



WEB

Academia Indiana



Introdução a WEB

- Internet é o conjunto de diversas redes de computadores que se comunicam entre si.
- Internet não é sinônimo de WWW (World Wide Web - Rede de alcance mundial). O WWW é um dos diversos serviços oferecidos através da Internet



Introdução ao HTML

- A linguagem HTML possui seus comandos de feitos com tags (etiquetas) que equivalem aos comandos de formatação da linguagem. Uma tag, em HTML, não é mais do que marcas padrões utilizadas para fazer as indicações necessárias ao browser. Todo documento HTML apresenta as tags cercadas por um sinal de menor ('<') e maior ('>').



Introdução ao HTML

- Existem dois tipos de tags. As compostas por uma tag de abertura e outra de fechamento, como por exemplo: `<body>` ou `<div>` . E as tags mais simples, em que se usa apenas uma única tag: `
` `<meta charset="UTF-8"/>` ``



Estruturas e Tags Básicas do HTML5

A estrutura básica do HTML5 é parecida as estrutura do XHTML e do HTML 4, contendo elementos HEAD e BODY.

```
<!DOCTYPE html>  
<html lang="pt-br">  
<head>  
  
</head>  
<body>  
  
</body>  
</html>
```



Estruturas e Tags Básicas do HTML5

O DOCTYPE define qual tipo de arquivo você está escrevendo, ela é responsável por informar ao navegador qual é este tipo. É o primeiro código a ser adicionado em uma página.

Enquanto o XHTML 1.0 trazia mais informações, o DOCTYPE do HTML5 é mais simples. Segue abaixo exemplo do DOCTYPE do XHTML:

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



Estruturas e Tags Básicas do HTML5

A tag **html** define o início da sua página, nela podemos adicionar o lang definindo a linguagem (português, inglês, espanhol, etc) da página. Este atributo ajuda aos leitores de tela a ler corretamente os textos contidos na página, além dos buscadores poderem identificar o idioma e para que o navegador possa utilizar o dicionário gramatical mais correto.



Estruturas e Tags Básicas do HTML5

No **head** ficarão informações conhecidas como metadados, que normalmente são globais, não aparecem na página, mas podem efetuar modificações na mesma. Aqui adicionamos o title para definir o título da página e as metas viewport, charset, description, keywords, cards para as rede sociais, links de arquivos css e javascript.

Dentro do head um dos comandos mais importantes é o title. Ele define o título da página e ajuda os buscadores, influenciando o Google por exemplo, a mostrar os conteúdos mais relevantes. O ideal é que o title tenha até 65 caracteres e que cada página dentro do seu site tenha uma diferente, identificando em qual página do site a pessoa está.



Estruturas e Tags Básicas do HTML5

As **metas** ajudam o seu site a descrever seu conteúdo aos buscadores, como o Google. Aqui você poderá dizer qual a codificação (charset), descrição e até palavras-chaves referentes a sua página.

Abaixo algumas das metas mais importantes:

- description: guarda a descrição do site, utilizado nos motores de busca.
- keywords: palavras chaves relacionadas ao site, utilizada nos primórdios pelos motores de busca.
- author: identifica o autor do site.



Estruturas e Tags Básicas do HTML5

Por fim, no **body** ficarão as tags e informações que serão mostradas ao seu usuário, interpretada pelo navegador.



Tags HTML e seus Significados

- **DOCTYPE:** define o tipo de documento
- **html:** define o início do documento html, todos as demais tags devem estar dentro desta tag
- **head:** cabeçalho onde ficar os metadados do documento, como metatags, css e javascript
- **title:** define o título do documento, mostrado na barra de títulos do navegador e como link nos mecanismos de busca
- **base:** define a url base do documento
- **link:** utilizado para linkar um arquivo de imagem ou css a página html
- **script:** utilizado para escrever ou anexar um arquivo normalmente javascript a página
- **meta:** metatags, como charset, description, viewport ou keywords
- **style:** utilizada para adicionar um código css a página
- **body:** corpo ou área principal do documento, dentro deste deve ficar o conteúdo da sua página
- **header:** define um cabeçalho da página ou de uma seção, normalmente pode conter um menu (nav), um logo ou link
- **nav:** define uma área de navegação contendo links, para a formação de menus com hiperlinks
- **footer:** define o rodapé de uma página ou seção
- **main:** define o conteúdo principal da página, deve ser utilizado apenas uma única vez em cada página
- **section:** define uma seção do documento
- **article:** pode ser um post ou notícia de um fórum ou blog, define um conteúdo independente
- **h1 a h6:** representam 6 níveis de títulos, sendo o h1 de maior importância
- **p:** define um parágrafo
- **br:** define uma quebra de linha
- **div:** define uma camada sem formatação, genérico
- **ul:** define uma lista não ordenada
- **li:** define um elemento ou item da lista não ordenada
- **a:** define um link que deve ser apontado com o atributo href
- **img:** define a utilização de uma imagem (JPG, GIF, PNG e outras). A imagem deve ser apontada com o uso do atributo src
- **strong:** apesar de deixar em negrito, representa a importância deste texto no meio da frase



Introdução ao CSS

- O CSS foi desenvolvido pela primeira vez em 1997, como uma maneira para os desenvolvedores da web para definir a aparência de suas páginas web. Destinava-se a permitir que os desenvolvedores, para separar o conteúdo de projeto de modo que o HTML pudesse executar mais de uma função sem se preocupar com o design e layout. *Assim como o HTML, o CSS não é realmente uma linguagem de programação. Também não é uma linguagem de marcação — é uma linguagem de folhas de estilos.*



Introdução ao CSS

- Um arquivo CSS (Cascading Style Sheet) permite que você separe seus sites do conteúdo HTML. Você usa seu arquivo HTML para organizar o conteúdo, mas toda a apresentação (fontes, cores de fundo, bordas, formatação de texto, efeitos de ligação e assim por diante...) são realizadas dentro de um CSS.



