

## 1. Introdução ao HTML

Neste documento, abordamos os fundamentos absolutos do HTML, para você começar. Definimos elementos, atributos e todos os outros termos importantes que você pode ter ouvido e onde eles se encaixam na linguagem. Também mostramos como os elementos HTML são estruturados, como uma página HTML típica é estruturada e explicamos outros recursos importantes da linguagem básica. Ao longo do caminho, mostraremos um pouco de HTML para interessá-lo!

### 1.1 O que é HTML?

HTML (Hypertext Markup Language) não é uma linguagem de programação; é uma linguagem de marcação usada para informar ao seu navegador como estruturar as páginas da web que você visita. Pode ser tão complicado ou simples quanto o desenvolvedor da web deseja. O HTML consiste em uma série de elementos, que você usa para incluir, agrupar ou marcar diferentes partes do conteúdo para que ele apareça ou aja de uma certa maneira. As tags anexas podem transformar um pouco de conteúdo em um hiperlink para vincular a outra página da Web, colocar itálico em palavras e assim por diante. Por exemplo, considere a seguinte linha de conteúdo:

```
Meu gato é muito brincalhão
```

Se quiséssemos que a linha se sustentasse, poderíamos especificar que é um parágrafo colocando-a envolvida pela tag (elemento) `<p>` :

```
<p>Meu gato é muito brincalhão</p>
```

**Observação:** As tags em HTML não diferenciam maiúsculas de minúsculas, ou seja, podem ser escritas em maiúsculas ou minúsculas. Por exemplo, uma tag `<title>` pode ser escrita como `<title>`, `<TITLE>`, `<Title>`, `<TiTIE>`, etc., e ela vai funcionar bem. A melhor prática, no entanto, é escrever todas as tags em minúsculas por consistência, legibilidade e outros motivos.

### 1.2 Estrutura de um elemento HTML

Vamos explorar um pouco mais o nosso elemento de parágrafo:



As principais partes do nosso elemento são:

**A abertura de tag:** consiste no nome do elemento (neste caso, p), envolto em colchetes de abertura e fechamento. Indica onde o elemento começa - nesse caso, onde está o início do parágrafo.

**O fechamento de tag:** é igual a tag de abertura, exceto pelo fato de incluir uma barra antes do nome do elemento. Indica onde o elemento termina - nesse caso, onde está o final do parágrafo. Não incluir uma tag de fechamento é um erro comum para iniciantes e pode levar a resultados estranhos.

**O conteúdo:** este é o conteúdo do elemento, que neste caso é apenas texto.

**O elemento:** a tag de abertura mais a tag de fechamento mais o conteúdo é igual ao elemento.

### 1.2.1 Criando seu primeiro elemento HTML

Crie um arquivo chamado: index.html. Note que a extensão tem que estar como html e não como txt, caso contrário o navegador não reconhecerá o arquivo.

No arquivo index.html, insira o seguinte conteúdo, o qual é a estrutura básica de qualquer página html:

```
<html>
  <head> </head>
  <body>

  </body>
</html>
```

Logo mais entenderemos o que cada tag destas significam. Agora, adicione a linha abaixo entre as tags <body>, assim:

```
<html>
  <head> </head>
  <body>
    Meu texto html
  </body>
</html>
```

Salve o arquivo e dê dois cliques no ícone dele para abrir no navegador.

Veja que o resultado é uma linha no navegador escrito Meu texto html. Agora vamos deixá-lo em itálico. Para isso, coloque as tags <em> no início da frase e </em> no fim da frase, ficará assim:

```
<html>
  <head> </head>
  <body>
    <em> Meu texto html </em>
  </body>
</html>
```

Salve e execute. Você verá que o texto estará em itálico.

