

Bootcamp Java Developer

Fase 2 - Java Web Developer
Módulo 14

Tipografías con CSS

Propiedades de fuente

`font-family`

Es una propiedad que hace referencia a la **tipografía**.

Las **familias de tipografías se dividen en varios tipos**. Hace un tiempo, cada vez que trabajábamos en un entorno web, debíamos tener cuidado de **diferenciar fuentes seguras de web o de sistema y fuentes de fantasía**.

Hoy, ese problema ha sido superado, pero veamos un ejemplo de aquel cuadro al cual debíamos prestar atención en su momento.

[Este link ayudará a entender ese proceso.](#)

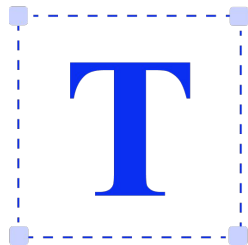
```
p {font-family: arial;}
```

Generar sugerencias

Actualmente, es posible trabajar con varios elementos alternativos para poder **utilizar la tipografía que nos guste** sin la limitación anterior.

Incluso, podemos sugerir con qué tipografía trabajar si el navegador tiene **problemas para encontrar la primera opción tipográfica**.

```
p { font-family: Verdana, Geneva, Tahoma, sans-serif }
```



Fuentes descargables

Propiedades de fuente

@font-face

Es una regla que **permite incorporar tipografías a nuestro proyecto desde una ruta, incluso en nuestro propio entorno de trabajo.**





```
@font-face { src: url(../fuentes/BodoniFLF-BoldItalic.ttf);  
  font-style: italic; font-weight: bold;  
  font-family: miFuente;}
```

Siempre que trabajamos con un **recurso externo** a nuestra hoja de estilo debemos trabajar con el **protocolo url** y luego **indicar la ruta.**



Generar variantes

Las tipografías tienen variantes, y **debemos explicitarlas** para poder trabajar, por ejemplo la **tipografía anterior consta de las siguientes variantes**:

	BodoniFLF-Bold
	<i>BodoniFLF-BoldItalic</i>
	<i>BodoniFLF-Italic</i>
	BodoniFLF-Roman



@font-face

Uso de variantes

Cada variante que nos interese utilizar debe estar mencionada de la siguiente forma:

```
✓ @font-face { src: url(../fuentes/BodoniFLF-BoldItalic.ttf);  
    font-style: italic; font-weight: bold;  
    font-family: miFuente;}  
  
✓ @font-face { src: url(../fuentes/BodoniFLF-Bold.ttf);  
    font-weight: 400;  
    font-family: miFuente;}
```


Luego, aquel elemento que utilice esas variantes tomará ese recurso específico, por ejemplo:

```
p {font-family: miFuente; font-style: italic; font-weight:bold; }
```



Google Fonts

Google Fonts

Uso de un CDN

Un **CDN** es un **Content Delivery Network**. El trabajo con este tipo de metodología permite **incorporar un recurso a nuestra interfaz sin necesidad de descargarlo**.

Si bien existen múltiples recursos posibles, mayormente encontramos el trabajo con **Google Fonts** como el **más usual**.

La página es <https://fonts.google.com/> y nos permite incorporar cualquier tipografía disponible en el inmenso catálogo que nos ofrece.

Al elegir una **tipografía** podemos encontrar **también diversas variantes**.

En los ejemplos posteriores trabajaremos alrededor de la **tipografía Texturina**.

Texturina

Variable

Guillermo Torres, Omnibus-Type



Esta tipografía de ejemplo tiene muchas **variables**, a la derecha se muestran algunas de ellas.

Para elegir con qué variante trabajar haremos clic en el signo **+**, al lado del nombre de cada una.

ExtraLight 200 ⊕

Styles

Type here to preview text

Texto de ejemplo

39px

Optical Size ⓘ



Thin 100

Texto de ejemplo

Thin 100 ⊕

Thin 100 Italic

Texto de ejemplo

Thin 100 Italic ⊕

ExtraLight 200

Texto de ejemplo

ExtraLight 200 ⊕

ExtraLight 200 Italic

Texto de ejemplo

ExtraLight 200 Italic ⊕

Light 300

Texto de ejemplo

Light 300 ⊕

Al elegir la tipografía así como sus variantes, **Google Fonts** nos ofrece **dos formas de trabajar**.

