

# Google fonts

Utilización de google fonts para títulos y párrafos, mencionar también que se pueden descargar fuentes de [www.dafont.com/es/](http://www.dafont.com/es/) y [www.1001fonts.com](http://www.1001fonts.com) para importar fuentes descargadas:

```
@font-face{  
src: url('./path-to-font/');  
font-family: 'the_name_of_my_font';  
}
```

## Listas

Las listas no ordenadas tienen algunas propiedades como por ejemplo modificar la viñetas de los elementos:

```
.list{  
list-style-type: disc, circle, decima,  
decimal-leading-zero, y algunos alfabetos;  
}
```

Reseteo de la lista para personalizarlas o para hacer menús.

```
.list{  
margin-top: 0;  
margin-bottom: 0;  
padding-left: 0;  
list-style: 0;  
}
```

## Tablas

Una tabla tiene su propio layout o su propio formato, hay cosas que no se pueden manejar como en CSS normal.

- **table-layout:**
  - **automatic:** comportamiento por defecto, ocupa lo que puede y colapsa, las columnas con más contenido son más anchas.
  - **fixed:** se establece un width para dividir todas las columnas con un mismo ancho
- **caption-side:** define en donde se coloca el caption de la página web
  - **top:** se posiciona en la parte superior
  - **bottom:** se posiciona en la parte inferior.
- **border-spacing:** controla el espacio entre las celdas
- **border-collapse:** la presentación de las celdas, si son seguidas o no
  - **spacing:** las celdas son separadas y puede modificarse su separación con la propiedad anterior
  - **collapse:** las celdas van juntas similar a como se visualiza en excel.
- **empty-cells:** controla qué hacer con las celdas vacías
  - **hide:** esconde las celdas vacías

## Imágenes

Una imagen responsive requiere una sola propiedad

```
.selector{
max-width: 100%;
}
```

- Dado que las imágenes son elementos inline, es decir, ocupan solo su espacio pero cuenta con un line-height que hace que exista un espacio por debajo en el contenedor de la imagen, se puede corregir utilizando la propiedad display:block en la imagen o el line-height del contenedor a 0.

```
.img{
display: block;
}
.img-container{
line-height: 0;
}
```

- **Centrado de las imágenes:** lo mejor aquí es cambiar el display de la imagen a block y darle margen auto a ambos lados, así va a estar centrada horizontalmente sin problema, sea cual sea su espacio.

```
.img{
display: block;
margin-left: auto;
margin-right: auto;
}
```

- **Centrado vertical:** Se puede utilizar flex-blox en display: flex y se pondría align-items: center, pero algunos navegadores viejos no lo soportan, para esos se requeriría hacer display: table para el contenedor y display: table-cell para sus elementos, adicional agregar vertical-align: middle en los elementos.

Otros aspectos importantes a la hora de trabajar con imágenes es object-fit para ajustar la imagen a una ventana o contenedor sin deformar la imagen y object-position que define qué porción o desde qué sección de la imagen se va a visualizar.

También se tiene filter, se tiene valores como blur(px) para desenfoque, brightness(0-1 quitar brillo, 1 en adelante sobre-exponer) brillo de la imagen, contrast(%), drop-shadow(h-shadow v-shadow desenfoque y color) para dar sombra al exterior, grayscale(0-1), hue-rotate(deg), invert(0-1) para invertir los colores de la foto, opacity(0-1) para transparencia, saturate(0-1) saturación de la imagen, sepia(0-1).

Para máscaras se utiliza clip-path tiene opciones como circle, ellipse, inset y poligon, recomendación usar <https://bennettfeely.com/clippy/>

## Colores

### Plugin para VS Code color highlight

En CSS existen diferentes formas de añadir colores a un fondo o un texto, algunas permiten una manipulación leve del color, como su tonalidad y algunas permiten manipular otros aspectos más avanzados. Las opciones que se tienen en CSS para definir un color son:

#### > Color name

```
.selector{
color: red;
}
```

## ➤ RGB y RGBA

```
.selector{  
  /*Se crean los colores definiendo su porcentaje de Rojo,  
  Verde y Azul*/  
  color: rgb(0, 0, 0);  
  /*Igual al anterior pero el valor final define la transparencia  
  del color*/  
  color: rgba(0, 0, 0, 1);  
}
```

## ➤ Hexa

```
.selector{  
  /*rgb*/  
  color: #fff;  
  /*rrggbb*/  
  color: #ffffff;  
}
```

## ➤ HSL y HSLA

```
.selector{  
  /*Color o ángulo en la rueda cromática, saturación,  
  luminosidad*/  
  color: hsl(120, 100%, 50%);  
  /*un valor adicional que indica la transparencia*/  
  color: hsla(120, 100%, 50%, 1);  
}
```