

# UT02 Uso de estilos

09 de Septiembre de 2018



I.E.S. Virgen de la Paz

# 1 Introducción a CSS

CSS - Cascading Style Sheets

- W3C propone la creación de un lenguaje que se utilizará para describir la presentación de documentos HTML o XML
- Se presentaron nueve propuestas. Las dos propuestas que se tuvieron en cuenta fueron la CHSS (Cascading HTML Style Sheets) de Håkon Wium Lie y la SSP (Stream-based Style Sheet Proposal) de Bert Bos..
- Entre finales de 1994 y 1995 Lie y Bos se unieron para definir un nuevo lenguaje que tomaba lo mejor de cada propuesta y lo llamaron CSS (Cascading Style Sheets).



# 1 Introducción a CSS

CSS - Cascading Style Sheets

- 1996, W3C publica la primera recomendación oficial, CSS1, ahora obsoleta.
- 1998 se publica la primera recomendación CSS2
- 2011 se revisa como CSS2.1, es hoy en día utilizada por todos los navegadores actuales
- CSS3 en desarrollo desde 1999 , dividido en módulos más pequeños pretende ampliar la versión CSS2.
  - Esquinas redondeadas
  - sombras
  - gradientes
  - Transiciones o animaciones
  - Nuevos layouts multi-columnas, cajas flexibles o grid layouts

# 1 Introducción a CSS

CSS - Cascading Style Sheets

- CSS Nivel 2, Revisión 1 recomendación que realiza ajustes menores a CSS2 basándose en la experiencia de trabajo con él entre 1998 y 2004.
- CSS Nivel 3, basada en módulos
  - soporte para texto flotante vertical
  - mejor manejo de tablas
  - idiomas internacionales
  - mejor integración con otras tecnologías XML como SVG, Math ML y SMIL.
- El CSS WG también está trabajando en conjuntos CSS especiales orientados a medios específicos como: CSS Mobile, CSS Print y CSS TV.

# 1 Introducción a CSS

## Declaración CSS

- Una hoja de estilo CSS está formada por **reglas** que indican la manera en la que se visualizará la página (reglas de estilo)
- Una regla CSS está formada por un **selector** y un **bloque de declaraciones**
- Una **declaración** CSS está formada a su vez por una propiedad y su valor separados por dos puntos

*Declaración*

**color: #990000;**

*Propiedad                      Valor*

# 1 Introducción a CSS

## Declaración CSS

- La **propiedad**: es un identificador, un nombre legible por humanos, que define qué característica es considerada.
- El **valor**: describe como las características deben ser manejadas por el motor. Cada propiedad tiene un conjunto de valores válidos, definido por una gramática formal, así como un significado semántico, implementados por el motor del navegador

# 1 Introducción a CSS

## Bloques y selectores CSS

- Las declaraciones son agrupadas en **bloques**, una estructura delimitada por una llave de apertura y una de cierre en la que se añaden las declaraciones CSS separadas por punto y coma.
- Un **selector** indica a qué elemento o parte de una página se aplica un determinado estilo
- La manera más sencilla, es usar las propias etiquetas HTML como selectores. Junto a un bloque de declaraciones forma una regla CSS

# 1 Introducción a CSS

## Regla CSS





# 1 Introducción a CSS

## Regla CSS

- Si un navegador encuentra un selector cuya sintaxis no comprende, éste ignorará la declaración entera y continúa con la siguiente,
- CSS contempla sintaxis para definir atributos para varios selectores y selectores con varios atributos.

```
h1, h2 {color: blue; background-color:red }
```



# 1 Introducción a CSS

## Comentarios CSS

- Los comentarios se utilizan para explicar el código y pueden ayudar cuando se edita el código fuente en una fecha posterior
- Los navegadores ignoran los comentarios
- Un comentario CSS comienza con / \* y termina con \* /.
- Los comentarios también pueden abarcar varias líneas

# 1 Introducción a CSS

## Incluir CSS en un documento HTML

- Existen tres modos distintos de aplicar las reglas de estilo a un documento HTML:
  - Estilos en línea.
  - Hojas de estilos internas.
  - Hojas de estilos externas
- Se puede combinar de hojas de estilo externas, con especificaciones de estilo internas y estilos en modo local en una misma página
- El navegador priorizará el estilo más próximo aplicando dicho estilo, siendo el más prioritario el estilo en línea, después las hojas de estilo internas y por último las hojas de estilos externas,

## 1.1 Añadir estilos a un documento con CSS

### Estilos en línea

- La primera alternativa es usando el atributo style dentro de las etiquetas HTML
- Es el peor y el menos utilizado, ya que tiene los mismos problemas que la utilización de las etiquetas <font>

`<h1 style="color:blue;margin-left:30px;">Esto es un encabezado</h1>`

## 1.1 Añadir estilos a un documento con CSS

### Hojas de estilo internas

- Las hojas de estilo internas se emplean cuando se define un número pequeño de estilos o cuando se quieren incluir estilos específicos en una determinada página HTML
- El principal inconveniente es que, si se quiere hacer una modificación en los estilos definidos, es necesario modificar todas las páginas que incluyen el estilo que se va a modificar

