

```
mirror_mod.use_z = True  
operation == "MIRROR_Z"  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True
```

```
selection at the end -add  
mirror_ob.select= 1  
mirror_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier_name))  
mirror_ob.select = 0  
= bpy.context.selected_object  
data.objects[one.name].select  
print("please select exactly
```

```
-- OPERATOR CLASSES ----
```

Aula de Scripts

Agenda

- **Conhecendo o professor**
- **Apresentando a Disciplina**
- **Conhecendo os alunos**
- **Introdução**



Conhecendo o Professor



<https://www.linkedin.com/in/junior-eloy/>





Conhecendo os Alunos

```
mirror_mod.use_z = True  
operation == "MIRROR_Z"  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True
```

```
selection at the end -add  
mirror_ob.select= 1  
mirror_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier_name))  
mirror_ob.select = 0  
= bpy.context.selected_object  
data.objects[one.name].select  
print("please select exactly
```

```
-- OPERATOR CLASSES ----
```

HTML, CSS e JS: Introdução

Front-end



Back-end



HTML: O Esqueleto

Estrutura Básica

HTML fornece a estrutura fundamental de qualquer página web, incluindo elementos como cabeçalhos, parágrafos e links.

Elementos e Tags

Usa tags como `<div>`, `` e `<a>` para organizar o conteúdo. Cada tag desempenha um papel específico.

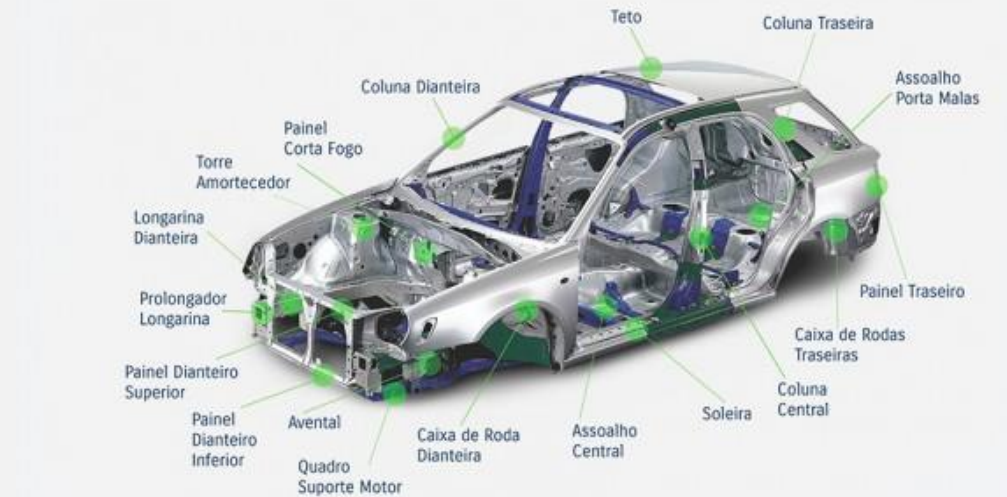
Semântica

Tags semânticas ajudam na compreensão do conteúdo. Exemplos são `<header>`, `<footer>`, e `<article>`.



HTML

Imagine a carcaça de uma carro
O HTML é como o esqueleto do carro.



CSS: Estilização

Seletores e Especificidade

Seletores definem quais elementos o CSS aplica. A especificidade determina prioridade entre regras.

O que é CSS?

CSS (Cascading Style Sheets) é usado para definir a apresentação de documentos HTML.

Propriedades CSS

Permite controlar cor, fonte, espaçamento e layout. Exemplos: color, font-size, margin.



CSS

O estilo do carro

Com cores, estilo, deixando-o mais bonito.



ThePhoto de PhotoAuthor está licenciada sob CCYISA.

JS: Funcionalidade

O que é JavaScript?

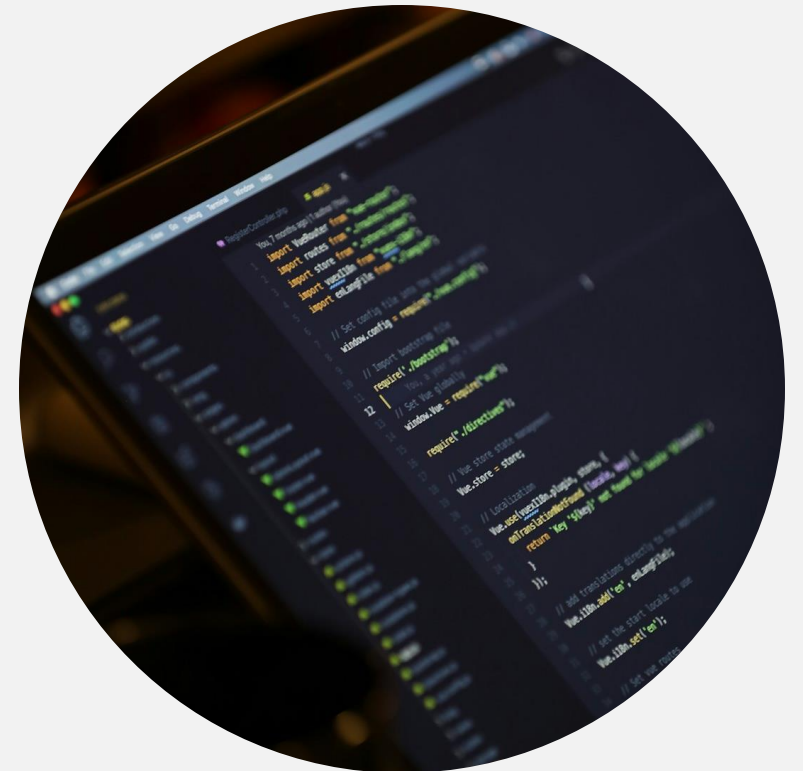
JavaScript é uma linguagem de programação que adiciona interatividade a páginas web.

