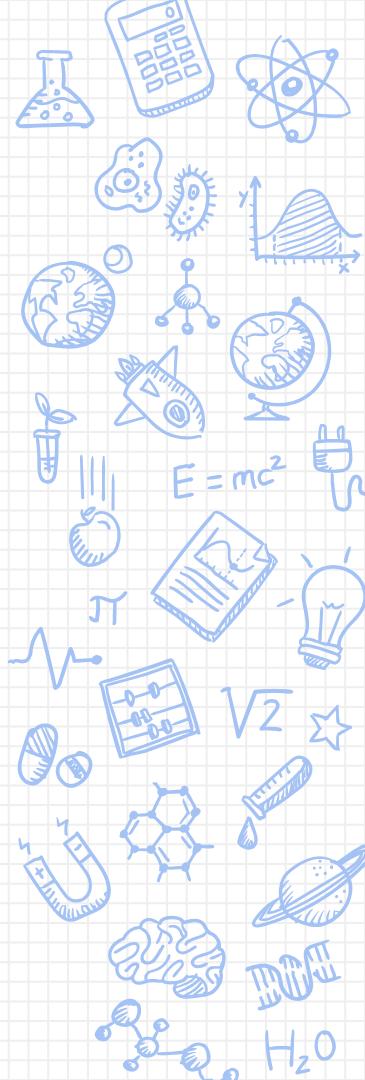


Clase 05

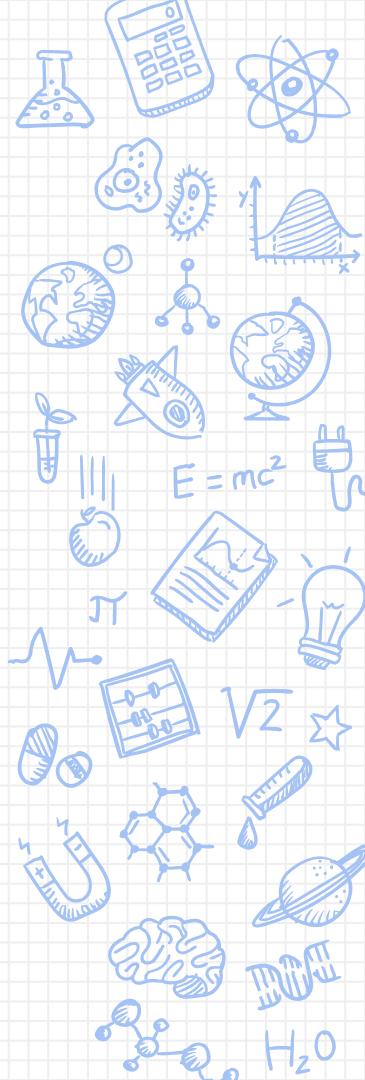
En la clase de hoy:

- CSS - Parte 1



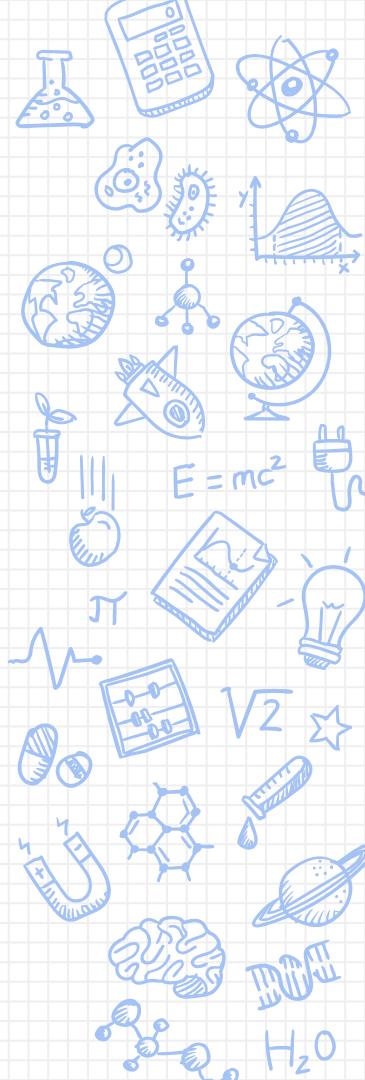
CSS

- CSS (Cascading Style Sheets), es el lenguaje que nos va a permitir cambiar el aspecto de nuestro sitio web.
- Si lo implementamos de manera correcta podemos hacer que dos estructuras HTML idénticas se vean completamente distintas.



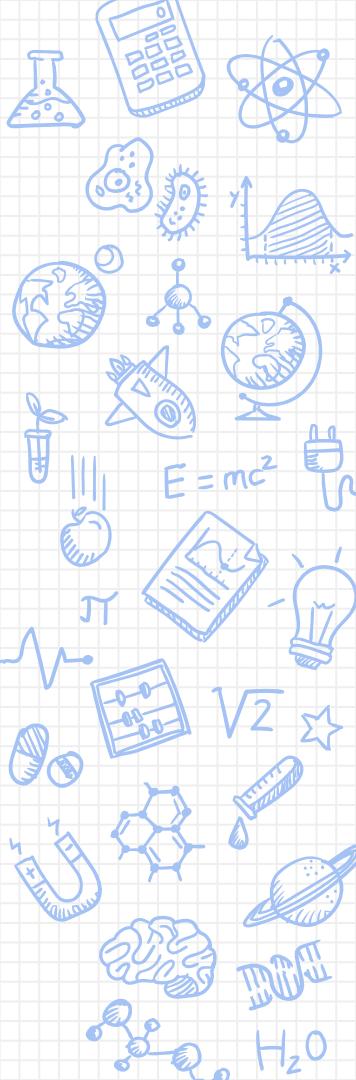
CSS

- Dentro de las posibilidades que nos brinda CSS, podemos identificar tres grupos:
 - Formato de texto.
 - Formato de cajas.
 - Ubicación de elementos.



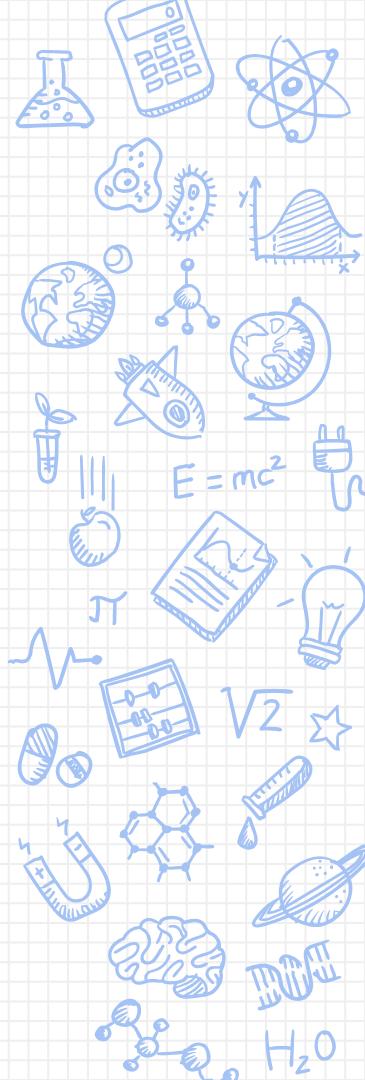
CSS

- Pero antes, debemos insertar CSS en nuestro HTML, para eso, existen tres formas:
 - CSS Inline **X**
 - Etiqueta style **X**
 - Archivo externo **=D**



