

# CSS

profesor Vladimir Bataller

## Selectores de clase

Permiten establecer clases con un estilo asociado, de forma que posteriormente podremos indicar que un bloque tenga el estilo correspondiente a dicha clase

### Sintaxis:

```
.nombre_clase{propiedad:valor; propiedad:valor; ..... ;}
```

### Ejemplo:

```
.entradilla {text-align:center;font-weight:bold;font-size:12px;}
```

```
.cita {font-style:italic;}
```

```
p.texto {text-align:justify;font-size:10px}
```

```
<h1>El precio de la vivienda, crece pero menos</h1>
```

```
<p class="entradilla">Aunque, el precio de la vivienda sigue subiendo a un ritmo muy alto se detecta una desaceleración. </p>
```

```
<p class="texto">Según el secretario de estado de vivienda <span class="cita">Juan García</span> en unos años se alcanzará .... </p>
```

# HTML

# CSS

profesor Vladimir Bataller

## Selectores de id

Permite establecer un estilo a un elemento d el documento, el cual queda determinado por su identificador (atributo id). No puede haber dos elementos con el mismo identificador.

### Sintaxis:

```
#nombre_id{propiedad:valor; propiedad:valor; ..... ;}
```

### Ejemplo:

```
#texto1 {color:green;}
```

```
#texto2 {color:red;}
```

```
p#texto3 {color:blue;}
```

```
<h1>Tres noticias</h1>
```

```
<p id="texto1">Primera noticia</p>
```

```
<p id="texto2">Segunda noticia</p>
```

```
<p id="texto3">Tercera noticia</p>
```

# HTML

# CSS

profesor Vladimir Bataller

## Selectores etiqueta-clase

Permiten especificar estilos para determinadas clases solamente cuando se aplican a una determinada etiqueta.

### Sintaxis:

```
etiqueta_exterior.clase {propiedad:valor;  
propiedad:valor; ..... ;}
```

### Ejemplo:

```
div.fondo{background-color:#EEEEEE;}
```

```
<div class="fondo"> Esto tiene fondo</div>  
<p class="fondo"> Esto tiene otro fondo distinto</p>  
<div> Esto tiene otro fondo fondo distinto o ninguno</p>
```

# HTML

# CSS

profesor Vladimir Bataller

## Selectores etiqueta-id

Permiten especificar estilos para determinadas clases solamente cuando se aplican a una determinada etiqueta.

### Sintaxis:

```
etiqueta#id etiqueta {propiedad:valor; propiedad:valor;  
..... ;}
```

### Ejemplo:

```
table#datos td {background-color:#EEEEEE;}
```

**Este estilo se aplica a todos los td que se hallen dentro de la tabla con id=datos**

.



# HTML

# CSS

profesor Vladimir Bataller

## Selectores contextuales

Permiten especificar estilos para determinadas etiquetas o clases, solamente cuando aparecen contenidas dentro de otras etiquetas.

### Sintaxis:

**etiqueta\_exterior etiqueta\_interior {propiedad:valor; propiedad:valor; ..... ;}**

### Ejemplo:

```
ul {list-style-image:url(imagenes/bullet1.gif);}
ul ul{list-style-image:url(imagenes/bullet2.gif);}
table#miTabla h1.titulo {font-size:12px}
```

```
<ul>      <li>Lunes </li>
          <ul>      <li> mañana </li>      <li> tarde </li>      </ul>
</ul>
<table> <tr>      <td> <h1> NOMBRE</h1></td>
          <td>Juan</td> </tr></table>
```

# HTML

# CSS

profesor Vladimir Bataller

## ! important

Cuando un elemento tiene la misma propiedad de estilo aplicada mediante diferentes selectores, se puede emplear !important para indicar cual prevalece.

```
div#caja{  
    border:1px solid black;background-color:yellow;}  
div.cuadro{width:400px;height:400px;  
    background-color:orange !important;}
```

```
<div id="caja" class="cuadro"></div>
```

.

# HTML

# CSS

profesor Vladimir Bataller

## Pseudo-clases

Aplican estilo a etiquetas que se hallan en un determinado estado. Básicamente se aplica a la etiqueta <A>

### Sintaxis:

**ETIQUETA:estado** {propiedad:valor; propiedad:valor; ..... ;}

### Ejemplo:

a:link { color: red } (enlace en reposo)

a:active { color: blue; font-size: 125% } (mientras se pulsa)

a:visited { color: green; font-size: 85% } (enlace visitado)

a:hover { color: black; font-size: 100% } (cuando el ratón está encima)

.