Manipulação do DOM

Permite alterar o conteúdo e a estrutura da página em tempo real. Exemplo: modificar texto ou estilo.

Eventos

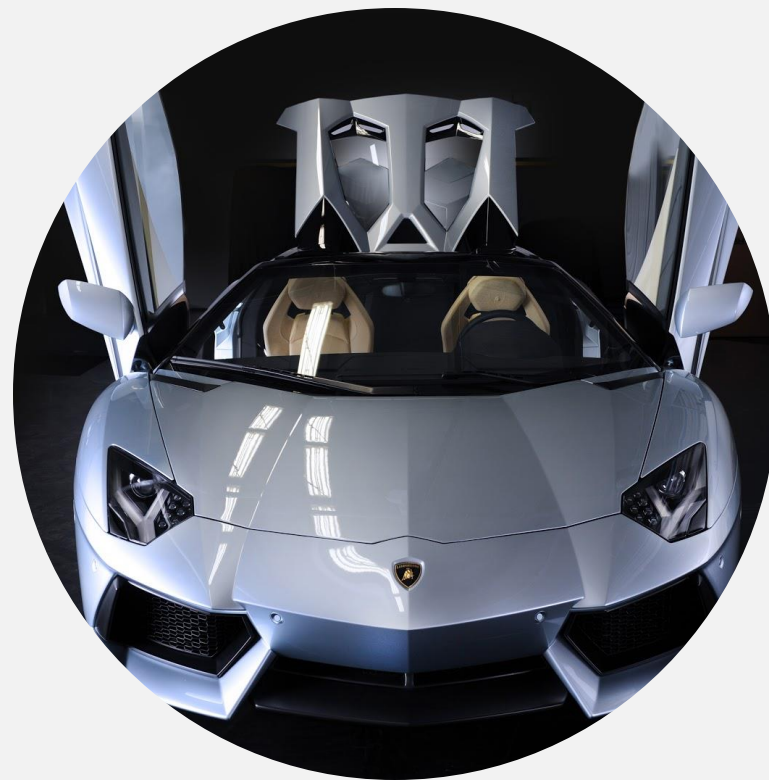
JS responde a interações do usuário, como cliques e teclas pressionadas, para ativar funções.



JS

Funcionalidades do carro

Abrir a porta, abrir os vidros, sistema de som.



ThePhoto de PhotoAuthor está licenciada sob CCYISA.

NODEJS

JavaScript no Servidor

Node.js permite que você utilize JavaScript não apenas para criar sites interativos no navegador, mas também para desenvolver a lógica do servidor. Isso facilita o desenvolvimento com uma única linguagem de programação em todo o stack.

NodeJS

O potente motor

Node é como o motor e o sistema de controle central do carro. É o que faz o carro andar, é como o motor e o sistema de controle central do carro. É o que faz o carro andar, respondendo aos comandos do motorista, controlando a velocidade e garantindo que todas as partes mecânicas estão funcionando harmoniosamente.



ThePhoto de PhotoAuthor está licenciada sob CCYISA.


```
mirror_mod.use_z = True  
operation == "MIRROR_Z"  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True
```

```
#selection at the end -add  
mirror_ob.select= 1  
mirror_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier_ob))  
mirror_ob.select = 0  
= bpy.context.selected_object  
data.objects[one.name].select  
print("please select exactly
```

```
-- OPERATOR CLASSES ----
```

Plano de Ensino

Conhecendo a Disciplina

Objetivos do Curso

Propõe desenvolver competências em scripting, automação e solução de **problemas** utilizando linguagens de programação.

Conteúdo Programático

O curso abrange tópicos como, **JavaScript, HTML, CSS e NodeJS**.

Metodologia de Avaliação

Avaliações serão compostas por **2 provas e 2 trabalhos**, visando a aplicação do conhecimento.



Como serei avaliado?

