



Tipografías

Familias

- Por medio de la propiedad **font-family** se puede establecer una lista de familias tipográficas
- Dado que todas las máquinas pueden no tener todas las tipografías instaladas podemos seleccionar más de una
- Las tipografías las separamos con comas. Ejemplo: Arial, serif
- El browser va a elegir cada tipografía de izquierda a derecha hasta encontrar una que tenga
- Los nombres de las tipografías se escriben con comillas dobles en caso de que el nombre contenga espacios, ejemplo: "Times New Roman", Georgia, serif
- Se recomienda poner una tipografía por default (serif, sans-serif, monospace, cursive o fantasy) para lograr al menos el estilo deseado
- También se pueden utilizar web fonts como pueden ser las de [Google](#)
- Para saber más sobre la propiedad **font-family** pueden ver el [MDN](#)

Ejemplos de tipografías por defecto



```
body {  
  font-family: Arial, sans-serif;  
  color: black;  
}  
  
h1 {  
  font-family: "Times New Roman", serif;  
  color: gray;  
}
```

- En este ejemplo definimos:
 - El texto del documento tiene que ser negro y con tipografía Arial o sans-serif



- Los títulos h1 son grises y con tipografía Times New Roman o serif

Tamaño

- Por medio de la propiedad **font-size** podemos establecer el tamaño de la tipografía
- Acepta como valor:
 - **Píxeles:** establecemos el número de píxeles y la palabra **px**
 - Es un valor fijo
 - Es ideal para pantallas de computadoras
 - No escalan bien para dispositivos mobile
 - No se pueden agrandar en browsers viejos
 - **Porcentaje:** es un % del valor establecido
 - Es relativo
 - Escala bien
 - Es fácil para cambiar la relación de todos los tamaños ya que definimos el valor inicial y después lo cambiamos en un solo lugar (el resto es todo relativo)
 - **EMS:** este valor representa una unidad del tamaño establecido
 - Si establecemos el tamaño a 10px, 1em representa 10px y 2em 20px
 - Si no se establece un valor toma el por default del browser (16px en general)
 - Se dice que toma el ancho y alto de la letra M
 - Es una unidad relativa al padre
 - Escala bien para mobile
 - Funciona en cascada
 - Un truco es establecer el tamaño del sitio en 62.5% tomando que el browser tiene 16px nos queda una tipografía inicial de 10px
 - Luego podemos utilizar valores como 1.8em para lograr una tipografía de 18px
 - **Points:** es una unidad para imprimir
 - 27 puntos representan una pulgada es decir 2,54 cm
 - Son precisos a la hora de imprimir
 - Para aprender más sobre el tema podemos seguir leyendo este buen artículo de alistapart.com (en inglés y muy buen sitio)
 - Una buena sección para mirar del mismo sitio: [Typography Web Fonts](#)





Font size

```
body { font-size: 100%; }
```

```
p { font-size: 12px; }
```

```
a { font-size: 1em; }
```

Bold

- Por medio de la propiedad **font-weight** podemos establecer si el peso de nuestra tipografía
- Podemos establecer si es **normal** o **bold**
- También acepta valores numéricos entre 100 y 900
- Otra opción es utilizar valores relativos al padre por medio de **lighter** (más liviano) o **bolder** (más pesado)
- Estas últimas opciones no son soportadas por todos los browsers
- Podemos aprender más sobre esta propiedad en el [sitio de MDN](#)

```
.negrita { font-weight: bold; }
```

```
h1 { font-weight: 900; }
```

```
<p>Texto en <span class="negrita">NEGRITA</span></p>
```

```
<h1>Texto con negrita en 900 de peso</h1>
```

Itálica

- Por medio de la propiedad font-style podemos establecer si queremos que nuestra tipografía sea normal, itálica o oblicua
- Los valores que acepta esta propiedad están en inglés:
 - normal
 - italic
 - oblique
- Podemos aprender más sobre esta propiedad en el [sitio de MDN](#)



```
.italica { font-style: italic; }
```

```
.oblicuo { font-style: oblique; }
```

```
<p>Texto en <span class="italica">Italica</span></p>  
<h1>Texto en <span class="oblicuo">Oblicuo</span></h1>
```

Transformación de texto

- Por medio de la propiedad **text-transform** podemos transformar las mayúsculas y minúsculas de nuestros textos
- A diferencia de lo que venimos viendo en este caso se utiliza text en lugar de font ya que hace referencia al texto y no a la tipografía
- Acepta los siguientes valores: **uppercase**: pasa todo el texto a mayúscula **lowercase**: pasa todo el texto a minúscula **capitalize**: convierte la primer letra de cada palabra a mayúscula
- Podemos aprender más sobre esta propiedad en el [sitio de MDN](#)

```
h1 { text-transform: uppercase; }
```

```
p { text-transform: lowercase; }
```

```
<h1>texto todo en mayúscula</h1>  
<p>TEXTO TODO EN MINUSCULA</p>
```

