

## Digital Guide

Productos IONOS

# Less CSS: presentación y tutorial



Equipo editorial de IONOS

11.6.2019



Índice

A cualquiera que pretenda programar o diseñar páginas web modernas le son indispensables las [CSS](#). El lenguaje de las hojas de estilo, que al igual que HTML es uno de los lenguajes clave de la World Wide Web, permite **separar de forma limpia el contenido de una página web de su formato gráfico**. Así, características como la disposición, los colores o la tipografía pueden modificarse en cualquier momento sin tener que alterar para ello todo el código fuente. Cuanto más grande sea un proyecto web, más **complejas e inabarcables** se vuelven sus **hojas de estilo** o stylesheets, es decir, los archivos que contienen las directrices de formato gráfico en código CSS, y, con ellas, el trabajo con el lenguaje informático. Es aquí donde Less, preprocesador de CSS, entra en acción.

## ¿Qué es Less?

[Less](#) (**Leaner Style Sheets**) es una ampliación retrocompatible del lenguaje CSS o, en otras palabras, un preprocesador del lenguaje de hoja de estilo. Eso quiere decir que cualquier código CSS es automáticamente también un código Less válido (pero no al revés). El objetivo de Less es **hacer más eficiente la escritura en código CSS**. Para ello, este lenguaje influido por [SASS](#) ofrece diferentes complementos a los comandos CSS tradicionales como, por ejemplo, variables y funciones, las cuales, entre otras cosas, **facilitan el trabajo con las hojas de estilo** y permiten **prescindir de la engorrosa duplicación de códigos**.

CSS Preprocessors in 60 Seconds



## Digital Guide

de estilo, así como del proceso de compilación, es **JavaScript-Tool Less.js**, lo cual le da a Less la ventaja de poder ser **compilado** tanto **del lado del servidor** como **del cliente** (en el navegador).

## Less versus CSS: ¿cuál es la diferencia?

Tanto CSS como Less son considerados lenguajes de hoja de estilo. Se trata pues, en ambos casos, de **lenguajes formales** que determinan el aspecto de interfaces o documentos. Para ello, las hojas de estilo con las **directrices de formato** tienen que estar referidas tan solo a los elementos HTML de una página web, siendo el navegador quien se encarga de analizarlos. La diferencia esencial entre Less y CSS es que **CSS** es un **lenguaje estático**, mientras que **Less** es un **lenguaje dinámico** y, por lo tanto, cuenta con elementos dinámicos como variables, operaciones, funciones, mixins o anidamientos, que no se encuentran en CSS.

De este modo, mientras que los comandos en CSS tienen que seguir un orden concreto, Less ofrece a los desarrolladores mucha más **flexibilidad**: se pueden definir, por ejemplo, **reglas personalizadas** para un tipo concreto de elementos que se apliquen a toda la hoja para, de este modo, no tener que realizar tediosas repeticiones de código. También la sintaxis es diferente en ambos lenguajes. La sintaxis de Less puede ser considerada, básicamente, como una **metasintaxis de CSS**, ya que los códigos CSS válidos también lo son en Less con la misma semántica.

## Del lado del cliente o del servidor: ¿cómo se usa Less?

Si has elegido Less para tu proyecto, tienes dos opciones: usar el navegador que quieras para compilar las Less stylesheets con ayuda de **Less.js** por el lado del cliente; o instalar la **aplicación de JavaScript** en la computadora de desarrollo y traducirlo allí con ayuda de Less.js y del sistema de tiempo en ejecución de JavaScript [Node.js](https://nodejs.org/) a través de la línea de comandos.

## Less CSS: uso del lado del cliente

El uso de Less del lado del cliente es la forma más rápida y fácil para **trabajar con el lenguaje de las hojas de estilo**. Sin embargo, no se recomienda esta opción para el posterior entorno live, ya que la compilación adicional de Less a CSS causaría claras

## Digital Guide

Para compilar Less en el navegador, indica primero en el documento en cuestión que deseas usar Less stylesheets (hojas de estilo con la extensión `.less`). Para ello, integra la extensión mediante el atributo `rel/«stylesheet/less»`:

```
<link rel="stylesheet/less" type="text/css" href="st
```

Luego descarga la **versión actual de Less.js**, que puedes encontrar, por ejemplo, en el directorio oficial [GitHub](#) del preprocesador CSS. La herramienta JavaScript la integra entonces en la cabecera ( `<head>` ) del proyecto:

```
<script src="less.js" type="text/javascript"></scrip
```

### ⚠ Nota

Es importante integrar primero la hoja de estilo y luego el script, ¡de lo contrario pueden surgir problemas de procesamiento!

