

Acessibilidade é proporcionar a interação de pessoas portadoras de deficiência ou mobilidade reduzida com o mundo.

## Padrões Criados pelo W3C

Você sabe quais são os benefícios de seguir os padrões criados pelo W3C?

**O uso dos padrões criados pelo W3C no desenvolvimento torna os *sites* mais visíveis aos motores de busca, como Google, Bing e Yahoo.**

Além disso, os usuários também são beneficiados, já que os padrões W3C facilitam o acesso dos *softwares* leitores de texto e [sintetizadores de voz](#).

**O HTML é considerado a linguagem base da internet. Ele é o [acrônimo](#) de *Hypertext Markup Language* e significa Linguagem de Marcação de Hipertexto, em português.**

O HTML está estruturado no conceito de hipertexto desde sua criação. Desse modo, elementos são conectados por palavras, imagens, vídeos, áudios e documentos, formando uma grande rede de informação.

A conexão dos elementos do hipertexto é algo imprevisível, já que uma fantástica comunicação de dados é disponibilizada. Essa comunicação enriquece o conhecimento e estabelece novas informações relacionadas.

Em 2004, um grupo de desenvolvedores das empresas Mozilla, Apple e Opera fundaram a WHATWG – Web Hypertext Application Technology Working Group. Com isso, teve início o desenvolvimento da nova versão do HTML, chamada de HTML5 atualmente.

No início de 2009, o W3C anunciou a primeira especificação do HTML5. Depois de alguns anos de trabalho, a versão final do HTML5 foi prometida para 2012.

Entre as alterações encontradas na nova versão do HTML5, estão:

- Novas APIs, para o desenvolvimento de gráficos bidimensionais, por exemplo
- Aprimoramento do uso *off-line*
- Melhoria na depuração de erros
- Controle embutido de conteúdo multimídia, entre outras propostas.

Os rótulos HTML são conhecidos como *tags*. As *tags* iniciam com o sinal de menor (<) e terminam com o sinal de maior (>).

# Linguagem HTML

Você já viu uma *tag*?

Como mencionamos, existem dois tipos básicos de *tags*:

- **Tag de abertura**
- **Tag de fechamento.**

As duas se diferenciam apenas pela barra /. A barra indica o fechamento da *tag*, de modo que a *tag* de abertura não tem a barra.

Vejamos:

**<aberturadatag> </fechamentodatag>**

## Doctype

Quando o navegador identifica o tipo de HTML, pode utilizar o tipo de renderização adequada.

Vamos adiante para descobrir como declarar o *doctype*!

Vejamos como é feita a declaração para os diferentes tipos de HTML.

### HTML 5

```
<!DOCTYPE html>
```

### HTML 4.01

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//PT-BR"  
http://www.w3.org/TR/html4/loose.dtd>
```

### XHTML 1.0

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

### Dica!

Você pode encontrar uma lista de como declarar seu *doctype* no [site do W3C](http://www.w3.org).

<https://www.w3.org/QA/2002/04/valid-dtd-list.html>

## Metatags

As *metatags* são *tags* que fornecem dados aos navegadores e motores de busca sobre o documento HTML em que estão inseridas.

As *metatags* não aparecem em sua tela, pois são informações específicas para os navegadores.

Vejamos alguns exemplos de *metatags*:

*Descrição da página*, *palavras-chaves*, *autor do documento*.

```
<meta name="description" content="Descrição da página">
<meta name="keywords" content="Palavras-chaves">
<meta name="author" content="nome do Autor">
```

Agora, vamos aprender a inserir comentários nos códigos HTML!

## Definição de Comentários

No HTML, os comentários possuem uma abertura e um fechamento, assim como as *tags*.

Os comentários são muito importantes para ajudar o desenvolvedor a realizar manutenções.

Vejamos como é fácil inserir um comentário em HTML:

```
<!-- seu comentário vai aqui -->
```

**Simples, não é? Agora, vamos conhecer as especificações e os atributos.**

## Especificações e Atributos

Os atributos são informações que inserimos nas *tags* para que elas tenham um comportamento desejado. Existem atributos que utilizamos em quase todas as *tags*. Já outros atributos podem ser exclusivos para um tipo de *tag*.

Os atributos são inseridos junto com a *tag* de abertura. Vejamos:

```
<nomedatag atributo1="valor do atributo 1" atributo2="valor do atributo 2"> </nomedatag>
```

Alguns exemplos de atributos são:

**Ainda não está claro? Não se preocupe! Veremos a aplicação dos atributos na prática.**

## Publicação do Documento

Carlos é um programador iniciante. Ele já sabe escrever códigos em HTML, mas ainda não descobriu como publicar os documentos na *web*.

E você? Já sabe como publicar documentos em HTML? Navegue pelas setas e veja algumas informações necessárias para que Carlos consiga executar sua publicação.

Para tornar os documentos em HTML disponíveis na *web*, precisamos fazer uma publicação ou *upload* do conteúdo.

A publicação dos documentos consiste no envio de todo o material produzido para um servidor. Isso significa criar um *site* que será disponibilizado por meio de um endereço *URL*.

O nome dado ao armazenamento dos arquivos é hospedagem. A hospedagem costuma ser feita por empresas especializadas.

Existem hospedagens gratuitas. No entanto, elas são vinculadas à propaganda ou podem gerar transtorno.

Ao publicar um material, é indicado que você adquira um domínio para tornar mais fácil a busca e a exposição do conteúdo publicado.

Agora, Carlos já pode executar a publicação de seus documentos na *web*!

## Publicação do Documento

Antes de começarmos a praticar, precisamos identificar cada *tag* da estrutura básica do HTML. Para isso, vamos analisar a estrutura que servirá de base para página que vamos desenvolver.

<!-- seu comentário vai aqui --> tag comentario

Estrutura básica

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Título da página</title>

</head>

<body>

</body>

</html>
```

Estrutura básica <head> <meta charset="utf-8"> <title>Título da página</title> </head> <body>  
</body></html>

- 1 = <!DOCTYPE html>

## Doctype

Observe que a primeira linha de nossa estrutura HTML é a declaração do *doctype*.

Você lembra que o *doctype* informa o tipo de documento que o navegador está acessando?

- 2 = <html>

## <html>

A *tag* <html> envolve, praticamente, todo o conteúdo da página.

Assim como o *doctype*, a *tag* <html> também informa ao navegador que o conteúdo dentro dela é HTML.

- 3 = <head>

## <head>

A *tag* <head> é a *tag* cabeça ou o cabeçalho da nossa página. Nesse sentido, a *tag* <head> desempenha um papel fundamental.

Dentro dessa *tag*, inserimos as informações que o navegador vai ler, como títulos, descrições, palavras-chaves e outras.

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

## <meta>

As *tags* meta ou *metatags* são *tags* que fornecem dados aos navegadores e aos motores de busca sobre o documento HTML em que estão inseridas. Tais *tags* não aparecem em sua tela, pois são informações específicas para os navegadores.

Você se lembra dos atributos? No caso apresentado, observe que a *tag* <meta> possui o atributo *charset* e o valor 'utf-8'. O atributo *charset* informa ao navegador o tipo de codificação que deve ser usada.

Com isso, precisamos informar ao navegador o tipo de codificação de caracteres que ele deve usar. O modelo de codificação UTF-8 pode representar qualquer caráter universal padrão Unicode.

- 5 = <title>
- 

## <title>

A *tag* <title>, dentro do *head*, é a *tag* que informa o título da página ao navegador.

Atualmente, a *tag* <title> é considerada umas das *tags* mais importantes.

Além de adicionar um título à aba no navegador, ela também auxilia os motores de busca a encontrarem o *site* no momento de uma pesquisa.

- 6 = <body>

## <body>

A *tag* *body* é o corpo da página. Desse modo, a *tag* <body> é responsável por expor o conteúdo para o usuário final.

# Escrita de Código

Já está na hora de começarmos a praticar! Mas, primeiro, você sabe que ferramenta usar para começar a escrever códigos em HTML?

**Códigos HTML podem ser escritos em qualquer editor de texto. Existem muitos editores grátis para *download* na internet.**

Agora, usaremos o **Bloco de Notas** (editor de texto nativo do sistema operacional Windows 10). Caso você não tenha essa versão do Windows, não se preocupe. Versões anteriores também possuem esse editor de texto instalado.

## Demonstração

Acesse a demonstração **Estrutura do código HTML**, clicando no ícone.

## Separadores – Quebra de Linha

No desenvolvimento de conteúdos, é comum haver textos com grande quantidade de linhas. De modo geral, os navegadores quebram as linhas automaticamente. No entanto, pode ser necessário quebrar uma linha em determinado ponto.

Nesse caso, usamos o elemento `<br>` para quebrar as linhas. No local em que inserirmos a tag `<br>`, o texto pulará para a próxima linha.