Calculo

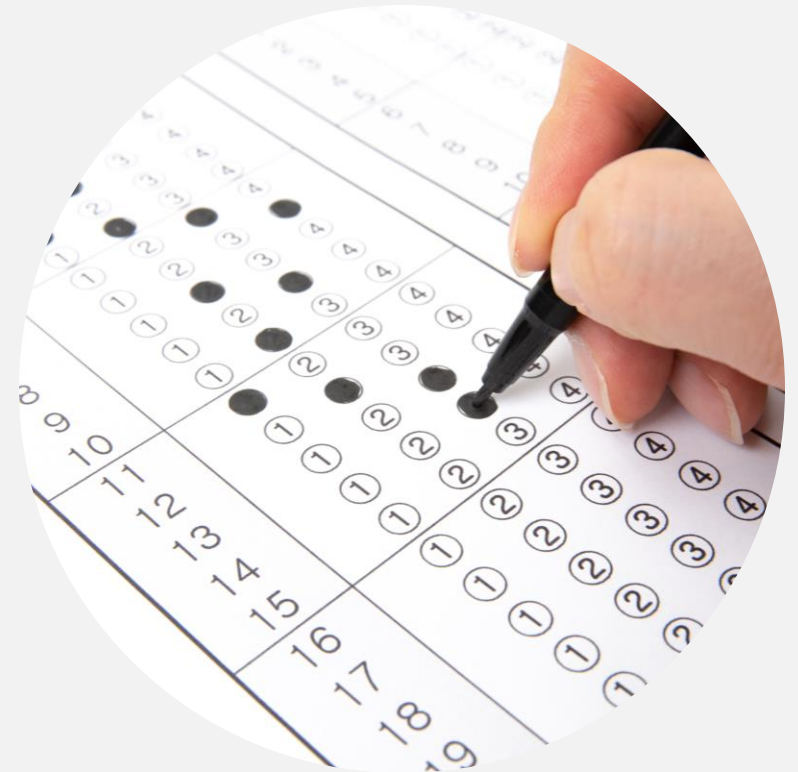
$$(((P1 * 0.3) + (T1 * 0.7)) / 2) + (((P2 * 0.3) + (T2 * 0.7)) / 2)$$

1:

HTML, CSS e JS

2:

NodeJS e TypeScript



```
mirror_mod.use_z = True  
operation == "MIRROR_Z"  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True
```

```
#selection at the end -add
```

```
mirror_ob.select= 1
```

```
mirror_ob.select=1
```

```
context.scene.objects.active
```

```
("Selected" + str(modifier_ob
```

```
mirror_ob.select = 0
```

```
= bpy.context.selected_object
```

```
data.objects[one.name].select
```

```
print("please select exactly
```

```
-- OPERATOR CLASSES ----
```

HTML

Visão geral do HTML 5

- De acordo com o W3C a Web é baseada em 3 pilares:
 1. Um esquema de nomes para localização de fontes de informação na Web, esse esquema chama-se URI.
 2. Um Protocolo de acesso para acessar estas fontes, hoje o HTTP.
 3. Uma linguagem de Hypertexto, para a fácil navegação entre as fontes de informação: o HTML.

Hypertexto

- HTML é uma abreviação de Hypertext Markup Language - Linguagem de Marcação de Hypertexto.
- HTML é uma linguagem para publicação de conteúdo (texto, imagem, vídeo, áudio e etc) na Web.
- HTML é baseado no conceito de Hipertexto. Hipertexto são conjuntos de elementos – ou nós – ligados por conexões.

O Criador

- Desenvolvido originalmente por Tim Berners-Lee o HTML ganhou popularidade quando o Mosaic - browser desenvolvido por Marc Andreessen na década de 1990 - ganhou força. A partir daí, desenvolvedores e fabricantes de browsers utilizaram o HTML como base, compartilhando as mesmas convenções.

O começo e a interoperabilidade

- Entre 1993 e 1995, o HTML ganhou as versões HTML+, HTML2.0 e HTML3.0, onde foram propostas diversas mudanças para enriquecer as possibilidades da linguagem.
- Em 1997, o grupo de trabalho do W3C responsável por manter o padrão do código, trabalhou na versão 3.2 da linguagem, fazendo com que ela fosse tratada como prática comum.
- Desde o começo o HTML foi criado para ser uma linguagem independente de plataformas, browsers e outros meios de acesso. Interoperabilidade significa menos custo. Você cria apenas um código HTML e este código pode ser lido por diversos meios, ao invés de versões diferentes para diversos dispositivos.

Versões HTML

Version	Year
HTML	1991
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML	2000
HTML5	2014

WHAT Working Group

- Enquanto o W3C focava suas atenções para a criação da segunda versão do XHTML, um grupo chamado Web Hypertext Application Technology Working Group ou WHATWG trabalhava em uma versão do HTML que trazia mais flexibilidade para a produção de websites e sistemas baseados na web.
- O WHATWG (<http://www.whatwg.org/>) foi fundado por desenvolvedores de empresas como Mozilla, Apple e Opera em 2004. Eles não estavam felizes com o caminho que a Web tomava e nem com o rumo dado ao XHTML. Por isso, estas organizações se juntaram para escrever o que seria chamado hoje de HTML5.

