

Clase 9

CSS para textos

ESTILOS CSS PARA TEXTOS

En esta clase conoceremos algunos de los tantos estilos que podemos aplicar a textos de HTML con CSS.

Les recomendamos utilizar esta Web donde para profundizar cada tema de CSS que veamos en la materia:

<https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>

A continuación desarrollaremos:

- **Font-size**
- **Unidades de medida**
- **Line-height**
- **Text-align**
- **Text-decoration**
- **Font-weight**
- **Font-style**
- **Font-family**
- **Styling lists**
- **List-style-position**

FONT-SIZE

`font-size` es utilizado para determinar el tamaño del texto que se encuentra dentro de un tag. Se compone de dos partes, la primera el número y la segunda la unidad de medida, normalmente se utilizan estas unidades de medida: `px`, `em`, `%`, `vh`.

Ejemplos:

```
p { font-size: 18px; }  
q { font-size: .8em; }  
blockquote { font-size: 10vh; }
```

También existen valores textuales predefinidos en el CSS que pueden ser aplicados en reemplazo del número y la unidad:

`xx-small`, `x-small`, `small`, `medium`, `large`, `x-large`, `xx-large`

O tamaños relativos como `larger`, `smaller`.

Ejemplos:

```
p { font-size: medium; }  
q { font-size: small; }  
blockquote { font-size: larger; }
```

UNIDADES DE MEDIDA (APLICA PARA CSS EN GENERAL)

Para definir tamaños en CSS se debe indicar el número y la unidad. Algunas de las unidades de medida que existen en CSS son las siguientes:

Medidas absolutas:

cm = Centímetros

mm = Milímetros

in = Pulgadas (2,54cm)

pt = Puntos (1/72 de una pulgada)

px = Píxeles (1/96 de una pulgada)

Medidas relativas (toman el valor base de un elemento que lo contiene):

em = Toma el valor del elemento contenedor y lo multiplica por el valor indicado por ejemplo 0.5em sería la mitad de tamaño que tenga definido el elemento que lo contiene, si fuera 1.5em sería un 50% más grande.

rem = Toma el valor del elemento base que lo contine y lo multiplica como lo hace 'em'.

% = Utilizado para dimensiones, permite determinar el porcentaje respecto al tamaño del elemento contenedor. Este elemento contenedor debe tener definido un tamaño ya sea 'width' o 'height' para que funcione.

vh y vw = Toma de base el tamaño de la pantalla y aplica un porcentaje respecto a ese tamaño. 'VH' para la altura (de Height) y 'VW' para el ancho (de Width)

Un ejemplo: <https://codepen.io/matiasgv/pen/MWQvgam>

LINE-HEIGHT

`line-height` permite definir el espacio entre líneas. Se puede indicar el número y la unidad o directamente el número sin unidad.

Además existen valores como por ejemplo 'normal', 'initial', etc. Les recomendamos buscar en el link indicado al comienzo de esta clase sobre 'line-height' para profundizar el tema.

Algunos ejemplos:

```
.ejemplo { line-height: 1.2; }
```

```
h1 { line-height: 1.4em; }
```

```
#parte1 { line-height: 120%; }
```

TEXT-ALIGN

`text-align` permite indicar la alineación del texto.
Los valores más comunes son los siguientes:

- `left`
- `center`
- `right`
- `justify`
- `justify-all`

Si se aplica a un elemento que tiene el mismo ancho que lo que ocupa el texto, posiblemente no funcione dado que esto es aplicable cuando el elemento es más grande que el lugar que ocupa el texto.

Lo mismo pasa si lo aplicamos a tags que se utilizan dentro de la línea como por ejemplo `'span'`, `'a'`, `'i'`, `'q'`, etc.

Debería ser aplicado a tags de bloques como por ejemplo `'p'`, `'li'`, `'ul'`, `'div'`, `'blockquote'`, etc.

