

## Unidad 3: CSS Básico - Modelo de Caja

|                                    |    |
|------------------------------------|----|
| Sintaxis                           | 2  |
| Formas de vincular css con el html | 3  |
| Selectores                         | 5  |
| 1. Selector Universal              | 6  |
| 2. Selector de etiqueta            | 6  |
| 3. Selector de Clase               | 7  |
| 4. ID                              | 7  |
| Propiedades CSS                    | 7  |
| Modelo de caja                     | 10 |
| Vínculos                           | 16 |
| Etiquetas de Bloque y de línea     | 18 |
| Propiedad Display                  | 19 |

# Sintaxis

El código css, tiene una sintaxis diferente al del HTML.

La sintaxis css, sigue el siguiente formato:

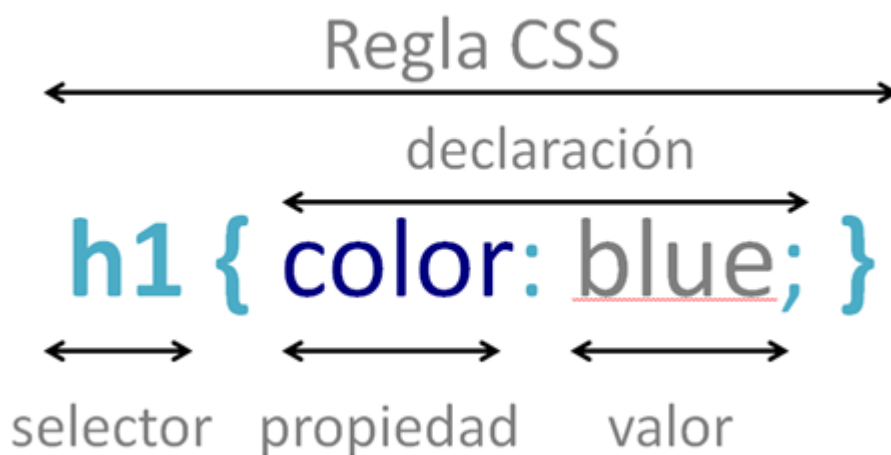
```
selector { propiedad: valor; }
```

El selector es aquello a lo que le queremos dar formato.

La propiedad es la característica visual que queremos modificar.

Por ejemplo:

```
h1 { color: blue; }
```



h1 sería el selector, la propiedad a la que le vamos a asignar un valor es el color (representa el color de fuente), y el valor en este caso es blue.

Esta regla css, establece que todas las etiquetas h1 de nuestros archivos html, van a tener color azul, como color de fuente.

Cada propiedad que yo quiera modificar representa una declaración.

Y todas las declaraciones en css, terminan con punto y coma.

Si quiero modificar varias propiedades para un mismo selector, separo cada declaración con un punto y coma.

En este ejemplo estoy estableciendo que todos los párrafos van a ser de color azul, y el tipo de fuente tahoma.



```
p {  
    color: blue;  
    font-family: tahoma;  
}
```

# Formas de vincular css con el html

Hay tres formas establecer un estilo CSS en un documento HTML:

1. En línea (Incrustado dentro de una etiqueta HTML)

```
<h2 style="color: blue;" > Subtítulo </h2>
```

Esta es la forma menos recomendada.

2. Dentro del mismo documento, dentro de las etiquetas <style></style>

Las etiquetas style, van dentro del head del documento.

```
<style>
```

```
h1{  
  
    color: blue;  
  
}
```

```
</style>
```

3. En un archivo .css externo. Esta es la forma más recomendada, ya que permite, poder vincular todos los archivos de un sitio a un mismo archivo css, y compartir estilos entre ellos, de forma más optimizada y mantenible.

```
<html>
```

```
<head>
```

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

```
</head>
```

```
<body>...
```

Lo primero que tengo que hacer es crear un archivo con la extensión .css

Y después dentro de la etiqueta head del html, lo vinculo a través de la etiqueta

link.

La etiqueta link tiene dos atributos:

- rel  
establece que tipo de archivo voy a vincular. El valor para las hojas de estilos es stylesheet  
quedando: `rel="stylesheet"`
- href  
en este atributo la ruta a la ubicación del archivo css  
`href="css/estilos.css"`

# Selectores

## 1. Selector Universal

Se utiliza para seleccionar todos los elementos de la página. El selector universal es el \*

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

El selector universal se indica mediante un asterisco (\*). A pesar de su sencillez, no se usa habitualmente, ya que es difícil que un mismo estilo se pueda aplicar a todos los elementos de una página. Sirve por ejemplo, para sacar el formato por defecto que tienen todas las etiquetas html.