**A tag `<br>` é uma tag nula. Dessa forma, não precisamos inserir o `</br>` para o fechamento.**

## Separadores – Parágrafos

Já sabemos que a utilização de tags `<br>` resulta em quebras de linhas. No entanto, não é recomendado usar muitas quebras de linha quando trabalhamos com uma grande quantidade de texto.

Como vimos, a tag `<br>` deve ser usada em determinados pontos. Quando trabalhamos com **textos longos** ou **com grande quantidade de parágrafos**, devemos usar a tag `<p>`.

A tag `<p>` tem a função de separar blocos de texto e significa **parágrafo**. Essa tag deve ser inserida da seguinte maneira:

## Separadores – Linha Horizontal

O último separador que iremos apresentar é a **linha horizontal**. Apesar de ser funcional, a linha horizontal não é muito utilizada atualmente.

Para inserir uma linha horizontal, basta adicionar a tag `<hr>`. Desse modo, onde a tag `<hr>` for inserida, será gerada uma linha.

**Você verá que o texto também será quebrado com a inserção da linha horizontal. Por conta disso, tome cuidado com a inclusão dessa tag no**

Assim como a tag `<br>`, a tag `<hr>` também é uma tag **nula**. Isso significa que não há necessidade de fechá-la.

## Formatações de Cabeçalho e Textos

O HTML possui uma grande quantidade de tags voltada para formatação de textos.

A função das tags de formatação é **facilitar e melhorar visualmente as páginas web**.

**As tags de formatação contribuem para um desenvolvimento mais rápido e eficiente de um conte**

Vamos conhecer as principais tags de formatação a seguir!

# Títulos – Cabeçalhos

Os títulos no HTML são divididos em seis níveis – de 1 a 6. Quando inseridos na página, os títulos se destacam dos demais textos pelo **tamanho da letra**.

No HTML, a *tag* que representa os títulos é a **<h>**. Observe a composição da *tag* **<h>**:

Exemplo: **<h1>** Seu Título **</h1>**

- **<h1>** Título nível 1 **</h1>**
- **<h2>** Título nível 2 **</h2>**
- **<h3>** Título nível 3 **</h3>**
- **<h4>** Título nível 4 **</h4>**
- **<h5>** Título nível 5 **</h5>**
- **<h6>** Título nível 6 **</h6>**

**Ainda está confuso? Vamos ver como aplicar tudo isso na prática. Negrito – bold, em inglês**

**Existem duas formas de aplicar o negrito com tags. Cada forma de aplicação possui uma função:**

**<b>** – Segundo a descrição do W3C, a tag **<b>** deve ser usada para destacar uma palavra.

**<strong>** – A tag **<strong>** também destaca a palavra. No entanto, essa tag deve ser usada para representar algo importante, sério ou urgente

## Itálico

**Assim como o negrito (bold), o itálico também possui duas formas de ser aplicado.**

**<i>** – Segundo as orientações do W3C, a tag **<i>** deve ser usada em termos técnicos. Por exemplo: O nome da flor é Orquídea.

**<em>** – Já a tag **<em>** deve ser usada para dar ênfase a uma palavra. Por exemplo: Ela não passou por aqui ontem.



### Sublinhado, subscrito e sobrescrito

As tags para sublinhar, subscrever e sobrescrever são:

Tag `<u>` – A tag `<u>` adiciona sublinhado ao texto.

Tag `<sub>` – O texto dentro da tag `<sub>` se torna subscrito (letras pequenas abaixo de um texto). Ex: H<sub>2</sub>O.

Tag `<sup>` – Coloca o texto como sobrescrito (letras pequenas acima de um texto). Ex: X<sup>4</sup>.

### tags html

`<sub>`

O texto dentro da tag `<sub>` se torna subscrito.

`<sup>`

Coloca o texto como sobrescrito.

`<em>`

Aplica itálico ao texto. É usada para dar ênfase a uma palavra.

## Aplicação de Cores – *Background* e *Foreground*

A aplicação de cores no texto, no fundo de páginas e nos *links* é de grande ajuda para distinguir os textos importantes nas páginas *web*. Para isso, passaremos a utilizar **atributos**, em vez de *tags*.

Vamos iniciar com o atributo **bgcolor**. Tal atributo tem como função adicionar cor de fundo (*background*) onde é inserido. Usaremos a tag `<body>` para exemplificar.