### 1.2.2 Elementos aninhados

Os elementos também podem ser colocados dentro de outros elementos - isso é chamado de aninhamento. Se quiséssemos afirmar que nosso gato é muito brincalhão, poderíamos envolver a palavra "muito" em um elemento `<strong>`, o que significa que a palavra deve ser enfatizada fortemente:

```
<html>
  <head> </head>
  <body>
    <p>Meu gato é<strong>muito brincalhão</strong></p>
  </body>
</html>
```

No entanto, você precisa garantir que seus elementos estejam aninhados corretamente: no exemplo acima, abrimos o elemento `<p>` primeiro, depois o elemento `<strong>`; portanto, precisamos fechar o elemento `<strong>` primeiro e depois o elemento `<p>`. O seguinte está incorreto:

```
<p>Meu gato é <strong>muito brincalhão.</p></strong>
```

Os elementos precisam abrir e fechar corretamente, para que estejam claramente dentro ou fora um do outro. Se eles se sobrepuserem como acima, seu navegador tentará adivinhar o que você estava tentando dizer e poderá obter resultados inesperados. Então não faça isso!

### 1.2.3 Elemento de bloco versus elemento embutidos ou de linha

Existem duas categorias importantes de elementos em HTML que você deve conhecer. Eles são elementos em **nível de bloco** e **elementos em linha**.

Os elementos no nível do bloco formam um bloco visível em uma página - eles aparecerão em uma nova linha a partir de qualquer conteúdo que foi anterior a ela, e qualquer conteúdo que for depois dela também aparecerá em uma nova linha. Os elementos no nível do bloco tendem a ser elementos estruturais na página que representam, por exemplo, parágrafos, listas, menus de navegação, rodapés etc. Um elemento no nível do bloco não seria aninhado dentro

de um elemento embutido, mas poderia ser aninhado dentro de outro elemento no nível do bloco.

Elementos embutidos (ou de linhas) são aqueles que estão contidos nos elementos no nível do bloco e envolvem apenas pequenas partes do conteúdo do documento, não parágrafos e agrupamentos inteiros de conteúdo. Um elemento embutido não fará com que uma nova linha apareça no documento; eles normalmente apareceriam dentro de um parágrafo de texto, por exemplo, um elemento <a> (hiperlink) ou elementos de ênfase como <em> ou <strong>.

Veja o seguinte exemplo:

```
<em>primeiro</em><em>segundo</em><em>terceiro</em>  
<p>quarto</p><p>quinto</p><p>sexto</p>
```

<em> é um elemento embutido. Como você pode ver a saída abaixo, os três primeiros elementos ficam na mesma linha um do outro, sem espaço entre eles. Por outro lado, <p> é um elemento no nível do bloco, portanto, cada elemento aparece em uma nova linha, com espaço acima e abaixo de cada (o espaçamento é devido ao estilo CSS padrão que o navegador aplica aos parágrafos).

*primeirosegundoterceiro*

quarto

quinto

sexto

#### 1.2.4 Elementos vazios

Nem todos os elementos seguem o padrão acima de uma tag de abertura, conteúdo e uma tag de fechamento. Alguns elementos consistem apenas em uma única tag, que geralmente é usada para inserir ou incorporar algo no documento no local desejado pelo programador. Por exemplo, o elemento <img> incorpora um arquivo de imagem em uma página na posição em que está incluído:

```

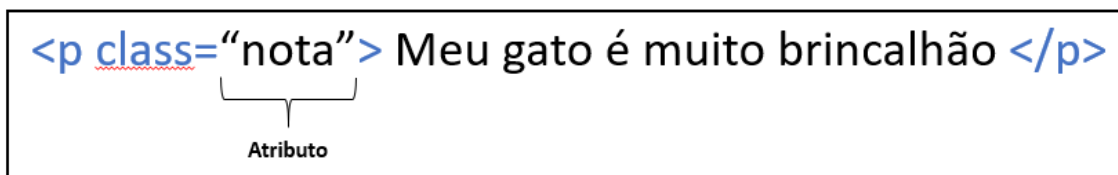
```

Isso produziria o seguinte em sua página:



### 1.2.5 Atributos

Os elementos também podem ter atributos, assim:



Os atributos contêm informações extras sobre o elemento que você não deseja que apareçam no conteúdo real. Nesse caso, o atributo `class` permite que você atribua um nome de identificação ao elemento, que pode ser usado posteriormente para direcionar o elemento com informações de estilo e outras coisas.

Um atributo deve ter:

- Um espaço entre ele e o nome do elemento (ou o atributo anterior, se o elemento já tiver um ou mais atributos).
- O nome do atributo, seguido por um sinal de igual.
- Um valor de atributo, com aspas de abertura e fechamento em volta dele.