## 1.1 Añadir estilos a un documento con CSS

### Hojas de estilo internas

- Se añaden usando la etiqueta `<style>` dentro del mismo documento HTML y solamente se pueden incluir en la cabecera del documento (dentro del elemento `<head>`).

```
<head>
  <style>
    body {
      background-color: red;
    }
    h1 {
      color: blue;
      margin-left: 40px;
    }
  </style>
</head>
```

## 1.1 Añadir estilos a un documento con CSS

### Hojas de estilo externas

- Todos los estilos CSS se incluyen en un archivo de tipo CSS que se enlazan mediante la etiqueta <link> que se coloca en el <head> de la página
- Un archivo CSS es un archivo de texto plano cuya extensión es .css
- Se pueden crear todos los archivos CSS que sean necesarios y cada página HTML puede enlazar tantos archivos CSS como necesite.

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

# 1.1 Añadir estilos a un documento con CSS

## Hojas de estilo externas

- La etiqueta `<link>` incluye cuatro atributos cuando enlaza un archivo CSS:
  - **rel:** Indica el tipo de relación que existe entre el recurso enlazado y la página HTML. Para los archivos CSS, siempre se utiliza el valor `stylesheet`.
  - **type:** Se usa para indicar que se aplica un estilo formato css, su valor siempre es `text/css`.
  - **href:** Indica la URL del archivo CSS que contiene los estilos. La URL indicada puede ser relativa o absoluta y puede apuntar a un recurso interno o externo al sitio web.
  - **media:** Atributo opcional que indica sobre que dispositivo se aplicaran los estilos. `Handheld`, `print`, `projection`, `screen` `tv` `braille` o `all` .



## 1.1 Añadir estilos a un documento con CSS

regla CSS @import

- La regla CSS @import permite importar reglas desde otras hojas de estilo
- Estas reglas deben preceder a todos los otros tipos de reglas

```
@import "navigation.css"; /* Usando un string */
```

```
@import url("navigation.css"); /* Usando una url */
```

<https://developer.mozilla.org/es/docs/Web/CSS/@import>

## 1.2 Conceptos clave de CSS

### Herencia en CSS

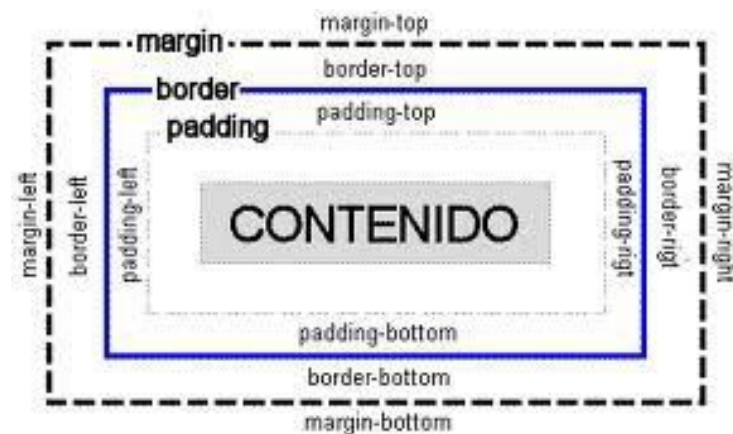
- Es la característica de CSS que al aplicar un estilo a un elemento, todos sus elementos hijos heredan este estilo
- Esto simplifica la creación de hojas de estilos complejas.
- Por ejemplo, la tipografía base de la página se debe establecer en el elemento `<body>` de la página y el resto de elementos la heredarán de forma automática
- Hay algunas características que no se heredan, por ejemplo
  - **border**, para evitar que se fuera duplicando los bordes en cada hijo
  - **font-size** si se fija como porcentaje, para evitar que cada hijo fuera reduciendo/aumentando el tamaño

## 2. El modelo de cajas de CSS

- Para poder estructurar nuestra página web de forma correcta vamos a dividirla en diferentes capas
- Cada una de estas capas va a poder recibir un formato diferente a través de las hojas de estilo
- Hasta ahora todos nuestros documentos estaban formados por una sola capa
- Para crear más capas en un documento usaremos la etiqueta `div`, que delimita un contenedor que incluye texto en párrafos, imágenes, tablas, etc
- La etiqueta `<div>` provoca un salto de línea.
- Cada capa es una “caja” que puede situarse en cualquier parte de la página y en la que podemos insertar contenido HTML.

