

UD2_2_4.2 CSS. HOJAS DE ESTILO CSS 3 Texto

CONTENIDOS

1. Tipos de letras personalizados.
2. Las letras huecas
3. Manejar títulos largos.
4. Los guiones automáticos.
5. Redimensionar un elemento.

1. Tipos de letra personalizados.

Las hojas de estilo CSS3 permiten importar y utilizar una tipografía personalizada incluso si no está instalada en el puesto del usuario. Esta propiedad **@font-face** constituye una innovación significativa para los diseñadores Web que podrán incluir, de este modo, tipografías más creativas.

La adopción de un tipo de letra particular se realiza en dos fases.

Primero hay que descargar el tipo de letra para, a continuación, incorporarlo en la página.

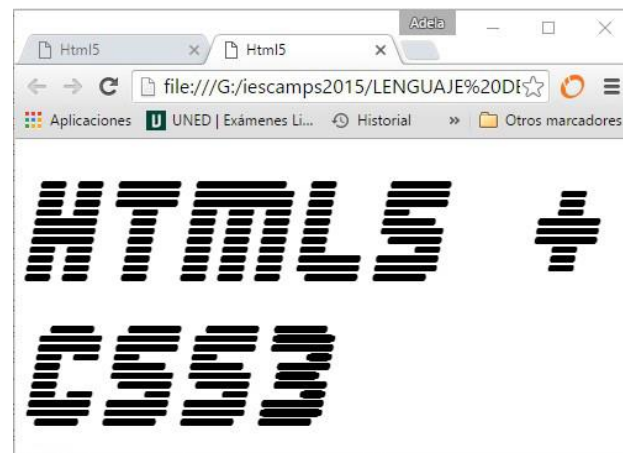
```
@font-face { font-family: "nombre_tipografia";  
url("nombre_tipografia.extension_tipografia"); }
```

- `font-family: "nombre_tipografia"` indica con qué nombre se va a referenciar al tipo de letra en la página.
- `url("nombre_tipografia.extension_tipografia")` indica la url, mediante direccionamiento relativo o absoluto, del tipo de letra personalizado.
- **extension_tipografia** determina la extensión del tipo de letra, por ejemplo la extensión ttf (True Type Fonts).

2. Tipos de letra personalizados.

Una vez hemos descargado el tipo de letra mediante **@font-face**, es posible aplicarla a algún elemento, como con cualquier otro tipo de letra, mediante la propiedad `font-family`.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
@font-face { font-family: "mitipografia";
src: url('fantasia.ttf');}
.fontface { font: 100px mitipografia,
fantasy;}
</style>
</head>
<body>
<div class="fontface">Html5 + CSS3</div>
</body>
</html>
```



3. Tipos de letra personalizados.

Existen diferentes formatos de fuentes adaptadas a la Web (Web fonts):

Formato .eot	El formato eot (Embedded OpenType) es un formato compacto de fuente OpenType. Este formato es propietario de Microsoft.
Formato .ttf	El formato ttf (True Type Font) es el formato de fuente de caracteres más extendido.
Formato .otf	El formato otf (OpenType Font) es una evolución del formato ttf en el que se han enriquecido las posibilidades tipográficas.
Formato .woff	El formato woff (Web Open Font Format) es un formato de fuente de caracteres comprimido para uso en la Web. Este nuevo formato, desarrollado en 2009, ha sido adoptado rápidamente por diferentes navegadores.

4. Tipos de letra personalizados.

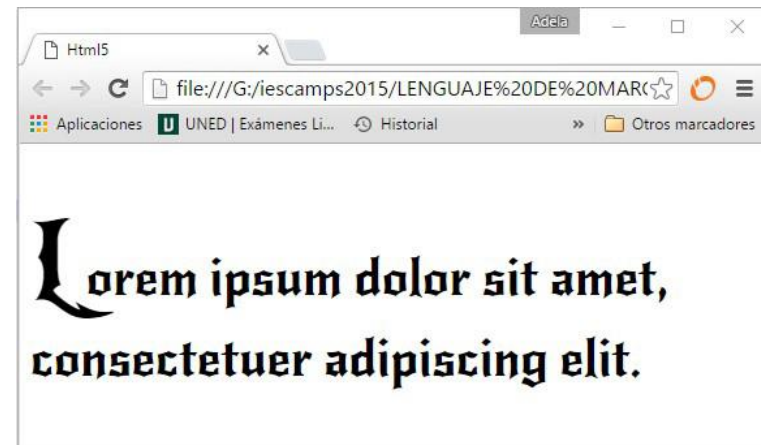
Una propiedad de estilo compatible podría ser:

```
@font-face{ font-family: 'mitipografia';  
src: url('rocker.eot') format('embedded-opentype'),  
url('rocker.woff') format('woff'),  
url('rocker.ttf') format('truetype');}
```

Esto sin contar con que Internet Explorer 9 tiene ciertas dificultades con el atributo src cuando se hace referencia a un tipo de letra eot y después a un tipo de letra ttf, o woff. El problema se ha resuelto con Internet Explorer 10

5. Tipos de letra personalizados.

