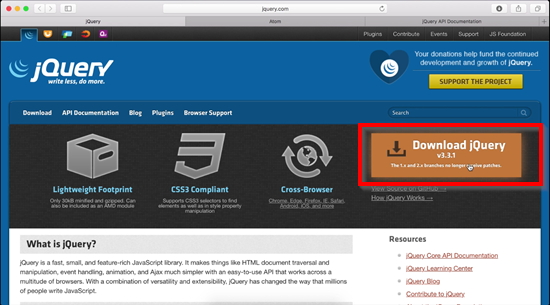
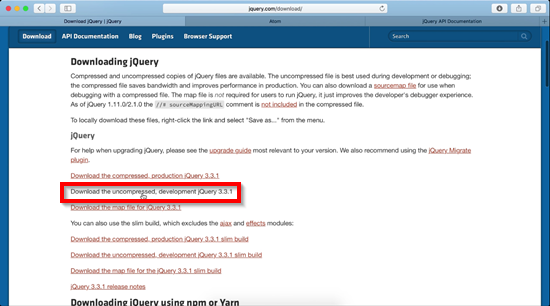
### Aula 1: Introdução

### jQuery Introdução

Quando se trata de **jQuery** , existem dois métodos para adicioná-lo ao seu site. O primeiro método é simplesmente acessar o **site jQuery** e baixar o **arquivo jQuery** completo . Ao incluir este arquivo na pasta do seu host, você tem acesso à **biblioteca jQuery** . O segundo método é usar um **jQuery CDN** , ou **Content Delivery Network** , para carregar dinamicamente o **arquivo jQuery** de outro lugar. Este segundo método tem a vantagem de tornar seu site mais rápido se o usuário já encontrou o **CDN** antes e baixou os arquivos necessários em segundo plano. Para aprender a usar um **CDN jQuery**em seus próprios projetos, você pode acessar  <https://code.jquery.com/> para obter instruções.

**Para nosso projeto** , vamos usar o primeiro método e **baixar o arquivo jQuery** . Acesse o **site principal do jQuery** ( <https://jquery.com/> ), clique em **Download jQuery** e **baixe** o arquivo de **desenvolvimento descompactado** (esta versão será a mais fácil de ler).



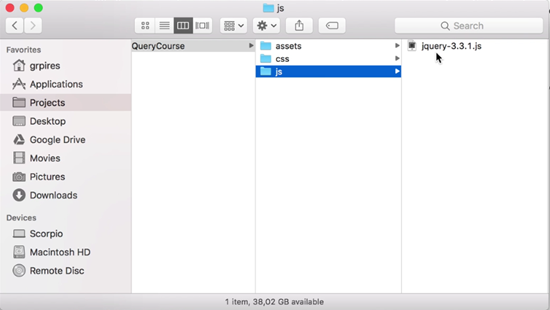


### Configuração da estrutura do projeto

Agora vamos configurar a pasta do nosso projeto. Primeiro, clique com o botão direito do mouse e crie uma **nova pasta** chamada **jQueryCourse** onde você deseja salvar o projeto. Dentro desta pasta, vamos criar mais três pastas:

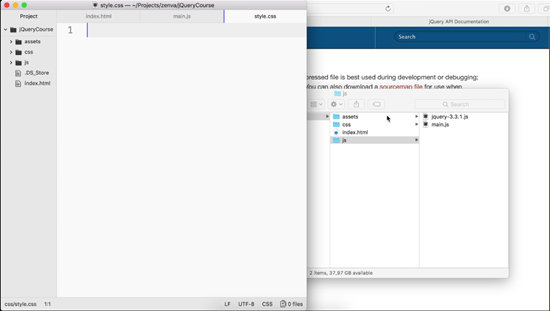
* **assets**
* **CSS**
* **js**

Na **pasta js** , vamos adicionar o **arquivo jQuery** que acabamos de baixar **recortando e colando** de onde ele foi baixado no computador.



