

01
Vídeo 1

Transcrição

Olá, seja bem-vindo!

Vamos começar a analisar o site da **Apeperia**.

Basicamente, a página tem um cabeçalho, uma área de destaque, uma seção de "Sobre", os valores dos "Planos".

Quando o cursor do mouse passa por uma das planos, a caixa muda de cor. Observe que a cor roxa é bastante usada, podendo ser considerada como um padrão. Mais abaixo encontramos as seções de "Blog", "Contato" e o rodapé.

Iremos abrir o código do site, é possível fazer o download do projeto inicial [aqui](https://s3.amazonaws.com/caelum-online-public/less-na-pratica/site-apeperia.zip) (<https://s3.amazonaws.com/caelum-online-public/less-na-pratica/site-apeperia.zip>). Iremos abrir o `index.html` no Sublime, mas você pode usar qualquer editor de texto da sua preferência.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <title>Index</title>
    <link rel="stylesheet" href="css/normalize.css">
    <link rel="stylesheet" href="css/estilos.css">
    <link rel="stylesheet" href="css/media-queries.css">
    <link href='http://fonts.googleapis.com/css?family=Roboto+Slab:400,300,700' rel='stylesheet' type='text/css'>
</head>
<body>
    <header>
        <div class="container">
            <h1>
                
            </h1>
            <nav class="menu-opcoes">
                <ul>
                    <li><a href="#">sobre</a></li>
                    <li><a href="#">planos</a></li>
                    <li><a href="#">blog</a></li>
                    <li><a href="#">contato</a></li>
                </ul>
            </nav>
        </div>
    </header>
```

Observe que usamos `css/media-queries.css`. Ele geralmente é utilizado quando queremos que as páginas sejam flexíveis quanto mudanças de tamanho de tela.

Sobre

O que fazemos

Múltiplas plataformas

No Apeperia, nós criamos aplicativos

Podemos ver, que quando reduzimos a janela do site, o menu do cabeçalho é reposicionado. Isso significa que ele funciona bem em dispositivos móveis.

No arquivo `estilos.css`, vemos que o código as parte foram divididas com comentários. Por exemplo, veremos um trecho referente a seção de "Sobre":

```
/** Sobre **/ 
.sobre { 
background: #ebebeb; 
} 

.duas-colunas { 
-webkit-column-count: 2; 
-moz-column-count: 2; 
column-count: 2; 
columns: 1; 
}
```

Sobre

O que fazemos

No Apeperia, nós criamos aplicativos personalizados de uma maneira diferente, preocupados com o modelo atual em que um aplicativo é criado entregue e a manutenção é vista como um custo extra para o cliente, decidimos criar um novo modelo de negócios, em que você assina nosso serviço, nós criamos o aplicativo e a manutenção já é inclusa no serviço, dessa maneira você não se preocupa em ter o app sempre na versão mais atualizada.

Com equipes especializadas, equipes exclusivas para projetos especiais agora é a hora da sua empresa ter o próprio aplicativo com qualidade.

Múltiplas plataformas

Na Apeperia não temos preconceito com nenhuma plataforma, fazemos apps fantásticos para qualquer device!

Não importa se você quer um app para iOS, Android, Windows Phone ou Blackberry. Nós sabemos fazer!



Então, o designer principal pediu que alterássemos a cor principal do site, que antes era roxo (#8837CC) e agora será a seguinte: #c24e4b . Iremos fazer a substituição em algumas parte do código, primeiramente no Header :

```
/** Header **/
header {
    border-top: 5px solid #c24e4b;
    background: rgba(30, 44, 53, 0.8);
    height: 90px;
    width: 100%;
    position: absolute;
}
```

Depois no .destaque button .

```
.destaque button {
    margin-top: 1em;
    background: #c24e4b;
    border: 0;
    padding: .6em;
    font-size: 1.2em;
    -webkit-border-radius: 0.3em;
    border-radius: 0.3em;
    box-shadow: 0 2px 6.65px 0.35px rgba(0, 0, 0, 0.3);
    font-weight: bold;
}
```

Poderíamos fazer a substituição automaticamente usando o comando **Replace**, mas faremos isto manualmente nesta aula. Os seguintes trechos ainda serão alterados.

Na parte de **Planos**, vamos mudar o .plano h3 :

Primeira parte:

```
.plano h3 {
    color: #fff;
```

```
font-size: 1.8em;
background: #c24e4b;
text-align: center;
height: 2em;
padding-top: .6em;
margin-top: 0;
}
```

Segunda parte:

```
.plano button {
background: #c24e4b;
border: 0;
color: #fff;
width: 210px;
height: 38px;
font-size: 1.2em;
-webkit-border-radius: 0.3em;
border-radius: 0.3em;
box-shadow: 0 2px 6.65px 0.35px rgba(0, 0, 0, 0.3);
margin-left: 2em;
}
```

A terceira parte é referente ao botão "enviar":



No código, adicionaremos o hexadecimal `#c24e4b` dentro do botão de contato

```
.contato button {
margin: 0 auto;
background:#c24e4b;
border: 0;
color: #fff;
width: 210px;
height: 38px;
font-size:1.2em;
-webkit-border-radius: 0.3em;
border-radius: 0.3em;
box-shadow: 0 2px 6.65px 0.35px rgba(0, 0, 0, 0.3);
}
```

Se tentássemos agilizar o processo, como por exemplo, criando uma variável de `cor-padrão`, apenas estragariamos o CSS. Atualmente, não é possível fazer isso no CSS. Teríamos que recorrer a uma linguagem de **pré-processador**. No nosso caso,

usaremos o **Less**. E para criarmos uma variável, adicionaremos o `@`. Então, antes de `Geral`, vamos incluir a variável `@cor-padrao` para fazer as alterações do hex `c24e4b`.

```
@cor-padrao: #c24e4b;
```

Praticamente, seguiremos a mesma regra de declarações CSS. Mas para que a primeira linha seja aceita, teremos que renomear o arquivo e adicionar o `.less`. Agora, o nosso arquivo irá se chamar `estilos.less`.

Vamos ajustar também o nome do arquivo no `index.html`.

```
<head>
    <meta charset="UTF-8">
    <title>Index</title>
    <link rel="stylesheet" href="css/normalize.css">
    <link rel="stylesheet" href="css/estilos.less">
    <link rel="stylesheet" href="css/media-queries.css">
    <link href='http://fonts.googleapis.com/css?family=Roboto+Slab:400,300,700'
        rel='stylesheet' type='text/css'>

</head>
```

No arquivo LESS, nós alteramos o `border-top` do `header` adicionando a variável `@cor-padrao`:

```
/**Header*/
header {
    border-top: 5px solid @cor-padrao;
    background: rgba(30, 44, 53, 0.8);
    height: 90px;
    width: 100%;
    position: absolute;
}
```

Mas se testarmos visualizar a página no navegador, veremos que a variável ainda não foi aceita. Isto acontece porque o browser só aceita CSS como linguagem de estilo e não entende o LESS. Primeiramente teremos que compilar o LESS para gerar o CSS, só então o `index.html` aproveitará o arquivo. Logo, vamos retornar a referência no arquivo HTML para `estilos.css`.

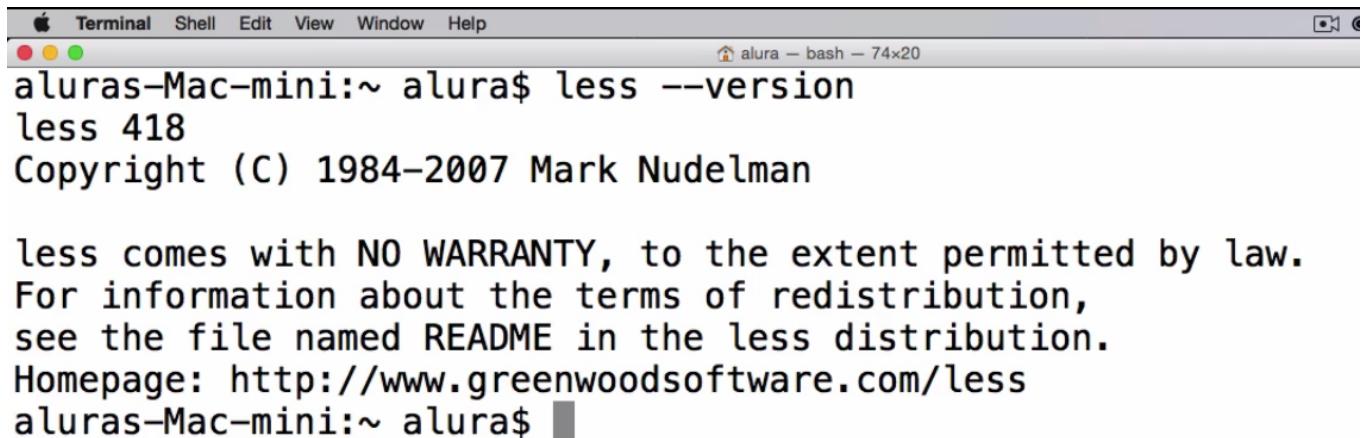
```
<head>
    <meta charset="UTF-8">
    <title>Index</title>
    <link rel="stylesheet" href="css/normalize.css">
    <link rel="stylesheet" href="css/estilos.css">
    <link rel="stylesheet" href="css/media-queries.css">
    <link href='http://fonts.googleapis.com/css?family=Roboto+Slab:400,300,700'
        rel='stylesheet' type='text/css'>