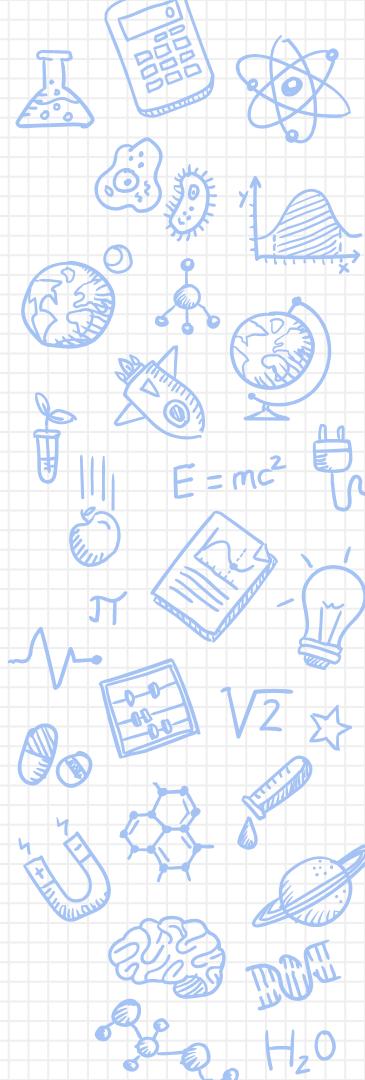
CSS Inline

- El CSS inline, consiste en agregar un atributo **style** a cada etiqueta y escribir las propiedades que quiero modificar dentro de cada una.
- Esto es absolutamente imposible de mantener, y está completamente **PROHIBIDO**.
- El único lugar en donde se sigue utilizando es en el infierno, y en mailing, que los clientes de correo electrónico tiene serios problemas de soporte.



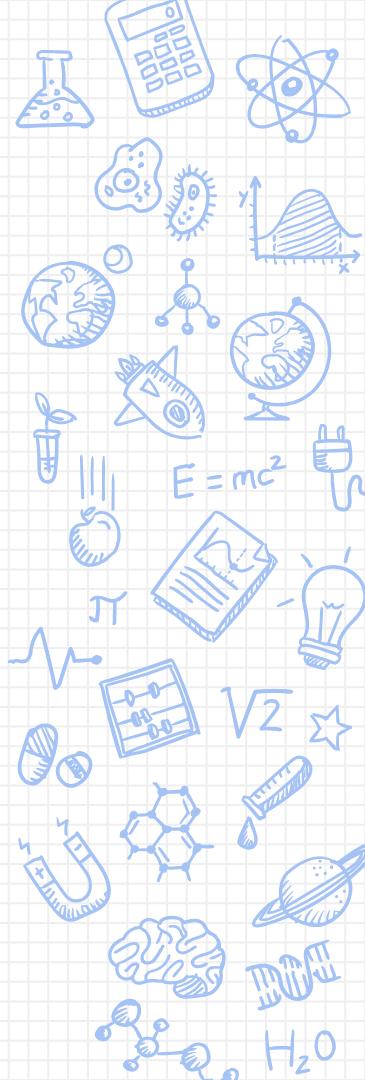
CSS – Etiqueta style

- La incorporación de CSS por medio de una etiqueta style, consiste en generar esta etiqueta dentro del head. Y dentro de ella escribir el código CSS.
- Si bien no es tan grave como el inline, sigue siendo sumamente costoso de mantener, por lo cual está **PROHIBIDA**.



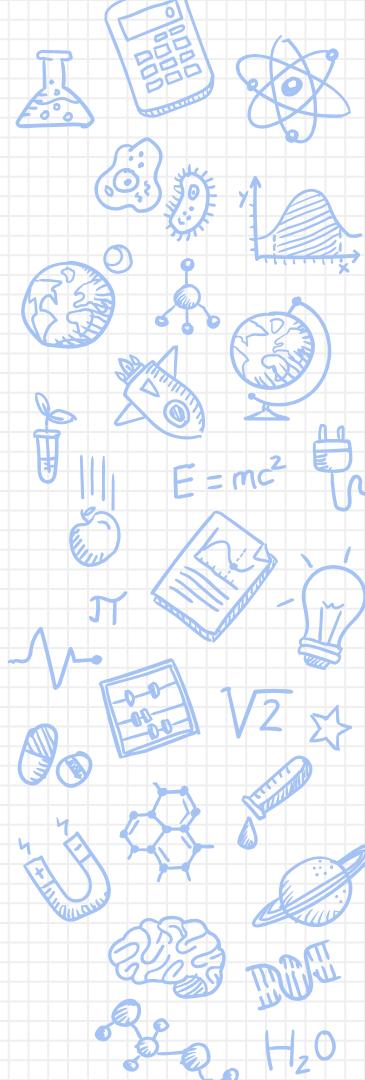
CSS – Archivo externo =D

- El archivo externo es la mejor opción que tenemos hoy en día.
- Se trata de un archivo con extensión CSS que vamos a tener ubicado en alguna carpeta de nuestro proyecto y va a aplicar sus estilos a todos nuestros archivos de HTML que se lo pidan.
- Escribimos el código una sola vez, y lo utilizamos en todos lados. **Óptimo.**



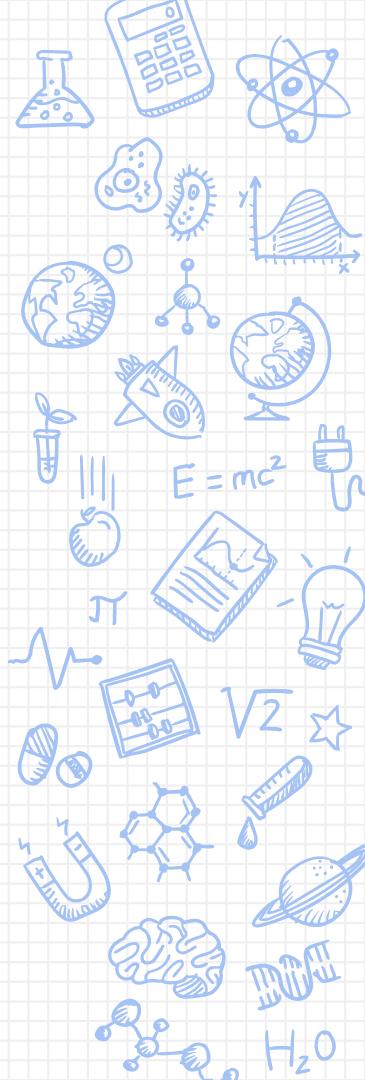
CSS – Archivo externo =D

- Para vincular al HTML con un archivo externo de CSS, vamos a hacer uno de la etiqueta **link**.
- Esta etiqueta va dentro del head, y recibe dos atributos obligatorios:
 - **rel**: siempre va a ser “stylesheet”.
 - **href**: la ruta donde debe ir a buscar el archivo de CSS.