## 2. El modelo de cajas de CSS

- Una capa no es más que una división lógica de nuestra página que nos va a permitir crear secciones y dotarlas de un “comportamiento” concreto e independiente del resto
- Una capa está compuesta por:
  - Un área de contenido: delimitada por los atributos width y height
  - El espacio alrededor de esta área (padding)
  - El borde
  - El espacio invisible alrededor del borde (margin)



## 2. El modelo de cajas de CSS

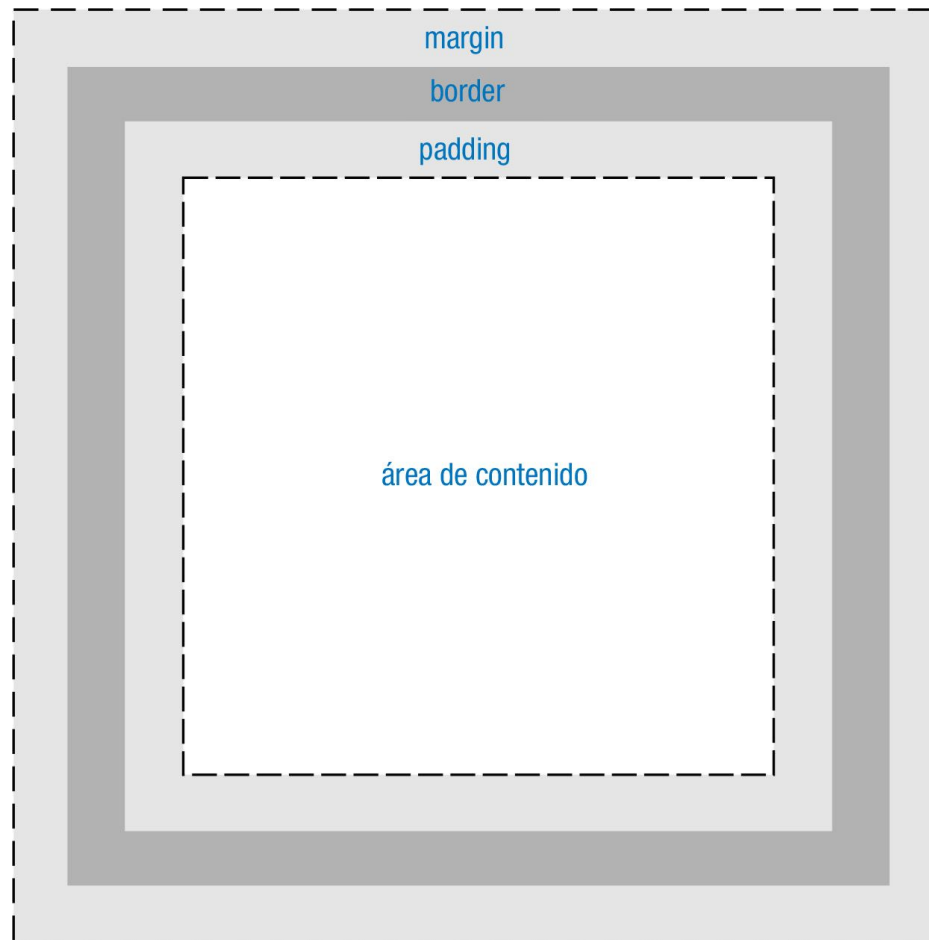
- **width**: indica la anchura de la capa
- **height**: indica la altura de la capa
- **margin**: margen
- **padding**: relleno
- **visibility**: indica si la capa se puede ver en la página o permanece oculta al usuario (deja vacío el hueco que ocupa la capa). Puede tener varios valores:
  - visible: la capa se muestra
  - hidden: la capa está oculta
  - inherit: valor por defecto, heredando la visibilidad de su objeto contenedor

## 2. El modelo de cajas de CSS

### Formatos de cuadro

- Se denomina cuadro a la zona visible o no que envuelve una etiqueta HTML
- Se calibran dos distancias: el margen (margin) y el relleno (padding – distancia del borde al texto)
  - **margin:**. Se pueden dar las cuatro medidas sólo con la etiqueta margin o una sola medida que se aplicará a las cuatro. margin-top, margin-right, margin-bottom, margin-left
  - **padding:** determina el espacio de relleno. Se usa igual que margin.
  - **border-width:** anchura del borde: border-top-width, border-right-width, border-bottom-width y border-left-width o border-width especificando las cuatro medidas.

## 2. El modelo de cajas de CSS



 Áreas transparentes

## 2. El modelo de cajas de CSS

### Formatos de cuadro

- **border-color**: especifica el color del borde.  
**border-style**: estilo de la línea: dotted, dashed , solid double, groove , ridge , inset , outset o none.
- **border-top, border-right, border-bottom y border-left**: permite indicar a la vez las tres propiedades de borde: anchura, estilo y color para cada línea del borde.
- **border**: igual que las anteriores pero para las cuatro líneas a la vez
- **border-collapse**: une los bordes de la tabla en uno o no: collapse | separate
- **width**: permite especificar la anchura (pensada para elementos que no tengan texto)
- **height**: permite especificar la altura del elemento  
**hover**: usado para aplicar efectos cuando pasamos el ratón por encima de un elemento



### 3. Selectores

#### Selectores

- Los selectores CSS son utilizados para "encontrar" (o seleccionar) elementos HTML
- Los selectores básicos CSS son
  - selector etiqueta
  - selector clase
  - selector id

### 3. Selectores

#### selector de clase

- El selector de clases selecciona elementos con un atributo de clase específico
- La sintaxis es

