**CSS (Cascading Style Sheet):**

• Sirven para estilizar el contenido HTML.

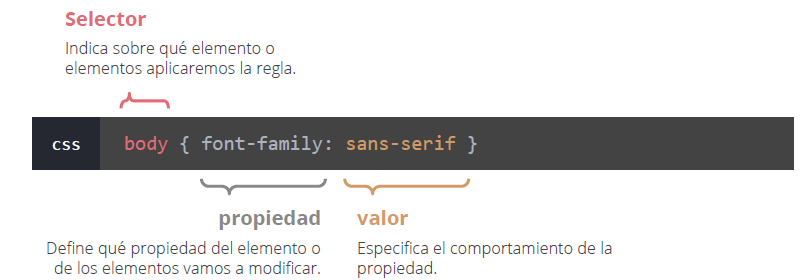
• Maneja el diseño y la presentación de las páginas web.

• Vinculación con HTML: <link href=”css/style.css” rel=”stylesheet”>.

**REGLAS**: conjunto de instrucciones que se aplican a un elemento determinado para agregarle estilos.

• Llaves: indican el comienzo de las reglas.

• Punto y coma (;): se puede poner más de una regla por elemento y se usan para separarlas.



**PROPIEDADES**: permiten manipular los elementos. Si se tuvieran que agrupar serian: tipografías, fondos, tamaños, posicionamiento, visualización, comportamiento, interfaz, entre otros.

**SELECTORES**:

• ID: atrapa al elemento HTML que tenga asignado el atributo id con el valor correspondiente. Se recomienda usar *nombres únicos* para cada elemento y no repetirlos a lo largo del documento. Para llamarlo usamos el *#* seguido del nombre del *ID*. Ej.:

* *En HTML*: <h3 id=”saludo”> Hola! </h3>.
* *En CSS*: #saludo {color: blue;}.

• Class: atrapar al elemento HTML que tenga asignado el atributo class con el valor correspondiente. Podemos asignarle la *cantidad de clases que queramos* a un mismo elemento, solo hace falta separarlas con un espacio. Para llamarlo usamos el *.* seguido del nombre de la *clase*. Ej.:

* *En HTML*: <h3 class=”noticia”> Una noticia </h3>.
* *En CSS*: .noticia {font-size: 22px;}.

• Etiqueta: atrapar al elemento HTML con el mismo nombre de etiqueta que llamemos desde nuestro CSS. Para llamarlo usamos el nombre de la *etiqueta*. Ej.:

* *En HTML*: <p> Lorem ipsum </p>.
* *En CSS*: p {color: grey;}.

• Combinados: son la *combinación* de todos los anteriores, también podemos usar varios selectores del mismo tipo. Ej.:

* *En HTML*: <h2 class=”subtitulo”> Un subtítulo </h2>.
* *En CSS*: h2.subtitulo {color: yellow;}. (Combinacion de selector de etiqueta con uno de clase).

• Descendientes: agregan *especificidad*. Para llamarlos escribimos los selectores separados por un espacio (el de la derecha siempre será el que está dentro del de la izquierda). Ej.:

* *En HTML*: <ul id=”lista”> <li> Primer item </li> </ul>.
* *En CSS*: ul#lista li {text-align: center}.

• Importante: CSS siempre va a priorizar a los selectores más específicos para aplicar los estilos.

**COLORES**: se pueden aplicar 4 tipos de formatos en cualquier propiedad que reciba color: *nombre* (purple), *hexadecimal* (#f05331), *RGB* (rgb(255, 100, 50)) y *RGBA* (rgba(122, 50, 125, 0.5), el ultimo representa la opacidad).

**Propiedades**: reciben como valor cualquiera de los formatos permitidos.

• Color: color al texto de un elemento.

• Background-color: color de fondo a un elemento.

• Opacity: transparencia a todo el elemento. Va del 0 al 1 y mientras menor será el nº mayor será la transparencia. Ej.: 0.5 representa una opacidad del 50%.

**FUENTES**:

• Fuentes seguras: conjunto de familias tipográficas que pueden ser usadas en cualquier página web sin problemas porque están preinstaladas en todos los sistemas operativos que utilizan nuestros dispositivos.