Ejemplos:

```
p { text-align: left; }  
blockquote { text-align: right; }
```

TEXT-DECORATION

Con text-decoration podemos subrayar, tachar, colocar una línea por sobre el texto o aplicar varios de estos juntos.

Los valores posibles son:

- underline
- overline
- line-through
- none

```
p { text-decoration: underline; }  
a { text-decoration: none; } /* le sacamos el subrayado a los hipervinculos */  
span { text-decoration: overline; }  
span { text-decoration: underline overline; } /* aplicamos ambos */  
span { text-decoration: underline overline line-through; } /* aplicamos todos juntos */
```

También existen propiedades para customizar la decoración como por ejemplo

`text-decoration-color` `text-decoration-style` pero no son soportados por todos los navegadores.

Los invitamos a que investiguen sobre esas propiedades.

FONT-WEIGHT

En las clases anteriores conocimos `` y `` para dejar en negrita a los textos. Con `font-weight` podemos aplicar esto desde CSS y además nos permite definir mediante valores numéricos el tamaño del borde (digamos cuán negrita lo queremos).

Podemos aplicar valores como:

- Normal
- Bold
- Lighter
- Bolder
- Valor numérico: 100, 200, 300, 400, 500, 600, 700, 800 o 900

Por ejemplo:

```
p { font-weight: bold; }  
blockquote { font-weight: 900; }
```


FONT-STYLE

Como vimos en clases anteriores podemos aplicar `<i>` o `` para verlo en cursiva o Italic.

Con Font-style podemos aplicar desde CSS este mismo efecto colocando el valor `'italic'`.

También podemos optar por valores como `'oblique'` o también agregar un numero luego de `'oblique'` que indique el ángulo de inclinación.

Para poder ver a los casos de `'oblique'` la fuente utilizada debe tener este tipo de estilo, caso contrario lo veremos como `'italic'`. Pueden ver un ejemplo de ese caso con `'oblique'` en la Web recomendada.

Un ejemplo:

```
p { font-style: italic; }
```

FONT-FAMILY

`font-family` nos permite indicar la fuente a utilizar en el texto a mostrar.

Existen pocas opciones que están aseguradas en los navegadores, son estas:

`serif`, `sans-serif`, `monospace`, `cursive` y `fantasy`.

El resto de fuentes dependerán de que se encuentre instalada en la computadora que navega el sitio. Por las dudas que la fuente no se encuentre disponible, podemos indicar más de una fuente para que si no encuentra la primera intente con la siguiente.

Por ejemplo podemos definir esto:

```
p { font-family: "Helvetica", "Verdana", "Arial", sans-serif; }
```

Si la computadora no tiene “Helvetica” intentará con “Verdana”, luego con “Arial” y si no encontró ninguna terminamos colocando una de las que si o si están, en este caso sans-serif.

También existen fuentes Web que pueden ser cargadas mediante CSS, pueden buscar ‘Web Fonts’ en Google y desarrollar este tema.

STYLING LISTS

Como vimos en HTML podemos crear listas con ``, `` y ``

Ahora con CSS podemos aplicar estilos con `list-style-type` y podemos cambiar la ubicación con `list-style-position` (valores posibles: `inside` o `outside`)

Valores posibles para `list-style-type`:

Para `'ul'`:

`disc, circle, square, none`

Por ejemplo:

```
li { list-style-type: disc; }
```

Para `'ol'`:

`decimal, decimal-leading-zero, lower-roman, upper-roman, lower-Alpha, upper-alpha`

Por ejemplo:

```
li {  
    list-style-type: decimal-leading-zero;  
    list-style-position: inside;  
}
```

También podemos colocar una imagen, para eso debemos utilizar `list-style-image` como en el siguiente ejemplo:

```
li { list-style-image: url("my_triangle.png"); }
```

Ver ejemplo completo en:

<https://codepen.io/matiasgv/pen/MWQvaWX>