# HTML

# CSS

profesor Vladimir Bataller

## Pseudo-elementos

Aplican estilo a determinadas partes del contenido de una etiqueta como puede ser la primera letra o la primera línea.

### Sintaxis:

**ETIQUETA::parte** {propiedad:valor; propiedad:valor; ..... ;}

### La parte puede tomar los valores:

**first-letter** : para definir el estilo de la primera letra

**first-line** : para definir el estilo de la primera línea

### Ejemplos:

```
p{font-size:12px;}  
p::first-letter{font-size=20px;}  
h1::first-line{color:grey}
```

# HTML

# CSS

profesor Vladimir Bataller

## Posicionamiento

El posicionamiento es una característica que nos permite determinar con exactitud donde queremos situar cada elemento del documento.

**CAPAS:** Se pueden asimilar a rectángulos que podemos ubicar en la posición exacta que desee el diseñador. Dentro de estas capas se podrá insertar cualquier elemento html.

Para crear las capas se emplean las etiquetas <div> las cuales se verán afectadas por elementos de estilo dedicados al posicionamiento. En algunas ocasiones se puede emplear también la etiqueta <table> como capa.

Se pueden definir en cualquier parte del cuerpo del documento.

# HTML

# CSS

profesor Vladimir Bataller

## Propiedades de posicionamiento

Para determinar las coordenadas de una capa, empleamos las propiedades css left, top, height y width.

Ejemplo:

```
<div id="capa1" style="position:absolute;  
left:77px; top:46px; width:314px; height:156px;  
z-index:1;  
background-color: #9999FF; border: 1px solid #000000;">  
</div>
```

# HTML

# CSS

profesor Vladimir Bataller

## Display

Permite indicar si la etiqueta a la que se le incluye este estilo, agregará un salto de línea detrás de ella.

Valores posibles:

inline -> no agrega salto de línea.

block -> inserta un salto de línea después del elemento y su ancho, tiende a expandirse todo lo que le permita su contenedor a no ser que se le asigne un width.

inline-block -> Se le puede asignar un ancho y alto, pero permite que se ubiquen elementos a sus lados.

none -> No se muestra el elemento y deja libre su espacio para que lo ocupe el siguiente elemento.

```
<p style="display:inline">Este parrafo no produce un salto de linea.</p>
por lo tanto este texto sigue en la misma linea.<span
style="display:block"> Sin embargo este texto aparece en otra
linea</span>
```

# HTML

# CSS

profesor Vladimir Bataller

## Transparencia con rgba

En css, además de la función rgb para especificar el color, tenemos la función rgba que incluye la transparencia.

rgba(rojo, verde, azul, transparencia)

los colores se indican de 0 a 254

La transparencia se indica de 0.0 a 1.0

También está la propiedad de estilo "opacity" que permite asignar el canal alfa de forma independiente del color o imagen del elemento.

```
<div style="background-image:url(mifondo.jpg);">  
  <h3 style="background-color:rgba(250,150,150,0.5);">  
    Capitulo 1 </h3>  
</div>
```



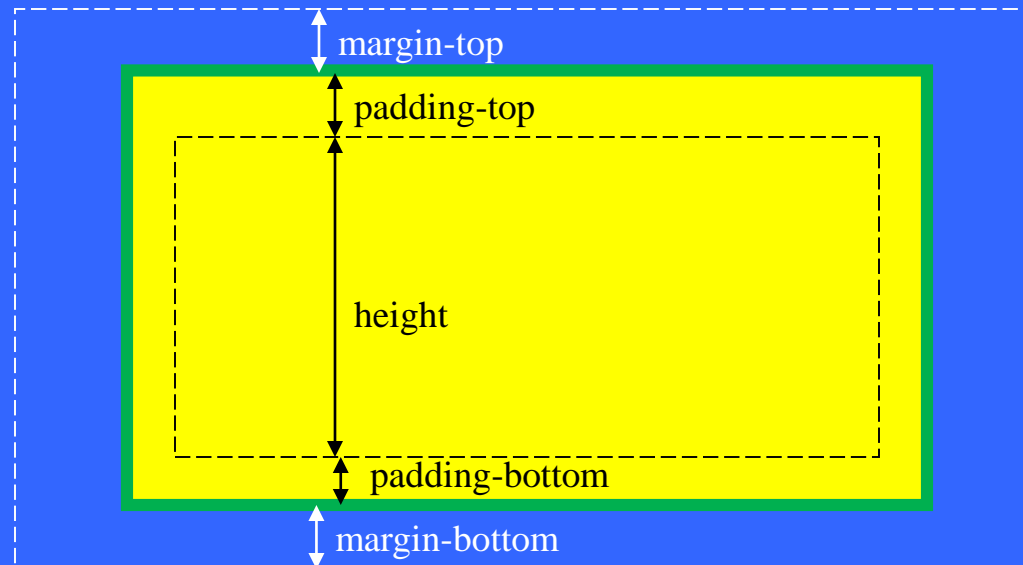
# HTML

# CSS

profesor Vladimir Bataller

## Tamaño de los bloques

El tamaño visible de un bloque es su height y width más los paddins vertical y horizontal.



