



Curso Less

# O que é LESS

- O pré-processador LESS foi criado em 2009 por Alexis Sellier e escrito em Ruby mas logo em seguida foi portado para JavaScript. A grosso modo, o LESS estende o CSS, fornecendo mecanismos disponíveis em linguagens de programação mais tradicionais que não estão disponíveis no CSS, através destes mecanismos, como variáveis, loops, funções e etc, é possível produzir o código de forma rápida, com manutenção fácil, prática e de baixo custo.
- A implementação oficial do LESS é open-source e codificada em JavaScript, no entanto, existem várias implementações em outras linguagens como Python, Java, PHP, JavaScript e muitas outras. Neste curso utilizaremos o módulo lessc do Node, para compilar nossos arquivos LESS.

# instalação - lessc

```
npm install less
```

# .less

```
@branco: #FFF;  
body{  
    color: @branco  
}
```

```
body {  
    color: #FFF;  
}
```

# “compilando” alterações

```
lessc exemplo1.less exemplo1.css
```

# selectors

```
nav {  
    width: 100%;  
    background: #000;  
}
```

```
nav ul {  
    list-style: none  
}
```

```
nav li {  
    color: #fff;  
    display: inline-block;  
}
```

# selectors

```
nav {  
  
    width: 100%;  
    background: #000;  
  
    ul {  
  
        list-style: none  
    }  
  
    li {  
  
        color: #fff;  
        display: inline-block;  
    }  
  
}
```

use com moderação!



# propriedades aninhadas

```
nav {  
    font-family: "Times New Roman", Times, serif;  
    font-style: italic;  
    font-size: 30px;  
    font-weight: bold;  
}
```

# propriedades aninhadas

```
nav {  
  font: {  
    family: "Times New Roman", Times, serif;  
    style: italic;  
    size: 30px;  
    weight: bold;  
  }  
}
```

# pseudo-elements

```
a {  
    color: #ffa;  
  
}  
  
a:hover {  
    color: #faf;  
}
```

# pseudo-elements

```
a {  
    color: #ffa;  
  
    &:hover {  
        color: #faf;  
    }  
  
}
```

# variables

```
@width: 100%;  
@font-size: 1.2rem;  
@background: #962acd;  
@color: #fff;  
@border-style: solid 1px @color;
```

```
nav {  
  background: @background;  
  color: @color;  
  width: @width;  
  font-size: @font-size;  
}
```

# variables - boolean e null

```
@color-light: false;
```

```
@color-dark: true;
```

```
@color-dark: null;
```

# variables - listas

```
@animals: puma, sea-slug, egret, salamander;
```

```
@animals: puma sea-slug egret salamander;
```

```
@animals: "puma", "sea-slug", "egret", "salamander";
```

# variables - maps

```
@social: (  
  facebook: #3b5998,  
  flickr: #0063db,  
  github: #4183c4,  
  googleplus: #dd4b39,  
  instagram: #517fa4,  
  linkedin: #007bb6,  
  pinterest: #cb2027,  
  soundcloud: #f60,  
  twitter: #00aced,  
  youtube: #b00  
);
```



interpolation

# selectors and property names

```
@name: destaque;
```

```
@attr: border;
```

```
p.~'@{$name}{  
  ~'@{$attr}-color: blue;  
}
```

```
p.destaque {  
  border-color: blue;  
}
```

# property values

```
p {
```

```
  @font-size: 12px;
```

```
  @line-height: 30px;
```

```
  font: ~'@{$font-size}/~'@{$line-height};
```

```
}
```

```
p {
```

```
  font: 12px/30px;
```

```
}
```

# extend

```
.msg {  
  border: 1px solid;  
  padding: 4px;  
  text-align: center;  
}
```

```
.msg-error {  
  .msg;  
  border-color: #ff0000de;  
  color: #ff0000de;  
}
```

```
.msg-alert {  
  .msg ;  
  border-color: #ff8e00de;  
  color: #ff8e00de;  
}
```

```
.msg-error {  
  border: 1px solid;  
  padding: 4px;  
  text-align: center;  
  border-color: #ff0000de;  
  color: #ff0000de;  
}
```

```
.msg-alert {  
  border: 1px solid;  
  padding: 4px;  
  text-align: center;  
  border-color: #ff8e00de;  
  color: #ff8e00de;  
}
```

# extend

```
a:hover {  
  
    text-decoration: underline;  
  
}
```

```
.hoverlink {  
  
    @extend a:hover;  
  
}
```

```
a:hover, .hoverlink {  
  
    text-decoration: underline;  
  
}
```

# extend

```
.error {  
  border: 1px #f00;  
  background-color: #fdd;  
}
```

```
.attention {  
  font-size: 3em;  
  background-color: #ff0;  
}
```

```
.seriousError {  
  .error;  
  .attention;  
  border-width: 3px;  
}
```

```
.error {  
  border: 1px #f00;  
  background-color: #fdd;  
}
```

```
.attention {  
  font-size: 3em;  
  background-color: #ff0;  
}
```

```
.seriousError {  
  border: 1px #f00;  
  background-color: #fdd;  
  font-size: 3em;  
  background-color: #ff0;  
  border-width: 3px;  
}
```

# operators

1. Number Operations
2. Color Operations
3. String Operations
4. Boolean Operations
5. List Operations

# subtraction & negative numbers

O - pode significar muitas coisas em CSS e em Sass:

Um operador de subtração (como na `5px - 3px`)

O início de um número negativo (como em `-3px`)

Um operador de negação unária (como em `- @var`)

Parte de um identificador (`fonte-size`)



# division and /

Se o valor ou qualquer parte dele é armazenado em uma variável ou retornado por uma função.

