

LESS - Básico



Diogo Antonio Sperandio Xavier



SUMÁRIO

[INTRODUÇÃO](#)

[INSTALANDO](#)

[Less na versão JavaScript.](#)

[Instalando o Grunt - OPCIONAL](#)

[LESS](#)

[Variáveis](#)

[Seletores aninhados](#)

[Mixins](#)

[Mixins com argumentos / Mixins dinâmicos](#)

[Herança](#)

[Cores](#)

[Números](#)

[Condicionais e estruturas de controle](#)

[Import files](#)

[Formatação de saída](#)

[Comentários](#)

[Namespace](#)

[Extend](#)

[Media](#)

[SITES DE APOIO](#)

INTRODUÇÃO

Apostila com o objetivo de auxiliar no início dos trabalhos com Less. Para informações mais completas, recomendo checar a [documentação](#). Espero que esse pequeno guia possa auxiliar a recordar comandos e passos, comandos e acima de tudo, o aprendizado.

INSTALANDO

Less na versão JavaScript.

Para instalar o Less, é necessário ter o node instalado. Para checar basta abrir o prompt de comando e digitar:

```
node -v
```

caso possua o node, a resposta será:

```
C:\Users\diogo.xavier>node -v  
v14.15.1
```

Após isso, para instalar o less, digite:

```
npm install -g less
```

A resposta será similar a:

```
C:\Users\diogo.xavier\AppData\Roaming\npm\lessc -> C:\Users\diogo.xavier\AppData\Roaming\npm\node_modules\less\bin\lessc+ less@4.1.1  
updated 1 package in 1.472s
```

Em seguida, confira a versão do less.

```
lessc --v
```

A resposta será similar a:

```
C:\Users\diogo.xavier>lessc -v  
lessc 4.1.1 (Less Compiler) [JavaScript]
```

Se ainda ficou alguma dúvida, visite o site do [less](https://lesscss.org/) para obter mais informações.

Instalando o Grunt - OPCIONAL

Primeiramente vale dizer que o grunt é apenas um facilitador, pois compila seu arquivo.less a cada salvamento, mas não é obrigatório para a funcionalidade. O grunt tem muitas outras funcionalidades que podem ser vistas no site do [grunt](#).

Para instalá-lo execute no prompt os seguintes comandos:

```
npm install -g grunt
```

em seguida:

```
npm install -g grunt-cli
```

e:

```
npm install grunt grunt-contrib-less grunt-contrib-watch jit-grunt --save-dev
```

Após isso, certifique-se de ter esses dois arquivos [deste link](#):

0 primeiro: gruntfile.js

```
module.exports = function(grunt) {  
  require('jit-grunt')(grunt);  
  
  grunt.initConfig({  
    less: {  
      development: {  
        files: {  
          "css/estilos.css": "css/estilos.less" // where files are  
        }  
      }  
    },  
    watch: {  
      styles: {  
        files: ['**/*.less'], // files that grunt will looking for  
        tasks: ['less']  
      }  
    }  
  });  
};
```

```
grunt.registerTask('default', ['less', 'watch']);  
};
```

0 segundo: package.json

```
{  
  "name": "site-name",  
  "version": "1.0.0",  
  "main": "Gruntfile.js",  
  "scripts": {  
    "test": "echo \"Error: no test specified\" && exit 1"  
  },  
  "author": "",  
  "license": "ISC",  
  "description": "",  
  "devDependencies": {  
    "grunt": "^0.4.5",  
    "grunt-contrib-less": "^1.0.1",  
    "grunt-contrib-watch": "^0.6.1",  
    "jit-grunt": "^0.9.1"  
  }  
}
```

Em seguida, execute no terminal digitando “grunt”.

```
C:\Users\diogo.xavier\Desktop\models\grunt-for-less>grunt_
```

Se possuir alguma dúvida, acesse o site do [grunt](#).

LESS

A cada mudança nos arquivos.less é necessário ser efetuada a **compilação** para gerar o arquivo CSS que será **linkado ao head do seu HTML**.

Pelo prompt, acesse a pasta que tem seus arquivos.less e execute o comando:

```
lessc style.less style.css
```

No head de seu arquivo HTML, utilize normalmente o link do arquivo CSS que será gerado.

Para utilizar o less, deve-se criar um arquivo style.less e efetuar as alterações do CSS nele.

Variáveis

```
@color: red;
div {
  color: @color;
}
```

Suporte à definição de variável com escopo dentro de um contexto de seletor:

input:

```
@color: black;
.scoped {
  @bg: blue;
  @color: white;
  color: @color;
  background-color: @bg;
}
.unscoped {
  color: @color;
  // Would be an error
  // background: @bg;
}
```

Output:

```
.scoped {  
  color: white;  
  background-color: blue;  
}  
.unscoped { color: black; }
```

Variáveis devem ser declaradas no **contexto global**, caso contrário, funcionarão apenas dentro do escopo em questão.