#### 1.2.5.1 Adicionando atributos a um elemento

Outro exemplo de um elemento é `<a>`- isso significa "âncora" e fará com que o pedaço de texto que ele envolve em um hiperlink. Isso pode levar vários atributos, mas vários são os seguintes:

- **href:** O valor desse atributo especifica o endereço da web para o qual você deseja apontar o link; onde o navegador navega quando o link é clicado. Por exemplo `href="https://www.arieldias.com/"`.
- **title:** O atributo `title` especifica informações extras sobre o link, como a qual página está sendo vinculada. Por exemplo `title="Página do professor Ariel"`. Isso aparecerá como uma dica de ferramenta quando passar o mouse pelo elemento.
- **target:** O atributo `target` especifica o contexto de navegação que será usado para exibir o link. Por exemplo, `target="_blank"` exibirá o link em uma nova guia. Se você deseja exibir o link na guia atual, apenas omita esse atributo.

No arquivo `index.html` que criamos anteriormente, adicione a linha abaixo entre as tags `<body>`.

```
<p>Um link para meu <a href="https://www.arieldias.com/"  
title="Site do professor Ariel" target="_blank"> site favorito  
</a>. </p>
```

### 1.2.5.2 Adicionando atributos booleanos

Às vezes, você vê atributos escritos sem valores - isso é perfeitamente permitido. Esses são chamados de atributos booleanos e podem ter apenas um valor, geralmente o mesmo que o nome do atributo. Como exemplo, considere o atributo `disabled`, que você pode atribuir para formar elementos de entrada, se desejar que eles sejam desativados (acinzentados) para que o usuário não possa inserir nenhum dado neles.

```
<input type="text" disabled="disabled">
```

Como abreviação, é perfeitamente permitido escrever isso da seguinte maneira (também incluímos um elemento de entrada de formulário não desativado para referência, para dar uma ideia do que está acontecendo):

```
<!-- o uso do atributo desativado impede que o usuário final
insira texto na caixa de entrada -->
<input type="text" disabled>

<!-- O usuário pode inserir texto na entrada a seguir, pois não
contém o atributo desativado -->
<input type="text">
```

### 1.2.5.3 Omitindo aspas ao redor dos valores de atributo

Quando você olha na World Wide Web, encontra vários estilos de marcação estranhos, incluindo valores de atributos sem aspas. Isso é permitido em determinadas circunstâncias, mas interrompe sua marcação em outras. Por exemplo, se revisitarmos nosso exemplo de link anteriormente, poderíamos escrever uma versão básica apenas com o atributo `href`, assim:

```
<a href=https://www.arieldias.com/>site favorito</a>
```

No entanto, assim que adicionarmos o atributo `title` neste estilo, as coisas darão errado:

```
<a href=https://www.arieldias.com/ title=Site do professor
Ariel>site favorito</a>
```

Nesse ponto, o navegador interpretará mal sua marcação, pensando que o atributo `title` são realmente quatro atributos - um atributo de título com o valor "Site" e três atributos booleanos, do professor Ariel. Obviamente, não é isso que foi planejado e causará erros ou comportamento inesperado no código.

Nosso conselho é sempre incluir as aspas dos atributos - isso evita esses problemas e resulta em código mais legível também.