## Uso de Less CSS del lado del servidor

Less también se ejecuta rápida y fácilmente en la computadora de desarrollo. En lo que al sistema operativo se refiere, se puede elegir: el preprocesador CSS funciona en Windows, macOS y UNIX/Linux, siempre y cuando el **sistema de tiempo en ejecución de JavaScript Node.js** mencionado arriba esté **instalado**.

Descárgate la versión actual para tu sistema en la [web de Node.js](#) e instálala. Con npm, el gestor de paquetes del sistema de tiempo en ejecución de JavaScript, instala luego el lenguaje de hojas de estilo a través de la línea de comandos:

```
npm install -g less
```

## Digital Guide

```
lessc example.less example.css
```

## Less: tutorial con ejemplos sobre las características más importantes

Si a menudo se trabaja con CSS, aprender a manejar Less merece siempre la pena. Less no solo ofrece la posibilidad de integrar elementos dinámicos en los códigos de la hoja de estilo, sino también la de ahorrar mucho tiempo y esfuerzo. Para ello, eso sí, hay que observar primero las propiedades de esta ampliación de CSS. Y es que, para poder redactar hojas de estilo en Less, hay que conocer las **bases de su sintaxis**. En este breve tutorial de Less te mostramos, con prácticos ejemplos de Less CSS, las características más importantes, incluyendo la notación correspondiente.

### Variables

Uno de los puntos fuertes de Less es la posibilidad de crear variables, al igual que en otros lenguajes de programación. Dichas variables pueden contener **cualquier tipo de valores**, aunque los más relevantes son los que se usan más a menudo en la hoja de estilo en cuestión. Así, se suelen utilizar variables para definir de forma centralizada **colores, tipografías, dimensiones** (altura, anchura, profundidad), **selectores o URL**, así como sus variaciones (más claro, más oscuro, etc.). Los valores elegidos pueden usarse luego en cualquier punto de la hoja de estilo, de forma que los cambios globales solo tienen que introducirse en una única línea del código.

En el siguiente fragmento de código, por ejemplo, **se definen dos variables**: una para el **color de fondo** (*@background-color*) y otra para el **color del texto** (*@text-color*). Ambas contienen códigos hexadecimales:

```
// Less
@background-color: #ffffff;
@text-color: #000000;
p{
  background-color: @background-color;
  color: @text-color;
  padding: 15px;
}
ul{
```

## Digital Guide

```
}
```

El color de fondo, en este caso el blanco, se aplica tanto al texto en bloque (p) como a las listas sin numerar (ul). Como color de texto se ha escogido el negro, que se aplica al texto en bloque y a los elementos de las listas (li). Si se quiere **modificar** esta configuración de colores, y, por ejemplo, usar texto blanco sobre fondo negro en las listas y los párrafos, basta con **intercambiar los valores de las dos variables**. En una hoja de estilo CSS convencional habría que cambiar los valores de todos los elementos de uno en uno. Tras la compilación a CSS, el código adquiere la siguiente forma:

```
/* CSS */
p{
background-color: #ffffff;
color: #000000;
padding: 15px;
}
ul{
background-color: #ffffff;
}
li{
color: #1A237E;
}
```

## Mixins

Los mixins funcionan con un principio similar al de las variables. En este caso, en cambio, no se definen valores sueltos, sino clases enteras de manera centralizada, incluyendo los valores que contienen, para luego poder **transferirlos a otras clases en la** Less stylesheet. Además, los mixins también pueden funcionar como funciones y **aceptar parámetros** (también con valores por defecto). Un ejemplo de ello es este mixin para redondear las esquinas (*.rounded-corners*), que se aplica tanto a la cabecera (*#header*) como al pie de página (*#footer*). Mientras que para la cabecera se toma el valor predeterminado, el *#footer* aplica al *mixin* un valor propio (10px):

```
// Less
.rounded-corners (@radius: 5px) {
border-radius: @radius;
-webkit-border-radius: @radius;
-moz-border-radius: @radius;
```

