Contenidos teóricos

1. ¿Qué es CSS?

¿Qué es CSS?

El es lenguaje de la web con el que podremos dar estilos a nuestras páginas, sitios y webapps. Sus siglas significan **Hojas de estilos en cascada** y vienen del inglés *“Cascading Style Sheets”*.

Actualmente se encuentra en su **versión 3** la cual introdujo severas mejoras frente a sus predecesores y además se está en constante actualización con implementaciones nuevas al lenguaje que nos proporcionan cada vez más herramientas y propiedades para trabajar nuestros estilos.

La idea detrás de este lenguaje es separar las responsabilidades en el desarrollo de un sitio web. Como vimos anteriormente, HTML posee etiquetas que brindan cierto estilo visual, sin embargo actualmente lo utilizamos únicamente para generar la estructura del contenido de nuestro sitio, mientras que CSS nos ocuparemos de posicionar y modificar visualmente esa estructura.

Sintaxis

La estructura CSS se basa en reglas que tienen el siguiente formato:



Cada sentencia posee 3 partes básicas : selector, propiedad y valor.

Además se encuentra definida entre un bloque de llaves **{ }** y cada declaración de propiedad finaliza con un **;** (punto y coma). En una misma sentencia pueden haber tantas declaraciones como estilos necesitemos aplicar a ese elemento seleccionado.

**Selector**

El selector es el elemento HTML que vamos a seleccionar del documento para aplicarle un estilo concreto. Por ejemplo, con p seleccionamos todas las etiquetas <p> de nuestro documento.

**Propiedad**

Es la propiedad CSS que queremos cambiar de ese elemento seleccionado, por ejemplo el color de fuente, el tamaño o su disposición.

**Valor**

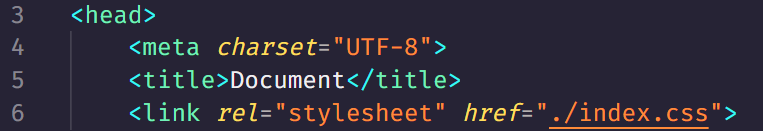
Es el valor que se le asigna a la propiedad y que dictará esa modificación.

Vinculación

Existen **3** métodos para aplicar o vincular estilos CSS a nuestro HTML: **interno**, **externo** y **en línea**.

**Externo**

En la etiqueta **<head></head>** de nuestro documento, podemos incluir una etiqueta **<link />** que vincule nuestro archivo HTML con un archivo externo de extensión **.css**.

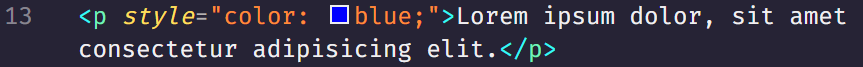


**Interno**

También podemos añadir nuestros estilos directamente en el documento HTML, a través de una etiqueta **<style></style>** también dentro de la etiqueta **<head></head>**.

**En Línea**

Se colocan directamente en el atributo ***style*** de la etiqueta a modificar.



### 2. Herencia

Herencia

Es el concepto por el cual algunas propiedades CSS son heredadas desde las etiquetas padres hacia sus etiquetas hijas. 

En este ejemplo, todas las etiquetas hijas de body tendrán por defecto el color de texto **silver** salvo que la herencia se corte por otro elemento padre hacia su hijo:



o por la asignación de un estilo propio:



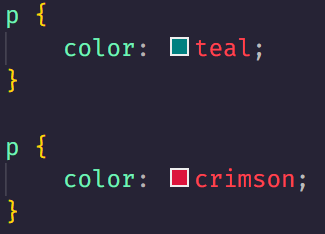
No todas las propiedades CSS son heredadas, porque algunas de ellas no tendría sentido que lo fueran. Por ejemplo, los márgenes no se heredan porque es poco probable que un elemento hijo necesite los mismos márgenes que su padre. Normalmente, el sentido común dicta qué propiedades se heredan y cuáles no.

Algunas de ellas son: **color**, **font-weight**, **font-family**, **float**, **line-height**, entre otras.

### 3. Cascada

Cascada

Es el mecanismo que controla el resultado final cuando se aplican varios estilos o propiedades CSS iguales al mismo elemento.



Como bien lo indica su nombre, cuando nos encontramos con estos casos, siempre prevalece el último en orden descendente. Como en la imagen, nuestros párrafos tendrán color **crimson**.

Sin embargo, esto no es lo único que se debe tener en cuenta a la hora de resolver qué sucede con estilos que entran en conflicto.

Hay tres conceptos principales que controlan el orden en el que se aplican las declaraciones de CSS:

**Precedencia**.

**Especificidad**.

**Orden en que se importan los recursos o estilos**.

### 4. Selectores

Selectores

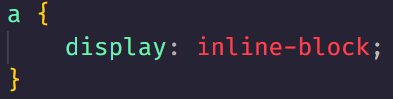
Cuando comenzamos a trabajar con CSS, es común aplicar estilos con selectores genéricos como una etiqueta HTML, por ejemplo un **div { }**. Sin embargo, lo que estamos haciendo en realidad es seleccionar todos los elementos del documento que sean dicha etiqueta, lo cual no siempre es lo que buscamos.