O HTML5 e suas mudanças

- Quando o HTML4 foi lançado, o W3C alertou os desenvolvedores sobre algumas boas práticas que deveriam ser seguidas ao produzir códigos client-side.
- Assuntos como a separação da estrutura do código com a formatação e princípios de acessibilidade foram trazidos para discussões e à atenção dos fabricantes e desenvolvedores.
- HTML4 ainda não trazia diferencial real para a semântica do código e também não facilitava a manipulação dos elementos via Javascript ou CSS.

O que é o HTML5?

- O HTML5 é a nova versão do HTML4.
- O WHATWG define as regras de marcação que usaremos no HTML5, eles também definem APIs que formarão a base da arquitetura web.
- Essas APIs são conhecidas como DOM (Modelo de Objetos do Documento).
- Um dos principais objetivos do HTML5 é facilitar a manipulação do elemento possibilitando o desenvolvedor a modificar as características dos objetos de forma não intrusiva e de maneira que seja transparente para o usuário final.

O que é o HTML5?

- HTML5 fornece ferramentas para a CSS e o Javascript fazerem seu trabalho da melhor maneira possível.
- HTML5 permite por meio de suas APIs a manipulação das características destes elementos, de forma que o website ou a aplicação continue leve e funcional.
- HTML5 modifica a forma de como escrevemos código e organizamos a informação na página. Seria mais semântica com menos código. Seria mais interatividade sem a necessidade de instalação de plugins e perda de performance.

O desenvolvimento modular

- Antigamente, para que uma nova versão do HTML ou do CSS fosse lançada, todas as ideias listadas na especificação deveriam ser testadas e desenvolvidas para então serem publicadas para o uso dos browsers e os desenvolvedores.
- Esse método foi mudado com o lançamento do HTML5 e o CSS3. A partir de agora, as duas tecnologias foram divididas em módulos.
- Isso quer dizer que a comunidade de desenvolvedores e os fabricantes de browsers não precisam esperar que todo o padrão seja escrito e publicado para utilizarem as novidades das linguagens.

Estrutura básica, doctype e charsets

```
1
2  <!-- O Doctype não é uma tag do HTML, mas uma instrução para que o browser
3  | tenha informações sobre qual versão de código a marcação foi escrita, ou seja, HTML5 -->
4  <!DOCTYPE html>
5
6  <!-- O código HTML é uma série de elementos em árvore onde alguns elementos são filhos de
7  | outros e assim por diante. O atributo LANG é necessário para que os browsers
8  | saibam qual a linguagem principal do documento. -->
9  <html lang="pt-br">
10
11  <!--A Tag HEAD é onde fica toda a parte inteligente da página. No HEAD ficam os metadados.
12  | Metadados são informações sobre a página e o conteúdo ali publicado.-->
13  <head>
14
15  | <!--Metatag Charset. No nosso exemplo há uma metatag responsável por chavear qual tabela de
16  | caracteres a página está utilizando.-->
17  | <meta charset="UTF-8">
18  | <meta name="viewport" content="width=device-width, initial-scale=1.0">
19  | <meta http-equiv="X-UA-Compatible" content="ie=edge">
20  | <title>Exemplo Aula 1</title>
21  </head>
22  <body>
23  | <h1>Olá Mundo</h1>
24  </body>
25  </html>
```

Modelos de Conteúdo

- Há pequenas regras básicas que estão no HTML desde o início. Estas regras definem onde os elementos podem ou não estar. Se eles podem ser filhos ou pais de outros elementos e quais os seus comportamentos.
- Dentre todas as categorias de modelos de conteúdo, existem dois tipos de elementos: **elementos de linha e de bloco**.
- Os **elementos de linha** marcam, na sua maioria das vezes, texto.
- Alguns exemplos: **a, strong, em, img, input, abbr, span**.

Modelos de Conteúdo

- Os **elementos de blocos** são como caixas, que dividem o conteúdo nas seções do layout.
- Abaixo segue algumas premissas que você precisa conhecer:
 - Os elementos de linha podem conter outros elementos de linha, dependendo da categoria que ele se encontra. Por exemplo: o elemento `a` não pode conter o elemento `label`.
 - Os elementos de linha nunca podem conter elementos de bloco.
 - Elementos de bloco sempre podem conter elementos de linha.
 - Elementos de bloco podem conter elementos de bloco, dependendo da categoria que ele se encontra. Por exemplo, um parágrafo não pode conter um `DIV`. Mas o contrário é possível.