</head>
```

A parte de compilação resolveremos no Terminal. Vamos começar verificando se o Less está instalado.

```
aluras-Mac-mini:~ alura$ less --version
```

Clicamos em "Enter" e teremos o seguinte retorno:



```
Terminal Shell Edit View Window Help
alura ~ bash 74x20
aluras-Mac-mini:~ alura$ less --version
less 418
Copyright (C) 1984-2007 Mark Nudelman

less comes with NO WARRANTY, to the extent permitted by law.
For information about the terms of redistribution,
see the file named README in the less distribution.
Homepage: http://www.greenwoodsoftware.com/less
aluras-Mac-mini:~ alura$
```

Agora, precisaremos compilar o arquivo LESS. Teremos que navegar da pasta `apeperia` para `css`. No Terminal, precisaremos entrar em `Documents`. Usaremos o comando `cd` para mudar o diretório.

```
aluras-Mac-mini:~ alura$ cd Documents
aluras-Mac-mini:Documents alura$ cd natan
aluras-Mac-mini:natan alura$ ls
```

Com o `ls`, ele irá listar. Vamos entra na pasta `less`.

```
aluras-Mac-mini:~ alura$ cd Documents
aluras-Mac-mini:Documents alura$ cd natan
aluras-Mac-mini:natan alura$ ls
less
aluras-Mac-mini:natan alura$ cd less
aluras-Mac-mini:less alura$ cd Arquivos\ do\ curso/
```

Precisávamos entrar em `Arquivps do curso`, como seria trabalhoso digitar tudo isso, eu digitei apenas `Ar + Tab`. Em seguida, entraremos na `apeperia` e `css`.

```
aluras-Mac-mini:~ alura$ cd Documents
aluras-Mac-mini:Documents alura$ cd natan
aluras-Mac-mini:natan alura$ ls
less
aluras-Mac-mini:natan alura$ cd less
aluras-Mac-mini:less alura$ cd Arquivos\ do\ curso/
aluras-Mac-mini:Arquivos do curso alura$ ls
apeperia
aluras-Mac-mini:Arquivos do curso alura$ cd apeperia/
aluras-Mac-mini:apeperia alura$ cd css
aluras-Mac-mini:css alura$ lessc
```

No fim, chamamos o Less com o `lessc`. Mas ao clicarmos em "Enter", receberemos uma mensagem absurda.

```

css
specify options afterwards e.g. --plugin=less-p
--plugin-clean-css="advanced"
or --clean-css="advanced"

-----
--line-numbers=TYPE      Deprecated -----
Outputs filename and line numbers.
TYPE can be either 'comments', which will output
the debug info within comments, 'mediaquery'
that will output the information within a fake
media query which is compatible with the SASS
format, and 'all' which will do both.
Compresses output by removing some whitespaces.
We recommend you use a dedicated minifier like less-plugin-clean-css

```

O Less não é adivinho, precisaremos especificar que o Less deve compilar o arquivo `estilos.less` em um CSS.

```
aluras-Mac-mini:css alura$ lessc estilos.less estilos.css
```

Especificamos qual arquivo será compilado e qual será gerado. Dessa vez, será gerado o arquivo `estilos.css`.

Se compararmos o arquivo criado recentemente com o de `.less`, veremos que eles são bem parecidos. Vamos configurar o arquivo LESS para ficar com a sintaxe de CSS.