Para nossas próximas etapas, você precisa fazer **o download** e **instalar o Atom** para que possamos escrever e editar nosso código. Você pode encontrar **o Atom** em seu site: <https://atom.io/>

Depois de baixar **o Atom** , **abra** o programa. Vamos criar **três novos arquivos** ( **Control N** ou **Command N** para os atalhos). Primeiro, criaremos **o index.html** e o salvaremos na **pasta principal do jQueryCourse** . Em seguida, crie **main.js** para nosso **javascript** e salve-o na **pasta js** . Por último, crie um arquivo chamado **style.css** para nosso CSS e adicione-o à **pasta css** .



### Configuração do código base do projeto

Agora vamos começar a configurar o código básico para o nosso projeto. Em **index.html** , primeiro adicionaremos nosso HTML clichê para iniciar a página. Uma vez dentro, você pode verificar se ele funciona abrindo **index.html em** seu navegador.

<html>

<head>

<script src="js/jquery-3.3.1.js"></script>

<script src="js/main.js"></script>

<link rel="stylesheet" type="text/css" href="css/style.css">

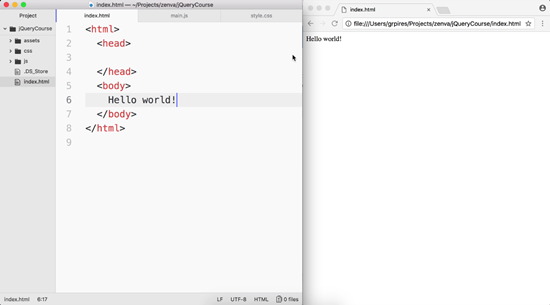
</head>

<body>

Hello world!

</body>

</html>



Para que possamos ter acesso aos demais arquivos que criamos, precisamos vinculá-los em **index.html** . Na **seção head** , adicionaremos o código abaixo. Ao fornecer e vincular nossos arquivos **javascript** , **jQuery** e **CSS** , agora temos acesso a eles.

<script src="js/jquery-3.3.1.js"></script>

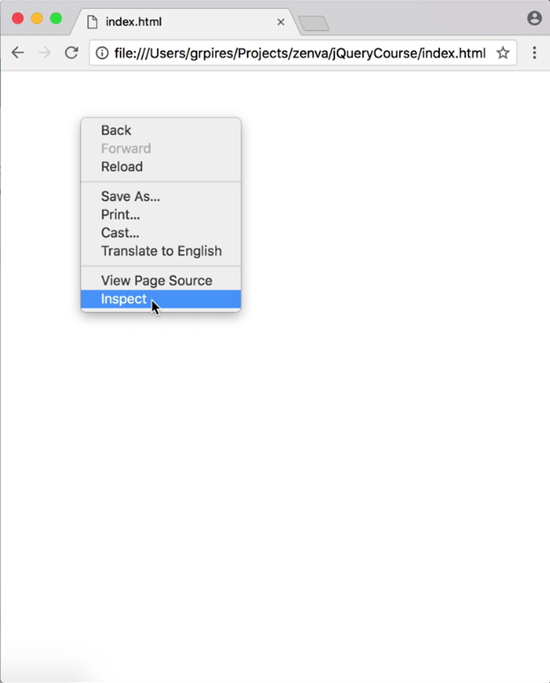
<script src="js/main.js"></script>

<link rel="stylesheet" type="text/css" href="css/style.css">

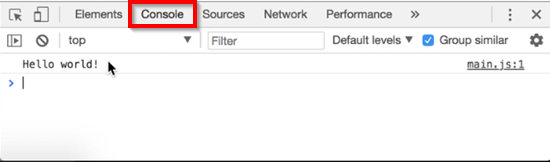
Vamos testar se nosso arquivo   **main.js está vinculado. Remover “** Olá, mundo!” De **index.html** e vá para **main.js** . Neste arquivo, adicione a seguinte linha para que o **javascript** escreva “Hello world!” em vez de.

console.log('Hello world!');

Para ver isso em nosso navegador corretamente, você precisa visualizá-lo no **Inspetor** . Se estiver usando **o Chrome** , você pode **clicar com o botão direito do mouse** e selecionar **Inspecionar** no menu.