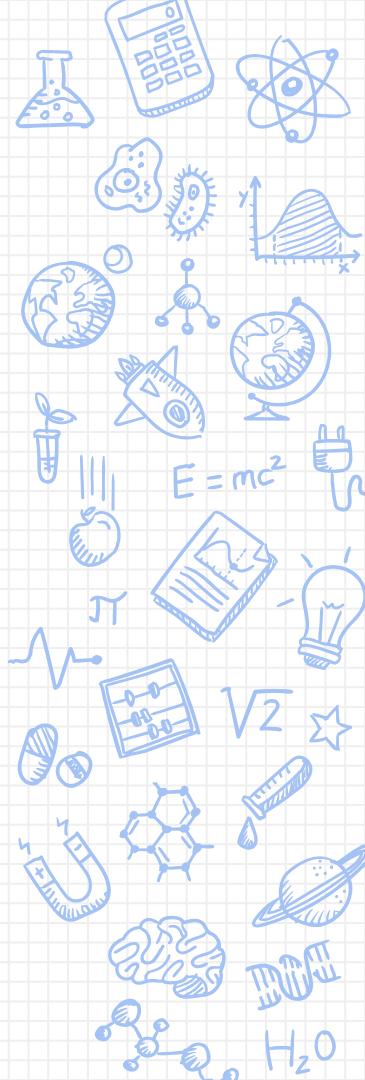
CSS

- Una vez hecho esto, vamos a conocer cómo trabaja CSS.
- En principio, vamos a tener tres formas de afectar a un elemento:
 - Nombre de etiqueta.
 - Id.
 - Clase.



CSS – Nombre de etiqueta

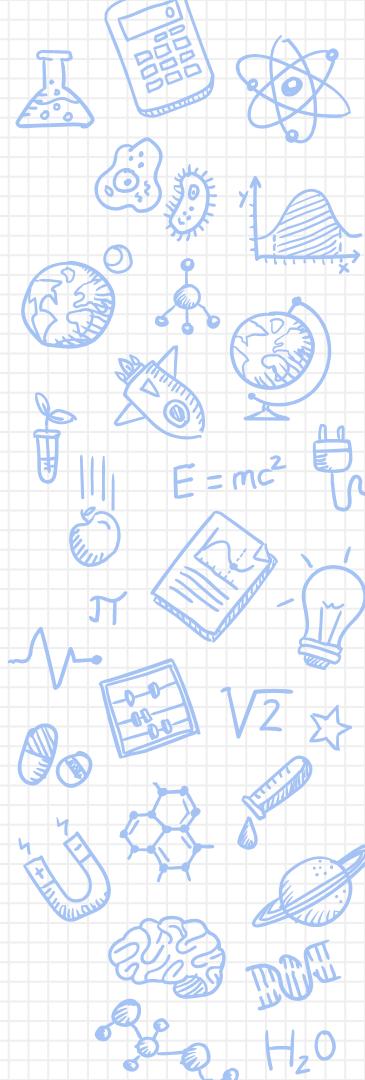
- Para afectar a un elemento por nombre de etiqueta hacemos lo siguiente:
 - ```
p {
 // código CSS
}
```
- Como verán, indicamos el nombre de la etiqueta que queremos modificar, y luego, entre llaves, escribimos el código CSS.



# CSS - Id

---

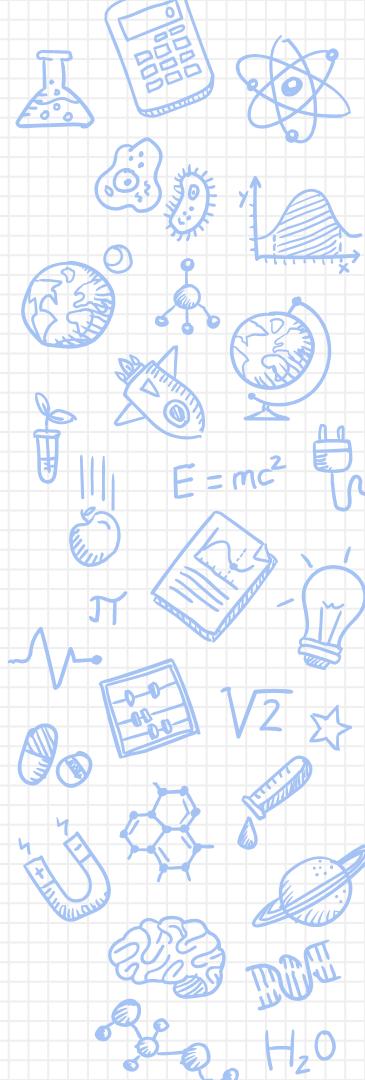
- El id es un atributo que puede recibir una etiqueta HTML.
- Cómo vimos con las anclas, el valor de un id no puede estar repetido en el archivo de html.
- Para llamar a un elemento por id desde css hacemos:
  - **#id\_elemento {  
                  // Código CSS  
}**



# CSS – Clase

---

- Las clases son similares a los ids, sirven para identificar elementos.
- La diferencia es que se pueden repetir las veces que queramos, y un elemento puede tener más de una clase.
- Para llamar a un elemento por su clase desde CSS escribimos:
  - **.clase\_elemento {**
  - // Código CSS
  - }



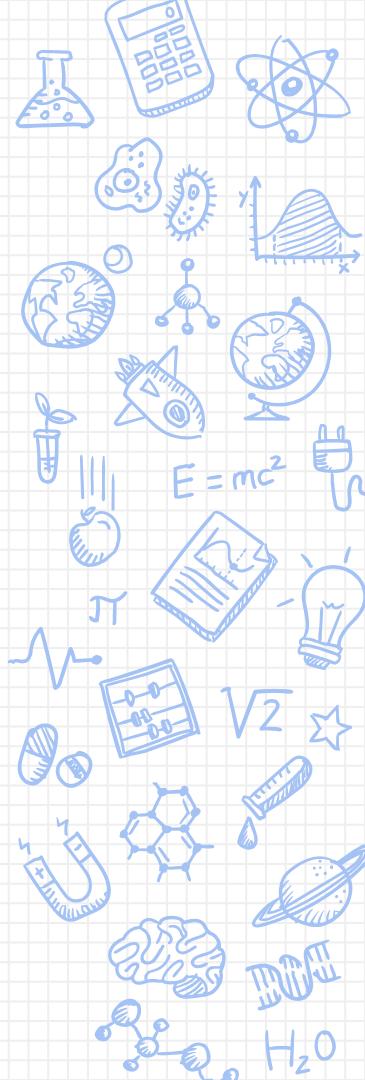
# CSS – Sintaxis

---

- Todas las reglas de CSS van a estar formadas por el elemento al que quiero afectar. Luego, el grupo de llaves, y dentro de ellas, las propiedades y los valores. Por ejemplo:

```
- p {
 color: red;
 background: green;
}
```

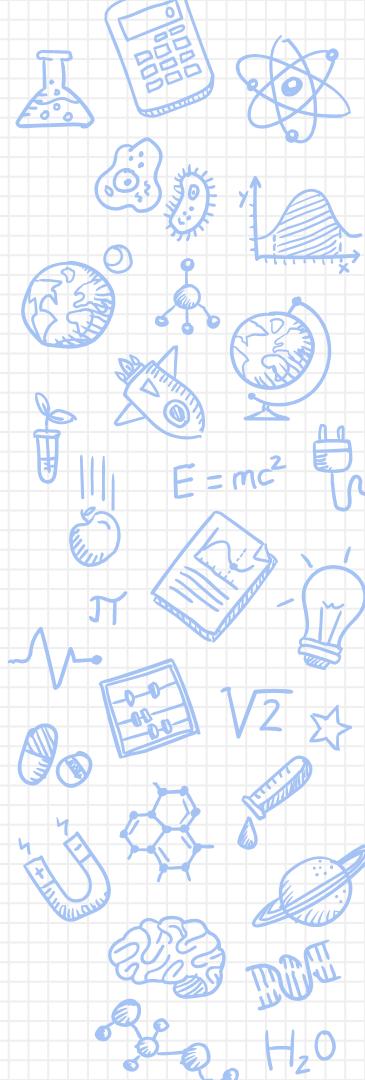
- Si necesitamos aplicar lo mismo a dos elementos, los sepáramos con una coma.