### 1.2.6 Anatomia de um documento HTML

Isso tudo que vimos até então envolve o básico de elementos HTML individuais, mas eles não são muito úteis por si só. Agora, veremos como os elementos individuais são combinados para formar uma página HTML inteira:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My test page</title>
  </head>
  <body>
    <p>This is my page</p>
  </body>
</html>
```

Aqui temos:

- <!DOCTYPE html>: O doctype. No tempo passado, quando o HTML era jovem (cerca de 1991/2), os doctypes deveriam funcionar como links para um conjunto de regras que a página HTML tinha que seguir para ser considerada um bom HTML, o que poderia significar verificação automática de erros e outras coisas úteis. Eles costumavam ser algo como isto:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

No entanto, atualmente, ninguém se importa com eles, e eles são apenas um artefato histórico que precisa ser incluído para que tudo funcione corretamente. <!DOCTYPE html> é a sequência mais curta de caracteres que conta como um doctype válido; é tudo o que você realmente precisa saber.

- <html></html>: O elemento <html> agrupa todo o conteúdo da página inteira e às vezes é conhecido como elemento raiz.
- <head></head>: O elemento <head> atua como um contêiner para todas as coisas que você deseja incluir na página HTML, que não é o conteúdo que você está mostrando aos visualizadores da sua página. Isso inclui coisas como palavras-chave e uma descrição da página que você deseja que apareça nos resultados da pesquisa, CSS para estilizar nosso conteúdo, declarações de conjunto de caracteres e muito mais. Você aprenderá mais sobre isso mais adiante.
- <meta charset="utf-8">: Este elemento especifica o conjunto de caracteres do seu documento para UTF-8, que inclui a maioria dos caracteres da grande maioria das linguagens escritas humanas. Essencialmente, agora ele pode lidar com qualquer conteúdo de texto que você possa colocar nele. Não há razão para não definir isso, e isso pode ajudar a evitar alguns problemas posteriormente.

- `<title></title>`: O elemento `<title>` define o título da sua página, que é o título que aparece na guia do navegador em que a página é carregada e é usado para descrever a página quando você a adiciona aos favoritos/favoritos.
- `<body></body>`: O elemento `<body>` contém todo o conteúdo que você deseja mostrar aos usuários da Web quando eles visitam sua página, seja texto, imagens, vídeos, jogos, faixas de áudio reproduzíveis ou qualquer outra coisa.

Se você quiser escrever um pouco de HTML no computador local, copie o exemplo da página HTML listado acima e cole no seu arquivo `index.html` criado anteriormente.

Agora você pode abrir esse arquivo em um navegador da Web para ver como é o código renderizado e, em seguida, editar o código e atualizar o navegador para ver qual é o resultado.

Portanto, neste exercício, você pode editar o código localmente no seu computador, conforme descrito acima.

### EXERCÍCIO PROPOSTO

Agora, logo abaixo da tag de abertura do elemento `<body>`, adicione um título principal para o documento. Isso deve estar dentro de uma tag `<h1>` de abertura e de uma tag `</h1>` de fechamento.

Edite o conteúdo do parágrafo para incluir algum texto sobre algo de seu interesse.

Faça com que as palavras importantes se destaquem em negrito, envolvendo-as em uma tag de abertura `<strong>` de uma tag `</strong>` de fechamento.

Adicione um link ao seu parágrafo, conforme explicado anteriormente.

Adicione uma imagem ao seu documento, abaixo do parágrafo, conforme explicado anteriormente. Você receberá pontos de bônus se conseguir vincular a uma imagem diferente (localmente no seu computador ou em outro lugar da Web).

#### 1.2.7 Espaço em branco em HTML

Nos exemplos acima, você deve ter notado que muitos espaços em branco estão incluídos nas listagens de código - isso não é necessário; os dois seguintes trechos de código são equivalentes:

```
<p>Cachorros são bravos.</p>
```

```
<p>Cachorros      são  
      bravos.</p>
```

Não importa quanto espaço em branco você use (que pode incluir caracteres de espaço, mas também quebras de linha), o analisador de HTML reduz cada um deles em um único espaço ao renderizar o código. Então, por que usar tanto espaço em branco? A resposta é legibilidade - é muito mais fácil entender o que está acontecendo no seu código, se você o tiver bem



formatado, e não apenas agrupado em uma grande confusão. Em nosso HTML, cada elemento aninhado é recuado por dois espaços a mais do que aquele que está dentro. Depende de você que estilo de formatação você usa (quantos espaços para cada nível de recuo, por exemplo), mas considere formatá-lo.

#### 1.2.8 Incluindo caracteres especiais no HTML

Em HTML, os caracteres <, >, ", ' e & são caracteres especiais. Eles fazem parte da própria sintaxe HTML. Então, como você inclui um desses caracteres no seu texto? Por exemplo, se você realmente deseja usar um e comercial ou um sinal menor que, e não deseja o interpretar como código.

Temos que usar referências de caracteres - códigos especiais que representam caracteres e podem ser usados nessas circunstâncias exatas. Cada referência de caractere é iniciada com um e comercial (&) e finalizada por um ponto e vírgula (;).

Caractere literal	Referência de caractere equivalente
<	& lt;
>	& gt;
"	?
'	& apos;
&	& amp;

No exemplo abaixo, você pode ver dois parágrafos, que estão falando sobre tecnologias da web:

```
<p>Em HTML, você define o parágrafo usando o elemento <p>. </p>  
  
<p> Em HTML, você define o parágrafo usando o elemento  
&lt;p&gt;. </p>
```

Na saída abaixo, você pode ver que o primeiro parágrafo está errado, porque o navegador pensa que a segunda instância <p> está iniciando um novo parágrafo. O segundo parágrafo parece bom, porque substituímos os colchetes angulares por referências de caracteres.

Em HTML, você define o parágrafo usando o elemento.

Em HTML, você define o parágrafo usando o elemento <p>.

### 1.2.8 Incluindo caracteres especiais no HTML

Em HTML, como na maioria das linguagens de programação, existe um mecanismo disponível para escrever comentários no código - os comentários são ignorados pelo navegador e são invisíveis para o usuário, e seu objetivo é permitir que você inclua comentários no código para dizer como seu código funciona, o que as diferentes partes do código fazem e assim por diante. Isso pode ser muito útil se você retornar a uma base de código em que não trabalhou há muito tempo e não se lembrar do que fez - ou se você entregar seu código para outra pessoa trabalhar.

Para transformar uma seção do conteúdo HTML em um comentário, você precisa agrupá-lo nos marcadores especiais `<!--` e `-->`, por exemplo:

```
<p>Parágrafo fora do comentário</p>  
<!-- <p>Parágrafo dentro do comentário</p> -->
```

### 1.3 Conclusão

Você chegou ao final deste capítulo - esperamos que tenha gostado do seu tour pelos princípios básicos do HTML! Neste ponto, você deve entender como é a linguagem, como ela funciona em um nível básico e ser capaz de escrever alguns elementos e atributos. Este é o lugar perfeito para estar agora, já que os próximos capítulos deste módulo abordarão algumas das coisas que você já examinou com muito mais detalhes e introduzirão alguns novos conceitos da linguagem. Fique ligado!

## 2. Cabeçalho e Metadados em HTML

O cabeçalho de um documento HTML é a parte que não é exibida no navegador da Web quando a página é carregada. Ele contém informações como a página <title>, links para CSS (se você optar por estilizar seu conteúdo HTML com CSS), links para favoritos personalizados e outros metadados (dados sobre o HTML, como o autor, e palavras-chave importantes que descrevem o documento). Neste capítulo, cobriremos todos os itens acima e muito mais, a fim de fornecer uma boa base para trabalhar com a marcação.

### 2.1 Cabeçalho HTML

Vamos revisar o documento HTML simples que abordamos no capítulo anterior:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Minha primeira página</title>
  </head>
  <body>
    <p>Este é o conteúdo da minha página. </p>
  </body>
</html>
```

O cabeçalho HTML é o conteúdo do elemento <head> - diferentemente do conteúdo do elemento <body> (que é exibido na página quando carregado em um navegador), o conteúdo do cabeçalho não é exibido na página. Em vez disso, o trabalho dele é conter metadados sobre o documento. No exemplo acima, o head é bem pequeno:

```
<head>
  <meta charset="utf-8">
  <title>Minha primeira página</title>
</head>
```

#### 2.1.1 Adicionando um título

Já vimos o elemento <title> em ação - isso pode ser usado para adicionar um título ao documento. No entanto, isso pode ser confundido com o elemento <h1>, que é usado para adicionar um cabeçalho de nível superior ao conteúdo do seu corpo - isso também é chamado de título da página. Mas são coisas diferentes!

O elemento <h1> aparece na página quando carregado no navegador - geralmente deve ser usado uma vez por página, para marcar o título do conteúdo da página (o título da história, o título da notícia ou o que for mais apropriado para o seu uso).

O <title>elemento são metadados que representam o título do documento HTML geral (não o conteúdo do documento).