```
DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
@font-face{
font-family: 'rocker';
src: url('rocker.eot');
src: url('rocker.eot?#iefix')
format('embedded-opentype'),
url('rocker.woff') format('woff'),
url('rocker.ttf') format('truetype');}
.police { font: 40px 'rocker', fantasy;}
p:first-letter { font-size:200%;}
</style>
</head>
<body>
<p class="police">
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
</p>
</body>
</html>
```



6. Tipos de letra personalizados.

Para obtener una fuente de caracteres en los diferentes formatos existen, afortunadamente, sitios dedicados a esta función. Por ejemplo :

www.font2web.com o www.fontsquirrel.com/fontface/generator.

.

Existe otra forma de utilizar tipos de letra particulares en una página Web sin tener que recurrir a @font-face y que es perfectamente compatible con los distintos navegadores. Consiste en utilizar Google Font Api. Las tipografías proporcionadas por este servicio están perfectamente adaptadas a la visualización en un navegador y tienen un peso reducido.

Basta con agregar una hoja de estilo a su página Html para obtener la nueva tipografía.

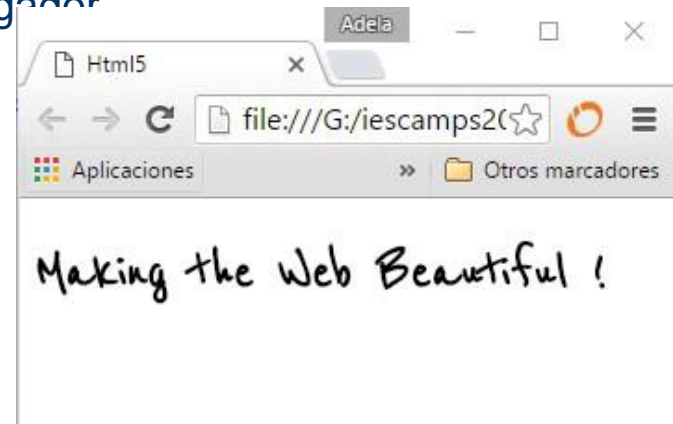
```
<link  
href='http://fonts.googleapis.com/css?  
family=Reenie+Beanie&subset='latin'  
rel='stylesheet' type='text/css'>
```

A continuación, basta con aplicar la tipografía en una clase CSS mediante la propiedad **font-family**.

7. Tipos de letra personalizados.

No es preciso, entonces, pasar por la propiedad @font-face, ni tener que ocuparse del formato del tipo de letra. Todo esto lo gestiona Google para asegurar la mejor visualización en cada navegador

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<link
href='http://fonts.googleapis.com/css?family=Reenie+Beanie&subset=
latin' rel='stylesheet' type='text/css'>
<style>
h1 { font-family: 'Reenie Beanie', arial, serif; }
</style>
<body>
<h1>Making the Web Beautiful !</h1>
</body>
</html>
```



8. Tipos de letra personalizados.

Para encontrar tipos de letra adaptados a la Web, es posible realizar una búsqueda en Google con la palabra clave **webfont** y obtendremos la dirección de numerosos sitios especializados. Por ejemplo www.fontsquirrel.com/

No olvides que puede existir un copyright sobre la tipografía y que, algunas de ellas, pueden requerir un pago.

9. Las letras huecas.

Durante mucho tiempo la tipografía ha sido el pariente pobre de las páginas Web. Con la propiedad CSS3 text-stroke, ya puede mostrar letras huecas. Esta propiedad está disponible con el prefijo -webkit- en Google Chrome 4.0, Safari3.1+, iOS Safari 4.0+ y Android Browser 2.1+.

<code>text-stroke-width: valor;</code>	Grosor del borde de las letras (en pixels).
<code>text-stroke-color: color;</code>	Color del borde de las letras.
<code>text-stroke: valor color;</code>	Contracción de las dos propiedades anteriores.
<code>text-fill-color: color;</code>	Color de rellenos de las letras (Se permite el valor transparent).