`.nombre_de_clase { propiedad de estilo: valor }`

`<h3 class="destacar">Encabezado rojo</h3>`

`<p class="destacar">párrafo rojo</p>`

---

`.destacar { color: red; }`

### 3. Selectores

#### selector de clase

- También se puede especificar que sólo los elementos HTML especificados serán afectados por una clase

```
<h3 class="destacar">Encabezado rojo</h3>
```

```
<p class="destacar">párrafo rojo</p>
```

---

```
p.destacar { color: red; }
```

- Los elementos HTML también pueden referirse a más de una clase

```
<p class="cuerpo destacar">A este párrafo se le aplican dos clases. </p>
```

### 3. Selectores

#### Selector id

- Se utiliza igual que el elemento class sólo que con otra nomenclatura, que se utiliza para diferenciar la escritura del JavaScript
- La sintaxis es  
`#nombre_de_id { propiedad de estilo: valor}`

```
#botonEnviar {  
    text-align: center;  
    color: red;  
}
```

- Se recomienda utilizar clases si no se utiliza JavaScript

### 3. Selectores

Seleccionar todos los elementos

- Existe un selector especial que permita aplicar un estilo CSS a todos los elementos de la página
- El selector \*, no es recomendable utilizarlo mucho al ser lento
- Ejemplo, para crear un borde negro de dos píxeles a todos los elementos de una página web añadirías:

```
* {  
    border: 2px solid black;  
}
```

### 3. Selectores

#### Anidar selectores CSS

- En HTML es posible anidar elementos HTML unos dentro de otros

```
<div>
```

```
    <div>
```

```
        <p>¡Me gustan las patatas!</p>
```

- Para seleccionar un párrafo <p> que está dentro de dos etiquetas <div> sin seleccionar todos los párrafos crear una etiqueta CSS como la siguiente:

```
div div p {  
    /*declaraciones CSS!*/  
}
```

### 3. Selectores

Selectores basados en relaciones

- **Todos los elementos contenidos dentro de otro**  
`p a { color: red; }`
- **Todos los elementos hijos directos de otro:**  
`p > a { color: red; }`
- **El primer hijo directo de otro:**  
`p : a { color: red; }`
- **Selector hermanos generales:** Se utiliza para seleccionar elementos que son hermanos (tienen el mismo padre) `h1 ~ h2 { color: red }`
- **Selector hermanos adyacentes:** elementos que son hermanos y están seguidos en el código  
`html h1 + h2 { color: red }`
- <https://developer.mozilla.org/es/docs/Web/CSS/Introducci%C3%B3n/Selectors>

### 3. Selectores

#### Selectores de atributos

- CSS3 incluye tres nuevos selectores para seleccionar elementos que contengan un atributo y un valor
  - **elemento[atributo^="valor"]**: contienen el atributo y comienza por la cadena de texto
  - **elemento[atributo\$="valor"]**: contienen el atributo y terminan por la cadena de texto.
  - **elemento[atributo\*="valor"]**: contienen el atributo y contiene la cadena de texto.

```
a[href*="daw"] {  
    color: purple;  
}
```



### 3. Selectores

#### Especificidad

- Especificidad son los medios por los que el navegador decide qué valores de una propiedad de CSS son más relevantes para un elemento y, por lo tanto, serán aplicados
- La especificidad está basada en las reglas que se aplican y están compuestas de diferentes formas de selectores CSS.
- Cuando la especificidad es igual en varias de las múltiples declaraciones, se aplicará la última declaración encontrada en el CSS

### 3. Selectores

Especificidad de menor a mayor

1. Selectores de etiqueta (h1) y pseudo-elementos (:before).
2. Selectores de clase (.boton), selectores de atributos ([type="radio"]) y pseudo-clases (:hover).
3. Selectores de ID (p.e., #logo).
4. Los estilos inline añadidos a un elemento (style="font-weight:bold")

### 3. Selectores

La excepción !important

- Cuando se emplea important en una declaración de estilo, esta declaración sobrescribe a cualquier otra.
- Aunque técnicamente !important no tiene nada que ver con especificidad, interactúa directamente con esta
- Sin embargo, el uso de !important es una mala práctica y debería evitarse porque hace que el código sea más difícil de depurar al romper la cascada natural de las hojas de estilo
- Cuando dos declaraciones en conflicto con el !important son aplicadas al mismo elemento, se aplicará la declaración con mayor especificidad.

### 3. Selectores

Cuando usar !important

- En declaraciones específicas de CSS que sobrescriban CSS externo (de librerías externas como Bootstrap o normalize.css).
- Tienes un archivo CSS que establece aspectos visuales de tu sitio de manera global.
- Tú (u otros) usas estilos inline en los propios elementos. Esto es considerado como una muy mala práctica, algunos plugins jQuery muy mal escritos que usan estilos inline

## 3.4 Pseudoselectores

### pseudo-class

- Permiten acceder a elementos HTML que no son parte del árbol del documento
- Por ejemplo si un enlace ha sido pulsado, si el ratón está sobre un elemento etc.
- Los selectores pseudo-class permiten dar estilo a estos cambios en nuestro documento HTML.

```
selector:pseudo-class_selector {  
    property: value;  
}
```