## 2. Selector de etiqueta

Selecciona todos los elementos de la página cuya etiqueta HTML coincide con el valor del selector.

Para utilizar este selector, solamente es necesario indicar el nombre de una etiqueta HTML correspondiente a los elementos que se quieren seleccionar.

El siguiente ejemplo aplica diferentes estilos a los encabezados y a los párrafos de una página HTML:

```
h2 {  
  
    color: red;  
  
}  
  
p {  
  
    color: black;  
  
}
```

### 3. Selector de Clase

El selector de clase, sirve para aplicar estilos a ciertos elementos de una página web. Por ejemplo si quisiera aplicarle un estilo bold a algunos párrafos de mi sitio. En ese caso, la etiqueta p, no me serviría ya que se los aplicaría a todos.

Para poder utilizar el selector de clase, hay que agregar el atributo class de HTML sobre el elemento que quiero seleccionar.

```
<p class="destacado"> Párrafo destacado</p>
```

```
<p>Párrafo común.</p>
```

Una vez que tengo el html, creo en el archivo CSS una nueva regla llamada .destacado con los estilos que quiero aplicar.

En el css, los selectores de clase se indican anteponiendo al nombre de la clase un punto (.)

```
.destacado { font-weight: bold; }
```

Aplica lo declarado a todos los elementos que posean la clase indicada. Puede repetirse muchas veces en una misma página.

### 4. ID

El atributo id, es otra de las maneras de poder aplicar estilos a un elemento específico de nuestra página web. A diferencia de las clases, que se pueden repetir varias veces en una misma página, el atributo id, es unívoco, es decir, solo un elemento en nuestro html puede tener un mismo valor de id.

En el html:

```
<div id="contenedor"> </div>
```

La sintaxis de los selectores de ID es muy parecida a la de los selectores de clase, salvo que se utiliza el símbolo del numeral (#) en vez del punto (.) como prefijo del nombre de la regla CSS:

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

## Propiedades CSS

| Propiedad                              | Valores  |
|--|--|
| <b>color</b><br><i>Color del texto</i> | Valores posibles:<br>1. Nombre de color<br>green   red   blue   orange ....<br><pre>p{   color: red; }</pre>   |
|  | 2. Código hexadecimal (# antes del código hexadecimal)<br><pre>p{   color: #000099; }</pre>  |
|  | 3. Color RGBA (red, green, blue, alpha)<br>Los valores, van separados por comas.<br>La A, representa la transparencia, los valores posibles van del 0 al 1 en decimales separados por punto.<br><pre>p{   color: rgba(255,0,25,1); }</pre> |
| <b>font-family</b>                     | Arial, san-serif;<br><br>Times New Roman, serif;   |



|  |  |
|--|--|
| <b>tipo de fuente</b><br><i>tipo de fuente</i> | <pre>p{     font-family: Tahoma, san-serif; }</pre>  |
| <b>font-size</b><br><i>tamaño de letra</i>     | <p>Medidas posibles :</p> <ul style="list-style-type: none"> <li>• px (píxeles)</li> <li>• em (medida relativa)</li> <li>• % (porcentajes)</li> </ul> <p>Ejemplo: 12px, 1em, 80%</p> |
|  | <pre>h1{     font-size:12px; }</pre>   |
| <b>font-style</b><br><i>estilo</i>             | <p>Valores Posibles: normal   italic   oblique</p>   |
|  | <pre>p{     font-style: italic; }</pre>  |
| <b>font-weight</b><br><i>grosor de letra</i>   | <p>Valores posibles: 100   200   400 600   normal   bold</p>   |
|  | <pre>p{     font-weight: bold; }</pre>   |

|   |   |
|---|---|
|   |   |
| <b>text-align</b><br><br><i>alineación de texto</i> | Valores posibles: left   right   center justify |
|   | <pre>p{     text-align:center; }</pre>          |

# Modelo de caja

El modelo de cajas es el comportamiento de CSS que hace que todos los elementos de las páginas se representan mediante cajas rectangulares.

Las cajas de una página se crean automáticamente. Cada vez que se inserta una etiqueta HTML, se crea una nueva caja rectangular que encierra los contenidos de ese elemento.

Las cajas de las páginas no son visibles a simple vista porque inicialmente no muestran ningún color de fondo ni ningún borde.