```

index.html    estilos.less    estilos.css
1 @cor-padrao: #c24e4b;    1 /** Geral **/
2                                     2 body {
3     /** Geral **/           3     font-family: "Roboto Slab", serif;
4     body {                 4 }
5         font-family: "Roboto Slab", serif;   5     .container {
6     }                           6         width: 940px;
7     .container {            7         margin: 0 auto;
8         width: 940px;        8     }
9         margin: 0 auto;      9     .titulo-secao {
10    }                      10    text-decoration: underline;
11    .titulo-secao {        11    font-size: 2em;
12        text-decoration: underline; 12    text-align: center;
13        font-size: 2em;        13 }
14        text-align: center; 14    .sobre,
15    }                      15    .planos,
16    .sobre,                16    .blog,
17    .planos,               17    .contato {
18    .blog,                  18    padding: 3em 0 6em 0;
19    .contato {             19 }
20    padding: 3em 0 6em 0; 20

```

Mas onde estava preenchido com `@cor-padrao` no LESS e no arquivo `estilos.css` está com o hexadecimal da cor do CSS:

```

/** Header ***/
header {
    border-top: 5px solid #c24e4b;
    background: rgba(30, 44, 53, 0.8);
    height: 90px;
    width: 100%;
    position: absolute;

```

}

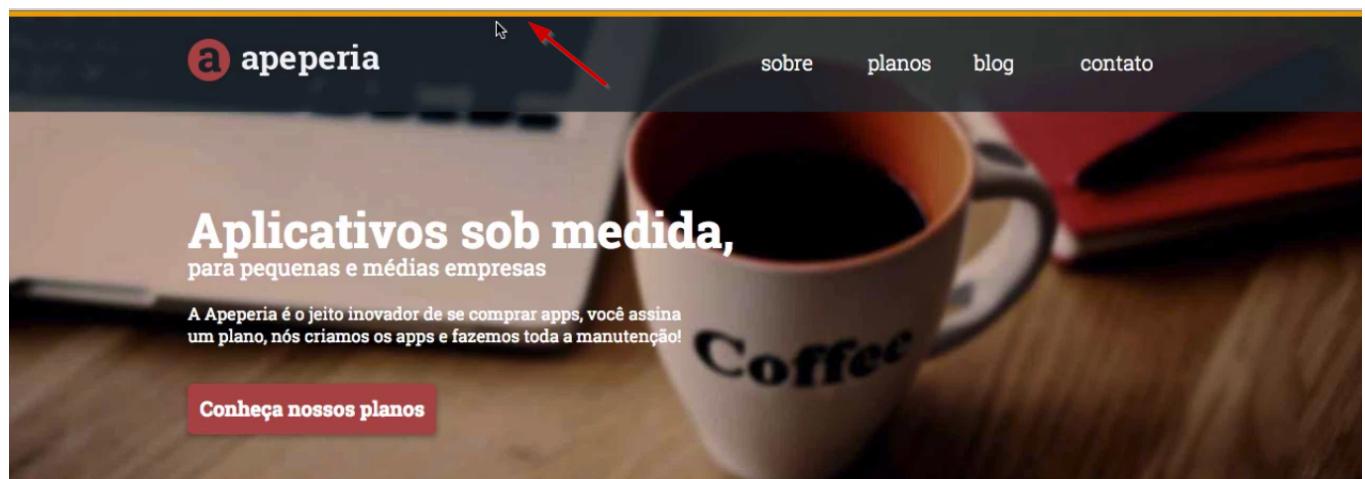
Agora, podemos alterar a cor no arquivo LESS, por exemplo para `orange`.

```
@cor-padrao: orange;
```

Mas além de mudarmos no código, vamos ter que compila-lo novamente. Para isto, usaremos o seguinte comando no Terminal.

```
aluras-Mac-mini:css alura$ lessc estilos.less estilos.css
```

Se voltarmos no editor de texto, veremos que agora o `border-top` será alterado para `orange`. Vamos ver como ficou no navegador?



A borda laranja indica que foi atualizado. Em todas as partes que apareceu a `@cor-padrao` no arquivo LESS, também foi substituído por `orange`. Lembre-se que de sempre compilar o arquivo LESS após fazer alterações no arquivo, para que elas sejam visualizadas no navegador.



Já fizemos o ajuste adicionando a variável `@cor-padrao`, vamos agora utilizar a cor `#c24e4b` solicitada pelo designer.

```
@cor-padrao: #c24e4b
```