Categorias

- Cada elemento no HTML pode ou não fazer parte de um grupo de elementos com características similares.
- As categorias estão a seguir. Manteremos os nomes das categorias em inglês para que haja um melhor entendimento:
 - Metadata content
 - Flow content
 - Sectioning content
 - Heading content
 - Phrasing content
 - Embedded content
 - Interactive content

Metadata content

- Este conteúdo vem antes da apresentação, formando uma relação com o documento e seu conteúdo com outros documentos que distribuem informação por outros meios.
- Os elementos que compõe a categoria Metadata são:
 - base
 - command
 - link
 - meta
 - noscript
 - script
 - style
 - title

Flow content

- A maioria dos elementos utilizados no body e aplicações são categorizados como Flow Content. São eles:
 - a
 - abbr
 - address
 - area (se for um decendente de um elemento de mapa)
 - article
 - aside
 - audio
 - b
 - bdo
 - blockquote
 - br
 - button

Flow content

- canvas
- cite
- code
- command
- datalist
- del
- details
- dfn
- div
- dl
- em
- embed
- fieldset

- figure
- footer
- form
- h1
- h2
- h3
- h4
- h5
- h6
- header
- hgroup
- hr
- i

Flow content

- iframe
- img
- Input
- ins
- kbd
- keygen
- label
- link (Se o atributo itemprop for utilizado)
- map
- mark
- math
- menu
- meta (Se o atributo itemprop for utilizado)
- meter
- nav
- noscript
- object
- ol
- output
- p
- pre
- progress
- q
- ruby

Flow content

- samp
- script
- section
- select
- small
- span
- strong
- style (Se o atributo scoped for utilizado)
- sub
- sup
- svg
- table
- textarea
- time
- ul
- var
- video
- wbr
- text

Sectioning content

- Estes elementos definem um grupo de cabeçalhos e rodapés. Basicamente são elementos que juntam grupos de textos no documento.
 - article
 - aside
 - nav
 - section

Heading content

- Os elementos da categoria Heading definem uma seção de cabeçalhos, que podem estar contidos em um elemento na categoria Sectioning.
 - h1
 - h2
 - h3
 - h4
 - h5
 - h6
 - hgroup

Phrasing content

- Fazem parte desta categoria elementos que marcam o texto do documento, bem como os elementos que marcam este texto dentro do elemento de parágrafo.
 - a
 - abbr
 - area (se ele for descendente de um elemento de mapa)
 - audio
 - b
 - bdo
 - br
 - button
 - canvas
 - cite
 - code

Phrasing content

- command
- datalist
- del (se ele contiver um elemento da categoria de Phrasing)
- dfn
- em
- embed
- i
- iframe
- img
- input
- ins (se ele contiver um elemento da categoria de Phrasing)
- kbd
- keygen
- label
- link (se o atributo itemprop for utilizado)
- map (se apenas ele contiver um elemento da categoria de Phrasing)
- mark
- math

Phrasing content

- meta (se o atributo itemprop for utilizado)
- meter
- noscript
- object
- output
- progress
- q
- ruby
- samp
- script
- select
- small
- span
- strong
- sub
- sup
- svg
- textarea
- time
- var
- video
- wbr
- text