Bajo este modelo, toda etiqueta html, cuenta con:

Un ancho (propiedad width)

- Un Alto (height)
- Un borde (que puede ser sólido, a rayas, con puntos, con algún grosor, con algún color o incluso sin borde, esta propiedad es border)
- Un margen interno (padding)
- Un margen externo (margin)
- Un fondo (que puede ser un color o una imagen)

| Propiedad                    | Valores  |
|------------------------------|--|
| <b>width</b><br><i>ancho</i> | Medidas posibles : px , em, % (porcentajes)<br><br><code>div{ width: 300px; }</code> |

|  |  |
|--|--|
| <b>height</b><br><br><i>alto</i>                       | <p>Medidas posibles : px , em, % (porcentajes)</p><br><pre>div{height: 100px;}</pre>   |
| <b>padding</b><br><br><i>Margen interior o relleno</i> | <p>Medidas posibles : px , em, % (porcentajes)</p>   |
|  | <pre>p{     padding: 5px; }</pre> <p>Agrega un padding en los 4 lados de la caja.</p>  |
|  | <pre>padding-top: 5px; (agrega un padding superior)  padding-left:5px; (agrega un padding izquierdo)  padding-bottom:5px; (agrega un padding inferior)  padding-right:5px; (agrega un padding derecho)</pre> |
|  | <p>Medidas posibles : px , em, % (porcentajes)</p>   |

|   |   |
|---|---|
| <p><b>margin</b></p> <p><i>margen externo</i></p>       | <pre>div{     margin: 5px; }</pre> <p>Agrega un margen en los 4 lados de la caja.</p> <hr/> <p>margin-top: 5px;<br/>(agrega un margen superior)</p> <p>margin-left: 5px;<br/>(agrega un margen izquierdo)</p> <p>margin-bottom: 5px;<br/>(agrega un margen inferior)</p> <p>margin-right: 5px;<br/>(agrega un margen derecho)</p> |
| <p><b>border-color</b></p> <p><i>color de borde</i></p> | <p>Valores posibles: Nombre de color, Código hexadecimal, RGBA</p> <pre>div{     border-color: #000000; }</pre>   |

|  |   |
|--|---|
| <b>border-style</b><br><i>estilo de borde</i>      | Valores posibles: solid   dashed   dotted   none   double   |
|  | <pre>div{     border-style:solid; }</pre>   |
| <b>border-size</b><br><i>grosor de borde</i>       | Medidas posibles : px , em, % (porcentajes)   |
|  | <pre>p{     border-size:2px; }</pre>  |
| <b>border</b><br><i>borde (propiedad genérica)</i> | <p>Esta propiedad permite combinar todas las propiedades que vimos vinculadas a bordes en una sola.</p> <p>Es un genérico que nos permite especificar en la misma propiedad, el grosor tipo y tamaño.</p> |
|  | <pre>p{     border:2px solid #000000; }</pre>   |

|  |  |
|--|--|
|  | <p>Al igual que con el margin y el padding, se puede especificar algún lado específico de la caja:</p> <p>border-left: 2px;</p> <p>border-bottom: 2px;</p> <p>border-right: 2px;</p> <p>border-top: 2px;</p>   |
| <p><b>background-color</b></p> <p><i>color de fondo</i></p>  | <p>Valores posibles:</p> <ol style="list-style-type: none"> <li>1. Nombre de color</li> <li>2. Código hexadecimal</li> <li>3. RGBA (en caso de querer un fondo de color con transparencia)</li> </ol> <pre>div{     background-color: #cc4433; }</pre> |
| <p><b>background-image</b></p> <p><i>imagen de fondo</i></p> | <p>La ubicación de la imagen se especifica con url(ubicación).Por ejemplo:</p> <pre>body{     background-image: url(imagen.jpg); }</pre>   |

|  |   |
|--|---|
|  | }   |
| <b>background</b><br><br><i>fondo (propiedad genérica)</i> | <p>Al igual que border, existe la propiedad genérica background</p> <pre>body{     background: #cc4433 url(imagen.jpg); }</pre> |



# Vínculos

La etiqueta `a`, de `html` es la que nos permite generar vínculos.

Los vínculos (links), tienen diferentes estados. El estado más conocido es el estado `hover`, que es cuando pasamos el mouse por encima de un vínculo, hay otros estados, como el estado `active`, que es cuando hacemos clic sobre el vínculo, o el estado `visited`, que es cuando ya visitamos un link.

Los vínculos por defecto son de color azul y subrayados.

Con la propiedad `color`, podemos modificarle el color a cualquier vínculo.

