

CSS3

Guia de estilos

AGENDA

CSS3

Selectores

Combinadores

Pseudoclases / Pseudoelementos

Especificidad

Gustavo Adolfo Garcia Blanco

propiedad display

propiedad position

BEM

Web Responsive

CSS3

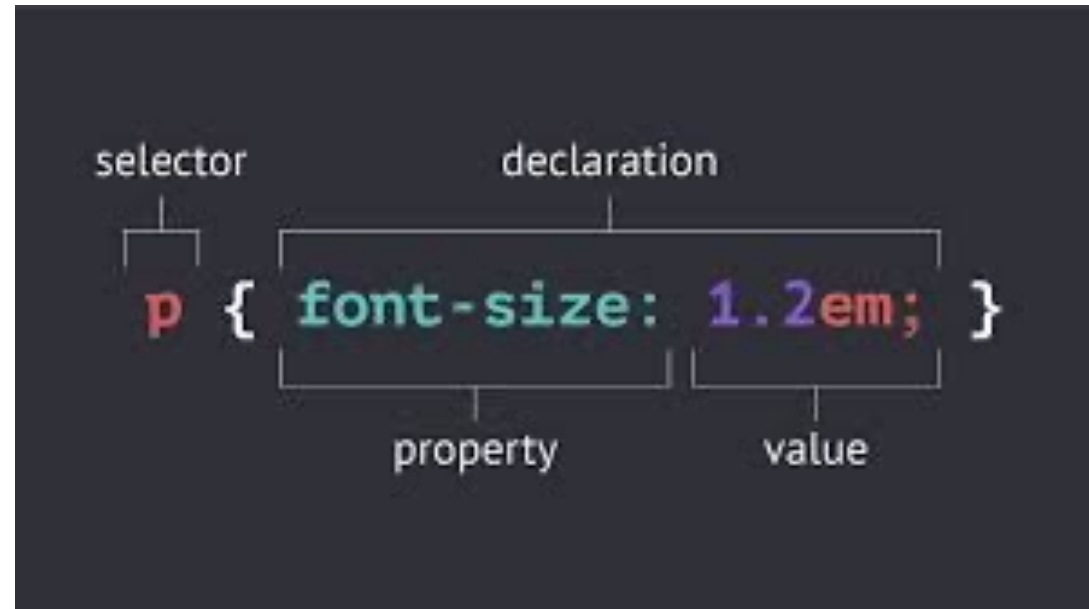
CSS3 es modular, ya que la especificación de estilos fue creciendo tanto, que **hubo que desarrollarla modularmente**. De hecho es una especificación que se viene desarrollando desde 1999.

Gustavo Adolfo García Blanco

SELECTORES

Un *selector* CSS es la primera parte de una regla CSS. Es un patrón de elementos y otros términos que indican al navegador qué elementos HTML se seleccionan para aplicarles una regla que incluye los valores de las propiedades CSS. El elemento o los elementos seleccionados por el selector se denominan *sujeto del selector*.

SELECTORES



SELECTORES

Tipos de selectores:

- Universal

```
*{ }
```

- De tipo

```
h1 { }
```

- De clase

Gustavo Adolfo Garcia Blanco

```
.title { }
```

- De ID

```
#unique {}
```

- De atributo

```
a[title] {}
```

COMBINADORES

Tipos de Operadores:

- Operadores de combinación descendentes

`article p`

- Operadores de combinación de elementos hijos

`article > p`

- Operadores de combinación de elementos hermanos adyacentes o hermanos directos

`h1 + p`

- Operadores de combinación de elementos hermanos generales

`h1 ~ p`

PSEUDOCCLASES

Una *pseudoclase* es un selector que marca los elementos que están en un estado específico, por ejemplo, los que son el primer elemento de su tipo, o aquellos por los que el cursor les pasa por encima. Tienden a actuar como si hubieras aplicado una clase en una parte determinada del documento y, a menudo, ayudan a reducir el exceso de clases y proporcionan un marcado más flexible y fácil de mantener.