10. Las letras huecas.

```
h1 { -webkit-text-stroke: 1px black;  
-webkit-text-fill-color: transparent;}
```

Es posible visualizar los diferentes parámetros en la siguiente url:

www.westciv.com/tools/textStroke/

```
<!DOCTYPE html>  
<html lang="es">  
<head>  
<title>Html5</title>  
<meta charset="UTF-8">  
<style type="text/css">  
.stroke { font-size: 72px;  
-webkit-text-stroke: 2px #000;  
-webkit-text-fill-color: transparent;}  
</style>  
</head>  
<body>  
<div class="stroke">Ediciones ENI</div>  
</body>  
</html>
```



11. Las letras huecas.

Algunos, sin embargo, pueden lamentar la falta de compatibilidad de este efecto. Por otra parte, es posible utilizar un truco para reproducir este efecto de letras huecas en Firefox, Opera, Internet Explorer 10 y, por supuesto, Chrome así como Safari.

Consiste en utilizar la propiedad `text-shadow` con un tamaño de 1 px sin propagación (`blur`) para las cuatro posiciones, arriba a la derecha, arriba a la izquierda, abajo a izquierda y abajo a derecha.

La propiedad `color` sustituye a `text-fill-color`.

```
h1 { color: white;
text-shadow:
-1px -1px 0 #000,
1px -1px 0 #000,
-1px 1px 0 #000,
1px 1px 0 #000;}
```

12. Las letras huecas.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style type="text/css">
.compatible {
font-size: 72px;
color: white;
text-shadow:
-1px -1px 0 #000,
1px -1px 0 #000,
-1px 1px 0 #000,
1px 1px 0 #000;}
</style>
</head>
<body>
<div class="compatible">EdicionesENI</div>
</body>
</html>
```



13. Manejar títulos largos.

La propiedad CSS `text-overflow` es una propiedad muy útil para ciertos contenidos recortados voluntariamente, como un resumen de artículo o comentarios de visitantes.

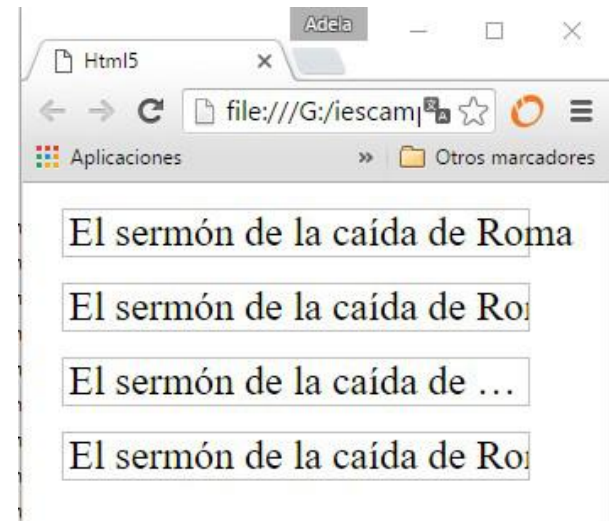
<code>text-overflow</code>	<code>clip;</code> <code>ellipsis;</code> <code>string;</code>
----------------------------	--

- El valor `clip` corta (a veces bruscamente) el texto excedente.
- El valor `ellipsis` termina el texto cortado con puntos suspensivos.
- El valor `string` termina el texto con una cadena de caracteres (string), por ejemplo ", etc.". Sólo funciona en Firefox.

14. Manejar títulos largos.

```
<!doctype html>
<html>
<head>
<title>Html5</title>
<meta charset="utf-8">
<style type="text/css">
div { margin: 15px;
font-size: 150%;
width: 272px;
padding-left: 3px;
border: 1px solid silver;}
.nada { white-space: nowrap;}
.clip { white-space: nowrap;
overflow: hidden;
text-overflow: clip;}
.ellipsis { white-space: nowrap;
overflow: hidden;
text-overflow: ellipsis;}
.string { white-space: nowrap;
overflow: hidden;
text-overflow: " , etc.";}
</style>
</head>
```