### EXERCÍCIO PROPOSTO

1. Para começar esse exercício proposto, gostaríamos que você fosse ao nosso repositório do GitHub (<https://github.com/arielsdias/DesenvolvimentoWeb>) e baixasse uma cópia da nossa página exemplo-titulo.html. Para fazer isso:
  - a. copie e cole o código da página em um novo arquivo de texto e salve-o em um local adequado.
  - b. Pressione o botão "Raw" na página do GitHub, que faz com que o código bruto apareça (possivelmente em uma nova guia do navegador). Em seguida, escolha o menu Arquivo do navegador > Salvar página como ... e escolha um local adequado para salvar o arquivo.
2. Agora abra o arquivo no seu navegador. Agora deve ser completamente óbvio onde o conteúdo <h1> aparece e onde o conteúdo <title> aparece!
3. Você também deve tentar abrir o código no seu editor de código, editar o conteúdo desses elementos e atualizar a página no seu navegador.

O conteúdo do elemento <title> também é usado de outras maneiras. Por exemplo, se você tentar marcar a página como favorito ( Favoritos > Marcar esta página ou o ícone de estrela na barra de URL do Firefox), verá o conteúdo do <title> preenchido como o nome sugerido do favorito.

O conteúdo <title> também é usado nos resultados da pesquisa, como você verá abaixo.

#### 2.2 Metadados: o elemento <meta>

Metadados são dados que descrevem dados, e o HTML possui uma maneira "oficial" de adicionar metadados a um documento - o elemento <meta>. Existem muitos tipos diferentes de elementos <meta> que podem ser incluídos no <head> da sua página, mas não tentaremos explicar todos eles neste estágio, pois isso seria confuso demais. Em vez disso, explicaremos algumas coisas que você normalmente vê, apenas para lhe dar uma ideia.

##### 2.2.1 Especificando a codificação de caracteres do seu documento

No exemplo que vimos acima, esta linha foi incluída:

```
<meta charset="utf-8" >
```

Esse elemento simplesmente especifica a codificação de caracteres do documento - o conjunto de caracteres que o documento pode usar. utf-8 é um conjunto de caracteres universal que inclui praticamente qualquer caractere de qualquer linguagem humana. Isso significa que sua página da web poderá exibir qualquer idioma; Portanto, é uma boa ideia definir isso em todas as páginas da Web que você criar! Por exemplo, sua página pode lidar com inglês e japonês muito bem.

Se você definir a codificação de caracteres como ISO-8859-1, por exemplo (o conjunto de caracteres do alfabeto latino), a renderização da página poderá aparecer toda bagunçada. Experimente a codificação de caracteres.

Para fazer isso, revise o modelo HTML simples que você obteve na seção anterior <title>(na página exemplo-titulo.html ), tente alterar o valor do meta charset para ISO-8859-1 e adicione o japonês à sua página. Este é o código que usamos:

```
<p>Japanese example: ご飯が熱い。 </p>
```

### 2.2.2 Adicionando um autor e uma descrição

Muitos elementos <meta> incluem os atributos name e content:

- name especifica o tipo de meta elemento que é; que tipo de informação contém.
- content especifica o meta conteúdo real.

Dois desses elementos meta que são úteis para incluir em sua página definem o autor da página e fornecem uma descrição concisa da página. Vejamos um exemplo:

```
<meta name="author" content="Chris Mills">  
<meta name="description" content="Este documento visa fornecer  
ao estudante tudo o que precisam saber para obter um bom começo  
em HTML.">
```

A especificação de um autor é benéfica de várias maneiras: é útil entender quem escreveu a página, se você tiver alguma dúvida sobre o conteúdo e quiser contatá-lo. Alguns sistemas de gerenciamento de conteúdo têm recursos para extrair automaticamente as informações do autor da página e disponibilizá-las para esses fins.

Especificar uma descrição que inclua palavras-chave relacionadas ao conteúdo da sua página é útil, pois tem o potencial de fazer com que a sua página apareça mais em pesquisas relevantes realizadas nos mecanismos de pesquisa (essas atividades são denominadas Otimização de Mecanismos de Pesquisa ou SEO ).

### EXERCÍCIO PROPOSTO

A descrição também é usada nas páginas de resultados dos mecanismos de pesquisa. Vamos fazer um exercício para explorar isso

- Vá para a página inicial [www.arieldias.com](http://www.arieldias.com) .
- Exibir a fonte da página (Clique com o botão direito do mouse na página, escolha Exibir fonte da página no menu de contexto).
- No arquivo fonte, encontre a meta tag de descrição.
- Agora pesquise "Ariel Dias" no seu mecanismo de pesquisa favorito (usamos o Google.) Você notará a descrição <meta> e o <title> do elemento usado no resultado da pesquisa - definitivamente vale a pena!

