

Módulo 17

Conhecendo o LESS

Gian Souza



Neste módulo vamos conhecer o LESS,
um pré-processador de código CSS.



Juntamente com o SASS, o LESS é um
dos pré-processadores mais populares e é
encontrado em requisitos para vagas de
desenvolvedor front-end.

Você pode consultar o código escrito
durante o módulo [clikando aqui](#).

Sintaxe



A sintaxe do LESS basicamente segue a mesma do CSS convencional, utilizando chaves e ponto e vírgula.

Um adicional que temos no LESS é o comentário em linha, onde utilizamos: //

```
body {  
    // background-color: red;  
}
```



Setup



Para configurarmos o LESS é necessário fazermos a instalação de forma global:

```
npm install --global less
```

E de forma local na pasta do módulo:

```
npm init // para inicializar um projeto do npm
```

```
npm install --save-dev less
```

Setup



Para configurarmos o LESS é necessário fazermos a instalação de forma global:

```
npm install --global less
```

E de forma local na pasta do módulo:

```
npm init // para inicializar um projeto do npm
```

```
npm install --save-dev less
```

Durante a primeira aula configuramos o package.json para expor o comando lessc.

Variáveis



Para declara uma variável no LESS
utilizamos o @, por exemplo:

```
@fontePrincipal: Roboto, sans-serif;
```

```
body {  
  font-family: @fontePrincipal;  
}
```



Variáveis



As variáveis no LESS podem ser alteradas à nível de escopo:

```
@fontePrincipal: Roboto, sans-serif;
body {
  @fontePrincipal: Helvetica;
  font-family: @fontePrincipal; // será Helvetica
}

h1 {
  font-family: @fontePrincipal; // será Roboto
}
```

Escaping



O LESS possui um recurso chamado escaping, com este recurso podemos armazenar uma string que poderá ser utilizada para construir uma regra de estilo.

```
@desktopBreakpoint: ~"(min-width: 1023px)"; // escaping
@media @desktopBreakpoint { // @media (min-width: 1023px)
  body {
    font-size: 1.2em;
  }
}
```



Mixins



Com os mixins podemos escrever porções de estilos que podemos reutilizar em diversos seletores, para escrever um mixin seguimos a estrutura:

```
.nomeDoMixin() {  
    // Estilização  
}  
  
.classe {  
    .nomeDoMixin(); // utilização  
}
```

Mixins



Um mixin pode receber argumentos, como uma função, e podemos atribuir um valor padrão para o argumento:

```
.cardDeProduto(@color, @padding: 16px) { // mixin
  color: @color;
  padding: @padding;
}

.produto {
  .cardDeProduto(#333, 8px); // uso
}
```

Maps



Com os maps podemos criar uma estrutura parecida com um objeto, onde temos um conjunto chave-valor, isso é útil para organizar dados como breakpoints e um esquema de cores para o site.



Maps



Para criar um map seguimos a estrutura:

```
@nomeDoMap: { // ou #nomeDoMap()  
  chave: valor;  
}
```

// exemplo

```
@breakpoints: { // ou #breakpoints()  
  mobile: 320px;  
  tablet: 768px;  
  desktop: 1024px;  
}
```



Maps



E para fazer uso de um map:

```
@seletor {  
    background-color: @nomeDoMap[corPrincipal];  
}  
  
// exemplo  
@media (min-width: @breakpoints[desktop]) { // 1024px  
    // ...  
}  
  
OU  
@media (min-width: #breakpoints[desktop]) {  
    // ...  
}
```



Links úteis



- [Documentação completa do LESS](#)