Si se quiere establecer para toda la página que los width y height indicados incluyan los padding y borders, se debe indicar lo siguiente:

```
* { box-sizing: border-box; }
```

# HTML

# CSS

profesor Vladimir Bataller

## Elementos Flotantes

La propiedad de estilo **float** permite a indicar un elemento en que parte se colocará dentro de un bloque en el que se encuentre con más elementos.

Con esto se consigue el efecto de una imagen incrustada en un bloque de texto de un periodico, en el que el texto rodea a la imagen.

Valores posibles para float:

left -> alineación del elemento flotante a la izquierda.

right -> alineación del elemento flotante a la derecha.

none-> elimina el efecto flotante del elemento.

```
<p> En este texto la imagen aparece a la derecha incrustada entre el texto como si fuera un periódico.</p>
```

# HTML

# CSS

profesor Vladimir Bataller

## DOCTYPE

Indica que versión de html o xhtml sigue el documento.

Los navegadores tienen tres modos de renderizado de páginas:

- 1.- Estandar.
- 2.- Estricto.
- 3.- Compatibilidad (Quirk).

```
<p> En este texto la imagen aparece a la derecha incrustada entre el texto como si fuera un periódico.</p>
```

# HTML

# CSS

profesor Vladimir Bataller

## DOCTYPE XHTML Strict

Excluye todos los elementos html relativos al formato, obligando a centrarse solamente en el contenido y expresar el formato con css.

```
<!DOCTYPE html  
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Atención Netscape, interpreta mal `<br/>`. Este problema se resuelve agregando un espacio despues de "br" y antes de "/":

```
<br />
```

# HTML

# CSS

profesor Vladimir Bataller

## DOCTYPE XHTML Transitional

Permite ciertos elementos html relativos al formato. Adecuado para navegadores antiguos que no admiten CSS.

```
<!DOCTYPE html  
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

▪

# HTML

# CSS

profesor Vladimir Bataller

## DOCTYPE XHTML Frameset

Permite declarar un conjunto de marcos que visualizarán en una ventana del navegador varias páginas cada una en un área. Sustituye la etiqueta Body por frameset.

```
<!DOCTYPE html  
PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

▪

# HTML

# CSS

profesor Vladimir Bataller

## XHTML Atributos del núcleo (I)

Todos los elementos xhtml excepto tienen una serie de atributos básicos aplicables. Los cuatro siguientes no se aplican a base, head, html, meta, param, script, style, y title

class	Clase de estilo aplicada al elemento.
id	Identificador único del elemento empleado en scripts y css.
style	Definición de estilo in-situ.
title	Texto "tool-tip" de ayuda al pasar el ratón por encima

▪

# HTML

# CSS

profesor Vladimir Bataller

## XHTML Atributos del núcleo (II)

Aplicable a todos los elementos xhtml excepto a base, br, frame, frameset, hr, iframe, param, and script . Pero sí aplicable a head, html, meta, style, y title.

dir      Dirección del texto. Valores: rtl   o   ltr  
lang      Código del idioma del elemento.

### Atributos de teclado

accesskey      Tecla de acceso rápido  
tabindex      índice de tabulación.

▪



# HTML

# CSS

profesor Vladimir Bataller

## XHTML Alineación de Capas

La forma css de alinear una capa es mediante la propiedad margin.

```
<div style="width:150px;background-color:yellow;margin:20px auto;">
```

Esta capa aparece centrada horizontalmente