Seletores aninhados

Less tem o seletor `&` que permite que o seletor aninhado se refira ao escopo do pai.

```
p {  
  a {  
    color: red;  
    &:hover {  
      color: blue;  
    }  
  }  
}
```

Mixins

```
.bordered {  
  border-top: dotted 1px black;  
  border-bottom: solid 2px black;  
}  
  
#menu a {  
  .bordered;  
}
```


Mixins com argumentos / Mixins dinâmicos

Ex1:

```
.bordered(@width: 2px) {  
  border: @width solid black;  
}  
  
#menu a {  
  .bordered(4px);  
}
```

Ex2: (usamos @ border-radius: 0.3em para definir um valor base, mas se você quiser outro valor, basta fazer: .rounded-edge (2em), por exemplo)

```
.rounded-edge(@border-radius: 0.3em){  
  -webkit-border-radius: @border-radius;  
  -o-border-radius: @border-radius;  
  border-radius: @border-radius;  
}
```

Ex3:

```
.standard-shadow {  
  box-shadow: 0 2px 6.65px 0.35px rgba(0, 0, 0, 0.3);  
}
```

Ex4:

```
.image-replacement {  
  text-indent: -9999px;  
  overflow: hidden;  
  background-repeat: no-repeat;  
}
```

Herança

Arquivo Less

```
header {  
  height: auto;  
  
  .container {  
    position: static;  
  }  
  h1 {  
    max-width: 50%;  
    margin: 0 auto;  
    img {  
      max-width: 100%;  
      margin: .5em auto;  
      display: block;  
    }  
  }  
}
```

Arquivo CSS

```
header {  
  height: auto;  
}  
  
header .container {  
  position: static;  
}  
  
header h1 {  
  max-width: 50%;  
  margin: 0 auto;  
}  
  
header h1 img {  
  max-width: 100%;  
  margin: 0.5em auto;  
  display: block;  
}
```

- OBS: Se você usar, por exemplo, `a: hover` ou `a: last-child`, você aplica a mesma lógica, mas com `&`.

EX:

```
...
&:last-child {
  border: 0;
}
```

Cores

O Less possui diversas funções para trabalhar com cores. Abaixo alguns exemplos:

1 - propriedade `fade` (algo tipo desvanecer):

```
background: fade(@color, 80); // fade the color in 80%
```

2 - propriedade `lighten` (clarear):

```
background: lighten(@cor-padrao, 5); // lightens the color in 5%
```

3 - propriedade `darken` (escurecer):

```
background: darken(@cor-padrao, 5); //darkens the color by 5%
```

Mais funções de cores na [documentação](#) referente.

Números

O less suporta números operações aritméticas, por exemplo:

```
1cm * 1em => Error
2in * 3in => 6in
(1cm / 1em) * 4em => Error
2in + 3cm + 2pc => Error
3in / 2in => 1.5in
```

Para resolver alguns problemas com dispositivos e compatibilidades, acesse o site do [can i use](#) e a [documentação referente a funções](#) do less para mais informações.

Condicionais e estruturas de controle

Less não fornece nenhuma estrutura condicional ou de loop. Em vez disso, ele fornece *protetores de mixin* e *correspondência de padrões* que podem ser usados para efeitos semelhantes.

```
.mixin (@color) when (lightness(@color) > 30%) {  
  background-color: black;  
}  
.mixin (@color) when (lightness(@color) =< 30%) {  
  background-color: white;  
}
```

O Less oferece suporte a looping por recursão, mas não a interpolação de seletor, portanto, é de uso limitado.

Import files

Less importará outros arquivos less.

Para organizar corretamente seus arquivos, você precisa dividir seus arquivos. A menos em mais de um arquivo, como em css. Para importar um arquivo less para outro basta usar o comando:

```
@import "fileName.less"; //with .less  
@import "anotherFileName; /without .less, both works
```

OBS: Você pode ou não utilizar a extensão .less, pois o comando dentro de um arquivo .less já considera tal extensão.

Formatação de saída

Less tem três formatos de saída: Normal, compactado ou yui-compressed.

Comentários

Less suporta o estilo (``/* */``) e (``//``).

Comentários do tipo (``//``) **não** serão levados para o CSS após compilados.

Namespace

Less:

```
#bundle () {  
  .red { background-color: red }  
  .green { background-color: green }  
}  
  
.foo {  
  #bundle > .red;  
}
```

CSS:

```
.foo {  
  background-color: red;  
}
```

Extend

Se você tiver o mesmo código para mais de uma classe, por exemplo, você pode usar o `extend`. Quando você compila menos arquivos, a propriedade `extend` será removida pelo bloco e organizada como no exemplo. Seu código será mais fácil de ler e analisar, porque seu código não é repetitivo, mas não permite que você defina parâmetros diferentes.

Ex:

Less

```
.contact button {  
  (another properties)  
  &:extend(.sombra-padrao);  
}
```

CSS

```
.standard-shadow,  
.spotlight button,  
.plan button,  
.contact button {  
  box-shadow: 0 2px 6.65px 0.35px rgba(0, 0, 0, 0.3);  
}
```

Media

Less

```
header .container {  
  position: relative;  
  
  @media (max-width: 980px) {  
    position: static;  
  }  
}
```

CSS

```
header .container {  
  position: relative;  
}  
@media (max-width: 980px) {  
  header .container {  
    position: static;  
  }  
}
```

SITES DE APOIO

1. <https://lesscss.org/>
2. <https://gruntjs.com/>
3. <https://lesscss.org/functions/#color-channel>
4. <https://caniuse.com/calc>
5. <https://lesscss.org/functions/#math-functions>