

CSS

Bases de las hojas de estilos

Introducción

Es el único lenguaje de estilos que existe, para cualquier web en la que se le quiera dar estilo se utiliza CSS. Sus siglas significan Cascade Style Sheet que traduce hojas de estilos en cascada.

Aunque no es un lenguaje de programación cuenta con funciones propias y variables.

Sintaxis

```
Selector{  
    propiedad: valor;  
}  
  
body{  
    background: black;  
}
```

A este conjunto entero se le llama regla, una regla puede tener tantas modificaciones del selector como se necesiten. Siempre cerrar una modificación con ; (punto y coma).

Tipos de selectores



Selectores básicos y selectores combinadores, son los más utilizados a la hora de realizar modificaciones y aplicar reglas.

Luego están los selectores de pseudo-elementos y selectores de pseudo-clases, que son de un tema un poco más avanzado.

Selectores Básicos

Selectores de etiqueta

```
p{  
color: red;  
}  
  
h1{  
font-size: 60px;  
}
```

Selectores de clase

```
.title{  
color: blue;  
}  
  
.text{  
font-size: 12px;  
}
```

Selectores Básicos

Selectores de id

No se recomienda para dar estilos, se utilizan más para diferenciar componentes y ser utilizados en HTML para las anclas (referencias a diferentes partes del mismo documento) y en JS para identificar específicamente un elemento en el DOM, recalmando que no se debe usar para estilos, un ejemplo sería

```
#title{  
color: blue;  
}
```

Selectores Combinadores

```
h1 + h2 {  
color: red;  
}
```

Selector de hermano adyacente:
selecciona a un hermano justo
debajo

Selector de hermano general:
buscar todos los hermanos que
compartan el mismo parente

```
h1 ~ h3 {  
color: green;  
}
```

Selectores Combinadores

Selectores descendentes: Aplica el estilo a todos los elementos que sean hijos

```
p > span{  
color: red;  
}
```

```
div span{  
color: blue;  
}
```

Selector de hijo directo: Se aplica al hijo directo del elemento seleccionado, el hijo directo es el que está en nivel 1 descendente:

Herencia, Cascada y Especificidad

Son los aspectos base del funcionamiento de CSS, esto permite conocer el porqué de un comportamiento de un elemento.

Actualmente se utilizan también arquitecturas o metodologías que hacen que la solución de conflictos sea más sencilla de resolver para el navegador.

Herencia



Se obliga a un elemento a heredar la propiedad de su elemento cercano

```
p{  
color: red;  
}  
  
a{  
color: inherit;  
}
```

Cascada



La manera como se leen las hojas de estilos y la forma en cómo se sobreescreiben los estilos a medida que se baja en el documento.

```
p{  
color: red;  
}  
  
a{  
color: inherit;  
}  
  
p{  
color: green;  
}
```

Especificidad

Peso de los estilos cuando
hay conflicto entre selectores

Etiqueta

1

Clases y pseudo clases 10
ID

100

Estilos en línea 1000
!important
++++

Especificidad

```
#parrafo{  
color: blueviolet;  
}  
  
.parrafo{  
color: red;  
}  
  
a{  
color: inherit;  
}  
  
p{  
color: green !important;  
}  
  
.parrafo-2{  
color: purple;  
}
```

Reescribir estilos de navegadores

Normalize -> <https://necolas.github.io/normalize.css/>

Metodología BEM

Es una forma de estructurar clases CSS de forma sencilla, escalable y reutilizable, es una de las metodologías más utilizadas en el mundo.

BEM viene de sus siglas en inglés Block, Elemento y Modifier

B - Bloque

.menu
.menu-principal

Un bloque (Block) es cualquier elemento autónomo y aislado del documento HTML, puede ser un menú, una galería, un formulario, una sección.
Se nombra con una palabra o una palabra seguida de un guión medio

E - Elemento

Un elemento (Element) es cada uno de los elementos del bloque, en un menú es un enlace, en una galería es una foto, en un formulario es un campo de texto o botón, en una sección puede ser el título.

Se nombra con el bloque al que pertenece seguido de dos guiones bajos y el nombre del elemento

```
.menu__item  
.menu-principal__link
```

M - Modificador

```
.menu__item--active  
.menu-principal__link--big
```

Un modificador (modifier) es cuando un bloque se repite en algún otro lugar de la web pero con modificación como color de texto, tamaño de la fuente, decoración.

Se nombra con el nombre del elemento seguido de dos guiones medios y el nombre del modificador

Box-Model

En la web todos los elementos se pueden representar como cajas, una manera de verlo es inspeccionando el código, lo que se resalta al momento de pararse sobre un elemento es un rectángulo, esta forma en la que el navegador dibuja estas cajas se llama el LAYOUT, que hace referencia a la geometría de la web, la posición de un elemento con respecto a.