1. **El elemento link es limitante.** Ante cualquier cambio, se deberían modificar todos los **HTML vinculados a él**.
2. **Es recomendable trabajar con @import**, ya que ante cualquier cambio, solo con **modificar el CSS** en cuestión, **todos los HTML relacionados con él podrán actualizarse** de forma más rápida y dinámica.

Use on the web

To embed a font, copy the code into the `<head>` of your html



`<link>`



`@import`

```
<style>
@import url('https://fonts.googleapis.com/css2?family=Texturina:ital,opsz,wght@1,12..72,200&display=swap');
</style>
```

Luego, sólo se debe copiar y pegar la propiedad y valor que ofrece **Google Fonts**:

```
font-family: 'Texturina', serif;
```

En nuestra hoja de estilos se verá de la siguiente forma:

```
@import url('https://fonts.googleapis.com/css2?family=Texturina:wght@100;200&display=swap');  
p {font-family: 'Texturina', serif;}
```

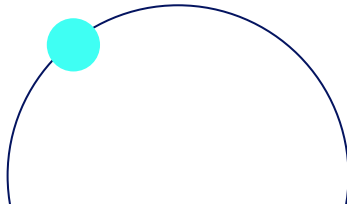
Pero... ¿qué es @import?

Vinculación dentro de la propia hoja de estilo

@import actúa igual que la **vinculación link**, pero a diferencia de ésta, **puede colocarse dentro de un archivo.css**.

De esta manera, ante cualquier cambio, **hace más rápido el mantenimiento de una interfaz**. Alcanza con ir a un archivo vinculado con todos los **.html** y modificar el elemento distinto. Por ejemplo, en el caso de **cambiar la tipografía de un sitio web entero**.

```
@import url('estilos.css')
```



Ejes tipográficos

Uso de ejes desde Google Fonts

En Google Fonts, encontramos **tipografías fijas y otras variables**.

Primero debemos entender que **las tipografías tienen ejes**. Por ejemplo:

- italic (eje de inclinación / *slant*)
- bold (eje de peso / *weight*)
- condensed (eje de ancho / *width*)
- bold-italic (eje de inclinación y peso)

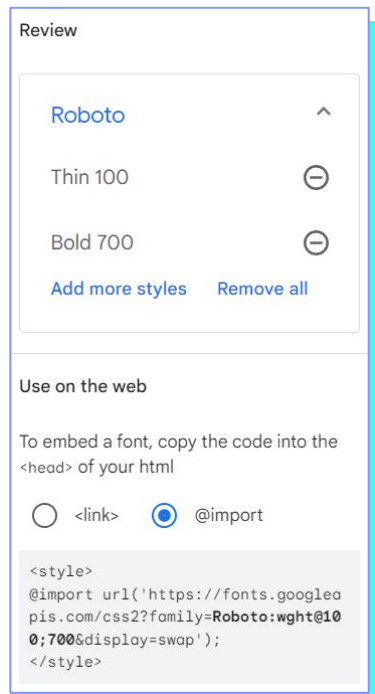
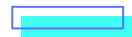
Cuándo trabajamos con los diferentes ejes y sus valores, por ejemplo en el **weight** ir de 400 a 700, debemos utilizar o llamar todos los archivos necesarios para poder hacerlo.



Uso en Roboto de Thin y Bold

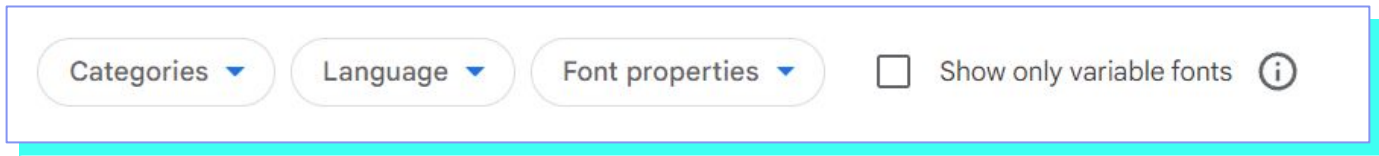
Una tipografía fija, como Roboto, obliga a vincular todas las variantes que se necesitan, por ejemplo, **thin** y **thin italic**.