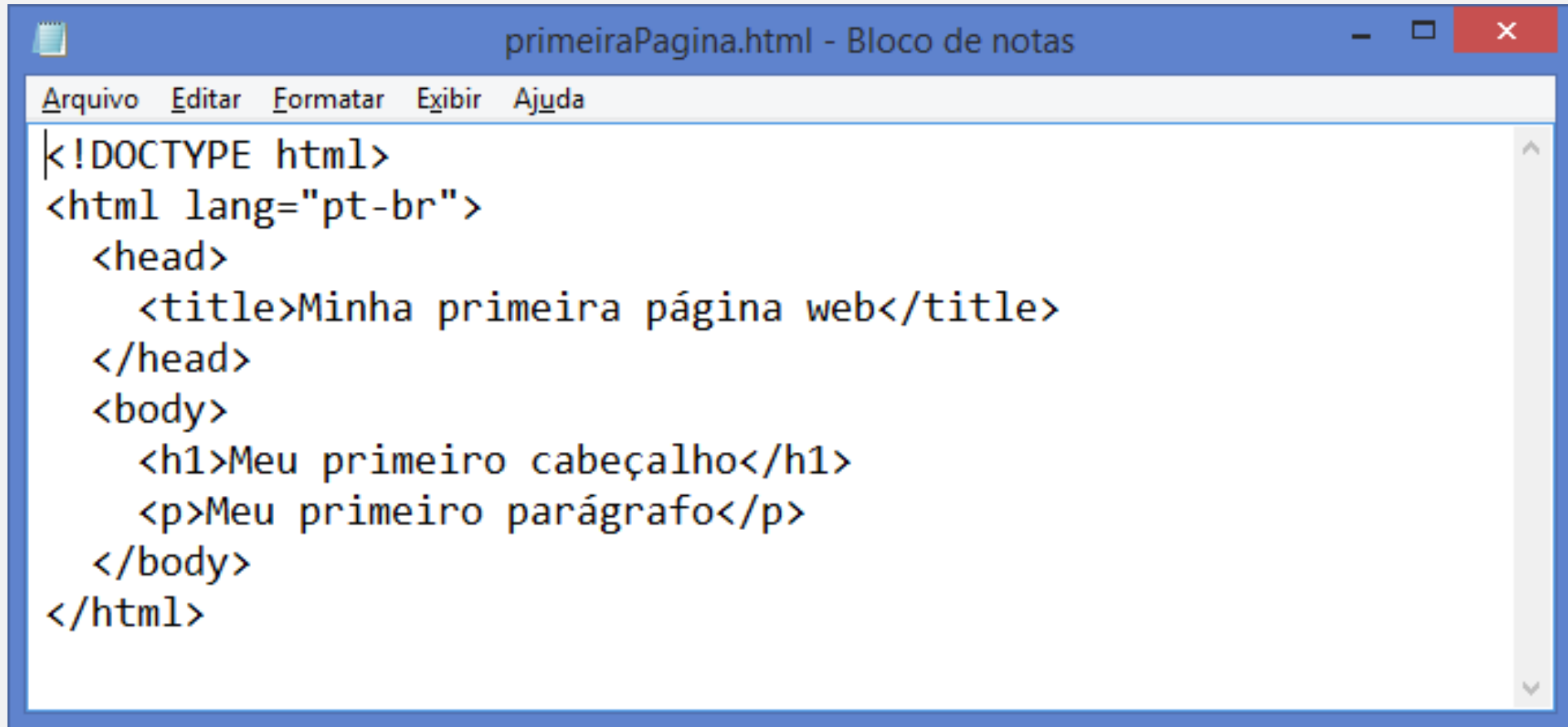
Embedded content

- Na categoria Embedded, há elementos que importam outra fonte de informação para o documento.
 - audio
 - canvas
 - embed
 - iframe
 - img
 - math
 - object
 - svg
 - video

Interactive content

- Interactive Content são elementos que fazem parte da interação de usuário.
 - a
 - audio (se o atributo control for utilizado)
 - button
 - details
 - Embed
 - iframe
 - img (se o atributo usemap for utilizado)
 - input (se o atributo type não tiver o valor hidden)
 - keygen
 - label
 - menu (se o atributo type tiver o valor toolbar)
 - object (se o atributo usemap for utilizado)
 - select
 - textarea
 - video (se o atributo control for utilizado)

Um documento HTML simples

A screenshot of a Windows Notepad application window. The title bar reads "primeiraPagina.html - Bloco de notas". The menu bar includes "Arquivo", "Editar", "Formatar", "Exibir", and "Ajuda". The text area contains the following HTML code:

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <title>Minha primeira página web</title>
  </head>
  <body>
    <h1>Meu primeiro cabeçalho</h1>
    <p>Meu primeiro parágrafo</p>
  </body>
</html>
```

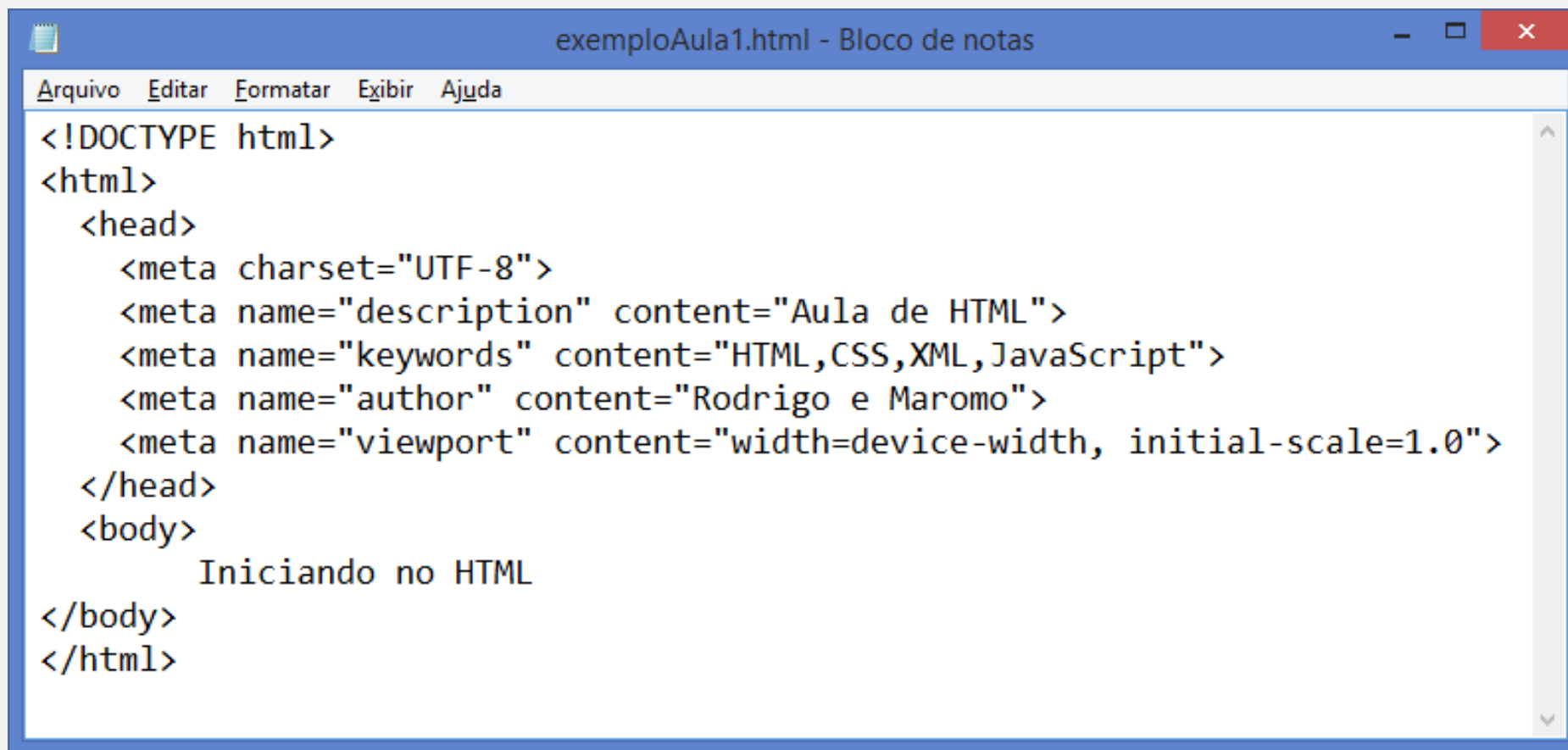
A tag <meta>

É muito importante conhecer e saber utilizar a tag <meta>.

Isso porque, além de outras informações em suas páginas, navegadores e buscadores se baseiam e utilizam informações contidas em seus metadados, que nada mais são do que informações e dados de sua página como autor, palavras chaves e outros.

A tag meta somente será utilizada dentro da tag <head>...</head>. Esta tag possui poderosos atributos aliados na construção e elaboração desse conteúdo.

A tag <meta>

A screenshot of a Notepad window titled "exemploAula1.html - Bloco de notas". The window has a menu bar with "Arquivo", "Editar", "Formatar", "Exibir", and "Ajuda". The text area contains the following HTML code:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <meta name="description" content="Aula de HTML">
    <meta name="keywords" content="HTML,CSS,XML,JavaScript">
    <meta name="author" content="Rodrigo e Maromo">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    Iniciando no HTML
  </body>
</html>
```