## Digital Guide

```
#footer {  
  .rounded-corners(10px);  
}
```

La CSS compilada a partir de estas líneas de código Less tiene la siguiente forma:

```
/* CSS */  
#header {  
  border-radius: 5px;  
  -webkit-border-radius: 5px;  
  -moz-border-radius: 5px;  
}  
#footer {  
  border-radius: 10px;  
  -webkit-border-radius: 10px;  
  -moz-border-radius: 10px;  
}
```

## Anidamientos

Para lograr herencias en CSS hay que teclear largos y costosos selectores. En Less, en cambio, se pueden **anidar selectores unos dentro de otros**, tantos como se quiera. Esto, por un lado, ahorra esfuerzo y, por otro, aporta más claridad a la estructura de la stylesheet. También para esta función del preprocesador CSS tenemos un ejemplo:

```
// Less  
#header {  
  h1 {  
    font-size: 26px;  
    font-weight: bold;  
  }  
  p {  
    font-size: 12px;  
    a {  
      text-decoration: none;  
      &:hover {  
        border-width: 1px  
      }  
    }  
  }  
}
```

Los selectores *p*, *a* y *:hover* en este caso han sido enlazados en la **Less stylesheet**, lo cual facilita mucho

## Digital Guide

```
/* CSS */
#header h1 {
  font-size: 26px;
  font-weight: bold;
}
#header p {
  font-size: 12px;
}
#header p a {
  text-decoration: none;
}
#header p a:hover {
  border-width: 1px;
}
```

## Operadores

También las operaciones aritméticas de suma (+), resta (-), multiplicación (\*) y división (/) se pueden usar en las Less stylesheets aplicando el operador correspondiente a cualquier valor numérico o de color. Así, con poco esfuerzo, se pueden establecer relaciones **entre los valores de los diferentes elementos**, relaciones que se mantienen aunque se modifiquen los valores iniciales. En caso de que la operación dictada por el operador no se pueda realizar o no tenga sentido en ese caso, el operador será ignorado automáticamente: por ejemplo, si se tratase de sumar un valor en píxeles con uno en centímetros. El siguiente ejemplo muestra las distintas posibilidades para operar en Less:

```
// Less
@the-border: 1px;
@base-color: #111;
#header {
  color: (@base-color * 3);
  border-left: @the-border;
  border-right: (@the-border * 2);
}
#footer {
  color: (@base-color + #003300);
}
```

Las definiciones de base para los **bordes** (1px) y el **color de base** (#111), que es un tono negro, también se aplican a la cabecera y al pie de página, de manera que los valores de base son modificados por tres operadores:

## Digital Guide

- operador de multiplicación \* 2, por lo que es el doble de ancho que el marco estándar (2px).
3. El color base del pie de página también es modificado con un operador: el valor hexadecimal #003300 **se suma** al valor de base #111, de manera que el pie de página adquiere un tono verde oscuro (#114411).

En el código CSS compilado los resultados de las operaciones también son sorprendentes:

```
/* CSS */
#header {
    color: #333;
    border-left: 1px;
    border-right: 2px;
}
#footer {
    color: #114411;
}
```

## Funciones

Less amplía CSS también con funciones que abren todo un abanico de posibilidades. Por ejemplo, en una *Less stylesheet* se pueden diseñar, por un lado, complejas **relaciones lógicas** con la función *SI (IF function)* o con una función booleana; o bien funciones no menos complejas para **cálculos matemáticos** como el coseno, el seno o la tangente. También se pueden usar, por otro lado, funciones simples para **definir el color** rápidamente (rgb, rgba, hsv etc.) o funciones con **operadores de color** como el contraste (*contrast*), la saturación (*saturate*, *desature*) o la claridad (*lighten*, *darken*). Para aumentar o disminuir la saturación de un elemento, por ejemplo, tan solo se necesita un valor de color y la función *saturate*. Luego puedes indicar, en forma de porcentaje (0-100%), cuánto quieres modificar la saturación:

```
// Less
@red: #842210;
#header {
    color: saturate(@red, 20%); ->#931801
}
#footer {
    color: desaturate(@red, 10%); ->##7d2717
}
```