Sin embargo, si se selecciona una **tipografía variable**, con un mismo archivo se pueden usar **todas las variantes**. De esta manera, se utilizarán menos recursos y será más fácil la implementación.



Uso de fuentes variables

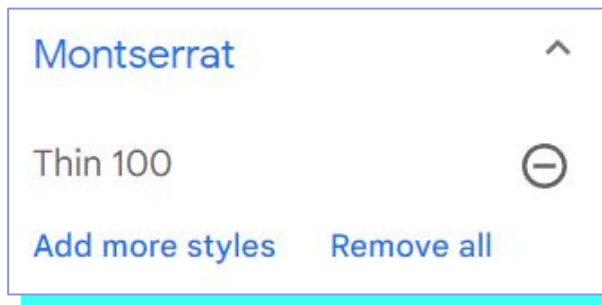
Al momento de realizar una búsqueda en Google Fonts, debemos tildar la casilla **Show only variable fonts** (Mostrar sólo fuentes variables) para así visibilizar sólo aquellas que posean esta característica.



A partir de este punto, podemos elegir la tipografía que nos guste. Para eso debemos reconocer cuántos ejes posee y sus variantes.

Trabajo de ejemplo con Montserrat

Para comenzar a trabajar, vamos a elegir Montserrat y, en este caso, un valor solo del eje *weight* por ejemplo:



Copiamos el link que nos ofrece Google Fonts tal cual está:

```
<style>
@import url('https://fonts.googleapis.com/css2?family=Montserrat:wght@100&display=swap');
</style>
```



Copiamos este **@import** en nuestro documento y le realizamos un pequeño cambio teniendo en cuenta que Montserrat tiene un *weight* desde el 100 al 900.



En base a eso, solo hay que modificar el `@import` de la siguiente manera:

```
@import url('https://fonts.googleapis.com/css2?family=Montserrat:wght@100..900&display=swap');
```

Esto es suficiente para poder utilizar todos los valores del 100 al 900, sin necesidad de vincular más que este único archivo.

De esta forma, si es posible **trabajar con fuentes variables se simplifica tanto la realización como el mantenimiento de la página** en cuanto al uso de tipografías.

Tipografías Tamaño

Propiedades de fuente

font-size

Es una **propiedad que hace referencia al tamaño de la tipografía**.

Los valores nos servirán incluso para otras propiedades como **width, height o cualquier propiedad que involucre medidas**.



Se pueden utilizar **medidas de longitud absolutas**, es decir, aquellas que **no cambian según el contexto, entorno, o dispositivo a través del cual las vemos**:

- cm
- mm
- in
- etc

También podemos utilizar **medidas relativas**, es decir aquellas que **dependen del medio, el tamaño del padre o cualquier otra referencia**.

- px
- %
- em
- rem

Estas medidas permiten **trabajar, en el caso de los px**, en referencia a la resolución de pantalla o, en el caso del %, **em o rem**, en base a una medida de referencia de un elemento padre.



```
body { font-size: 80%;}  
h1 { font-size: 4em;}
```


**Adaptabilidad,
rem y em**

¿Em o rem?

Diferencias en el uso de em o rem en tipografía

La principal característica de **em** es que es **equivalente a la medida base actual del documento**. Por ejemplo, si un elemento posee un valor de 2 em significa que, si tomamos un documento HTML por defecto, cuya base tipográfica está fijada en 16 px, este elemento entonces será el de 32 px, es decir 2 em es igual a 16 px x 2 en este caso.