A tag <meta>

Os principais atributos da tag **<meta>** são:

- **charset**: responsável por chavear qual tabela de caracteres está sendo usada.
- **name**: especifica um nome para o metadados.
- **content**: especifica um valor associado ao atributo name.

Valores do atributo **name**:

- **author**: especifica o nome do autor do documento.

Exemplo:

<meta charset="UTF-8">

<meta name="author" content="Rodrigo">

A tag <meta>

Os principais atributos da tag **<meta>** são:

- **description**: Especifica uma descrição da página. Os motores de busca podem pegar esta descrição para mostrar como resultados das pesquisas.

Exemplo:

```
<meta name="description" content="Aula de HTML">
```

A tag <meta>

Os principais atributos da tag <meta> são:

- **viewport:** O viewport é a área onde seu website aparece. É a área branca da janela quando você abre o browser. O viewport sempre vai ter o tamanho da janela. Mas a forma como os elementos são renderizados vai depender bastante do dispositivo. Em máquinas desktop nós não precisamos nos preocupar muito, já estamos acostumados com um determinado tamanho de tela e resolução média utilizada pelos usuários. Mas quando começamos a variar muito o tamanho das telas, a largura do viewport começa a ser uma preocupação porque afeta diretamente a forma como o usuário utiliza seu website.

A tag <meta>

Os principais atributos da tag **<meta>** são:

Exemplo:

<meta name="viewport" content="width=device-width, initial-scale=1.0">

Os valores de **content** são os que seguem abaixo:

- **width:** Define uma largura para o viewport. Os valores podem ser em PX ou “device-width”, que determina automaticamente um valor igual a largura da tela do dispositivo.

A tag <meta>

Os valores de **content** são os que seguem abaixo:

- **height:** Define uma altura para o viewport. Os valores podem ser em PX ou “device-height”, que determina automaticamente um valor igual a altura da tela do dispositivo.
- **initial-scale:** Define a escala inicial do viewport.

A tag <meta>

Os principais atributos da tag <meta> são:

- **keywords**: especifica uma lista de palavras chaves separadas por virgulas que são muito importantes e relevantes para a sua página HTML. Nunca deixe de especificar suas palavras chaves, pois serão necessárias para que os motores de busca possam incluir e organizar suas informações em possíveis buscas.

Exemplo:

```
<meta name="keywords" content="HTML,CSS,XML,JavaScript">
```

HTML Cabeçalhos

Cabeçalhos HTML são definidos com as tags **<h1>** ao **<h6>**.

São utilizadas para formatar um título ou um tópico de seção.

Existem seis níveis, sendo o primeiro nível de maior tamanho e de primeira importância, e o último nível de menor tamanho e conseqüentemente de menor importância.

A tag <p>

Parágrafos HTML são definidos com a tag <p>.

