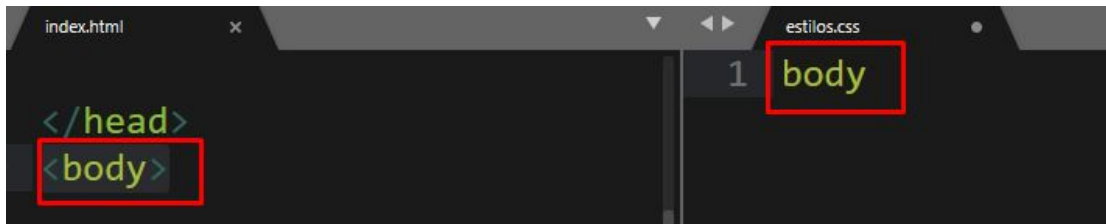


Selectores

Elemento

El selector de elemento se compone a partir del propio elemento de HTML al cual afecta. Por esa razón en nuestro HTML no debemos hacer más que crear el propio elemento, veamos un ejemplo:



ID

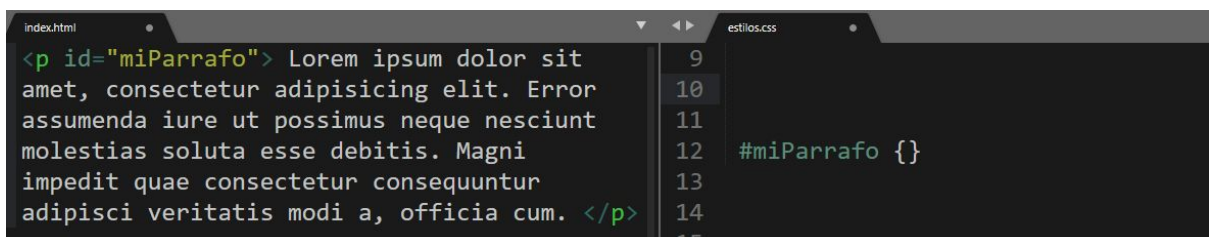
Este tipo de selector permite modificar un elemento de manera puntual. En realidad en nuestro CSS lo veremos de la siguiente forma, ejemplo para copiar y pegar en el HTML:

Este tipo de selector se compone en nuestra regla con un # que precede el nombre de id. El nombre de ID debe estar formado por letras o números, - o _ (guión de abajo underscore o guión del medio) , no puede contener espacios y tampoco números al comienzo (esto es una regla no colocar nunca números al principio, pues así sino no funcionará en algunos navegadores).

Luego es importante informar a qué elemento modificaré con este ID, pues a diferencia del selector de elemento o etiqueta anteriormente visto, no sabemos a qué afectamos con este ID salvo que lo informemos.

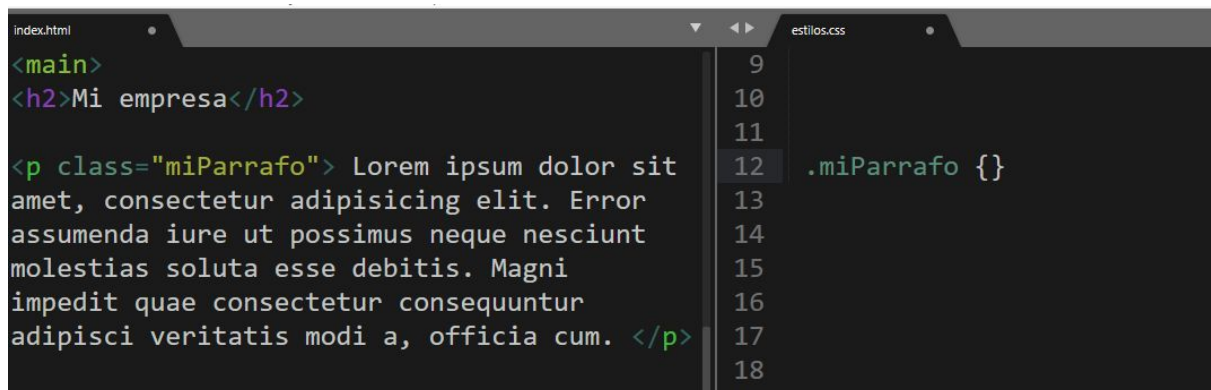
El ID tiene una característica peculiar que es básicamente que sólo puede utilizar un mismo nombre de ID una sola vez por HTML.