## 3.4 Pseudoselectores

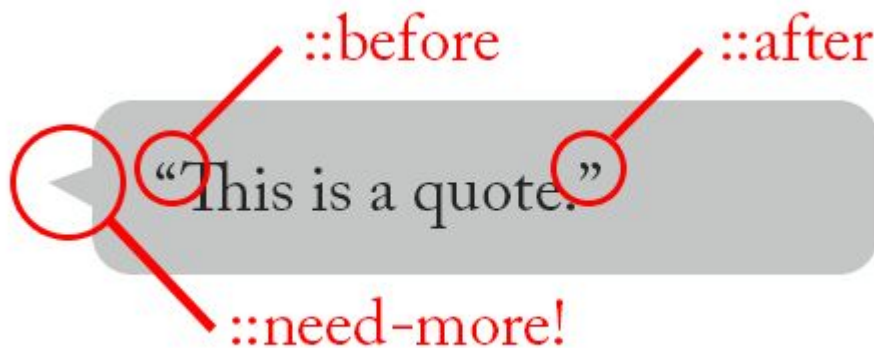
### Ejemplos pseudo-class

- a:link permite atribuir estilos al vínculo no visitado
- a:active permite atribuir estilos al vínculo activo
- a:visited permite atribuir estilos al vínculo visitado
- a:hover permite atribuir estilos al pasar por encima del vínculo (ponerlo en negrita, cursiva, cambia de color, etc)
- cursor permite cambiar la forma del cursor: auto, crosshair, default, hand, move, wait o help
- [https://www.w3schools.com/css/css\\_pseudo\\_classes.asp](https://www.w3schools.com/css/css_pseudo_classes.asp)

## 3.4 Pseudoselectores

### pseudo-elementos

- Al igual que las pseudo-classes, los pseudo-elementos se añaden a los selectores,
- No describen un estado especial sino que, permiten **añadir estilos** a una parte concreta del documento



## 3.4 Pseudoselectores

### Ejemplos pseudo-elementos

- **:after** Inserta contenido después del selector
- **:before** Inserta contenido antes del selector
- **:first-letter** Inserta contenido en la primera letra de cada selector
- **:first-line** Inserta contenido en la primera línea de cada selector
- **:selection** Selecciona la porción de un elemento que ha seleccionado el usuario
- [https://www.w3schools.com/css/css\\_pseudo\\_elements.asp](https://www.w3schools.com/css/css_pseudo_elements.asp)



## 4. Propiedades de fuente y texto

### Formato de fuente (font)

- **font-size**: tamaño de la letra
- **font-family**: tipo de letra
- **font-weight**: grosor de la fuente (normal|bold)
- **font-style**: estilo de letra (normal|italic|oblique)
- **font-variant**: normal | small-caps | inherit
- **font**: permite desde una sola propiedad cambiar todas las anteriores. Deben seguir un orden estricto:

`font: font-style font-variant font-weight  
font-size/line-height font-family;`

- **color**: color de la fuente

## 4. Propiedades de fuente y texto

### Formato de texto

- **word-spacing:** distancia entre las palabras del texto
- **letter-spacing:** distancia entre las letras del texto
- **text-decoration:** underline|overline|line-through|blink
- **vertical-align:** baseline|sub|super|top|text-top|middle|bottom|text-bottom|porcentaje
- **text-align:** left|right|center|justify
- **text-indent:** tamaño de la sangría de la primera línea
- **line-height:** distancia entre líneas
- **text-transform:** capitalize|uppercase|lowercase | none

## 4. Propiedades de fuente y texto

Formatos de lista para los elementos UL y OL

- **list-style-type:** indica el tipo de número o viñeta a utilizar.
  - Para UL: disc | square | circle | none.
  - Para OL: decimal | lower-roman | upper-roman | lower-alpha | upper-alpha | none
- **list-style-image:** permite indicar una imagen que se utilizará como viñeta (url)
- **list-style-position:** posición de los números o viñetas de la lista: inside (hacia dentro del relleno) | outside (hacia fuera)

## 4. Propiedades de fuente y texto

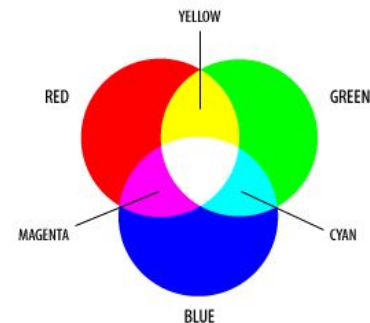
### Medidas texto

- **em.** Se corresponde al tamaño de la fuente para el párrafo en concreto al que se aplique. Si definimos que el texto tendrá un tamaño de 14 puntos, el valor de 1em será equivalente a 14 puntos
- **ex.** Similar a em, pero hace corresponder el tamaño actual de la fuente, al que tendría una equis minúscula (x)
- **px, (píxel)** relativa respecto de la resolución de la pantalla del dispositivo en el que se visualiza la página HTML.