### 2.2.3 Adicionando outros tipos de metadados

Enquanto você navega pela Web, também encontra outros tipos de metadados. Muitos dos recursos que você verá nos sites são criações proprietárias, projetadas para fornecer a determinados sites (como sites de redes sociais) informações específicas que eles podem usar.

Por exemplo, o Open Graph Data é um protocolo de metadados que o Facebook inventou para fornecer metadados mais ricos para sites. No código fonte do site [www.arieldias.com](http://www.arieldias.com), você encontrará:

```
<meta property="og:locale" content="pt_BR">
<meta property="og:type" content="website">
<meta property="og:title" content="Professor Ariel Dias | Notas de Aula">
<meta property="og:description" content="Ariel Dias é professor universitário, consultor em Tecnologia da Informação e Segurança, ministra cursos VIPs, em grupo ou in company. Além de palestras empresariais e acadêmicas.">
<meta property="og:url" content="http://www.arieldias.com/">
<meta property="og:site_name" content="Professor Ariel Dias">
```

Um efeito disso é que, quando você vincula o site Ariel Dias ao Facebook, o link aparece junto com uma imagem e descrição. Faça o teste!

### 2.3 Aplicando CSS e JavaScript ao HTML

Quase todos os sites que você usa nos dias de hoje empregam CSS para torná-los visualmente atrativos e JavaScript para aumentar a funcionalidade interativa, como reprodutores de vídeo, mapas, jogos e muito mais. Eles são mais comumente aplicados a uma página da Web usando os elementos `<link>` e `<script>`, respectivamente.

O elemento `<link>` sempre fica dentro do cabeçalho do seu documento. São necessários dois atributos, `rel = "stylesheet"`, que indica que é a folha de estilo do documento, e `href`, que contém o caminho para o arquivo da folha de estilo:

```
<link rel="stylesheet" href="meu-arquivo-css.css">
```

O elemento `<script>` não precisa entrar no cabeçalho; de fato, geralmente é melhor colocá-lo na parte inferior do corpo do documento (logo antes da tag `</body>` de fechamento), para garantir que todo o conteúdo HTML tenha sido lido pelo navegador antes de tentar aplicar o JavaScript a ele (se JavaScript tenta acessar um elemento que ainda não existe, o navegador lançará um erro.)

```
<script src="meu-arquivo-javascript.js"></script>
```

O elemento `<script>` pode parecer um elemento vazio, mas não é, e, portanto, precisa de uma tag de fechamento. Em vez de apontar para um arquivo de script externo, você também pode optar por colocar seu script dentro do elemento `<script>`.

### **EXERCÍCIO PROPOSTO**

1. Para iniciar esse exercício, pegue no github uma cópia dos nossos arquivos meta-exemplo.html, primeiro-script.js e primeiro-estilo.css e salve-os no computador local no mesmo diretório. Verifique se eles foram salvos com os nomes e extensões de arquivo corretos.
2. Abra o arquivo HTML no seu navegador e no seu editor de texto.
3. Seguindo as informações fornecidas acima, adicione os elementos `<link>` e `<script>` ao seu HTML, para que seu CSS e JavaScript sejam aplicados ao seu HTML.

Se feito corretamente, ao salvar seu HTML e atualizar seu navegador, você poderá ver que as coisas mudaram.

O JavaScript adicionou uma lista vazia à página. Agora, quando você clica em qualquer lugar da lista, uma caixa de diálogo será exibida solicitando que você digite algum texto para um novo item da lista. Quando você pressiona o botão OK, um novo item da lista será adicionado à lista que contém o texto. Quando você clica em um item da lista existente, uma caixa de diálogo é exibida, permitindo que você altere o texto do item.

O CSS fez com que o fundo ficasse verde e o texto se tornasse maior. Ele também estilizou parte do conteúdo que o JavaScript adicionou à página (a barra vermelha com a borda preta é o estilo que o CSS adicionou à lista gerada por JS).