Se o valor estiver entre parênteses, a menos que esses parênteses estejam fora de uma lista e o valor dentro.

Se o valor é usado como parte de outra expressão aritmética.

# division and /

```
p {  
  font: 10px/8px bold italic serif;  
  @width: 1000px;  
  width: @width/2;  
  width: round(1.5)/2;  
  height: (500px/2);  
  margin-left: 5px + 8px/2px;  
  font: (italic bold 10px/8px);  
  font-size: ~'@{ $base-size }/~'@{ $line-height };  
}
```

```
p {  
  font: 10px/8px;  
  width: 500px;  
  height: 250px;  
  margin-left: 9px;  
  font-size: 20px/22px;  
}
```

# color operations

Todas as operações aritméticas são suportadas para valores de cor , onde trabalham por partes. Exemplo:

```
p {  
    color: #010203 + #040506;  
}
```

01 + 04 = 05, 02 + 05 = 07 e 03 + 06 = 09:

```
p {  
    color: #050709;  
}
```

# color operations

```
p {  
    color: #010203 * 2;  
}
```

$01 * 2 = 02$ ,  $02 * 2 = 04$ , and  $03 * 2 = 06$

```
p {  
    color: #020406;  
}
```

# color operations

```
p {  
  color: rgba(255, 0, 0, 0.75) + rgba(0, 255, 0, 0.75);  
}
```

```
p {  
  color: rgba(255, 255, 0, 0.75);  
}
```

\* Cores em RGB e RGBA devem ter o mesmo valor de alfa para que as operações sejam realizadas com eles

# String operations

```
p {  
  cursor: e + -resize;  
}
```

```
p {  
  cursor: e-resize;  
}
```

# mixins

Mixins permitem definir estilos que podem ser reutilizados em toda a folha de estilo.

```
@mixin border-radius(@radius) {  
  -webkit-border-radius: @radius;  
  -moz-border-radius: @radius;  
  -ms-border-radius: @radius;  
  border-radius: @radius;  
}
```

# utilizando mixins

```
.box {  
    .border-radius(10px);  
}
```

```
.box {  
    -webkit-border-radius: 10px;  
  
    -moz-border-radius: 10px;  
  
    -ms-border-radius: 10px;  
  
    border-radius: 10px;  
}
```



# functions

## Algumas funções do LESS

ceil()

floor()

percentage()

round()

sqrt()

abs()

sin()

asin()

cos()

acos()

tan()

atan()

pi()

pow()

mod()

min()

max()

<http://lesscss.org/functions/#list-functions>

# criando uma função

```
@calc-fluid(@target, @container) {  
    return (@target / @container) * 100%;  
}
```

# usando uma função

```
div {  
    width: @calc-fluid(650px, 1000px);  
}
```

```
div {  
    width: 65%;  
}
```

# conditionals

```
@template: dark;
```

```
header {  
  when(@template: == dark) {  
    color: #fff;  
    background: #222;  
  }  
  when(@template: == light) {  
    color: #333;  
    background: #fff;  
  }  
}
```

```
header {  
  
  color: #fff;  
  background: #222;  
  
}
```

# iteration

```
.loop(@counter) when (@counter > 0) {  
  .loop((@counter - 1));    // next iteration  
  width: (10px * @counter); // code for each iteration  
}
```

```
div {  
  .loop(5); // launch the loop  
}
```

```
div {  
  width: 10px;  
  width: 20px;  
  width: 30px;  
  width: 40px;  
  width: 50px;  
}
```

# iteration

```
.generate-columns(4);
.generate-columns(@n, @i: 1) when (@i =< @n) {
  .column-@{i} {
    width: (@i * 100% / @n);
  }
  .generate-columns(@n, (@i + 1));
}
```

```
.puma-icon {
  background-image: url('/images/puma.png');
}

.sea-slug-icon {
  background-image: url('/images/sea-slug.png'); }

.egret-icon {
  background-image: url('/images/egret.png');
}

.salamander-icon {
  background-image: url('/images/salamander.png');
}
```

# @media

```
.sidebar {  
  width: 30%;  
  @media screen and (orientation: landscape) {  
    width: 40%;  
  }  
}
```

```
.container {  
  width: 90%;  
  @media screen and (orientation: landscape) {  
    width: 100%;  
  }  
}
```

```
.sidebar {  
  width: 30%;  
}  
.container {  
  width: 90%;  
}
```

```
@media screen and (orientation: landscape) {  
  .sidebar {  
    width: 40%;  
  }  
  .container {  
    width: 90%;  
  }  
}
```

# mixins + media + interpolation + content

```
@tablet-width: 768px;  
@desktop-width: 1024px;
```

```
.landscape {  
  @media screen and (orientation: landscape) {  
    @content;  
  }  
}  
.tablet {  
  @media (min-width: @tablet-width) {  
    @content;  
  }  
}  
.desktop {  
  @media (min-width: @desktop-width) {  
    @content;  
  }  
}
```



# mixins + media + interpolation + content

```
header {  
  
    padding: 3em;  
  
    @tablet {  
        padding: 2em;  
    }  
  
}
```

```
header {  
  
    padding: 3em;  
  
}  
  
@media (min-width: 768px ) {  
    header {  
        padding: 2em;  
    }  
}
```

# @import

```
@import "foo.css";           //importando um arquivo CSS
@import "foo.less";          //importando um arquivo LESS
@import "http://foo.com/bar"; //importa arquivo via http
@import @import url("http://fonts.googleapis.com/css?family=Droid+Sans"); //importa via url()
```

Obrigado!