## 5. Los colores y los fondos

### Formatos de fondo

- Muchas etiquetas indican colores,
- Hay tres formas de indicar el color:
- Nombre en inglés entre comillas  
"blue", "yellow", "red"
- Código hexadecimal seis cifras hexadecimales del 0 al FF , las dos primeras indican el nivel de rojo, las dos siguientes el de verde y las dos últimas el de azul  
#FF0000
- Código RGB. (RGB) 2 cifras decimales del 0 al 255 para el rojo, verde y azul.  
rgb(255, 0, 0)



## 5. Los colores y los fondos

### Formatos de fondo

- **background-color**: código de color|transparent
- **background-image**: url(url)
- **background-repeat**: indica cómo se va a repetir la imagen: repeat| repeat-x| repeat-y| no-repeat
- **background-attachment**: indica si la imagen queda fija al mover las barras de desplazamiento: scroll (se mueve)| fixed(queda fija)
- **background-position**: posición inicial (horizontal y vertical) de la imagen. Vertical: top|center|bottom y
- **horizontal: left|center|right**. También se puede especificar en porcentajes o en píxeles.
- **background**: fija en una propiedad todas las anteriores.

Debe seguir un orden estricto: background:

background-color background-image ..

I.E.S.

Virgen de la Paz

## 6 Flotar y posicionar position

- indica el tipo de posicionamiento de la capa. Primero ha de especificarse el tipo de posicionamiento y luego los valores a través de las propiedades top, bottom, right y left.
  - static: Valor por defecto
  - relative
  - absolute
  - fixed

## 6 Flotar y posicionar

position: relative

- Se comporta de la misma manera que static a menos que le agregues otras propiedades
- Aplica top, bottom, right y left para cambiar su posición natural

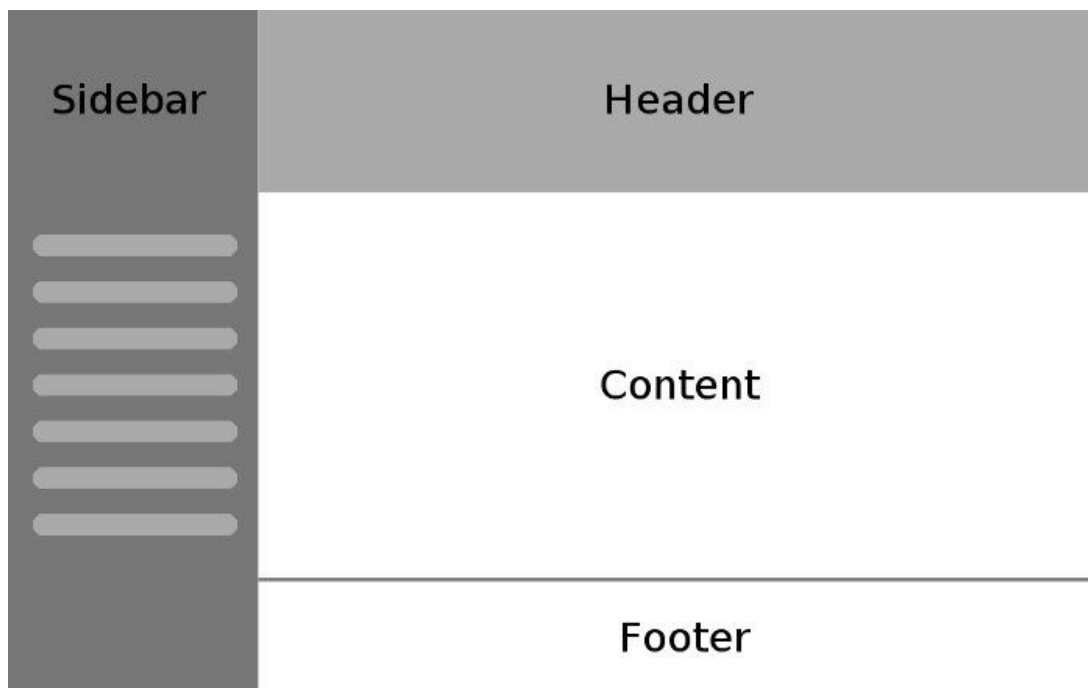




## 6 Flotar y posicionar

position: fixed

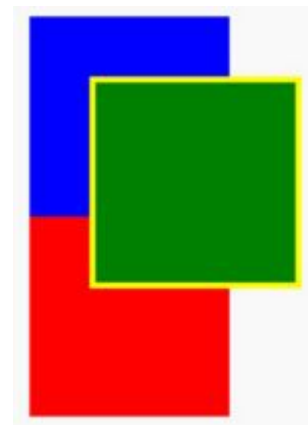
- Se posiciona a la ventana del navegador de manera relativa, lo que significa que se mantendrá en el mismo lugar incluso después de hacer scroll en la página.
- Muy útil para header, footer o menú de navegación



## 6 Flotar y posicionar

position:absolute

- Se comporta como fixed pero es relativo a su **ancestro posicionado más cercano** en lugar de ser relativo a la ventana del navegador.
- Si un elemento con position: absolute; no tiene ancestros posicionados, usará el elemento body del documento, y se seguirá moviendo al hacer scroll en la página.
- Recuerda, un elemento "posicionado" es aquel cuyo valor es cualquiera excepto static.