Sin embargo, en el ejemplo debajo, el párrafo no será el doble de 30 px, sino que acumulará las medidas de sus elementos padres, es decir 180 px en total.

```
body { font-size: 30px;}  
header { font-size: 2em;}  
div { font-size: 1.5em;}  
p { font-size: 2em; }
```

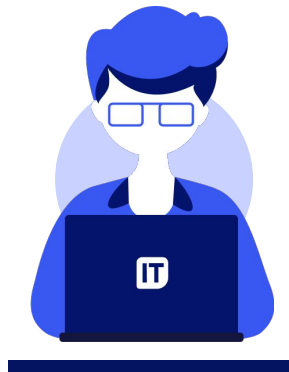


En el caso del **rem**, solo permite que se tome una medida base, es decir la medida del *root* o etiqueta *html*. Por lo tanto, en el ejemplo de la derecha, el valor del **h2** no será el cuádruple de la medida base (como pasaría si hubiésemos utilizado **em**), sino solo el doble. **Los valores no se acumulan y siempre refieren al root.**

El uso de **rem** es ideal para elementos que no queremos que se modifiquen en tamaños distintos de pantalla o *viewports* o cuando en éstos cambiamos el tamaño del *body* para hacer responsivos algunos elementos tipográficos o vinculados al tamaño tipográfico.

```
div {font-size: 4em;}
```

```
h2 {font-size: 2rem;}
```



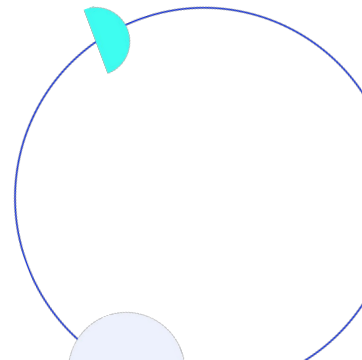
Conversiones

Si bien existen diferentes herramientas para convertir medidas de longitud, en nuestro caso, podemos hacer una **regla de tres simple** muy sencilla para **convertir una medida en px a una medida en em/rem**.

Si el diseñador nos brinda una medida de 20px esta podrá convertirse a em de la siguiente forma:

$$20\text{px} * 100 / 16\text{px}(\text{medida base real}) = 1.25\text{em}$$

De esta forma, **todas las medidas de nuestro diseño original podrán ser convertidas** sin problemas para así lograr una **mayor adaptabilidad** de nuestras tipografías, márgenes, paddings y demás elementos.



Tipografías responsivas: em y rem

Una forma de trabajar con tipografías de modo responsivo es que en cada **@media** asignemos una base distinta, es decir, trabajamos en el **body** y el resto de los elementos utilizan **em** para usar esa base que se agranda o achica como referencia, o asignamos específicamente un tamaño a cada elemento según el rango de pantalla (ver ejemplos a la derecha).



```
@media (max-width: 992px) {  
  body {  
    font-size: 150%;  
  }  
}  
  
@media (max-width: 768px) {  
  body {  
    font-size: 80%;  
  }  
}
```

Tipografías fluidas: vw y vh

El ejemplo de la pantalla anterior, donde para cada **@media** asignamos un valor específico de base o de tamaño para cada texto, no es siempre la mejor opción.

Muchas veces tenemos tamaños de pantalla intermedios donde los **breakpoints** utilizados generan saltos donde la tipografía no logra el tamaño adecuado. Corregir esos problemas, genera mucho trabajo o múltiples **breakpoints** innecesarios en un proceso donde la idea es simplificar todo al máximo.



Una alternativa es utilizar, entonces, **valores vinculados al ancho o alto de pantalla**, por ejemplo, **vh** o **vw**. De esta forma, nuestra **tipografía toma el tamaño de forma fluida**.

Nota: recomendamos seguir la guía de un diseñador con conocimiento tipográfico para la elección correcta de la familia tipográfica dependiendo el medio, pero si quieres conocer un poco más sobre tipografías y cómo usarlas, puedes visitar:

- [Cómo escoger la tipografía correcta en 5 pasos](#)
- [¿Cómo usar la tipografía en diseño web correctamente?](#)
- [Tipografía para Front Ends - Elena Ramírez \(LSE\)](#)

Revisión

- Repasar los conceptos vistos de **familia de fuentes**.
- Revisar los conceptos de **tamaños de fuente**.
- Implementar estas nuevas propiedades.
- Ver todos los videos y materiales necesarios antes de continuar.



**¡Sigamos
trabajando!**