### Aula 16: Criando itens dinamicamente

Nesta lição, vamos aprender como adicionar itens dinamicamente usando o **jQuery**.

**Preparando uma Interface Administrativa**

Para nossa loja, queremos criar uma interface administrativa que permita ao administrador adicionar novos itens à página dinamicamente, sem precisar editar diretamente o arquivo **index.html**. Para começar, vamos adicionar um novo **div** acima do nosso **div** **container** chamado**admin-container**. Dentro desse novo div, vamos adicionar um campo de **entrada** e um **botão**.

<div id="admin-container">

    <input id="input-create-item" type="text"></input>

    <button id="button-create-item">Add</button>

</div>

Através dessa interface, queremos ser capazes de digitar um novo destino no campo de entrada e clicar no botão para adicioná-lo à nossa página. Existem algumas coisas que precisamos fazer em **main.js** para realizar isso. Primeiro, precisamos criar um **evento onclick** para o nosso botão usando o **seletor** correto para o **ID** do **botão**.

$(document).ready(function(){

    $('#button-create-item').on('click',function() {

    });

});

Dentro da função do nosso evento de clique, precisamos armazenar o **valor** recebido no campo de **entrada**. Faremos isso configurando uma variável, **name**, e usando o **seletor** para o campo de entrada para obter o valor. No entanto, abaixo disso, também podemos alterar o **valor** para uma **string vazia** para limpar o campo de entrada após armazenar a informação.

$(document).ready(function(){

    $('#button-create-item').on('click',function() {

        let name = $('#input-create-item').val();

        $('#input-create-item').val('');

    });

});

**Adicionando novo HTML**

Agora estamos prontos para trabalhar com a adição real do HTML à página. Na função do nosso **evento onclick**, e após os elementos de armazenamento do valor de entrada, primeiro queremos configurar uma **variável** **html** que será definida como uma**string vazia.**

let html = ' ';

Em seguida, vá para o **index.html** e copie tudo dentro do nosso **contêiner div item**. Precisamos **colar** isso abaixo da declaração da variável html em **main.js**. Obviamente, o **jQuery** não pode ler isso, então precisamos envolver todas as nossas linhas entre aspas para transformá-las em strings. Para adicionar tudo isso à nossa **variável** **html** para que possamos escrever todo o HTML de uma vez, precisamos usar o **operador +=** e também adicionar **pontos e vírgula** para fechar as linhas.

html += '<div class="item">';

html += '<div class="name">Santa Monica</div>';

html += '<div class="description">Lorem ipsum dolor sit amet, consectetur adipisicing

 elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</div>';

html += '<div class="price">499</div>';

html += '<button>Add to cart</button>';

html += '<br />';

html += '<a href="#">More info</a>';

html += '<div class="more-info">Lorem ipsum dolor sit amet</div>';

html += '</div>';

(**Observação**: No Atom, você pode destacar a linha, pressionar a aspa simples e pressionar a tecla de espaço para rapidamente colocar a linha entre aspas).

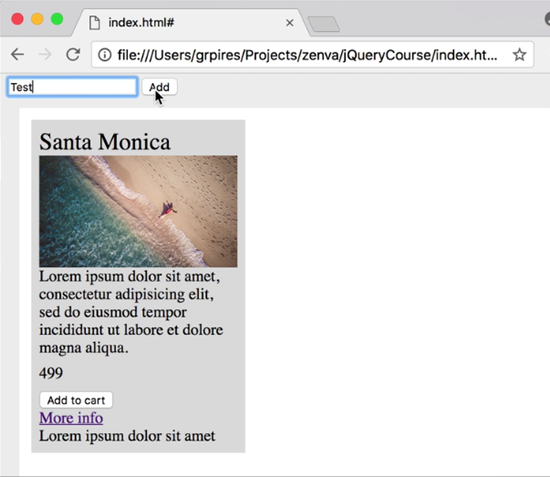
No entanto, agora precisamos realizar o passo mais importante: precisamos transformar a linha com a **classe “name”** em uma **string concatenada** que usa a **variável “name”** que armazena nosso novo destino do campo de entrada.

html += '<div class="name">' + name + '</div>';

Por último, mas não menos importante, queremos adicionar as strings de dados/HTML em nossa **variável “html”** ao **contêiner div** em **index.html**. No final da nossa função do botão, adicionaremos a seguinte linha. Tenha em mente que usamos a função **jQuery** **append** aqui, pois isso irá inserir corretamente, ou “anexar”, o item na página sem apagar outros elementos.

$('#container').append(html);

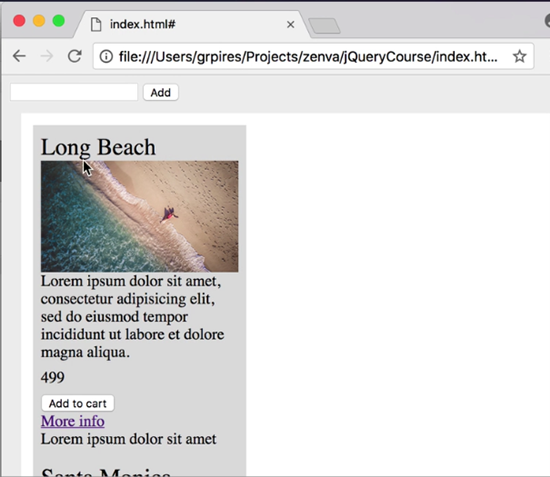
Agora você pode testar a página com diferentes itens e ver que eles são adicionados dinamicamente.



**Adições Finais**

Digamos que você queira adicionar novos destinos acima do item atual “Santa Monica”. Você pode fazer isso facilmente usando a função **prepend** em vez de **append**. Isso irá inserir novos elementos acima dos outros elementos no **contêiner div**.

$('#container').prepend(html);



Antes de encerrarmos esta lição, faremos uma pequena alteração no arquivo **style.css** e adicionaremos uma regra **float left** e **margin** à **classe “item”**. Isso permitirá que os itens adicionados apareçam à esquerda com um espaçamento melhor.

.item {

    width: 200px;

    background-color: #DDD;

    padding: 8px;

    float: left;

    margin: 4px;

}