## 6 Flotar y posicionar

float

- Se usa sobre todo para envolver imágenes con texto

```
img {
  float: left;
  margin: 0 0 1em 1em;
}
```



### A floating image

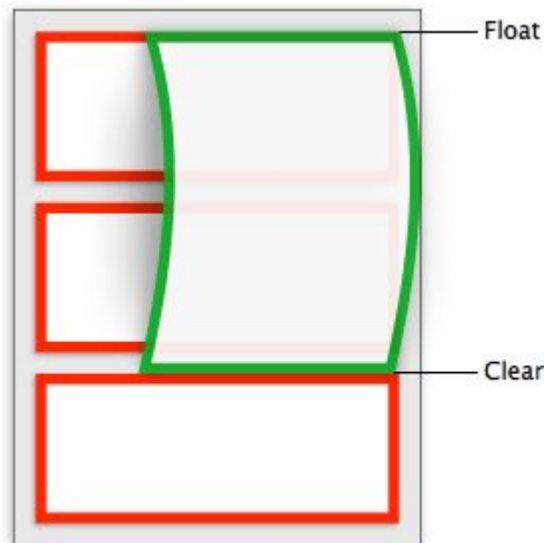
Iste quidem veteres inter ponetur honeste, qui vel mense brevi vel toto est iunior anno. Utor permisso, caudaeque pilos ut equinae paulatim vello unum, demo etiam unum, dum cadat elusus Interdum volgus rectum videt, est ubi peccat. Si veteres ita miratur laudatque poetas, ut nihil anteferat, nihil illis comparet, errat. Si quaedam nimis antique, si peraque dure

Interdum volgus rectum videt, est ubi peccat. Si veteres ita miratur laudatque poetas, ut nihil anteferat, nihil illis comparet, errat. Si quaedam nimis antique si peraque dure

## 6 Flotar y posicionar

clear

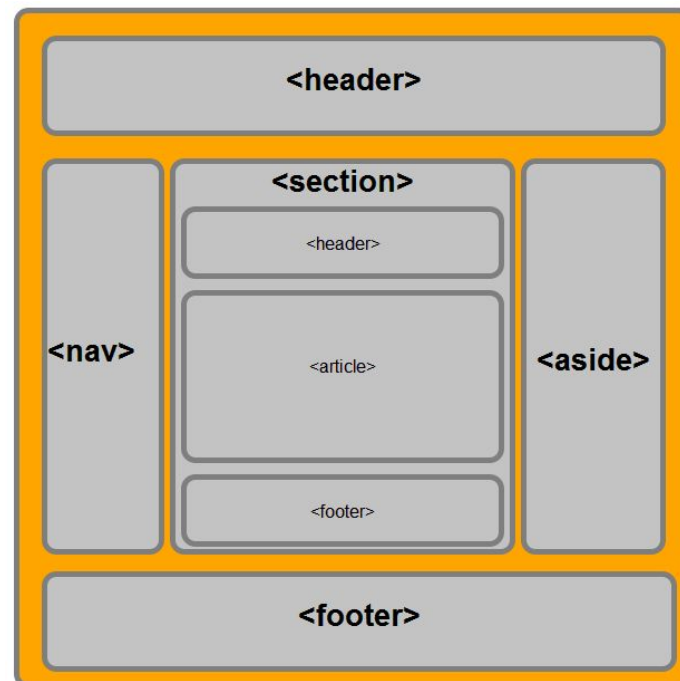
- indica el lado en el que no se admiten elementos flotantes:
  - **none**: se permite cualquier elemento flotante
  - **left**: no se admiten elementos flotantes a la izquierda
  - **right**: no se admiten elementos flotantes a la derecha
  - **both**: no se permiten a ningún lado
  - **inherit**



## 6 Flotar y posicionar

### Layout

- Layout es el diseño o estructura general de la página web creada mediante capas <div>
- Se pueden clasificar los diseños en función del número de columnas según dónde se coloque el menú de navegación
- **Layout líquido** es aquél que tiene el ancho de capas variable (usando porcentajes).



# 6 Flotar y posicionar

## Layout fijo con float y clear

```
<div id="container">
  <header>
    <h1>Título</h1>

    <h2>Subtítulo</h2>
  </header>
  <nav>
    <h3>Sección</h3>
    <p>Bla bla ...</p>
  </nav>
  <article>

    <h3>Sección</h3>
    <p>Contenido principal</p>
  </article>
  <footer>
    <p>Pie de página</p>
  </footer>
</div>
```

```
body {
  text-align: center;
}
header {
  text-align:center;
  clear: both;
  background-color:orange;
}
#container {
  width: 700px;
  margin: 0px auto;
  text-align: left;
}
aside {
  width: 200px;
  height:400px;
  padding: 10px;
  float: left;
  background-color:red;
}
article {
  margin-left: 225px;
}
footer {
  clear: both;
  background-color:green;
}
```