# CSS – Unidades de medida

---

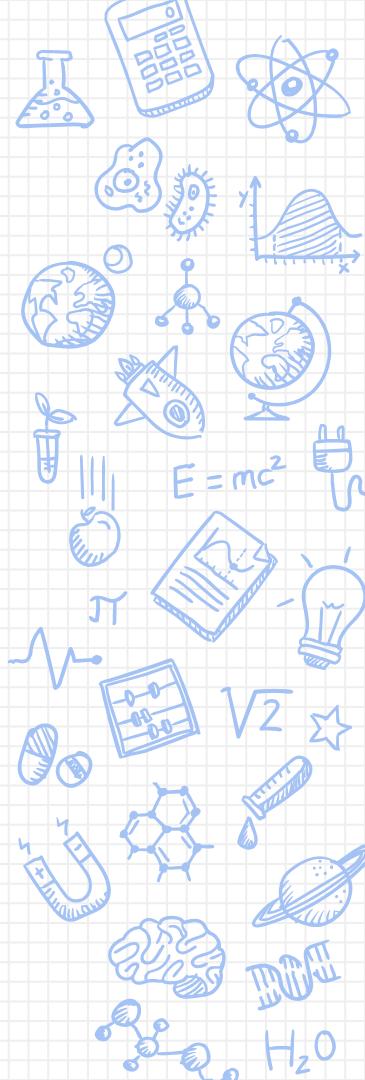
- Para cambiar el tamaño a un elemento, tenemos diversas unidades de medidas. Estas unidades se dividen en dos partes:
  - **Unidades absolutas:** son las que tienen un tamaño fijo independientemente del dispositivo.
  - **Unidades relativas:** se calculan según el tamaño del elemento contenedor.



# CSS – Unidades de medida

---

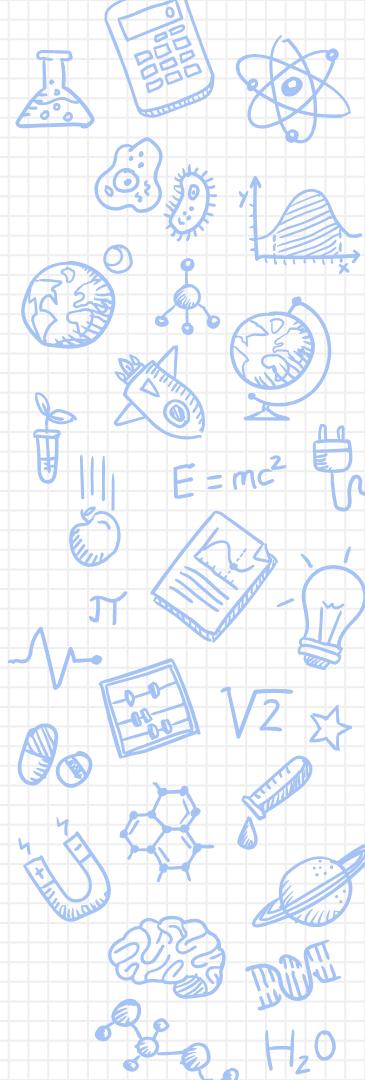
- Entre las unidades absolutas encontramos al cm, mm, in o pt. No suelen usarse.
- Del lado de las relativas tenemos al px, %, em y rem.
- El em es una medida que viene del diseño gráfico, y calcula el tamaño en base a la letra M que es la más ancha del abecedario.



# CSS – Unidades de color

---

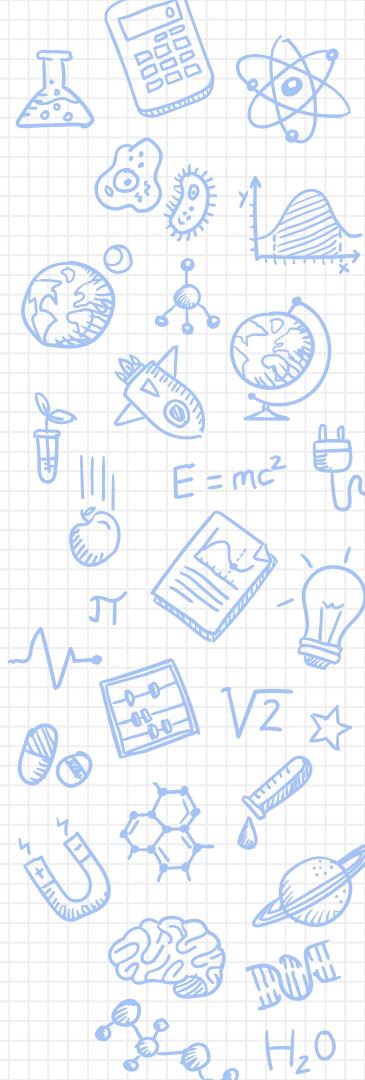
- En cuanto a los colores, en web tenemos cuatro formas de utilizarlos:
  - Palabra clave.
  - RGB.
  - Hexadecimal.
  - RGBA.



# CSS – Palabra clave

---

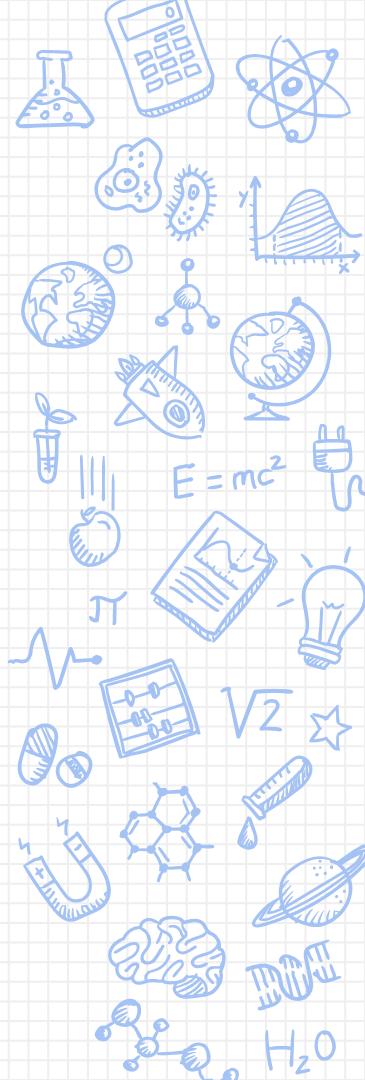
- El uso de colores por palabra clave es súper simple, utilizamos el color por su nombre en inglés → black, white, purple, pink, orange, blue, red, etc.