Gustavo Adolfo García Blanco

Las *pseudoclases* son palabras clave que comienzan con dos puntos:

:pseudo-class-name

PSEUDOELEMENTOS

Los *pseudoelementos* se comportan de manera similar a las *pseudoclases*. Sin embargo, actúan como si hubieras añadido un elemento HTML totalmente nuevo en el marcado, en lugar de haber aplicado una clase nueva a los elementos presentes. Los *pseudoelementos* empiezan con un doble signo de dos puntos `::`.

`::pseudo-element-name`

ESPECIFICIDAD

La **especificidad** es la manera mediante la cual los navegadores deciden qué valores de una propiedad CSS son más relevantes para un elemento y, por lo tanto, serán aplicados. La especificidad está basada en las reglas de coincidencia que están compuestas por diferentes tipos de selectores CSS.

CALCULAR ESPECIFICIDAD

La *especificidad* es un peso (importancia o valor) que se le asigna a una declaración CSS dada, determinada por el número correspondiente de cada tipo de selector. Cuando varias declaraciones tienen igual especificidad, se aplicará al elemento la última declaración encontrada en el CSS. La especificidad solo se aplica cuando el mismo elemento es objetivo de múltiples declaraciones. Según las reglas de CSS, en caso de que un elemento sea objeto de una declaración directa, esta siempre tendrá preferencia sobre las reglas heredadas de su ancestro.

PROPIEDAD DISPLAY

La propiedad CSS **display** especifica si un elemento es tratado como un elemento *en bloque* o *en línea* y el diseño usado por sus hijos, como *flow layout* (Diseño de Flujo), *grid* (Cuadrícula) o *flex* (Flexible).

Gustavo Adolfo García Blanco

Formalmente la propiedad *display* establece los tipos de visualización interna y externa de un elemento. La tipo externa establece la participación de un elemento en *flow layout*; la tipo interna establece el layout (Diseño) de los hijos. Algunos valores de *display* están totalmente definidos con sus especificaciones propias; por ejemplo el detalle de que pasa cuando **display: flex** es declarado y definido en la especificación de Modelo Flexible de Caja (Flexible Box Model specification) de CSS.

PROPIEDAD DISPLAY

Además de los Diferentes Tipos de caja de Visualización, el valor de `none` permite desactivar la visualización de un elemento; cuando no se utiliza `none`, todos los elementos descendentes también quedan desactivados. El documento se procesa como si el elemento no existiera en el árbol de documentos.

PROPIEDAD POSITION

La propiedad **position** de CSS especifica cómo un elemento es posicionado en el documento. Las propiedades `top`, `right`, `bottom`, y `left` determinan la ubicación final de los elementos posicionados.

PROPIEDAD POSITION

Tipos de posicionamiento

Un **elemento posicionado** es un elemento cuyo valor computado de `position` es `relative`, `absolute`, `fixed`, o `sticky`. (En otras palabras, cualquiera excepto `static`).

Gustavo Adolfo Garcia Blanco

Un **elemento posicionado relativamente** es un elemento cuyo valor computado de `position` es `relative`. Las propiedades `top` y `bottom` especifican el desplazamiento vertical desde su posición original; las propiedades `left` y `right` especifican su desplazamiento horizontal.

PROPIEDAD POSITION

Un **elemento posicionado absolutamente** es un elemento cuyo valor computado de `position` es `absolute` o `fixed`. Las propiedades `top`, `right`, `bottom`, y `left` especifican el desplazamiento desde los bordes del bloque contenedor del elemento. (El bloque contenedor es el ancestro relativo al cual el elemento está posicionado). Si el elemento tiene márgenes, se agregarán al desplazamiento. el elemento establece un nuevo contexto de formato de bloque para su contenido

Gustavo Adolfo García Blanco

Un **elemento posicionado fijamente** es un elemento cuyo valor de `position` computado es `sticky`. Es tratado como un elemento posicionado relativamente hasta que su bloque contenedor cruza un límite establecido (como por ejemplo dando a `top` cualquier valor distinto de `auto`), dentro de su flujo principal (o el contenedor dentro del cual se mueve), desde el cual es tratado como "fijo" hasta que alcance el borde opuesto de su bloque contenedor.