Em nossa nova interface, podemos selecionar **Console** , e lá você deve ver nossa mensagem.



### Aula 2: Preparando o documento

Nesta lição, vamos aprender conceitos necessários sobre como usar o jQuery para escrever dinamicamente “Hello world!” no corpo (body).

**Sobre o Document Ready**

Ao contrário de aplicativos executáveis que são fornecidos prontos com todo o código, sites e aplicações web dependem do constante download e execução do código linha por linha. Portanto, precisamos ter cuidado e muitas vezes esperar que o documento esteja pronto antes de injetar qualquer código jQuery. Para adicionar “Hello world!” ao nosso corpo, por exemplo, precisamos garantir que as tags do corpo (body) estejam carregadas antes da execução do nosso jQuery.

Felizmente, existe uma chamada comum do jQuery que podemos usar para garantir que o documento esteja pronto. No arquivo main.js, vamos excluir nosso código anterior e adicionar o seguinte:

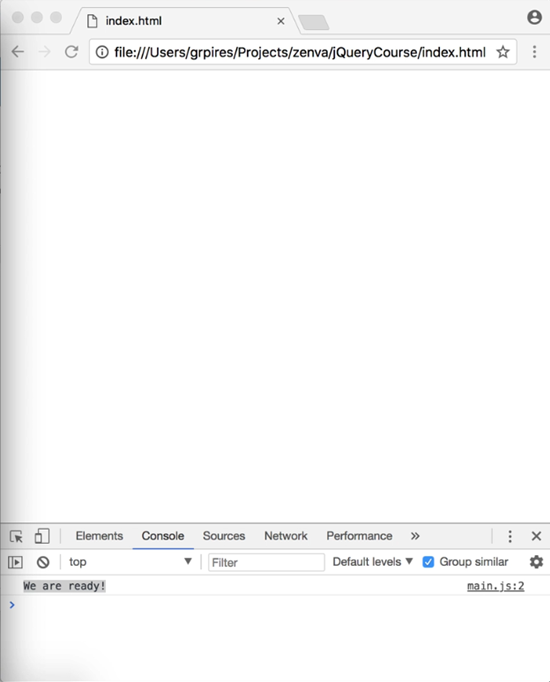
$(document).ready(function(){

    console.log('We are ready!');

});

(**Observação**: você pode escrever jQuery em vez do símbolo de dólar para chamar as funções do jQuery, mas o símbolo de dólar dá uma aparência mais limpa).

Com este código, selecionamos o documento e garantimos que ele esteja pronto antes de chamar a função que escreve em nosso console. Se você testar index.html, deverá ver a nova mensagem no Console.



### Aula 3: HTML e texto

Nesta lição, finalmente vamos usar o **jQuery** no arquivo **main.js** para escrever “Hello world!” no corpo (**body**) da nossa página web.