# CSS – RGB

---

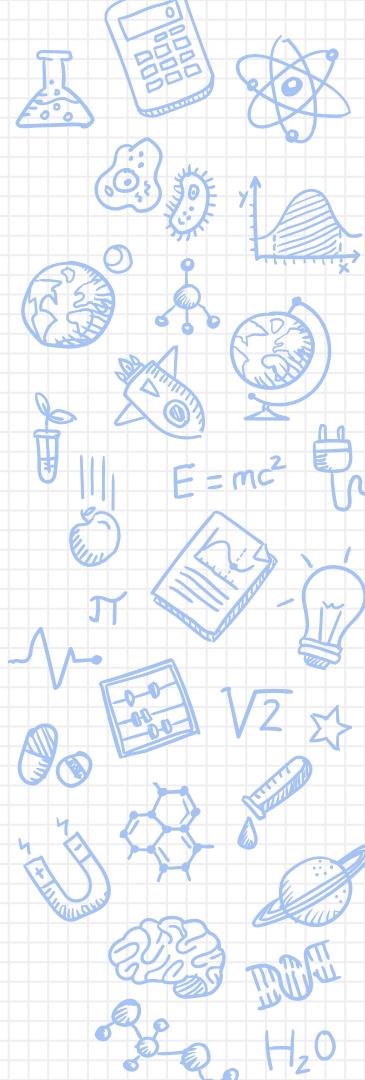
- En el caso de RGB, es una sigla que representa Red-Green-Blue.
- Para formar el color, le vamos a decir cuánto de cada uno queremos. Para eso, vamos a utilizar números del 0 al 255, donde 0 es totalmente negro y 255 totalmente blanco.
- Por ejemplo, escribiríamos:
  - `p {`
  - `color: rgb(38, 240, 02);`
  - `}`



# CSS – Hexadecimal

---

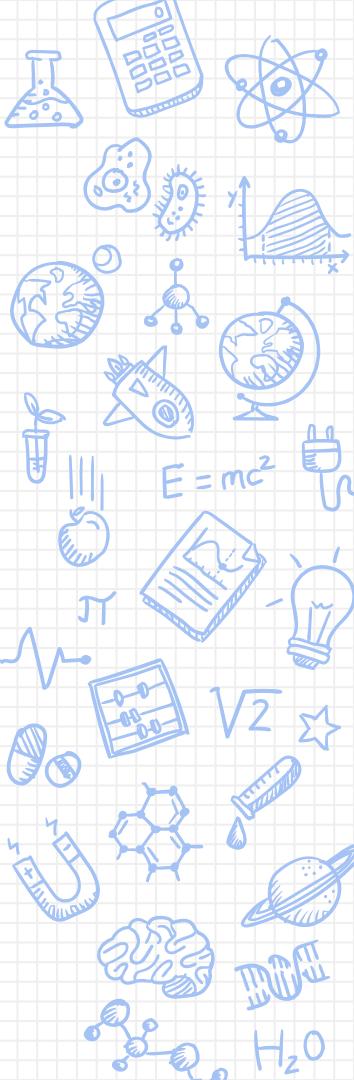
- Los hexadecimales son parecidos al RGB, de hecho, usa la misma lógica. Solamente que utiliza 6 dígitos, formados por tres pares de elementos. Y en lugar de ir de 0 a 255, va del 0 al 9 y de la A a la F.
- En este caso, el 0 sigue siendo completamente negro, y la F totalmente blanco.
- Por ejemplo → color: **#FF0000**



# CSS – Cascada

---

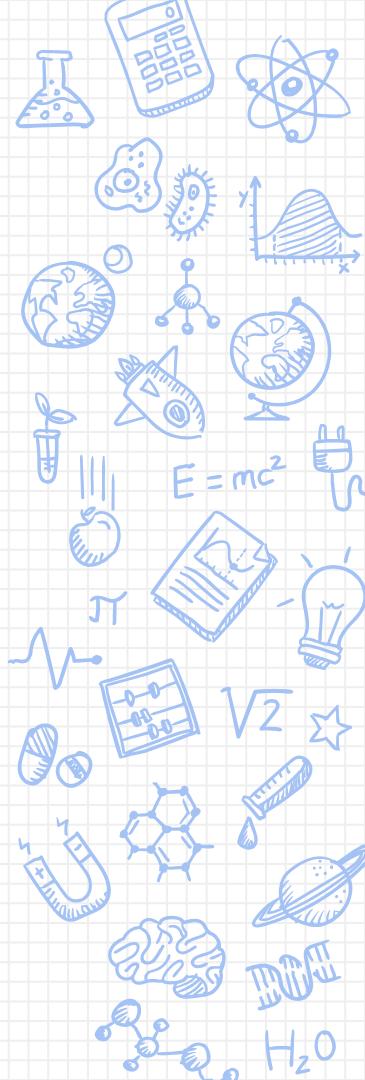
- El término cascada en el nombre de CSS, hace referencia a la estructura de HTML.
- Esto se utiliza para hacer reglas más específicas y no afectar a todos los elementos de mi sitio con una regla de etiqueta.
- Por ejemplo: Nosotros quedamos en que la estructura de un menú siempre iba a ser el nav, dentro una ul con sus li que dentro tendrían un a con el texto a clickear.



# CSS – Cascada

---

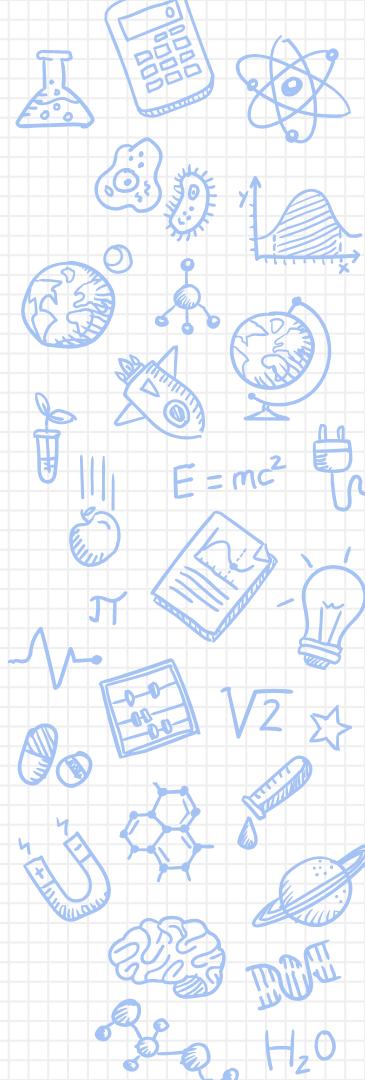
- Si en CSS hacemos una regla para los **a**, pero solo queríamos afectar a los que estaban dentro del menú, necesitamos cascada.
- Y eso se escribiría de la siguiente manera:
  - **nav ul li a {**  
          // Código CSS  
        }
- Con este código, va a afectar a los a que estén dentro de un li que esté dentro de una ul que esté dentro del nav.



