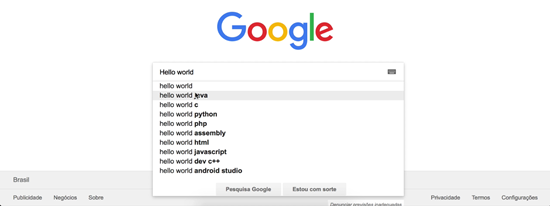
### Aula 21: AJAX

Nesta lição, vamos trabalhar na adição de mais responsividade para que os elementos possam reagir ao usuário.

**Responsividade e AJAX**

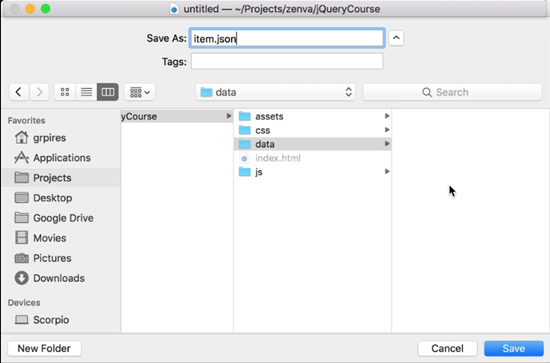
Ao usar a internet, cada página com a qual você interage é essencialmente baixada para o seu computador. No entanto, esse não é o caso para todos os arquivos. Por exemplo, o Google é capaz de sugerir consultas de pesquisa para você, mas você não precisa baixar um arquivo com cada sugestão de pesquisa possível.



Em vez disso, o Google processa essas informações em seus servidores e é capaz de enviar as informações dinamicamente. Isso é feito usando o **AJAX**, ou **Javascript Assíncrono e XML**. Por meio disso, as funções podem ocorrer no backend e enviar informações ao usuário mesmo após o carregamento da página.

**Configuração de Arquivo Externo**

Em nosso projeto, vamos usar o **AJAX** para carregar nossos itens de um arquivo externo. Para começar, vamos criar uma **nova pasta** em nosso **projeto jQueryCourse** chamada **“data”**. Nesta pasta, vamos criar um **novo arquivo** usando o **Atom** e nomeá-lo como **“item.json”**.



No**main.js**, na parte inferior de nossa**função document ready** e após todas as outras funções, vamos adicionar uma chamada **AJAX** em nosso arquivo externo. Nesta chamada, também vamos adicionar uma **função** que recebe a **resposta** como parâmetro e imprime a resposta no console.

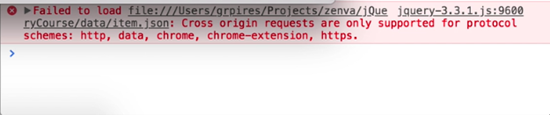
$.ajax('data/item.json', function(response){

    console.log(response);

});

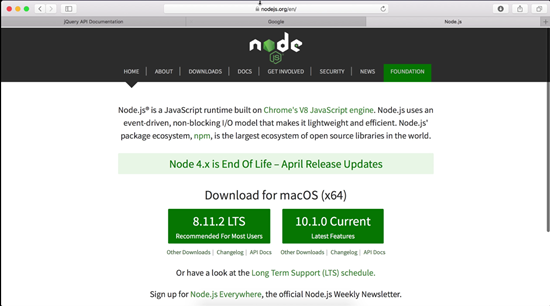
**Simulação de Servidor e Node.js**

Se você testar a chamada agora, receberá um erro.



Esse erro ocorre porque o **index.html** está tentando obter o arquivo localmente. Isso não é permitido, pois poderia ser usado maliciosamente para injetar código nocivo em seu computador. Em vez disso, precisamos simular um servidor para poder testar nossa página. Para fazer isso, usaremos o **node.js**, que pode ser encontrado aqui: https://nodejs.org/en/

Na página, faça o **download** e **instale** a versão do **node.js** apropriada para o seu sistema operacional.



O **node.js** vem automaticamente com o **npm**, que é o que precisaremos para simular um servidor. Para iniciar a simulação, abra o **Terminal** no **Mac** ou o Prompt de Comando no Windows. Na janela, precisamos baixar os arquivos necessários para nossa simulação de servidor usando o seguinte comando:

$ npm install -g node-http-server

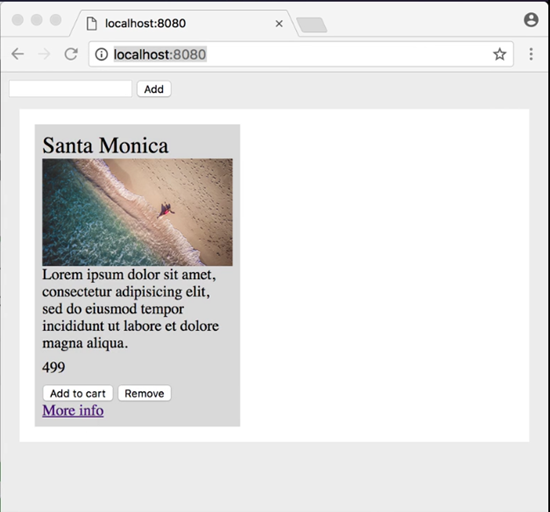
Em seguida, precisamos navegar até a pasta do nosso projeto no **Terminal** ou no **Prompt de Comando**.

$ cd Projects/codi/CursojQuery/

Por fim, agora digitamos o seguinte comando para configurar a simulação do servidor:

$ node-http-server

Sua simulação de servidor está sendo executada agora. Você pode testar isso acessando **localhost:8080** no seu navegador. Você deverá ver nossa página da web, mas agora servida para nós em um servidor local.



(**Observação**: o Terminal ou o Prompt de Comando deve ser mantido aberto para executar seu servidor enquanto você trabalha neste curso.)