Por eso contamos con selectores adicionales que nos permiten afinar nuestra puntería con el objetivo de aplicar estilos a un elemento o un grupo determinado aunque estos no tengan relación aparente o sean de distintas etiquetas.

En este sentido, los selectores son principalmente: **etiquetas**, **clases** o **IDs**.

##### ****Etiquetas****

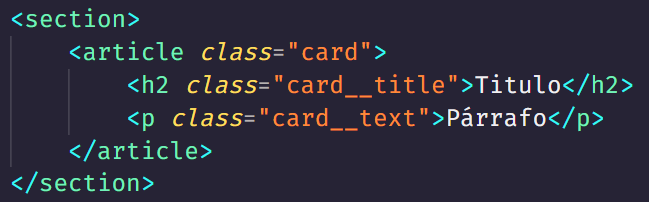
omo vimos en ejemplos anteriores, es a través del nombre de la etiqueta que deseamos seleccionar.



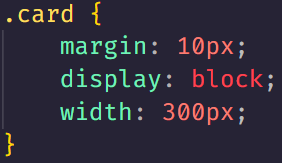
##### ****Clases****

A medida que vamos aplicando estilos y creando nuestros documentos HTML, comprobaremos que necesitamos cosas más flexibles y cómodas, aquí es donde entran en juego las clases de CSS.

Estas se colocan en la etiqueta mediante el atributo **class** y puede ser reutilizada tantas veces como sea necesario.



Al momento de utilizarlas en el CSS, se debe anteponer un punto (**.**) al nombre de la clase para diferenciarlo de un nombre de etiqueta.

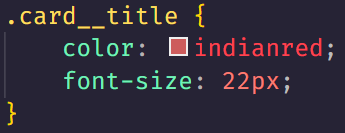


Con las clases podemos aplicar **2** conceptos diferentes. Por un lado, crear estilos **utilitarios** que apliquen un diseño específico independiente del elemento o el lugar que ocupe en la página. Un ejemplo de esto podría ser:



En este caso, cualquier elemento de **botón** en el sitio tendrá el mismo estilo por defecto y unificado.

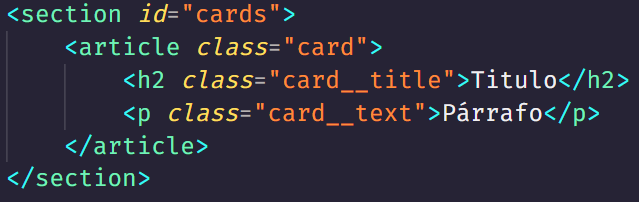
En cambio cuando creamos estructuras HTML complejas como una sección con productos de una tienda, las clases tienden a utilizarse para identificar cada elemento y su rol en ese grupo o sección.



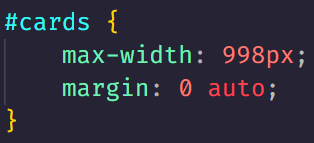
##### ****IDs****

Todas las etiquetas HTML pueden tener un atributo id con un valor concreto, pero este **NO** podrá repetirse ni ser reutilizado en otras etiquetas en el mismo documento.

Normalmente se utilizan para identificar secciones del sitio que luego podrán ser accedidas mediante enlaces.



Para invocarlas desde nuestro CSS se utiliza el símbolo de numeral (**#**) con el mismo propósito que las clases.



### Precedencia

Cuando tenemos varias reglas CSS en cascada que afectan a un mismo elemento, el orden de prioridad que determina cómo se mostrará el elemento es el siguiente:



En caso que necesitemos cortar con la **precedencia**, podemos utilizar la palabra **!important** entre el valor de la propiedad CSS y el punto y coma (**;**) final de esa declaración.



### Especificidad

La especificidad en CSS es un grupo de reglas aplicadas a los selectores CSS para determinar qué estilo se aplica a un elemento. Cuanto más específico sea un selector, mayor será su valor en puntos y más probable será que esté presente en el estilo del elemento.

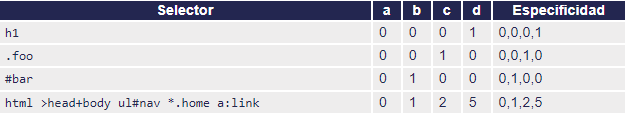
La especificidad tiene cuatro componentes; por ejemplo a, b, c y d. El componente "a" es el más distintivo y el "d", el que menos.

El componente **"a"** es bastante sencillo: es 1 para una declaración en un atributo **style**; si no, es 0.

El componente **"b"** es el número de selectores de **id** en el selector (los que empiezan con #).

El componente **"c"** es el número de selectores de atributo, incluidos los selectores de **clase** y pseudoclases.

El componente **"d"** es el tipo de etiqueta y pseudo elementos del selector.



Dependiendo de la suma de selectores y la jerarquía de cada uno es que se determina si un selector de un elemento es más o menos específico que otro.