# CSS – Precedencia

---

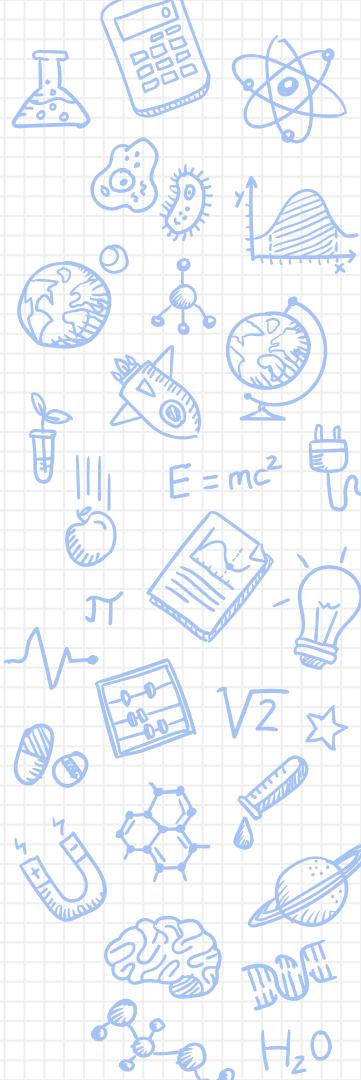
- La precedencia es un concepto que tenemos que tener en cuenta cuando escribimos CSS.
- Primero, vamos a decir que lo último que escribimos es lo que prevalece. Es decir, si hacemos dos reglas iguales para un párrafo, pero con dos colores distintos, se aplicará el que esté más abajo.



# CSS – Precedencia

---

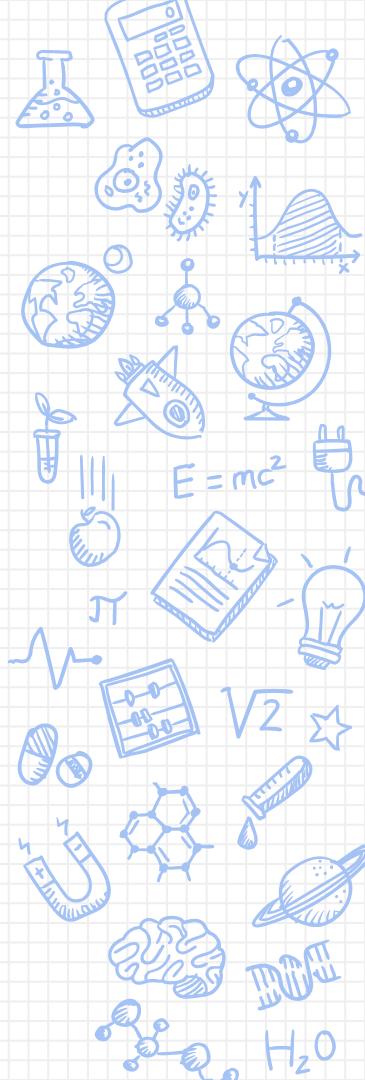
- Ese es el caso más simple, pero cuando trabajamos con etiquetas, clases y IDs se complejiza.
- Las reglas por etiqueta son las que tienen menos precedencia.
- Si utilizamos una regla de clase, aunque la de etiqueta esté por debajo, se aplicará ésta.
- Si utilizamos un ID, la precedencia es mucho mayor que la de una etiqueta o una clase.



# CSS – !important

---

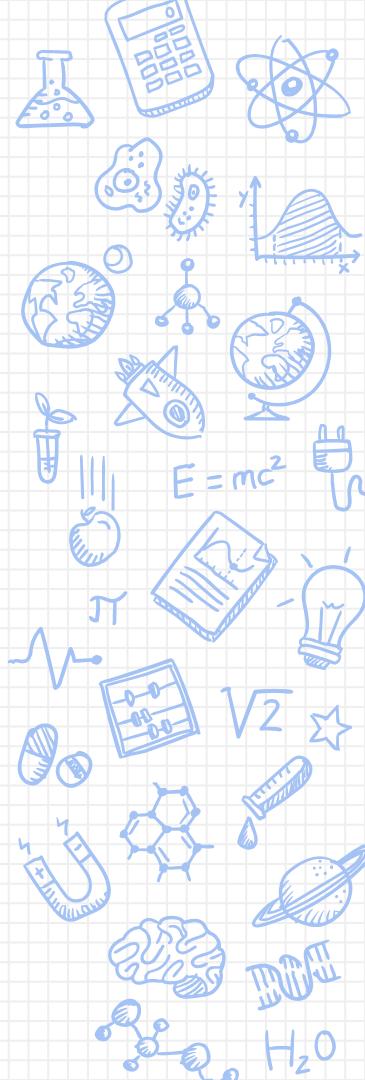
- Hay un pequeño hack en caso de que nos perdamos un poco con la precedencia.
- El uso de **!important**, anula toda precedencia y le da todo el protagonismo a esa regla.
- Por supuesto, el !important está **PROHIBIDO**.
- Si necesitás más de 5 important en tu CSS seguramente estés haciendo algo mal.



# CSS – Herencia

---

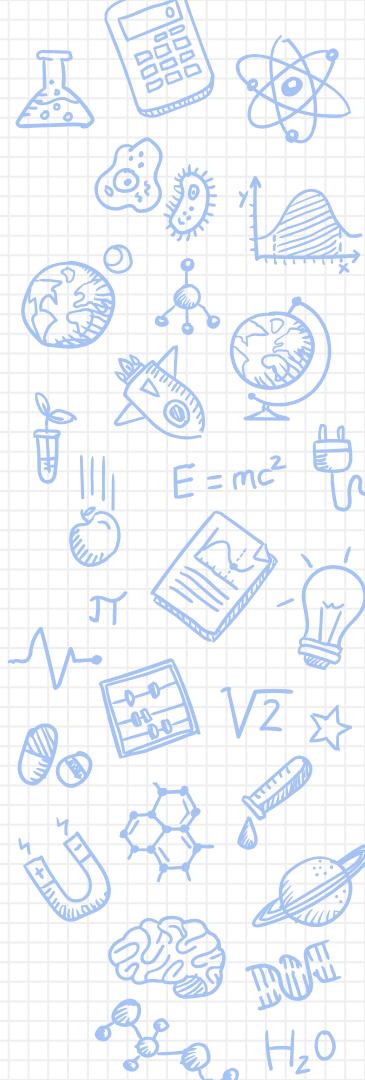
- Todas las propiedades de texto tienen algo llamado herencia.
- Esto quiere decir que si le aplicamos un color de texto a la etiqueta body, todo lo que haya adentro va a ser de ese color.
- El único elemento que escapa a esta regla es la etiqueta `<a></a>`



