CSS

*Cascading Style* Sheets - Lenguaje para dar estilo a una web HTML.

En cascada hace referencia a que toman prioridad aquellos estilos que estén más abajo en el archivo CSS.

Hay dos formas que se utilizan actualmente para utilizar CSS en un documento HTML.

**Estilos en línea**

<etiqueta style = “atributo:valor”> </etiqueta>

**Importar un archivo CSS *(Recomendado)***

<link rel=”stylesheet” type=”text/css” href=”Ruta del archivo CSS”>

Al utilizar este método

**Sintaxis básica**

selector {

propiedad: valor;

}

Donde **selector** es el elemento al que queremos agregar el estilo.

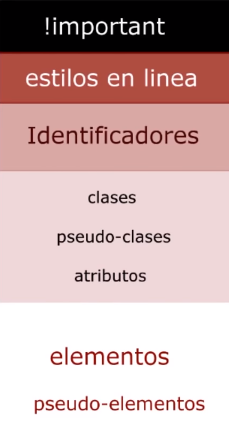
Y **Propiedad** es el atributo que se va a modificar dándole un nuevo **valor**.

**Selectores de CSS**

* Universal
  + En CSS utilizamos el selector \*
  + Aplica para todos los elementos HTML.
* De tipo
  + En CSS seleccionamos con Tipo\_de\_Elemento
  + Aplica para todos los elementos HTML de un mismo tipo.
* Clases
  + En HTML agregamos clases a los elementos con el atributo class = “Nombre\_de\_clase”
  + En CSS seleccionamos con .Nombre\_de\_Clase
  + Aplica para todos los elementos con una misma clase.
* ID
  + En HTML agregamos IDs a los elementos con el atributo id = “ID\_de\_elemento”
  + En CSS seleccionamos con #ID\_de\_elemento
  + Los IDs deben ser únicos por elemento.
* Por atributo
  + En HTML agregamos atributos con atributo = “valor”
  + En CSS seleccionamos con [atributo = “valor”]
* Descendiente
  + En HTML agregamos elementos dentro de otros elementos.
  + En CSS seleccionamos con elemento\_padre elemento\_hijo
  + Aplica para cualquier selector.
* Pseudo-clases
  + Sirve para aplicar estilos a un elemento en determinados eventos.
  + En CSS seleccionamos con selector:pseudo-clase

**Especificidad**

CSS aplica las propiedades a los elementos de acuerdo con una jerarquía de prioridades, a su vez, los cambios se aplican en cascada, por lo que, para cambios redundantes con selectores en el mismo nivel jerárquico, el resultado será aquel cambio que esté más cerca del final de la hoja de estilos.

****

**Metodología VIM**

A continuación, se presenta un ejemplo del uso de la metodología VIM.

<div class="contact-form">

<input type="text" class="contact-form\_\_input--active">

<input type="password" class="contact-form\_\_input">

<p class="contact-form\_\_p">

<h2 class="contact-form\_\_p-h2"></h2>

</p>

</div>

**Medidas**

* **Absolutas:**
  + px - pixeles
  + cm – centímetros
  + mm – milímetros
  + pt – puntos
* **Relativas:**
  + rem – Unidad efímera respecto a la raíz
  + em – Unidad efímera
  + vw – Viewport width (Porcentaje del ancho de la pantalla)
  + vh – Viewport height (Porcentaje de altura de la pantalla)
  + % - Porcentaje de la caja que contiene al elemento

1em es por defecto equivalente a 16px, pero pueden heredar su tamaño del elemento padre.

**Propiedades de texto**

* font-size: Tamaño de letra.
* font-family: Tipo de letra, se pueden especificar varias en caso de no encontrarse la deseada.
* line-height: Altura del interlineado (donde 1 es igual a la mitad del tamaño de la letra).
* font-weight: Grosor de la fuente, desde 100 a 900.
* font-stretch: Estira o comprime la fuente.
* font-variant: Variante de la fuente, por ejemplo, versalitas con small-caps

Recomendación: Utilizar Google Fonts para agregar fuentes a nuestros sitios web.

**Normalize**

Cada navegador utiliza estilos por defecto los cuales pueden generar inconsistencias en nuestras paginas web. Esto se puede solucionar utilizando una hoja de estilos que modifique estos valores en todos los navegadores.

[Normalize.css: Make browsers render all elements more consistently. (necolas.github.io)](https://necolas.github.io/normalize.css/)

**Padding**

La separación entre el contenido y la caja que lo contiene. padding es una *shorthand property* que condensa padding-top, padding-bottom, padding-left y padding-right.

Al definir **una** medida, el padding se aplica en todas direcciones.

Al definir **un par** de medidas, el padding se aplica en el eje vertical y el eje horizontal respectivamente.

Al definir **cuatro** medidas, el padding se aplica como se ve en la ilutración.

**Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente**

Y a continuación, todos los atributos que influyen en los márgenes y rellenos de los elementos HTML:

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

**Más propiedades de cajas**

* box-sizing: Cambia el comportamiento entre si agregar el padding al tamaño de la caja o no.
* text-align: Alinea el texto contenido.
* margin: La separación entre una caja y otras cajas adyacentes a esta.
* border-radius: Radio de redondeado del borde.
* border: Shorthand property para los bordes, se especifica el tamaño, el estilo y el color.
* box-shadow: Sombra de la caja, se especifica el desplazamiento horizontal, el desplazamiento vertical, el tamaño del desenfoque, el borde y el color de la sombra.
* text-shadow: Sombra del texto.
* transform: Transformaciones, por ejemplo, rotate(#deg)
* outline: Similar a border, pero no desplaza a los elementos adyacentes con el espacio que utiliza.

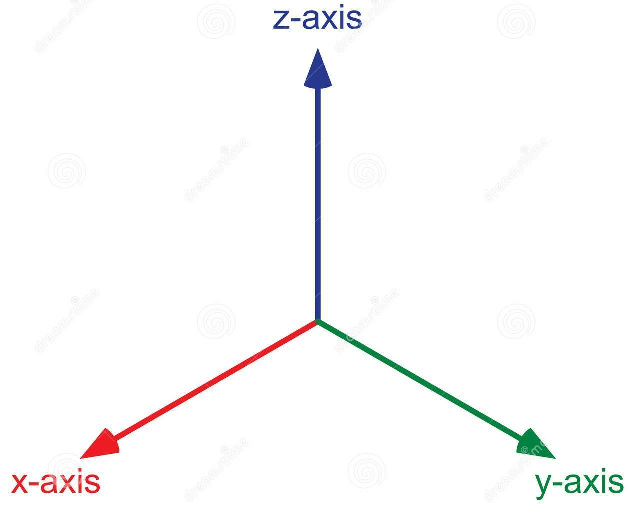
**Position**

Modifica las propiedades de posicionamiento del elemento.

* Relative
  + Nos permite mover el elemento respecto a la posición en la que se encontraría si no estuviera posicionada, reservando el espacio original.
  + Se utilizan los atributos top, bottom, left y right.
    - Top y left tienen más prioridad que bottom y right.
* Absolute
* Fixed
* Sticky

**Z-Index**

La altura en el eje z del elemento; si la altura de un elemento es mayor que la de otro elemento en el mismo lugar, el primer elemento se sobrepondrá al segundo. Sólo se puede utilizar cuando los elementos están posicionados (con el atributo *position*).

****

La única forma de poner un elemento padre por encima de uno de sus elementos hijos es asignarle al elemento hijo un z-index igual a -1 y no definir un z-index para el elemento padre.