## 6 Flotar y posicionar

Layout con float y clear



## 6 Flotar y posicionar overflow

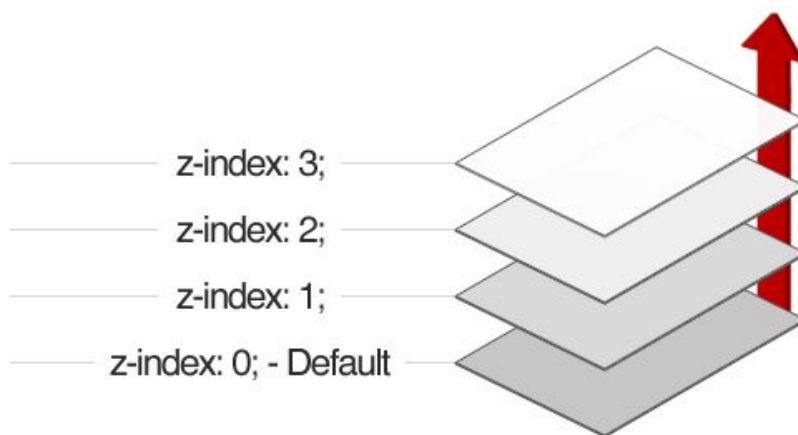
- Configura qué ocurrirá cuando el contenido de una capa no quepa en las dimensiones del div
  - **visible**: El contenido no es recortado, podría ser dibujado fuera de la caja contenedora
  - **hidden**: El contenido es recortado y no se muestran barras de posición.
  - **scroll**: Aparece una barra de desplazamiento para que sea visible el contenido
  - **auto**: Depende del navegador. En Firefox se muestra barras de desplazamiento si hay contenido excedente.



## 6 Flotar y posicionar

### z-index

- La propiedad **z-index** sirve para indicar la posición sobre el eje z de las capas, es decir, podemos indicar qué capas se verán por encima o debajo de otras, en caso de que estén superpuestas
- Toma valores numéricos y a mayor z-index, más arriba se verá la capa



## 6 Flotar y posicionar

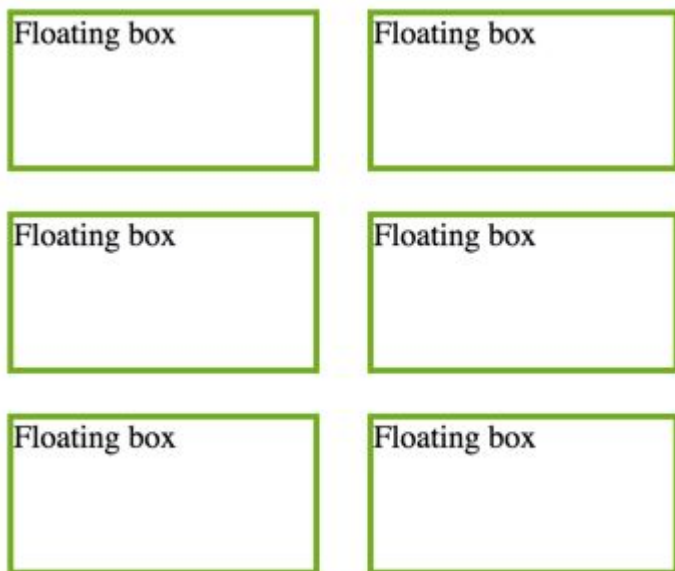
### display

- La propiedad **display** es la más importante para controlar estructuras.
- Cada elemento tiene un valor de display por defecto dependiendo de qué tipo de elemento sea.
  - **block**. Como los párrafos, los encabezados, las tablas, las listas, los div. Son los que fuerzan un salto de línea al final de los mismos.
  - **inline**. Como los enlaces, el énfasis o el span. Pueden ser hijos de cualquier otro elemento.
  - **none**. Usado en JavaScript para ocultar o mostrar elementos sin eliminarlos ni recrearlos.

## 6 Flotar y posicionar

display: inline-block

- La propiedad inline-block permite crear una cuadrícula de cajas que llene el navegador armoniosamente.
- Esto ha sido posible por mucho tiempo usando float, pero ahora con inline-block es aún más fácil.



```
.floating-box {  
  display: inline-block;  
  width: 150px;  
  height: 75px;  
  margin: 10px;  
  border: 3px solid #73AD21;  
}
```

## Buenas prácticas al escribir CSS

- Utilizar comentarios
- Los selectores se nombran en minúsculas, nunca empezando por caracteres especiales o numéricos
- Separa las palabras mediante guiones o mayúsculas, en las ultimas versiones se puede utilizar en el nombre el guión alto “-”
- El nombre de los selectores debe ser específico y claro, para que tenga una mayor capacidad expresiva.
- El nombre de clases e identificadores no debe describir una característica visual (color, tamaño o posición... seguir una visión semántica para facilitar los cambios.
- No hacer uso excesivo de clases.

## Buenas prácticas al escribir CSS

- Agrupar las reglas según su selector siempre que sea posible. Agruparlas unas debajo de otras:

```
[...]  
table {border:double}  
table.miembros {border:solid}  
table.empleados{border:grrobe}  
[...]
```

- Al principio del CSS definir los selectores de etiquetas.
- Estructurar visualmente los atributos. Si un elemento solo tiene 3 atributos se pueden poner en la misma línea. Pero si hay más se ponen en líneas diferentes sangrados con tabuladores.