Alineado de textos

- Por medio de la propiedad text-align podemos establecer como queremos alinear los textos de forma horizontal
- Acepta los siguientes valores:
 - **left**: alinea todo a la izquierda
 - **right**: alinea todo a la derecha
 - **center**: alinea todo al centro



- **justify**: hace que el texto ocupe toda la caja (justificado)
- Podemos aprender más sobre esta propiedad en el [sitio de MDN](#)

Decorado

- Por medio de la propiedad text-decoration establecemos si queremos decorar nuestras tipografía según los valores posibles:
 - **none**: saca todo tipo de decorado (es útil para hacer que los hipervinculos no tengan una raya debajo del texto)
 - **underline**: nos muestra el texto subrayado
 - **overline**: agrega una línea sobre el texto
 - **line-through**: el texto aparece tachado con una línea en el medio (es útil a la hora de tachar elementos de una lista)
 - **blink**: Volvemos a los 90's y nuestros textos parpadean
- Podemos aprender más sobre esta propiedad en el [sitio de MDN](#)

```
a { text-decoration: none; }  
h1 { text-decoration: underline; }
```

- En browsers más nuevo se puede establecer el color y tipo de línea que utilizamos
- Esta opción todavía **no es soportada por todos los browsers**
 - Primer parámetro el tipo de decorado
 - Segundo parámetro el tipo de línea (más sobre esto cuando vemos bordes)
 - Finalmente establecemos el color de la línea

```
h1 {  
  text-decoration: underline wavy red;  
}
```

Indentación

- Podemos indentar nuestro texto utilizando la propiedad **text-indent**
- Esta propiedad nos permite establecer un valor que va a funcionar como margen izquierdo de la primer palabra de la primer línea.
- Acepta valores negativos



- Podemos aprender mas sobre esta propiedad en el [sitio de MDN](#)

```
p { text-indent: 20px; }
```

Sombra

- Para establecer la sombra de nuestro texto utilizamos la propiedad **text-shadow**
- Esta propiedad utiliza varios valores:
 - El primer parámetro maneja el eje x de nuestra sombra, es decir que va de izquierda a derecha
 - El segundo parámetro maneja el eje y, es decir que va de arriba a abajo
 - El tercer parámetro maneja el blur (difuminar) de la sombra
 - El cuarto parámetro es el color de la sombra
- Podemos aprender más sobre esta propiedad en el [sitio de MDN](#)
- Los browsers actuales [soportan bien esta propiedad](#)

```
p { text-shadow: 4px 2px 0px #000000; }
```

- Para jugar con estas opciones podemos usar el [siguiente sitio](#)

Espaciado o aire

- Podemos establecer la distancia tanto entre letras como palabras
- Para manejar la distancia entre letras utilizamos **letter-spacing**
- Para manejar la distancia entre palabras utilizamos **word-spacing**
- Podemos aprender más sobre esta propiedad en el sitio de MDN:
 - [letter-spacing](#)
 - [word-spacing](#)

```
body {  
  letter-spacing: 1px;  
}
```

```
p {
```



```
word-spacing: 2px;  
}
```

Alto de línea

- Podemos establecer el alto de línea de un texto por medio de la propiedad **line-height**
- Acepta un valor numérico
- Podemos aprender más sobre esta propiedad en el [sitio de MDN](#)

```
p {  
  font-size: 12px;  
  line-height: 18px;  
}
```

Alineado Vertical

- Podemos alinear los textos de forma vertical utilizando la propiedad **vertical-align**
- Acepta los siguientes valores:
 - baseline
 - sub
 - super
 - top
 - text-top
 - middle
 - bottom
 - text-bottom
- Algo común es creer que esta propiedad nos permite alinear los textos de forma vertical en los elementos en bloque pero no funciona de esa forma
- Esta propiedad nos permite alinear el contenido de los elementos en línea
- Como sabemos las imágenes son elementos en línea al igual que los spans, hipervínculos, etc
- Si dentro de un párrafo tenemos una imagen, texto y otros elementos en línea, podemos establecer como va a ser la relación entre todos estos elementos que son hijos del párrafo
- De esta forma podemos hacer que un texto este bien alineado a una foto
- Un buen sitio para leer sobre este tema es [css tricks \(en inglés\)](#)



```
img { vertical-align: bottom; }
```

 [Edit this page](#)

Previous

« **Colores**

Next

Modelo de Caja »

Familias

Tamaño

Bold

Itálica

Transformación de texto

Alineado de textos

Decorado

Identación

Sombra

Espaciado o aire

Alto de línea

Alineado Vertical

Docs

Frontend

Community

Discord

Social



Linkedin

Instagram

Facebook

Copyright © 2021 Ada Itw.