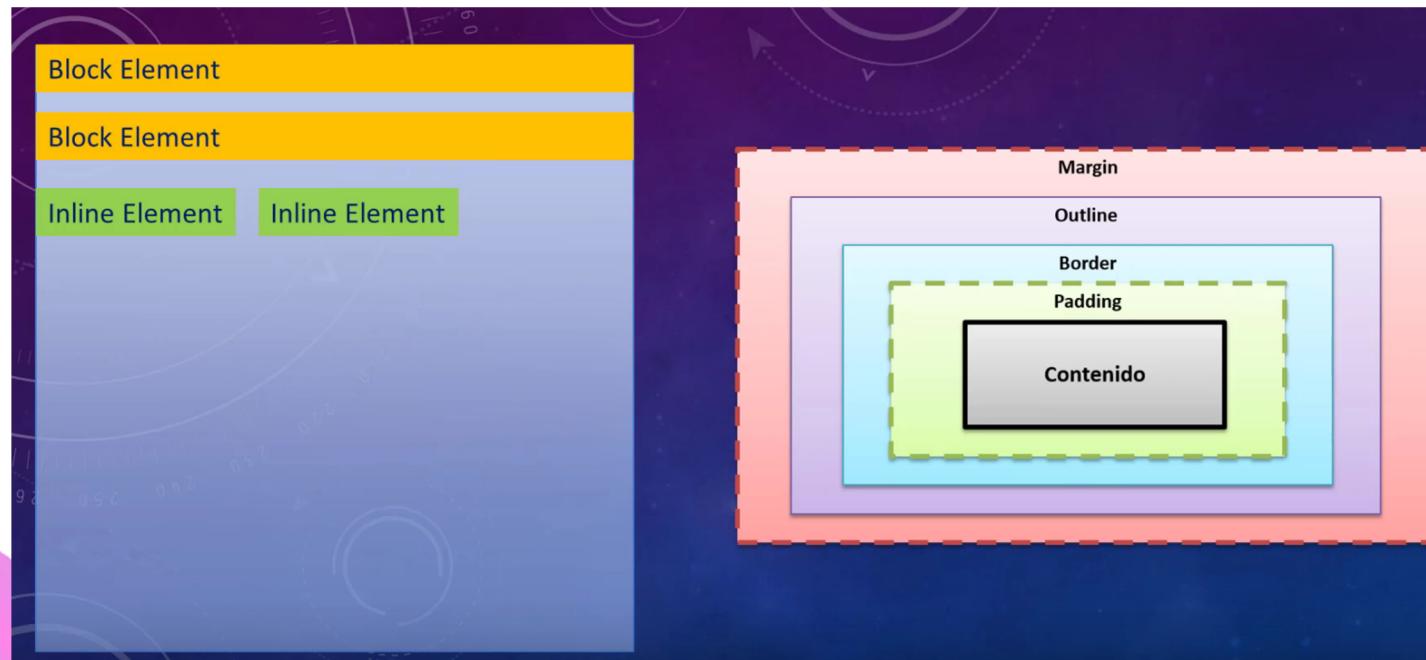
Las propiedades principales de cada caja es el ancho (width) y el alto (height).

Box-Model

Existen dos tipos de elementos HTML:

Elementos inline que solo ocupan su contenido y no se puede modificar su ancho o alto.

Elementos block que ocupan todo el ancho disponible y se les puede asignar un ancho y alto. ViewPort. por defecto el ancho es todo lo que puedan ocupar y el alto es lo que ocupe su contenido



Margin y Padding

Margin es la separación entre una caja y las cajas adyacentes, mientras que el padding es la separación entre una caja y su borde o límite.

Border

Es la línea que rodea la caja o contenedor, similar al margin y padding es un shorthand dado que el borde se puede representar para cada uno de los lados. Los bordes tienen propiedad Width, Style y Color

```
.Caja-1{  
border: 5px solid red;  
border-top: none;  
}  
  
.Caja-2{  
border-left-width: 5px;  
border-left-style: dotted;  
border-left-color: blue;  
}
```

Outline

```
.caja-1{  
outline: 5px solid red;  
outline-offset: -20px;  
}
```

Es una línea que rodea la caja entre el border y el margin, tiene las propiedades Width, Style, Color y Offset.

Overflow

Se usa cuando un elemento ocupa más del espacio de su contenedor y no es el comportamiento requerido

```
.selector{  
overflow: hidden; /*Para ocultar el  
contenido sobrante*/  
overflow: auto; /*para mostrar una  
barra de scroll donde haga falta*/  
overflow: scroll; /*para mostrar  
ambas barras de scroll así no hagan  
falta*/  
}
```

Float

Han sido reemplazado por flexbox, aún se utiliza en casos concretos como que un texto rodee a una imagen

```
.selector{  
float: left; /*para que el elemento  
flotado se ubique a la izquierda*/  
float: right; /*para que el elemento  
flotado se ubique a la derecha*/  
}
```

y como truco, se utiliza
overflow: hidden en el
contenedor padre para que
contenga a los elementos
flotados