Exemplo de Arquivo css

```
1  h1{
2    font-style: italic;
3  }
4  #fundo{
5    background-color: #a3a9fa;
6  }
7  .paragrafo{
8    color: red;
9  }
10 .paragrafo2{
11   color: #0018ff;
12 }
```



Incluindo CSS em uma Pagina HTML

Existem três formas para incluir o código CSS em seu projeto:

- Inline - A primeira forma de aplicar CSS a uma página é utilizando o atributo style em elementos do HTML:

```
1. <p style="color: blue">Parágrafo com fonte azul.</p>
2. <p>Esse outro parágrafo não é azul, a não ser que
3. exista <span style="color: red">CSS em outro lugar</span>.</p>
```



Incluindo CSS em uma Pagina HTML

- Tag Interna (interno) - A segunda forma é utilizar a tag **style** dentro do **head** da página HTML:

```
1. <head>
2.   <style type="text/css">
3.     seletor { propriedade: valor; }
4.   </style>
5. </head>
```




Incluindo CSS em uma Página HTML

- Externo - E a última - porém a mais utilizada - maneira de aplicar CSS é criar um ou mais arquivos com extensão **.css** e incluí-los na estrutura **head** do HTML:

```
1. <head>
2.   <link rel="stylesheet" type="text/css" href="reset.css">
3.   <link rel="stylesheet" type="text/css" href="styles.css">
4. </head>
```



Formulários HTML

O que é um formulário HTML?

O formulário HTML é um formulário de preenchimento de dados ou que resulta em uma ação desejada utilizando a linguagem de marcação HTML. É formado por um ou mais widgets. Esses widgets são campos de textos, caixas de seleção, botões, radio buttons e checkboxes utilizando ferramentas do próprio HTML. Dessa forma, o usuário pode interagir com a página ao executar ações através desses formulários.



Formulários HTML

O método GET

O método GET é usado pelo navegador para solicitar que o servidor envie de volta um determinado recurso. Portanto, é como se falássemos ao servidor “Servidor, eu quero obter este recurso.”. Nesse caso, o navegador envia um corpo vazio. Portanto, já que o corpo fica vazio, se um formulário for enviado através do método GET, os dados serão reconhecidos pelo servidor através da URL.



Formulários HTML

Algumas observações a respeito do método GET:

- O tamanho de uma URL é limitado a cerca de 3000 caracteres;
- Nunca use o GET para enviar dados confidenciais, pois esses dados ficarão visíveis na URL;
- É útil para envios de formulários em que um usuário deseja marcar o resultado;
- GET é melhor para dados não seguros, como strings de consulta no Google.



Formulários HTML

O método POST

O método POST é utilizado no navegador para conversar com o servidor. Os dados são enviados ao servidor através do corpo da solicitação HTTP. Também é realizada uma solicitação de resposta. Portanto, é como se falássemos ao servidor “Servidor, verifique esses dados e me retorne um resultado adequado”. Dessa forma, ao enviar um formulário através do método POST, os dados serão anexados ao corpo da solicitação HTTP.



Formulários HTML

Diferentemente do método GET, explicado anteriormente, o método POST não inclui o corpo na URL. Portanto, os dados enviados não ficarão visíveis na URL.

Algumas observações a respeito do método POST:

- Anexa dados de formulário dentro do corpo da solicitação de HTTP. Portanto, os dados não são mostrados na URL;
- Não tem limitações de tamanho;
- Os envios de formulários com o POST não podem ser marcados.



Formulários HTML

- `<input>` – Define um campo de entrada de dados;
- `<textarea>` – Define uma área de texto, podendo conter diversas linhas de texto;
- `<button>` – Define um botão;
- `<select>` – Define uma lista selecionável, também conhecida como drop-down;
- `<option>` – Define uma lista de opções dentro de um drop-down;
- `<optgroup>` – Define um grupo de opções;
- `<fieldset>` – Define um grupo de campos;
- `<label>` – Define um rótulo ou legenda para um campo ou controle do formulário;
- `<output>` – Define elementos de saída para o formulário;
- `<legend>` – Define um título para o conjunto de campos.



Validação e persistência de dados de um formulário



Sessões e Cookies

Cookies

Através de *cookies* o servidor *web* é capaz de trocar informações de estado com o navegador do usuário. Desse modo, somos capazes de adicionar produtos a um carrinho de compras, sem perder estas informações ao mudar de página, sair do *website* ou até mesmo fechar o navegador.

Tecnicamente falando, um *cookie* é uma pequena quantidade de informação persistida temporariamente pelo navegador. Os navegadores normalmente limitam o tamanho dos *cookies* em até 4KB, e apagam *cookies* com a data de “validade vencida”.



Sessões e Cookies

Sessões

As sessões têm um princípio similar aos *cookies*, só que o armazenamento do estado é feito pelo servidor *web*, e não pelo navegador.



Sessões e Cookies

Por exemplo, quando construímos uma aplicação que necessita de autenticação, no momento em que o usuário efetuar o *login*, podemos até permitir que algumas informações sejam armazenadas em um *cookie*, mas dados mais “sensíveis”, como usuário e *e-mail*, são mais interessantes de serem guardadas em sessões. Isto, pois **não é seguro** que esse tipo de informação fique “viajando” pela *web*.