## Digital Guide

**aumento del 20 %**, mientras que el código Less **disminuye** la saturación del pie de página **un 10 %**. En la hoja de estilo CSS, tanto las funciones como la variable de color (@red) se han transformado, por lo que solo se pueden ver los códigos hexadecimales junto con los niveles de saturación correspondientes:

```
/* CSS */
#header {
  color: #931801
}
#footer {
  color: #7d2717
}
```

## Less CSS: menos trabajo, más posibilidades

Nuestro breve tutorial para principiantes solo muestra una pequeña parte de las posibilidades que hacen del preprocesador CSS una herramienta tan útil. Una vez se han definido las variables, los mixins, etc., estos pueden **aplicarse en cualquier momento a nuevos elementos de la hoja de estilo** sin tener que empezar de cero, como suele ser el caso con el código CSS. Si se modifican valores como el color de base, las modificaciones se pueden introducir fácilmente en un documento Less con apenas unos clics, lo cual convierte al preprocesador CSS en una gran ayuda a la hora de realizar el proyecto a largo plazo de crear tu propia página web.

### Consejo

Encontrarás información más detallada sobre las características de Less en el manual online y en la página en inglés del proyecto: [guía en profundidad en lesscss.org](#).

¿Le ha resultado útil este artículo?



## Digital Guide

Productos  
asociados

Crear una  
página web

**Ver tarifas**

Artículos  
Favoritos

Cómo crear un correo  
con dominio propio

¿Cómo crear un  
correo con dominio  
propio? Transmite  
profesionalidad y  
favorece la...

[Leer más →](#)

¿Cómo comprar un  
dominio?

¿Cómo registrar un  
dominio web con los  
dominios de primer y  
segundo nivel  
deseados? ¿Cómo...

[Leer más →](#)

## Digital Guide

dominios hay en Internet? ¿Qué diferencia hay entre los dominios de primer...

[Leer más →](#)

**Prompt engineering: definición, ejemplos y buenas prácticas**

¿Qué es prompt engineering?, ¿cómo obtener mejores resultados de ChatGPT y otras...

[Leer más →](#)

**7 tipos de páginas web: ¿qué sitio web necesitas?**

Dentro de los tipos de página web, elegir un formato adecuado es fundamental para el éxito...

[Leer más →](#)

## Artículos similares

©

### Aprende SASS: tutorial definitivo con ejemplos de código

Si te gusta diseñar sitios web, pero CSS a veces te vuelve loco, SASS puede ser la solución para ti. El preprocesador facilita el trabajo en el diseño,...

[HTML](#)   [CSS](#)

[Leer más →](#)

## Digital Guide

## Cómo comprimir CSS y mejorar el rendimiento de una web

Encontrar elementos estructurales directamente en el código HTML no es muy común. Colores, fuentes y tamaños de otros elementos HTML s...

[Leer más →](#)

©

## Trucos CSS: snippets útiles para tu proyecto web

Actualmente, el lenguaje de las hojas de estilo CSS es una de las herramientas de diseño web más importantes. Este permite separar el...

CSS    Tutoriales

[Leer más →](#)

©

## Shadow DOM: subunidad DOM para bloques de códigos encapsulados

HTML, CSS y JavaScript se han asentado en el desarrollo web moderno. Pero al diseñar o reutilizar bloques de códigos por separado, los...

## Digital Guide

## Git: tutorial básico del sistema de control de versiones

Aunque Git fue creado por el equipo de Linux para cubrir sus propias necesidades, este sistema de control de versiones se convirtió...

**Tutoriales**   **Git**

[Leer más →](#)

[Sobre IONOS](#)

[Sala de prensa](#)

[Startup Guide](#)

[Centro de ayuda](#)

[Condiciones Generales](#)

[Política de privacidad](#)

[Tu solución digital](#)

---

© 2025 IONOS Cloud S.L.U.