O atributo **bgcolor** é adicionado da seguinte maneira:

`<body bgcolor = "COR"> </body>`

Já as cores podem ser inseridas de duas formas:

- Por seus nomes em inglês.
- Por seus códigos hexadecimais.

código hexadecimal	Nome
#FFFF00	Yellow
#808000	Olive
#FFFFFF	White
#000000	Black
#FF0000	Red
#0000FF	Blue
#00FF00	Green

## Cores

Veja outras cores no *site*: [www.w3schools.com/colors/colors\\_hex.asp](http://www.w3schools.com/colors/colors_hex.asp).

No *site* de busca, você também pode procurar por: **tabela de cores em HTML**. Você terá várias opções, sempre com o código e o nome da cor.

## Foreground

Utiliza a tag `<font> >/font>`

`<font color="cor"> </font>`

## Tamanho e Estilo do Texto

Já sabemos como modificar a cor do fundo e a cor da fonte. Agora, vamos mudar o **tamanho** e o **estilo** do nosso texto.

Vejamos:

`<font size= "Tamanho da fonte"></font>`

Você pode definir o tamanho da fonte de várias formas diferentes.

Por padrão, usamos o **px**. No entanto, você pode utilizar outras unidades de medida. Mais adiante, falaremos sobre as medidas.

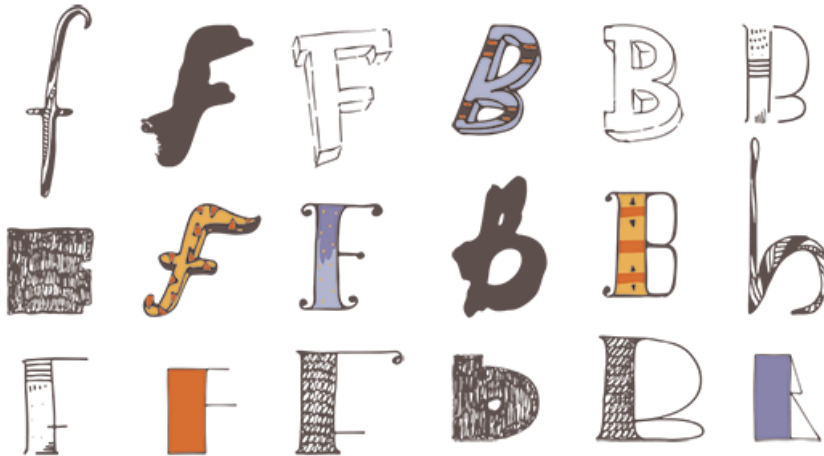
`<font size = "20px"> seu texto </font>`

## Tamanho e Estilo do Texto

Por meio da *tag* **<font>** e do atributo **face**, também conseguimos alterar o estilo da fonte que o texto terá.

**<font face = "Verdana"> seu texto </font>**

Vejamos como incluir o atributo **face** na *tag* **<font>**:



Com a chegada do **HTML5** e do **CSS3**, o W3C orienta que qualquer aplicação de estilos seja feita por meio das **folhas de estilo** ou **CSS**.

Mais adiante, veremos como aplicar o CSS.

## Listas

As listas são um recurso muito utilizado no HTML. Além do uso de listas na criação de tópicos, também é comum aplicar listas em outras funções, como em menus.

As listas podem ser caracterizadas de três maneiras. Clique nos ícones para conhecê-las.

### Lista não ordenada

As listas não ordenadas apresentam itens **sem uma ordem sequencial**. Podemos afirmar que são listas de pequenos marcadores.

Para criar uma lista não ordenada, basta adicionar a *tag* **<ul>**. Já os itens são incluídos a partir da *tag* **<li>**. Vejamos um exemplo:

**<ul>**

**<li> Fundação</li>**

**<li> Bradesco</li>**

```
<li> Brasil</li>
</ul>
```

## Lista ordenada

Conhecida também como lista numerada, a lista ordenada é utilizada para representar uma sequência ou numeração. A inserção da lista ordenada é feita com as *tags* **<ol>** e **<li>**.

Vejamos:

```
<ol>
  <li> Fundação</li>
  <li> Bradesco</li>
  <li> Brasil</li>
</ol>
```

## Lista de Definição

A lista de definição é constituída por um **termo** e uma **descrição**. A criação desse modelo de lista utiliza três elementos do HTML. Vejamos:

**<dl>** – cria a lista.  
**<dt>** – define o termo.  
**<dd>** – adiciona a definição.

## Inserção de *Links*

Conhecido também como *hiperlink*, o *link* é o segredo por trás da internet. A função do *link* é estabelecer a ligação com outros *sites*, páginas, imagens, sons e outros elementos.