```
<body>
<div class="nada">El sermón de la
caída de Roma</div>
<div class="clip"> El sermón de la
caída de Roma </div>
<div class="ellipsis"> El sermón de la
caída de Roma </div>
<div class="string"> El sermón de la
caída de Roma </div>
</body>
</html>
```



15. Los guiones automáticos.

La propiedad CSS **hyphens** indica al navegador el método que tiene que utilizar para separar sílabas con el fin de mejorar la presentación del texto.

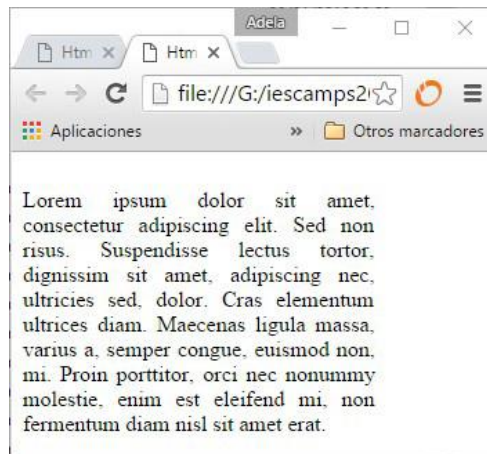
hyphens	<code>none;</code> <code>manual;</code> <code>auto;</code>
---------	--

- **none**: las palabras no se cortan, incluso si un carácter en el texto sugiere posibles puntos de corte.
- **manual**: las palabras se cortan solo a nivel de los caracteres que sugieren la posibilidad de un corte.
- **auto**: el navegador es libre de realizar los cortes en los sitios que considere apropiados, siguiendo la regla que haya escogido. Esta regla puede variar de un navegador a otro.

No todos los navegadores reconocen esta propiedad

16. Los guiones automáticos.

```
<!doctype html>
<html>
<head>
<title>Html5</title>
<meta charset="utf-8">
<style type="text/css">
.bloc { width: 255px;
text-align: justify;
-webkit-hyphens: auto;
-moz-hyphens: auto;
-ms-hyphens: auto;
hyphens: auto;}
</style>
</head>
```



```
<body>
<div class="bloc">
<br>
Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Sed non
risus. Suspendisse lectus tortor,
dignissim sit amet, adipiscing
nec, ultricies sed, dolor. Cras
elementum ultrices diam. Maecenas
ligula massa, varius a, semper congue,
euismod non, mi. Proin
porttitor, orci nec nonummy molestie,
enim est eleifend mi, non
fermentum diam nisl sit amet erat.
</div>
</body>
</html>
```

17. Redimensionar un elemento.

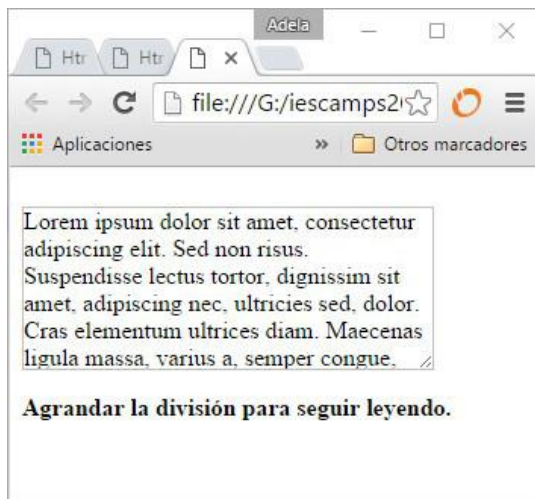
.

<code>resize</code>	<code>none;</code> <code>both;</code> <code>horizontal;</code> <code>vertical;</code>
---------------------	--

- **none:** el usuario no puede redimensionar el elemento.
- **both:** el usuario puede modificar la altura y la anchura del elemento.
- **horizontal:** el usuario puede modificar la anchura del elemento.
- **vertical:** el usuario puede modificar la altura del elemento.

18. Redimensionar un elemento.

```
<!doctype html>
<html>
<head>
<title>Html5</title>
<meta charset="utf-8">
<style type="text/css">
.bloque { resize:vertical;
overflow:hidden;
width:270px; height:60px;
border:1px solid silver;}
</style>
</head>
```



```
<body>
<br>
<div class="bloque">
Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Sed non
risus. Suspendisse lectus tortor,
dignissim sit amet, adipiscing
nec, ultricies sed, dolor. Cras
elementum ultrices diam. Maecenas
ligula massa, varius a, semper congue,
euismod non, mi. Proin
porttitor, orci nec nonummy molestie,
enim est eleifend mi, non
fermentum diam nisl sit amet erat.
</div>
<p><b>Agrandar la división para seguir
leyendo.<b></P>
</body>
</html>
```