# CSS – Estilos

---

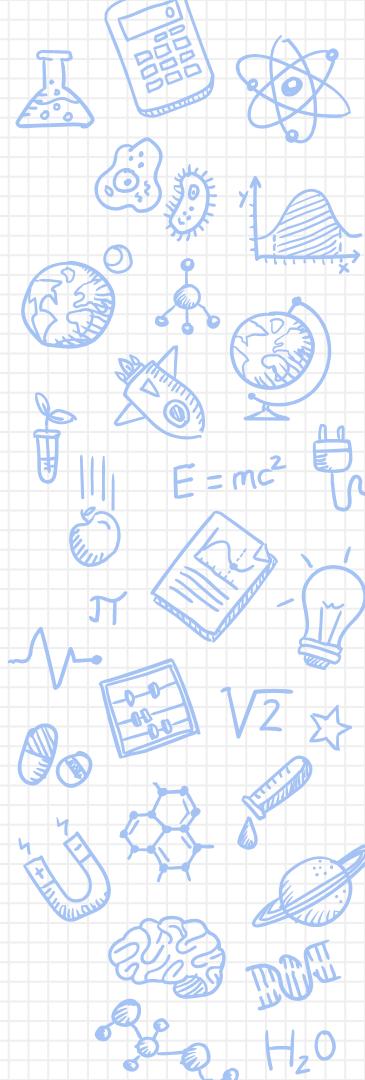
- Con la propiedad **font-style** podemos indicarle a un texto que esté en itálica.
- Con la propiedad **font-weight** podemos pasar un texto a negrita.
- Para cambiar la tipografía de un elemento, en principio, vamos a usar las fuentes del sistema.
- Esto se hace por medio de la propiedad **font-family** y recibe el nombre de la tipografía. Por ejemplo → font-family: Arial;



# CSS – Tipografía

---

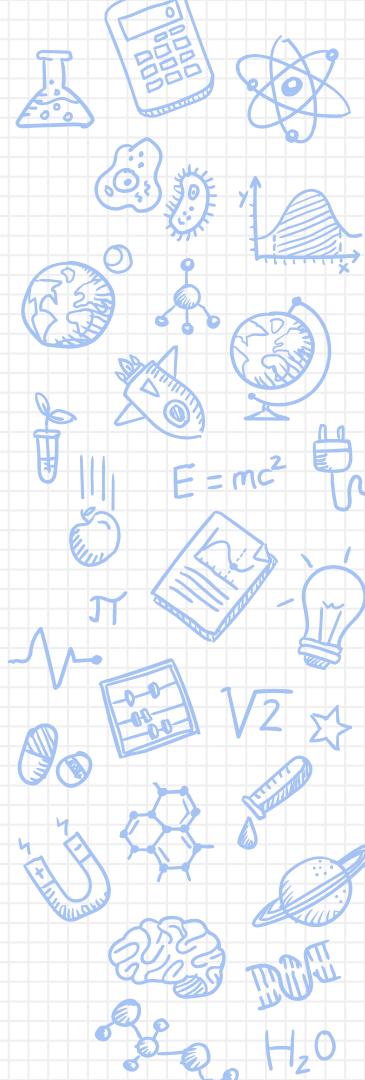
- Sin embargo, para tipografías es recomendable utilizar familias, y no una única fuente.
- Por ejemplo, Arial, lo completaríamos con Helvetica y sans-serif. Arial para Windows, Helvética para Mac y sans-serif es una clase de tipografía por defecto.



# CSS – Tipografía

---

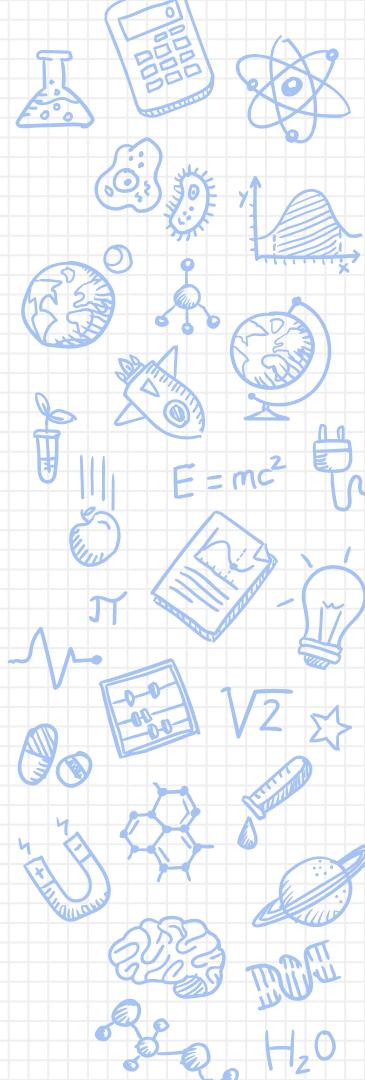
- Si queremos utilizar otro tipo de fuentes, vamos a indagar un poco en Google Fonts:
  - <https://fonts.google.com/>



# CSS – Tamaño

---

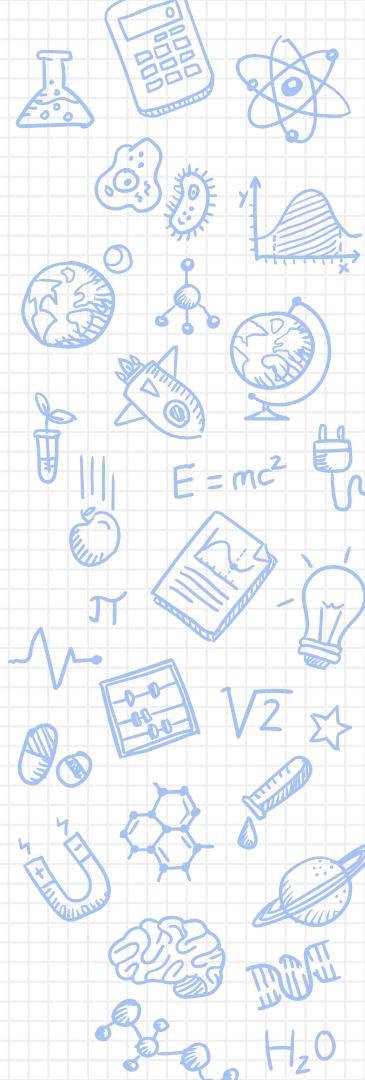
- Para modificar el tamaño de una fuente, contamos con la propiedad **font-size**.
- Esta propiedad puede recibir un valor en px, %, em o rem.
- Por lo general se utiliza px, em o rem.



# CSS – Espaciados

---

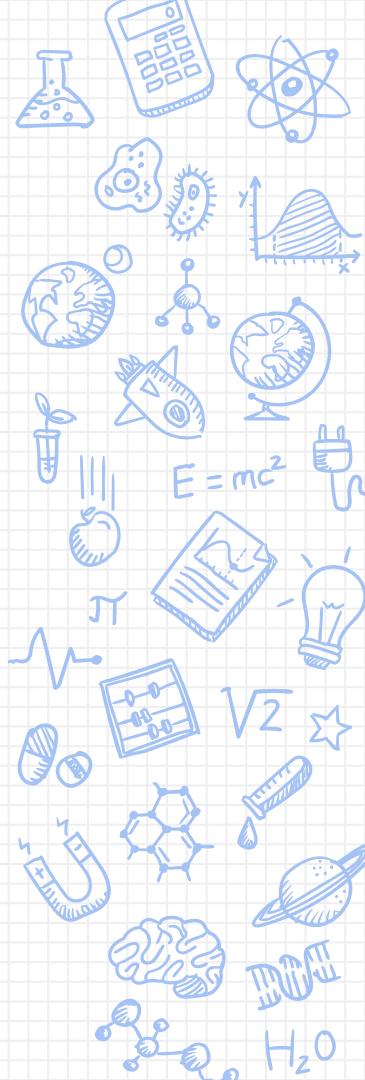
- Con las propiedades **letter-spacing** y **word-spacing**, que reciben un valor numérico ambas, podemos cambiar el espacio entre letras y entre palabras respectivamente.
- En cuanto a los espacios entre renglones, existe la propiedad **line-height**, que también puede recibir un valor en px, pero lo más común es pasarle un número mayor o igual a 110%.