Por tanto si hacemos caso del ejemplo anterior el ID párrafo en mi index.html será utilizado una sola vez. Si de pronto deseo utilizarlo en otro archivo.html por ejemplo contacto.html puedo hacerlo y no necesariamente debo aplicarlo a un párrafo puedo aplicarlo al elemento que desee pero siempre utilizándolo una vez por HTML. Veamos un ejemplo:



CLASS

Los dos selectores anteriores nos han generado mucho material para trabajar pero qué pasa si de pronto tengo no uno ni todos los elementos a modificar, sino solamente varios, que haré?. En este caso surge un selector que puede utilizarse la cantidad de veces que desee en un mismo HTML. De hecho es el selector que más se utilizará por su característica de ser de esta manera, pero esto no significa que no usaremos ID o SELECTORES DE ELEMENTO, estos se utilizan en menor medida pero siempre están presentes en nuestra hoja de estilo.

A screenshot of a code editor with two tabs: 'index.html' and 'estilos.css'. The 'index.html' tab is active, showing HTML code. The 'estilos.css' tab is also visible, showing CSS code. The HTML code includes a main tag, an h2 tag with the text 'Mi empresa', and a paragraph tag with a class attribute 'miParrafo' and a long Lorem ipsum text. The CSS code shows a selector '.miParrafo' followed by an empty block '{ }'.

```
<main>
<h2>Mi empresa</h2>

<p class="miParrafo"> Lorem ipsum dolor sit
amet, consectetur adipisicing elit. Error
assumenda iure ut possimus neque nesciunt
molestias soluta esse debitis. Magni
impedit quae consectetur consequuntur
adipisci veritatis modi a, officia cum. </p>

9
10
11
12 .miParrafo {}
13
14
15
16
17
18
```

COMPUESTOS

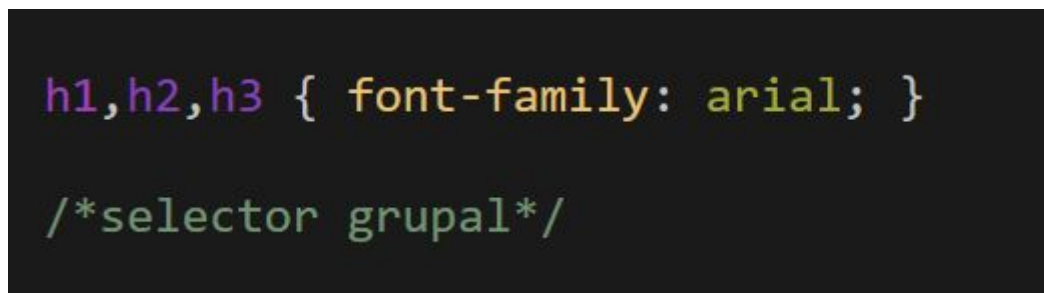
Qué pasaría si quisiera afectar a varios elementos o afecta a elemento según su posición su lugar , etc.

Bueno tenemos otros selectores, técnicamente son un producto de los selectores anteriores, eso significa que se generan a partir de la unión de dos o más de selectores de elemento , ID o CLASS.

La realidad es que tenemos varios de estos selectores y los iremos viendo a partir de que vayamos avanzado con propiedades y valores. Por ejemplo:

Grupal

El selector grupal, es un selector compuesto qué afecta a tantos selectores están en el grupo

A screenshot of a code editor showing CSS code. The code defines a group selector 'h1,h2,h3' with the property 'font-family' set to 'arial'. Below the code, there is a comment '/*selector grupal*/'.

```
h1,h2,h3 { font-family: arial; }

/*selector grupal*/
```

Por otro lado, puede ser conformado a partir de selectores de etiqueta, de clase o de id, e incluso pueden combinarse por caso:

```
h1,.miParrafo, #miCaja { font-family: arial; }  
  
/*selector grupal*/
```

Descendente

Afecta a ciertos elementos por encontrarse dentro de otros. Este tipo de selector nos ayuda a tener varios elementos iguales pero que sean afectados por su diferentes entornos o formas de estar anidados de manera distinta por ejemplo no es lo mismo decir:

```
p { color: blue; }
```

Qué decir:

```
#miCaja p { color: blue; }
```

En el primer caso afecto a todos los párrafos de forma genérica, en el segundo sólo aquellos que se encuentren dentro de un elemento cuyo id es caja, por ejemplo, el primer párrafo se verá afectado el segundo no, ya que el segundo si sostenemos la última regla de estilo, no se encuentra dentro de #caja

```
<div id="caja">  
<p> Lorem ipsum dolor sit amet, consectetur adipisicing elit. Error assumenda iure ut  
possimus neque nesciunt molestias soluta esse debitis. Magni impedit quae consectetur  
consequuntur adipisci veritatis modi a, officia cum. </p>  
</div>  
  
<div class="articulo">  
  <p> Lorem ipsum dolor sit amet, consectetur adipisicing elit. Natus illo consectetur  
  aut, asperiores repellendus unde, atque! Blanditiis est modi alias quis, veniam sit  
  iste maxime possimus beatae aut recusandae dignissimos!</p>  
</div>
```