

Gabarito

Transcrição

Aprendemos no curso que é uma boa prática utilizar um arquivo de CSS RESET para garantirmos que todos os navegadores utilizarão os mesmos valores base para as propriedades resetadas. É por isso que já importaremos o arquivo `css/reset.css` como primeiro CSS da nossa página.

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="UTF-8">
  <title>Aluraconf</title>
  <link rel="stylesheet" href="css/reset.css">
</head>
<!-- código posterior omitido>
```

O CSS Reset utilizado foi <https://meyerweb.com/eric/tools/css/reset/> (<https://meyerweb.com/eric/tools/css/reset/>).

Para cada grande área da nossa estrutura, teremos seu respectivo CSS. Essa estratégia ajuda na manutenção do nosso código. É por isso que criaremos o arquivo `aluraconf/css/header.css`.

Como arquivo criado, sem pestanejarmos, vamos importá-lo imediatamente no `head` de `index.html`:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Aluraconf</title>
  <link rel="stylesheet" href="css/reset.css">
  <link rel="stylesheet" href="css/header.css">
</head>
<!-- código posterior omitido -->
```

A primeira coisa que faremos é criar a classe `header`. Essa classe aplicada a imagem `destaque.jpeg` como fundo de todo o header.

```
/* css/header.css */
.header {
  background: url('../img/destaque.jpeg') no-repeat;
  background-size: cover;
  padding-top: 32px;
}
```

Usamos `background-size: cover` para que a imagem cresça o máximo possível para que toda a área do background seja preenchida. Quando estivermos em um dispositivo mobile, de menor largura, a imagem não será totalmente exibida e não há problema algum nisso, queremos apenas um fundo. Se aumentarmos a largura da tela gradativamente a imagem

será grande o suficiente para ocupar toda 100% da área, isto porque o padrão do `header`, um elemento `block`, é ocupar 100% da área disponível na tela.

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width">
  <link rel="stylesheet" href="css/reset.css">
  <link rel="stylesheet" href="css/header.css">
  <title>Aluraconf</title>
</head>
<body>
  <header class="header">
    <nav>
      <ul>
        <li>
          <a href="#">Sobre</a>
        </li>
        <li>
          <a href="#">Palestrantes</a>
        </li>
      </ul>
    </nav>
  </header>
</body>
</html>
```

Agora vamos centralizar o menu da aplicação através da classe `main-menu`. Desta vez, antes de criar a classe, já vamos posicioná-la no elemento correto dentro do HTML:

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="UTF-8">
  <title>Aluraconf</title>
  <link rel="stylesheet" href="css/reset.css">
  <link rel="stylesheet" href="css/header.css">
</head>
<body>
  <header class="header">
    <nav class="main-menu">
      <ul>
        <li>
          <a href="#">Sobre</a>
        </li>
        <li>
          <a href="#">Palestrantes</a>
        </li>
      </ul>
    </nav>
  <!-- código posterior omitido -->
```

Agora nosso CSS:

```
/* css/header.css */
/* CSS anterior omitido */

.header .main-menu {
  text-align: center;
}
```

Nosso menu principal esta centralizado, mas como as tags `` possuem `display block` uma fica abaixo da outra. Podemos resolver isso facilmente mudando seu `display` para `inline-block`. Não podemos usar `inline`, pois ainda precisaremos alterar mudar as margens de cada `` e apenas elementos `block` ou `inline-block` permitem isso:

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="UTF-8">
  <title>Aluraconf</title>
  <link rel="stylesheet" href="css/reset.css">
  <link rel="stylesheet" href="css/header.css">
</head>
<body>
  <header class="header">
    <nav class="main-menu">
      <ul>
        <li class="menu-item">
          <a href="#">Sobre</a>
        </li>
        <li class="menu-item">
          <a href="#">Palestrantes</a>
        </li>
      </ul>
    </nav>
  <!-- código posterior omitido -->
```

Nosso CSS até agora fica assim:

```
/* css/header.css */

.header {
  background: url('../img/destaque.jpeg') no-repeat;
  background-size: cover;
  padding-top: 32px;
}

.header .main-menu {
  text-align: center;
}

.header .main-menu .menu-item {
  display: inline-block;
```

```
margin: 0 2px 32px 0;
}
```

Quando passamos 4 parâmetros estamos passando os valores de `margin-top`, `margin-right`, `margin-bottom` e `margin-left` respectivamente.

Agora, vamos estilizar os itens de menu. A cor da letra será branca e o fundo, preto praticamente transparente. Podemos usar `rgba` para conseguir a transparência do fundo apenas. Ele também terá a borda um pouquinho arredondada e 16px de espaçamento entre o conteúdo e a borda. Não queremos cara de link por isso usamos `text-decoration: none`. Por fim, queremos garantir que o item do menu esteja sempre em caixa alta, sem depender que conteúdo tenha sido digitado neste formato. Fazemos isso através de `text-transform: uppercase`:

```
/* código anterior omitido */

.header .main-menu .menu-item a {

  color: white;
  padding: 16px;
  background: rgba(255, 255, 255, 0.1);
  border-radius: 4px;
  text-decoration: none;
  text-transform: uppercase;
}
```

Queremos ainda que ao passar o mouse sobre o item do menu (um efeito mais voltado para Desktop e não mobile), o menu fique como fundo na cor #48A3A6 e a cor da letra branca:

```
/* código anterior omitido */

.header .main-menu .menu-item a:hover {

  background-color: #48A3A6;
}
```

Excelente, mas para deixar ainda melhor a experiência do usuário, vamos aplicar uma transição para a cor do fundo aplicada pelo efeito `hover`. Queremos uma transição também na volta, por isso vamos adicionar a propriedade `transition: background-color 0.5s` no seletor que afeta a âncora.

```
/* código anterior omitido */

/* modificando o seletor já existente */
.header .main-menu .menu-item a {

  color: white;
  padding: 16px;
  background: rgba(0, 0, 0, 0.1);
  border-radius: 4px;
  text-transform: uppercase;
  text-decoration: none;
  transition: background-color 0.5s;
}
```

```
.header .main-menu .menu-item a:hover {  
  background-color: #48A3A6;  
}
```

Excelente. Agora podemos ir para o painel!