# CSS – Alineación

---

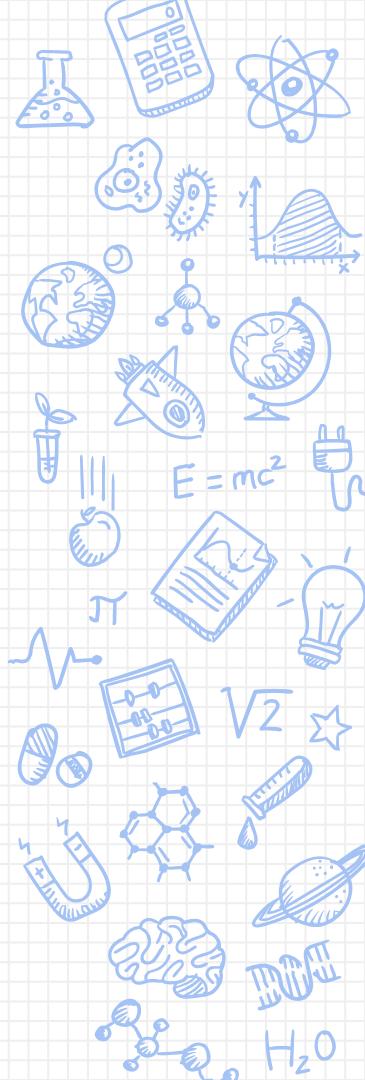
- Una de las cosas más importantes para trabajar con texto es su alineación. CSS nos da cuatro posibilidades por medio de la propiedad **text-align**:
  - Left.
  - Right.
  - Center.
  - Justify **X**



# CSS – Transformación

---

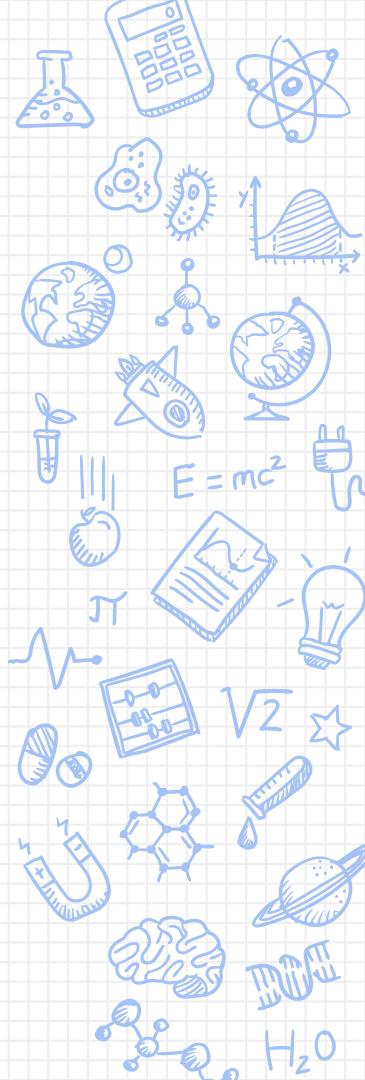
- Cómo dijimos la clase pasada, vamos a poder pasar un texto a mayúscula o minúscula. Eso se hace con la propiedad **text-transform**, que recibe uno de los siguientes valores:
  - Uppercase.
  - Lowercase.
  - Capitalize.



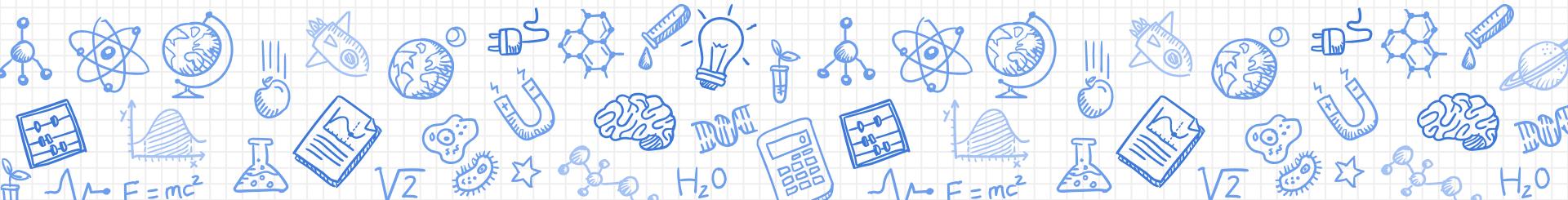
# CSS – Listas

---

- Para trabajar con listas, y más específicamente con sus viñetas. Tenemos la propiedad `list-style`.
- Los valores más comunes son **disc**, **circle** y **square**.
- Pero existen otros que nos permiten poner números romanos, o letras.
- Además, existe el valor **none**, que oculta las viñetas.



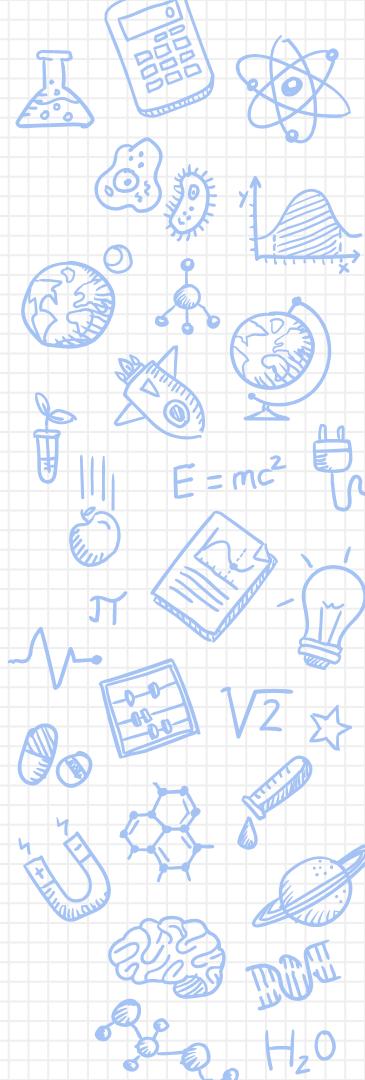
# íconos

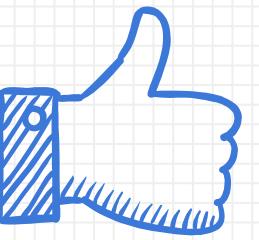


# CSS - Íconos

---

- Los íconos de nuestras webs, salvo que sean de diseñador, van a venir de alguno de los siguientes sitios:
  - <https://fontawesome.com/>
  - <https://www.flaticon.com/>





# SE ACABÓ!

## Dudas?

Escriban a:

✗ [federico.mirandaa@gmail.com](mailto:federico.mirandaa@gmail.com)