```
</div>
```

Nota la declaración de documento tiene que ser Strict si no, no funciona en IE6.

▪

# HTML

# CSS

profesor Vladimir Bataller

## Posicionamiento de Capas: El flujo

Para entender correctamente el posicionamiento, primero hay que conocer el concepto de flujo de los elementos en un documento html.

Los elementos html se distribuyen por defecto de acuerdo al flujo de la página. Es decir según están definidos en el documento html, se van mostrando en el navegador en líneas de arriba a abajo y el contenido de cada línea de izquierda a derecha.

El texto y las imágenes tienen un comportamiento "in-line" es decir solamente ocupan el ancho necesario y permiten que se ubiquen otros elementos "in-line" a sus lados. En este caso la altura de la línea será la altura del elemento más alto.

Los "divs", los "p" y los "h" tienen un comportamiento "block". En este caso el elemento intenta expandirse a todo el ancho de su elemento contenedor. Además incluye un salto de línea detrás de él.

## Posicionamiento : static

.

Cuando la propiedad Css "position" tiene asignado el valor static, el elemento se posiciona de acuerdo al "flujo" normal del documento y por lo tanto se ignoran los valores de las propiedades left, top, right o bottom.

Static es el valor por defecto de la propiedad position.

.

# HTML

# CSS

profesor Vladimir Bataller

## **Posicionamiento : absolute**

Cuando la propiedad Css "position" tiene asignado el valor absolute, se pueden dar dos casos

### **Si elemento esta dentro de otro posicionado como static**

En este caso las coordenadas del elemento hacen referencia a los límites del documento.

### **Si elemento esta dentro de otro posicionado como absolute, relative o fixed**

En este caso las coordenadas del elemento hacen referencia a los límites del elemento contenedor.

En cualquiera de los dos casos, el elemento se saca del flujo del documento, por lo que su espacio pasa a ser ocupado por los elementos que estan a continuación.

# HTML

# CSS

profesor Vladimir Bataller

## Posicionamiento : relative

Cuando la propiedad Css "position" tiene asignado el valor relative.

En este caso, las coordenadas del elemento indican el desplazamiento de este respecto a la posición que ocuparía en el flujo del documento (si fuera static).

El espacio ocupado por este elemento en el flujo del documento, **no** es ocupado por los elementos que están a continuación.

.

# HTML

# CSS

profesor Vladimir Bataller

## Posicionamiento : fixed

Cuando la propiedad Css "position" tiene asignado el valor fixed.

En este caso, las coordenadas del elemento indican el desplazamiento de este respecto a los límites del viewport (area visible del documento dentro del navegador) independientemente del scroll (desplazamiento con las barras de desplazamiento )que tenga la página en cualquier momento.

Estos elementos se sacan completamente del flujo del documento y es indistinto que estén dentro de otro elemento o sueltos en el body.

-

# HTML

# CSS

profesor Vladimir Bataller

## Página a tres columnas

Crear una página que tenga arriba una cabecera con el título, luego dos columnas y abajo del todo un pie de página.

Titulo

Menu

Contenido

Pie

# HTML

# Geolocalización

profesor Vladimir Bataller

## Utilidades para Geolocalización

Los navegadores móviles permiten mediante la propiedad viewport indicar el tamaño virtual del área visible del navegador.

Api de google que proporciona mapas a partir de urls como esta:

<http://maps.googleapis.com/maps/api/staticmap?center=40.5,-3.5&zoom=5&size=400x300&sensor=false>

▪



# HTML

# RWD

profesor Vladimir Bataller

## Flexbox

CSS3 introduce un conjunto de propiedades que permiten determinar el comportamiento (tamaño, posición y distribución) de un conjunto de cajas dentro de una contenedora.

### Propiedad *display* del contenedor

Cuando la propiedad `display` toma el valor *flex* o el valor *inline-flex* indica que ese elemento es un contenedor de cajas que se distribuirán de acuerdo al modelo Flexbox.

valor	comportamiento
flex	El contenedor a efectos externos se comporta como un block.
inline-flex	El contenedor a efectos externos se comporta como un inline.

Por defecto, los elementos se distribuyen en una sola línea y si no caben, el ancho de los elementos se adapta al total que tienen para repartir. Si les sobra espacio, entonces el sobrante queda a la derecha del último.\_

# HTML

# RWD

profesor Vladimir Bataller

**Flexbox:  
flex-direction**

Indica la dirección de los elementos interiores al contenedor.

**row -> se distribuye por filas (principio=izquierda, final=derecha).**

**row-reverse ->**

**column**

**column-reverse**

# HTML

# RWD

profesor Vladimir Bataller

## Flexbox: justify-content

Indica como se distribuyen los elementos interiores al contenedor si no usan todo el ancho disponible.

**flex-start -> Se pegan al principio (izquierda).**

**flex-end -> Se pegan al final (derecha).**

**center -> Se juntan todos en el centro, dejando el espacio a los lados.**

**space-between -> Dejan espacio entre los elementos .**

**space-around -> Dejan espacio entre los elementos y al principio y final.**

# HTML

# RWD

profesor Vladimir Bataller

## Flexbox: align-items

Indica como se alinean verticalmente los elementos cuando estos no ocupan toda la altura disponible.

**stretch -> los elementos interiores toman la altura de su contenedor si no tienen asignada la propiedad height.**

**flex-start -> se alinean al principio.**

**flex-end -> se alinean al final.**

**center -> se alinean centrados.**

**baseline -> se alinean las líneas base de los textos de los elementos (si hace falta el elemento sube o baja para conseguir esa alineación).**

# HTML

# RWD

profesor Vladimir Bataller

## Flexbox: flex-wrap

Indica si los elementos internos deben saltar (wrap) a la siguiente línea si no tienen espacio suficiente para mostrarse.

**nowrap ->**

**wrap ->**

**wrap-reverse -> saltan a la siguiente línea pero en orden inverso.**

# HTML

# RWD

profesor Vladimir Bataller

## Flexbox: align-content

Indica como se distribuye el espacio entre líneas.

**stretch->** Cada línea toma el mismo espacio y los elementos se alinean en la parte superior.

**flex-start ->** Las líneas se ubican lo más arriba posible que les permita la altura de sus elementos.

**flex-end ->** Las líneas se ubican lo más abajo posible que les permita la altura de sus elementos.

**center ->** Las líneas se ubican lo más cerca posible del centro del contenedor.

**space-between ->** distribuyen uniformemente el espacio entre líneas

**space-around ->** igual que between, pero con espacio al principio y final.

# HTML

# RWD

profesor Vladimir Bataller

## Flexbox: flex-box

Permite asignar, separados por espacios, los valores de flex-direction y flex-wrap.

Ejemplo:

```
flex-flow: row-reverse wrap;
```

# HTML

# RWD

profesor Vladimir Bataller

## Elementos interiores al flexbox: order

Indica un número por el que se ordenarán los elementos del mismo controlador. Por defecto todos tienen order:0.

Ejemplo:

**order:** -1; /\*Si los demás no tienen order, este se irá al principio..\*/



# HTML

# RWD

profesor Vladimir Bataller

## Elementos interiores al flexbox: align-self

Puede tomar los mismos valores que align-items y permite modificar la alineación de un elemento interior en concreto.

**stretch -> los elementos interiores toman la altura de su contenedor.**

**flex-start -> se alinean al principio.**

**flex-end -> se alinean al final.**

**center -> se alinean centrados.**

**baseline -> se alinean con la línea base.**

# HTML

# RWD

profesor Vladimir Bataller

**Elementos interiores al  
flexbox:  
margin:auto**

Los elementos interiores pueden emplear las propiedades margin, margin-left y margin-right con el valor auto para separarse todo lo posible de los demás. Absorbiendo todo el espacio libre.

**/\*consigue que todo el espacio libre de la fila quede a su derecha\*/  
margin-right:auto;**

**También se puede emplear cuando un único elemento está dentro del contenedor y queremos que el elemento esté centrado tanto horizontal como verticalmente. Para ello emplearemos:**

**margin: auto;**

# HTML

# RWD

profesor Vladimir Bataller

## Elementos interiores al flexbox: flex

Permite indicar un número que representa peso en el ancho que se le quiere aplicar a un elemento.

### Ejemplo:

```
div#capa1{flex:2}
```

```
div#capa2{flex:1}
```

```
div#capa3{flex:1}
```

La capa1 tendrá el doble de ancho que la 2 y la 3.

Si se indica la propiedad flex, se ignora la propiedad width del elemento.

Si queremos que un elemento ocupe toda una línea, se puede conseguir poniendo su propiedad flex-basis:100%. En este caso flex no debe haber sido asignado.

# HTML

# RWD

profesor Vladimir Bataller

## Meta - Viewport

Los navegadores móviles permiten mediante la propiedad viewport indicar el tamaño virtual del área visible del navegador.

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

**width:** anchura virtual (emulada) del viewport.

**height:** altura virtual del viewport.

**initial-scale:** la escala inicial del documento.

**minimum-scale:** la escala mínima que el usuario puede poner en el documento.

**maximum-scale:** la escala máxima que el usuario puede poner en el documento.

**user-scalable:** si se permite o no al usuario hacer zoom.

RWD – Responsive Web Design(Diseño sensible al dispositivo)

# HTML

# RWD

profesor Vladimir Bataller

## @media (media queries)

La instrucción css @media permite delimitar estilos aplicables solamente a dispositivos que cumplan ciertas condiciones.

```
body {  
    background-color: red;  
}  
  
@media only screen and (max-width: 500px) {  
    body {  
        background-color: blue;  
    }  
}
```

orientation: landscape / portrait