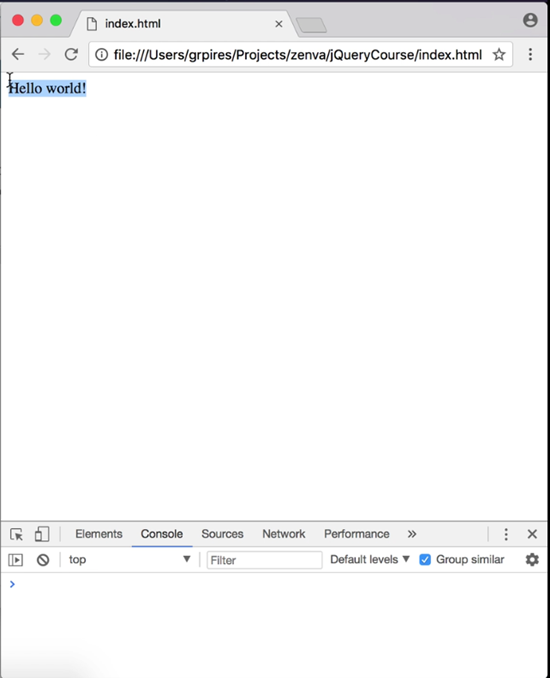
**Adicionando texto ao corpo**

Além de transformar nossos objetos, como o documento, em **objetos jQuery**, também podemos procurá-los usando um seletor. No main.js, fazemos isso simplesmente envolvendo o elemento que queremos procurar entre aspas simples para transformá-lo em um **seletor**. Esse seletor então indica ao nosso código para procurar por um objeto correspondente, neste caso, a **tag body** no código abaixo. Usando a chamada **.text** do **jQuery**, podemos adicionar o texto “Hello world!” ao corpo.

$(document).ready(function(){

    $('body').text('Hello world!');

});

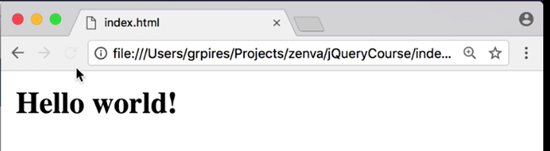


No entanto, digamos que queremos que nosso texto seja em negrito. Em vez de adicionar o texto via **.text** (que renderiza tudo como uma string e simplesmente escreve o texto literalmente), podemos alterar o próprio **HTML** com **.html**. Usando essa chamada, podemos adicionar HTML ao nosso conteúdo que está sendo escrito no documento e fazer com que esse HTML seja executado da mesma forma que o HTML regular seria.

$(document).ready(function(){

    $('body').html('<strong>Hello world!</strong>');

});



### Aula 4: Selecionando pelo ID

Nesta lição, vamos aprender como selecionar um elemento por **ID** usando **jQuery**. Antes de começarmos, **apague** o conteúdo da nossa função **“document ready”** para termos espaço para criar uma nova função.

$(document).ready(function(){

});

**Listas Não Ordenadas e Seleção por ID**

Na **seção body** do arquivo **index.html**, vamos criar dois **títulos** e algumas **listas não ordenadas**. Estas serão listas simples sobre itens quentes e frios. No entanto, para nossos itens frios, também vamos adicionar uma lista dentro da nossa lista para mostrar diferentes tipos de chuva.

<h4>Hot</h4>

<ul>

    <li>Fire</li>

    <li>Lava</li>

    <li>Steam</li>

</ul>

<h4>Cold</h4>

<ul>

    <li>Snow</li>

    <li>Ice</li>

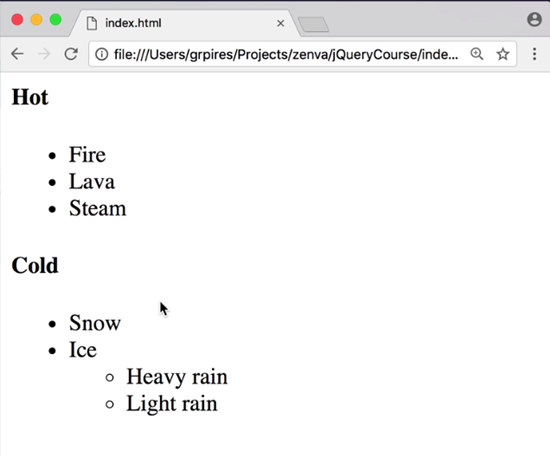
    <ul>

        <li>Heavy rain</li>

        <li>Light rain</li>

    </ul>

</ul>



Com a nossa lista básica pronta, vamos agora trabalhar com seletores e estilos. Vamos começar adicionando um **ID** ao nosso **item de lista** para “Fire”.

<li id="fire">Fire</li>

Vá para o arquivo **style.css**. Vamos adicionar uma **regra de classe “highlight”** que altera a cor do texto para vermelho.

