

Thumbnail images e mexendo nos estilos

Transcrição

Temos dois imóveis cadastrados, e vimos como renderizá-los na tela, fazendo com que aparecessem em `index.php`, incluímos um loop para que todos os posts fossem pegos do banco de dados, e que então fossem impressos seus títulos e conteúdos. O próximo passo consiste em incluirmos pelo menos uma imagem para cada um dos imóveis.

Estas imagens precisam fazer parte do conteúdo das postagens, certo? Se abrirmos um deles via painel de administrador, poderíamos incluir uma imagem no post, por exemplo, clicando no botão "Adicionar Mídia". Isso funcionaria, de certa forma, mas de que maneira poderíamos manipular a imagem no código?

A imagem é renderizada por meio de `the_content()` em `<p><?php the_content(); ?></p>`, assim como o texto descritivo que escrevemos anteriormente. Se quisermos desvinculá-los, precisaríamos de duas funções diferentes. Quem já é familiarizado com o Wordpress sabe que a maioria de seus temas possui suporte para o que se chama *thumbnail image*, que seria uma miniatura desassociada do conteúdo textual da postagem.

Atualmente, o tema que utilizamos não tem suporte para este tipo de imagem, portanto iremos configurar isso. Criaremos um arquivo denominado `functions.php` com o código referente a tudo que tiver a ver com funcionalidades extras ao tema. E como teremos apenas código PHP nele, não precisaremos nos preocupar em fechar a tag.

```
<?php  
  
add_theme_support('post-thumbnails');
```

Com isso, ao voltarmos ao dashboard, no canto inferior direito teremos "Imagem destacada", que é o suporte para *thumbnail image*. Com isso, a partir de então, todo post que criarmos poderá ter uma imagem destacada. Vamos testar clicando em "Definir imagem", escolhendo uma das imagens disponíveis e pressionando "Atualizar".

Atualizando a página no navegador, porém, não teremos nenhuma atualização visível. Como dito anteriormente, queremos desvincular `the_content()` da imagem, mas para isto será necessário trazê-la para `index.php`, e faremos isto acrescentando a linha `<?php the_post_thumbnail(); ?>` antes das tags `<h2>` que trazem o título da postagem.

A imagem passará a ser exibida como gostaríamos, ocupando toda a tela, pois no momento não estamos nos preocupando com o CSS. Para começarmos a ajustá-lo, começaremos incluindo um trecho de código HTML em `index.php`:

```
<?php  
    }  
}  
?>  
  
<main class="home-main">  
    <div class="container">  
  
        <ul class="imoveis-listagem">  
  
            <li class="imoveis-listagem-item">
```

```

        </li>

    </ul>
</div>
</main>

```

Em seguida, moveremos os códigos correspondentes ao back end dentro deste trecho, de modo a deixá-lo assim, com o loop dentro da lista, pois cada item seu será renderizado em um post:

```

<main class="home-main">
    <div class="container">

        <h1>Bem vindo ao Maluras!</h1>
        <ul class="imoveis-listagem">

            <?php
                if( have_posts() ) {
                    while( have_posts() ) {
                        the_post();

                    <li class="imoveis-listagem-item">
                        <?php the_post_thumbnail(); ?>
                        <h2><?php the_title(); ?></h2>
                        <p><?php the_content(); ?></p>
                    </li>

                <?php
                    }
                }
            </ul>
        </div>
    </main>

```

Assim, após o título da página, temos uma lista de imóveis e, para cada post ou imóvel que tivermos, o Wordpress renderizará um item da lista, "recheado" com o que queremos, ou seja, a *thumbnail* image, o título e o conteúdo. Vamos salvar o arquivo e verificar as alterações na página?

Não há nenhuma alteração, exceto por um pontinho (*bullet*) ao lado da imagem e do subtítulo "Casa de Argila". Além disso, a primeira postagem tem thumbnail, enquanto a outra não. Definiremos, então, uma imagem destacada para o post "Casa de Argila". O *bullet* é o CSS que o próprio navegador inclui em nosso site.

O Google Chrome possui um CSS padrão, e quando temos uma lista não ordenada, ele inclui um *bullet* automaticamente. Mas isso nos atrapalha, e quem já trabalha com CSS deve conhecer o `reset`, dentre os quais o do Meyer é um dos mais famosos, cujo código pode ser copiado [aqui \(http://meyerweb.com/eric/tools/css/reset/\)](http://meyerweb.com/eric/tools/css/reset/).

Este `reset` retira todo e qualquer estilo que os navegadores definem por padrão, e isso nos traz maior autonomia, além da flexibilidade e controle sobre as escolhas. Colaremos o código disponível no site em `style.css` e atualizaremos a página no navegador. Ao fazermos isso, porém, nada é alterado em nosso site.

Isso porque não importamos o CSS em `header.php`, e faremos isso usando um atalho do Sublime, digitando "link" e pressionando a tecla Tab, que auto preencherá a tag para nós, que incluiremos entre as tags `<head>`: `<link rel="stylesheet" type="text/css" href="style.css">`.

No entanto, mesmo assim nada foi resetado na página. Ao inspecionarmos o elemento e clicarmos na aba "Network", verificamos que o arquivo CSS não está sendo encontrado, e o caminho utilizado para a busca é

`http://localhost/wordpress/style.css`, e se consultarmos os nossos diretórios locais, teremos que o caminho deveria ser `wordpress/wp-content/themes/malura/style.css`.

Sendo assim, alteraremos o `href` da linha de código que acabamos de incluir para que aponte para `<?php get_template_directory_uri(); ?>`. Esta função irá nos trazer a pasta raiz do nosso tema, com o qual acessaremos `style.css`, e ela nos retorna uma *string*. Para que ela seja impressa, usaremos o atalho de PHP `?=` no lugar de `?php`.

O trecho de código em `header.php` ficará da seguinte forma:

```
<head>
  <meta charset="utf-8">
  <link rel="stylesheet" type="text/css" href="<?= get_template_directory_uri(); ?>/style.css"
  <?php wp_head(); ?>
</head>
```

Salvaremos o arquivo e voltaremos ao navegador para atualizarmos a página. Desta vez, o CSS foi resetado; iremos melhorá-la esteticamente incluindo o conteúdo do arquivo `estilo.css` em `style.css`. Utilizamos o Flexbox em CSS, e tudo está funcionando conforme esperado. Agora, de que maneira poderemos organizar melhor o nosso código? Poderíamos criar dois arquivos distintos, um para o `reset` e outro para o restante do CSS.

Faremos isto criando `reset.css` e copiando e colando o conteúdo correspondente. Para importarmos este arquivo, teremos que acrescentar a linha `<link rel="stylesheet" type="text/css" href="<?= get_template_directory_uri(); ?>/reset.css">` em `header.php`, **antes** da tag `<link` de `style.css`. Testaremos a página no navegador mais uma vez, e teremos o efeito desejado, isto é, tudo continua funcionando.

No entanto, passamos a redobrar o trabalho do Wordpress duplicando a linha com `<link>`. O que poderemos fazer para refatorar o código?

Em vez de deixarmos duas linhas `<link>`, vamos guardar o retorno da função `get_template_directory_uri()` na variável `home`:

```
<head>
  <?php $home = get_template_directory_uri(); ?>
  <meta charset="utf-8">
  <link rel="stylesheet" type="text/css" href="<?= $home; ?>/reset.css">
  <link rel="stylesheet" type="text/css" href="<?= $home; ?>/style.css">
  <?php wp_head(); ?>
</head>
```

Vimos bastante conteúdo, como incluir *thumbnail images*, organizar o código, utilizamos CSS.