Os *links* podem direcionar o usuário para outra parte de uma mesma página, para outra página no próprio *site* ou para um arquivo.

Para criar um *link*, precisamos de duas informações muito importantes:

- O **local de destino** de seu *link* (*target*)
- O **local de origem**, em que devemos clicar.

## Inserção de *Links*

- Quando inserimos um *link*, a *tag* **<a>** faz o papel de direcionar a navegação entre as páginas. O elemento **a** da *tag* significa **action** e realiza uma ação quando acionado.
- Para criar o *link*, o atributo **href** é obrigatório. Tal atributo guarda o novo endereço para onde o usuário será direcionado.
  - **<a href="Endereço destino"> Texto para seu link </a>**

- Vamos inserir um *link* em nossa página?

## Inserção de Imagens

O uso de imagens em páginas *web* é indispensável tanto por uma questão estética quanto para compreensão do conteúdo. O HTML possui uma *tag* específica para a inclusão de imagens.

A *tag* **<img>** possibilita trabalhar com imagens. Os atributos dessa *tag* auxiliam na definição do tamanho, da posição e da descrição da imagem.

**<img src= "Local onde está a imagem">**

## Tabelas

As tabelas sempre foram de extrema importância no HTML. No dia a dia de um profissional que trabalha com desenvolvimento *web*, o uso da tabela já foi essencial.

A facilidade ao incluir imagens e textos tornava o uso das tabelas bem menos trabalhoso, sobretudo na organização dos *layouts*.

Com a chegada dos aparelhos móveis, o uso de tabelas na composição de *layout* diminuiu consideravelmente. Ainda assim, as tabelas são usadas para exposição de dados e criação de relatórios.

Para criar uma tabela, precisamos de três *tags* **<table>**, **<tr>** e **<td>**.

Não há como criar uma tabela sem usar a *tag* **<table>**. *Table* significa tabela, em inglês.

Vamos praticar!?

## Tabelas

Agora, iremos criar uma tabela básica. Com esse exercício, vamos entender o funcionamento e a montagem de uma tabela em HTML.

Como vimos, para criar uma tabela, são três as *tags* básicas. Clique nas *tags* para vê-las representadas na estrutura da tabela a seguir.

- [\*\*<table>\*\* - Tabela](#)
- [\*\*<td>\*\* - Coluna](#)
- [\*\*<tr>\*\* - Linha](#)

linha 1, célula 1 linha 1, célula 2 linha 2, célula 1 linha 2, célula 2

Com essas três *tags* básicas, conseguiremos criar a seguinte tabela:

Aula dia 03/05/2021

```
<table border="1">

  <tr>

    <th>Titulo 1</th> => define titulo da coluna 1

    <th>Titulo 2</th> => define o titulo da coluna 2

  </tr>

  <tr>

    <td> Fundação Bradesco </td>

    <td> Apostila </td>

  </tr>

  <tr>

    <td>HTML</td>

    <td>CSS</td>

  </tr>

</table>
```

## Formulários – *Forms*

A interação de um *site* com o usuário pode ser feita de diversas maneiras. Por exemplo, uma animação que chama atenção para um fato, um botão que aciona músicas ou um formulário que envia mensagens, *e-mails*, *chats* e dados ao servidor.

Você sabe o que é um formulário?

**Às vezes, é preciso enviar informações ao servidor. Os formulários cumprem esse papel. Por meio dos formulários, os dados são coletados e enviados ao servidor posteriormente.**

De modo geral, o formulário é composto de:

- *Text area* – campo de texto.
- *Radio buttons* – botões radiais.
- *Checkboxes* – caixas de seleção.
- Outros *widgets* – componentes.

### Dica rápida!

Quando construímos um formulário, devemos lembrar que quanto menor for o número de campos a serem preenchidos, maior será a chance de manter o usuário no *site* ou de converter o interesse do usuário em cadastro.

## Formulários – Forms

Para criar um formulário, temos de usar uma *tag* específica. A *tag* **<form>** é a responsável por essa função. Vejamos:

```
<form action=" " method = " "> </form>
```

Os atributos na *tag* **<form>** não são obrigatórios. No entanto, os padrões recomendam a definição de valores para os dois atributos a seguir:

### Action

O atributo *action* define o destino dos dados que foram coletados. De modo geral, o valor inserido nesse atributo é um URL.