.highlight {

        color: #F00;

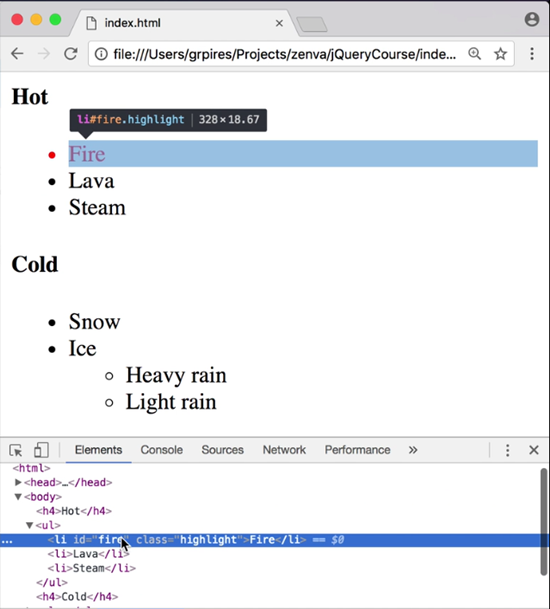
}

No entanto, sem a **classe** sendo adicionada ao nosso **item de lista** “Fire”, a regra de destaque não pode estilizá-lo. Usando a capacidade do **jQuery** de usar seletores, podemos adicionar dinamicamente essa classe ao nosso **item de lista**. Ao inserir “#fire” como seletor, informamos ao jQuery para procurar especificamente um elemento com o **ID** **“fire**“. Usando **.addClass**, podemos então adicionar a **classe “highlight”.**

$(document).ready(function(){

    $('#fire').addClass('highlight');

});



(**Observação**: Usando o Inspector, você também pode ver a classe adicionada no HTML agora).

### Aula 5: Selecionando pela Classe

Nesta lição, vamos aprender a selecionar elementos em nossa lista por **classe** usando o **jQuery**.

**Adicionando Classes Necessárias**

No **arquivo index.html**, vamos alterar um pouco nossa lista classificando nossos**itens de lista** como sólidos ou não-sólidos. Faremos isso adicionando uma **classe** de **“non-solid”** ou **“solid”** para cada elemento.

<h4>Hot</h4>

<ul>

    <li class="non-solid" id="fire">Fire</li>

    <li class="solid">Lava</li>

    <li class="non-solid">Steam</li>

</ul>

<h4>Cold</h4>

<ul>

    <li class="solid">Snow</li>

    <li class="solid">Ice</li>

    <ul>

        <li class="non-solid">Heavy rain</li>

    <li class="non-solid">Light rain</li>

    </ul>

</ul>

**Experimentando com Seletores**

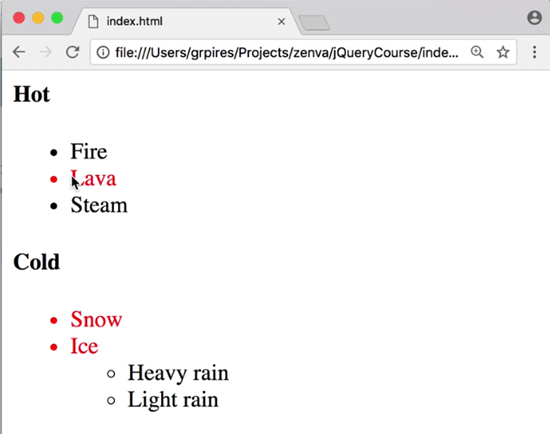
Usando nosso seletor no **jQuery**, podemos trabalhar facilmente com todos os objetos sólidos ou não-sólidos, mesmo que eles estejam em listas diferentes. No arquivo **main.js**, vamos alterar nosso seletor de **#fire** para **.solid**.