Y si quisiéramos sacarle el subrayado, la propiedad que tendríamos que utilizar es `text-decoration`.

| Propiedad                    | Valores  |
|------------------------------|--|
| <code>text-decoration</code> | Valores Posibles: <code>none</code>   <code>underline</code>   <code>overline</code><br><br><code>a{</code><br><br><code>text-decoration: none;</code><br><br><code>}</code> |

Si quisiéramos que algo se modificara en un vínculo al pasar el mouse, tendríamos que poder darle formato, a un estado específico del vínculo, que es el estado `hover`.

Para poder darle formato, tenemos que poder seleccionarlo. El selector de estados, tienen el siguiente formato, `nombreDeEtiqueta:estado`

Por ejemplo `a:hover {}`

```
a{ text-decoration:none;}
```

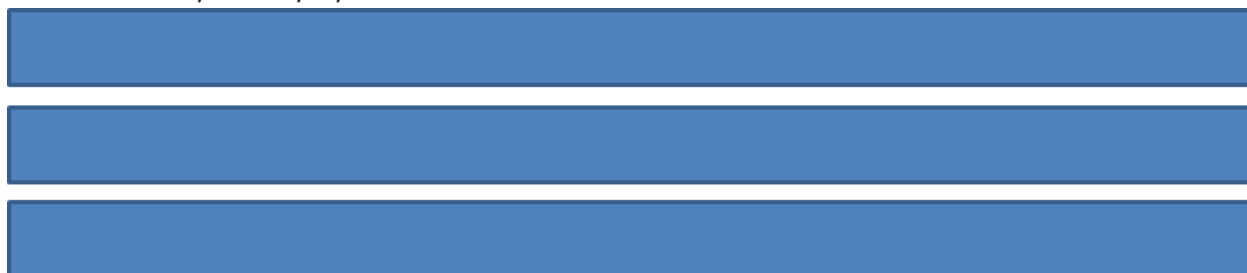
```
a:hover{  
  
color:red;
```

```
text-decoration:underline;  
}
```

# Etiquetas de Bloque y de línea

Existen dos tipos de etiquetas: las etiquetas de bloque y las etiquetas de línea.

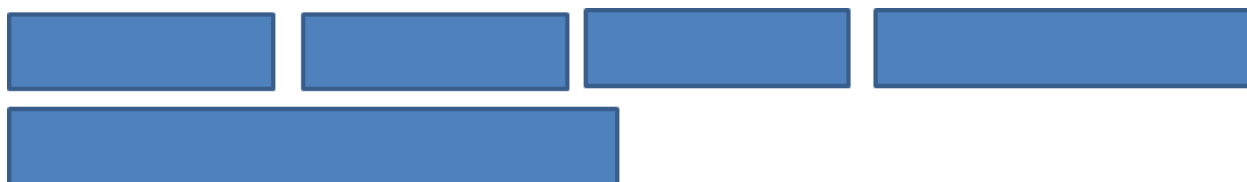
- Las etiquetas de bloque, ocupan el 100% del ancho, y el alto de su contenido.  
Ejemplos de etiqueta de bloque: div, p, h1-h6, article, section, main, nav, aside, header, footer, ul, li.



Al ocupar el 100% del ancho de la pantalla, no pueden ponerse una al lado de la otra, sino que van una debajo de la otra.

- Las etiquetas de línea, no tienen ancho por defecto, y su ancho es el ancho de su contenido. Por ende se ponen una al lado de la otra, si tuviera varias.  
A diferencia de las de línea, no pueden empujarse por los márgenes de arriba y abajo, ni tampoco centrarse en la pantalla, requieren para eso estar dentro de una etiqueta de bloque contenedora.

Ejemplos de etiqueta de línea: a, img, label, span.



## Propiedad Display

La propiedad display, me permite cambiar una etiqueta de bloque a línea, o de línea a bloque u ocultar un elemento. También sirve para habilitar dos propiedades que vamos a estar viendo en las próximas clases que son flexbox y grid, que nos van a permitir maquetar a varias columnas.

Algunos de los valores posibles:

- **display: none;**  
Oculto el elemento
- **display: block;**  
Convierte un elemento en bloque
- **display: inline;**  
Convierte un elemento en línea
- **display: inline-block;**  
Convierte un elemento en línea bloque, esto en concreto es que el elemento, tiene solo el ancho de su contenido y se pone uno al lado del otro, como el comportamiento de las etiquetas de línea, pero toma el margin-top y margin-bottom, que no funcionan en etiquetas de línea
- **display: flex;**  
Habilita flexbox
- **display: grid;**  
Habilita grid