### Method

O atributo *method* tem a função de definir o método do HTTP que será usado para enviar os dados – GET ou POST.

Quando clicamos em um *link* da *tag* **<a>**, o *link* utiliza o método GET.

O método GET possui capacidade para 1.024 caracteres e é utilizado para passar pequenas informações ao servidor.



1) Observe que inserimos duas *tags* **<input>** com *types* diferentes para demonstrar a finalidade de cada uma. O atributo *type* da *tag* **<input>** é o mais importante. Tal atributo define como o elemento deve comportar-se

2) Também inserimos uma *tag* **<textarea>** para verificar a diferença dela com a *tag* **<input>** do tipo *text*.

3) Por fim, inserimos uma *tag* **<label>** para cada elemento, a fim de facilitar a identificação de cada um.

Para criar um botão, usaremos a *tag* **<button>**. Apesar de parecer óbvio, a *tag button* pode apresentar mais de um comportamento. O comportamento da *tag button* depende de seu *type*, que pode ser de três tipos

### **Type submit**

O *type submit* envia os dados do formulário à página-destino do atributo *actions*.

Ex:

```
<button type="submit">cadastrar</button>
```

## Type checkbox

De modo geral, o **input** com atributo **type** e valor **checkbox** são usados em questões com opções de escolha – sim ou não, por exemplo.

Podemos selecionar mais de um *checkbox*. Isso torna o valor *checkbox* muito recomendado para opções de múltiplas escolhas.

Ex:

```
<input type="checkbox" name="opcao" value="selectA" checked>clique para selecionar o A<br>
```

```
<input type="checkbox" name="opcao" value="selectB">clique para selecionar o B <br>
```

=> os atributos "name" e "value" são responsáveis para informar "actions" aos dados que devem ser enviados ao servidor

=> Caso o atributo "checked" esteja aplicado o elemento (espaço para selecionar) já aparecerá selecionado.

## Type Radio

Agora vamos entender como o *Type Radio* se comporta?

=> type radio é similar a checkbox porém somente 1 opção poderá ser selecionada

Ex:



```
<input type="radio" name="opcao" value="selectA">Masculino  
<input type="radio" name="opcao" value="selectB" checked>Feminino
```

## Tag `<select>`

A tag `<select>` é formada por um conjunto de opções `<option>`. Tais opções fornecem ao usuário uma lista organizada no formato *drop-down*.

Vejamos:

```
<select>  
  <option value="verde">Verde</option>  
  <option value="vermelho">Vermelho</option>  
  <option value="azul">Azul</option>  
  <option value="alpha">Alpha</option>  
</select>
```

A tag `<select>` é uma *tag* não nula. Isso significa que ela deve conter abertura `<select>` e fechamento `</select>`, além das opções inseridas. Veja um exemplo a seguir:

```
<option value="valor a ser enviado ao servidor"> Texto Aqui </option>
```

## Type Hidden – Oculto



O *type hidden* torna o campo *input* invisível para o usuário, permitindo, dessa forma, a passagem de dados para o servidor sem a necessidade de interação com o usuário.

Ex:

```
<label>campo visível</label>
```

```
<input type="test"><br>
```

```
<label>campo invisível</label>
```

```
<input type="hidden" name="pais" value="Brasil"> => campo "hidden" fica invisível ao usuário
```

## Composição de Página HTML

Ao visitar um *site*, as páginas não são apenas textos e formulários.

**Sem dúvida, a busca pela melhor experiência do usuário é a maior preocupação de empresas e profissionais da área de desenvolvimento *web*.**

Agora que você conhece as *tags* básicas para a composição de uma página HTML, é essencial que sua página tenha mais do que uma boa estrutura. A página deve ter, também, uma boa aparência. Mas isso já é assunto para o próximo tópico. Preparado?

Antes de aprender a modificar a aparência de uma página HTML, que tal reforçar o que aprendemos sobre a estrutura por meio de exercícios?

## Exercícios

Em HTML, existem *tags* que são consideradas nulas, ou seja, que possuem um autofechamento. **(V)**

As *metatags* são *tags* que fornecem dados aos navegadores e motores de busca sobre o documento HTML em que estão inseridas. **(V)**

O Doctype é uma declaração com o tipo de documento de sua página *web*, pois sem a especificação explícita, o navegador não seria capaz de renderizar o documento HTML. **(F)**

Os atributos são informações inseridas nas *tags* para que tenham o comportamento desejado, sendo os mesmos para todas, permitindo a melhor definição do código HTML.

**(F)**