BEM

Si tu propósito al momento de crear CSS es ser específico, claro, legible, fácil de entender y por supuesto implementar una **buena práctica al momento de crear estilos**, la metodología BEM (Block Element Modifier) te será de gran ayuda para crecer en el ámbito profesional.

Gustavo Adolfo García Blanco

El propósito de BEM que en español sería **modificador de bloques de elementos**, es que tú puedas crear una estructura de código más consistente, no duplicar estilos y traer claridad al código definiendo y estableciendo mejor jerarquía en tu proyecto. Si quieres saber más de cómo surgió esta metodología te recomiendo leer a fondo sobre su historia.

BEM

Lo anterior por supuesto nos ayuda a **trabajar más de manera ágil y eficiente**, tanto solo como en equipo, ya que como he dicho tendremos la oportunidad de crear un código totalmente legible hasta para los usuarios introvertidos que den clic derecho y en inspeccionar elemento.

Esta metodología surge para incluir en la creación de estilos CSS y HTML una serie de reglas para el desarrollo del frontend de manera general, la cual consiste en crear o nombrar clases de elementos independientes haciendo las **hojas de estilos más simples y fáciles de entender** divididas en bloque, elemento y modificador. Las cuales describiré cada una.

BEM

Bloque: Es esa parte independiente en la cual se escrutará nuestro proyecto y donde se verán contenidos todos los elementos de este. es decir, esos contenedores principales que nos ayudaran a dividir las partes de nuestro documento y que son fácilmente identificadas. Ejemplos de esto serian header, footer, contenedor principal, ítems y demás. En realidad, depende de ti y tu forma de estructurar resaltando que los bloque serán los contenedores de los elementos y así mismo deben ser nombrados de forma coherente.

BEM

Elemento: Es (valga la redundancia) el elemento que esta dentro de un bloque. En la metodología BEM no puede utilizarse por si solo si no se especifica a que bloque pertenece. Creando un marco de referencia general podemos resumir que el bloque contiene los elementos y así debe estructurarse el nombramiento de la clase. Estos se nombran con dos underscore después del nombre del bloque.

BEM

Modificador: Este juega un papel muy importante ya que nos referimos a modificador cuando queremos **cambiar la apariencia específica de un bloque o un elemento**, por ejemplo: si tenemos un menú en el header y estos tienen los mismos nombres de clase para que cada uno tome propiedades iguales, pero hay uno que quieres que sea de otro color y tome otras propiedades como focus, active o hover, pues **le agregas un modificador** para así de simple modificarlo a tu gusto junto con las propiedades de los demás elementos del menú. para crear un modificador agregamos dos guiones y su nombre.

WEB RESPONSIVE

Un diseño web responsive es el que es capaz de adaptarse a pantallas de diferentes tamaños con un solo sitio web. El sistema detecta automáticamente el ancho de la pantalla y a partir de ahí **adapta todos los elementos de la página**, desde el tamaño de letra hasta las imágenes y los menús, para ofrecer al usuario la mejor experiencia posible. ¡Parece magia!

Gustavo Adolfo Garcia Blanco

En ocasiones, se confunde el responsive con las **webs para móviles**, pero no se trata de lo mismo. En el caso del diseño responsive, tenemos un solo sitio web que puede adaptarse para dispositivos de todo tipo, desde ordenadores de escritorio con grandes monitores hasta móviles, pasando por tablets y otros. En cambio, crear un sitio móvil implica diseñar desde cero una web independiente, cuyos contenidos y formato están especialmente adaptados para funcionar mejor en dispositivos móviles.