$(document).ready(function(){

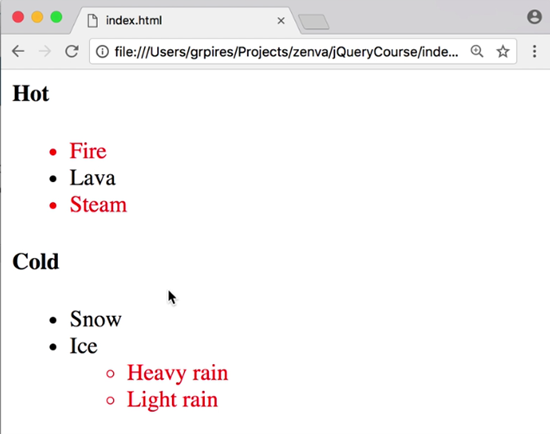
    $('.solid').addClass('highlight');

});

Ao testar o **index.html**, você verá que todos os objetos com a classe **“solid”** agora estão em vermelho.



Se você alterar o seletor para **.non-solid**, verá o efeito oposto.

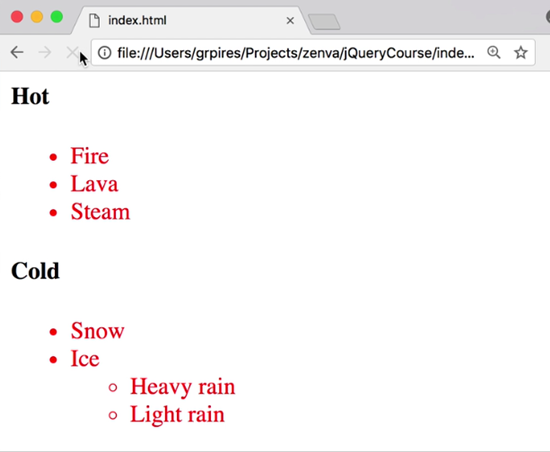


Também podemos usar uma vírgula para adicionar múltiplos seletores e selecionar ambas as classes ao mesmo tempo.

$(document).ready(function(){

    $('.non-solid, .solid').addClass('highlight');

});



No entanto, podemos fazer coisas ainda mais complicadas. Abaixo do item da lista para “Ice”, adicione mais um item de lista para “Rain”.

<li class="non-solid">Rain</li>

Se nosso **seletor** buscar apenas pela **classe “non-solid”**, iremos selecionar todas as chuvas em nossa lista de itens frios no momento. No entanto, vamos supor que só queremos selecionar “Heavy rain” e “Light rain”. Para fazer isso, precisamos primeiro alterar a estrutura do nosso HTML no arquivo **index.html**. Vamos começar envolvendo todo o **conteúdo do body** em uma **tag div** com um **ID** de **“container”**.

<div id="container">

    <h4>Hot</h4>

    <ul>

        <li class="non-solid" id="fire">Fire</li>

        <li class="solid">Lava</li>

    <li class="non-solid">Steam</li>

    </ul>

    <h4>Cold</h4>

    <ul>

        <li class="solid">Snow</li>

        <li class="solid">Ice</li>

        <li class="non-solid">Rain</li>

        <ul>

            <li class="non-solid">Heavy rain</li>

            <li class="non-solid">Light rain</li>

        </ul>

    </ul>

</div>

Agora podemos alterar nosso **seletor** no arquivo **main.js**. Ao adicionar o **ID do container** ao seletor e o símbolo maior que (>), informamos ao seletor para selecionar apenas os **filhos** do **ID do container**. No entanto, os filhos imediatos desse container são os títulos e as tags de lista não ordenada. Portanto, precisamos adicionar múltiplos **símbolos de maior** que para descer o número apropriado de níveis. Neste caso, adicionaremos três, já que “Heavy rain” e “Light rain” são o terceiro nível de filhos.

$(document).ready(function(){

    $('#container >>> .non-solid').addClass('highlight');

});

Você pode ver que agora apenas esses dois elementos recebem a **classe “highlight”.**