• Fuentes web: no siempre están preinstaladas en un dispositivo y, por lo tanto, deben ser descargadas en el navegador del usuario antes de ser mostradas. Ej.: Google Fonts o Adobe Fonts.

**Propiedades**:

• Font-family: familia tipográfica que queremos usar. Podemos poner más de una tipografía, separando las adicionales por *comas* (arial, sans-serif). En caso de que la primera no esté disponible, se cargará la 2º y así sucesivamente. Usamos *comillas* cuando el nombre de la tipografía este conformado por más de una palabra (“times new roman”). Para que funcione, la tipografía debe existir en la computadora del usuario o usar una webfont.

• Font-size: tamaño tipográfico. Recibe un valor numérico acompañado de la unidad de medida. Las unidades de medida más habituales suelen ser: *px, em* y *rem*.

• Font-style: estilo de la tipografía. Recibe los valores *italic*, *normal* y *oblique*. Para algunos elementos, como <em>, el valor por defecto será italic.

• Font-weight: peso de la tipografía. Recibe los valores *bold*, *lighter*, *normal*, entre otros. También puede recibir un valor numérico que se irá incrementando de 100 en 100. Para algunos tags el valor por defecto será bold.

• Text-align: alineación del texto. Los valores que recibe son *center*, *left*, *right*, *inherit* y *justify*. El valor por defecto para todos los elementos es left.

• Text-decoration: permite elegir un tipo de decoración para el texto. Recibe los valores *line-through*, *underline*, *overline* y *none*. Para algunos elementos, como los enlaces, el valor por defecto será underline.

• Line-height: define el alto de cada línea de textos, esto también suele llamarse *interlineado*. Recibe un valor numérico acompañado de la unidad de medida y, por lo general, está relacionado con el tamaño de la letra o font-size.

**FONDOS**:

**Propiedades con colores**:

• Background-color: color de fondo a un elemento.

**Propiedades con imágenes**:

• Background-image: permite asignarle una imagen de fondo al elemento, definiendo la ruta a través de la *URL*.

• Background-repeat: controla si se va a repetir, y de qué manera, la imagen dispuesta. Recibe los valores: *repeat*, *no* *repeat*, *repeat-x*, *repeat-y*, *round* y *space*.

• Background-position: mueve la imagen dentro del elemento y decidir dónde colocarla. Recibe como valores tamaños en *pixeles* y *porcentajes*, así como también *right*, *bottom*, *left*, etcétera. Podemos asignarle uno o dos valores. El primero para especificar la posición en el eje x y el segundo, la posición en el eje y.

• Background-attachment: establecer si la imagen de fondo se va a mover junto con la página al hacer scroll o si se va a quedar fija. Recibe como valor: *fixed*, *scroll*, *inherit* e *initial*.

• Background-size: establecer el tamaño de la imagen de fondo. Recibe como valor: *contain*, *cover*, *inherit*; así como también tamaños en *pixeles* y *porcentajes*, indicando con el primer valor el ancho y con el segundo, el alto.

**ICONOS**: se usan librerías como: *font awesome*, *iconic*, *material design*, etc. Se incluyen en HTML de la siguiente manera.

* *En head*: <link rel=”stylesheet” href=”https://use.fontawesome.com/release”>.
* *En body*: <i class=” fas fa-exclamation-triangle”> </i>.

**BUENAS PRACTICAS**:

• Crear 1ero el HTML: permite crear el CSS de manera más clara.

• Comentar: para acomodar las secciones (/\* \*/).

• Nombrar selectores correctamente: no comenzarlos con mayúscula, nº o caracteres especiales.

• Para separar palabras: usar camelcase o guion medio en su defecto.

• Usar nombres descriptivos: en los selectores permiten averiguan fácilmente a que elemento se le da estilo.

• No utilizar una característica visual como nombre de un selector: .menu-red en vez de .nav-menu {color:red}.

**DISPLAY**: podemos cambiar la disposición del elemento que queramos.

• Inline: define un elemento con comportamiento en línea, no recibe algunas propiedades del modelo de caja.

• Block: define un elemento con comportamiento de bloque, puede recibir propiedades del modelo de caja.

• Inline-block: define un elemento con comportamiento de semi-bloque, puede recibir propiedades del modelo de caja y también comparte propiedades de elementos de línea.

• None: oculta un elemento, no lo elimina de la estructura de HTML, solo desaparece de la vista.