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7      <title>Exemplo Parágrafos e Cabeçalhos</title>
8  </head>
9  <body>
10     <p>Parágrafo 1</p>
11     <h1>Cabeçalho de Nível 1</h1>
12     <p>Parágrafo 2</p>
13     <h2>Cabeçalho de Nível 2</h2>
14     <p>Parágrafo 3</p>
15     <h3>Cabeçalho de Nível 3</h3>
16     <p>Parágrafo 4</p>
17     <h4>Cabeçalho de Nível 4</h4>
18     <p>Parágrafo 5</p>
19     <h5>Cabeçalho de Nível 5</h5>
20     <p>Parágrafo 6</p>
21     <h6>Cabeçalho de Nível 6</h6>
22 </body>
23 </html>
```

O atributo title

O valor do atributo de título será exibido como uma dica, quando você passa o mouse sobre um elemento.

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7      <title>Exemplo Parágrafos e Cabeçalhos</title>
8  </head>
9  <body>
10     <p title="Você passou o mouse AQUI">Parágrafo 1</p>
11     <h1>Cabeçalho de Nível 1</h1>
12     <p>Parágrafo 2</p>
13     <h2>Cabeçalho de Nível 2</h2>
14     <p>Parágrafo 3</p>
15     <h3>Cabeçalho de Nível 3</h3>
16     <p>Parágrafo 4</p>
17     <h4>Cabeçalho de Nível 4</h4>
18     <p>Parágrafo 5</p>
19     <h5>Cabeçalho de Nível 5</h5>
20     <p>Parágrafo 6</p>
21     <h6>Cabeçalho de Nível 6</h6>
22 </body>
23 </html>
```

A tag <hgroup>

A tag <hgroup> representa o título de uma seção. O elemento é utilizado para agrupar um conjunto de várias tags <h1> até <h6> quando o título possui vários níveis tais como subtítulo, títulos alternativos ou slogans.

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7      <title>Exemplo Parágrafos e Cabeçalhos</title>
8  </head>
9  <body>
10     <hgroup>
11         <h1>Klose</h1>
12         <h2>O maior Artilheiro de todas as Copas</h2>
13     </hgroup>
14     <hgroup>
15         <h1>Pelé</h1>
16         <h2>O maior jogador de todas as Copas</h2>
17     </hgroup>
18 </body>
19 </html>
```

A tag <section>

A tag **<section>** define uma seção no documento e pode ser trabalhada em conjunto com outras tags como **<header>**, **<hgroup>**, **<footer>**, etc. Todas essas opções deverão ser definidas pelo desenvolvedor no momento da construção das marcações.

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <meta http-equiv="X-UA-Compatible" content="ie=edge">
8   <title>Exemplo de Seção</title>
9 </head>
10
11 <body>
12   <section>
13     <h1>HTML5</h1>
14     <h3>Desde o começo o HTML foi criado para ser uma linguagem independente de plataformas, browsers e outros
15       de acesso. Interoperabilidade significa menos custo. Você cria apenas um código HTML e este código pode
16       lido por diversos meios, ao invés de versões diferentes para diversos dispositivos.
17     </h3>
18   </section>
19   <section>
20     <h1>CSS</h1>
21     <h3>CSS é a sigla para o termo em inglês Cascading Style Sheets que, traduzido para o português, significa
22       de Estilo em Cascatas.</h3>
23   </section>
24 </body>
25 </html>
```

A tag <div>

A tag <div> serve para alterar o estilo em partes específicas da página e posicionar objetos.

```
<html>
<head>
<title>Divs</title>
</head>

<body>
<div style="background-color:#00BFFF;text-align:center">
<p>Primeira Div</p>
</div>
<div style="border:3px dotted #9F0">
<p>Segunda Div</p>
</div>

</body>
</html>
```


A tag

O HTML não possui quebra de linha automática, necessitando da tag
 para que haja este recurso.

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7      <title>Exemplo 5</title>
8  </head>
9  <body>
10     Não existe uma história<br>
11     Não existe uma saída<br>
12     Só existe a esperança de um futuro melhor
13 </body>
14 </html>
```

Tags de Formatação

- **** Especifica um texto em negrito, dando maior destaque ao conteúdo incorporado.
- **** Especifica um texto em itálico.
- **<u>** Especifica um texto sublinhado.
- **<dfn>** Esta tag dá uma definição de algum trecho ou sigla de texto.
- **<abbr>** Esta tag indica abreviação que pode ser muito útil aos navegadores, sistemas ortográficos e motores de busca.
- **<code>** Esta tag é utilizada quando há necessidade de inserir códigos de programas ou similares com o intuito de dar exemplos ou explicações durante a elaboração de uma página HTML.
- **<mark>** É utilizada para marcar uma parte do texto com fins de referência para aquele conteúdo.

Tags de Formatação

- **<samp>** Indica alguma saída de um programa, referente a um resultado ou até mesmo de um erro.
- **<kbd>** Permite uma indicação de entrada de dados via teclado, ou seja, indica o texto correto que deve ser digitado pelo usuário.
- **<var>** Esta tag é utilizada para indicar um exemplo de um aplicativo, expressão matemática ou alguma variável em questão.
- **<sub>** Define o texto como subscrito.
- **<sup>** Define o texto sobrescrito.

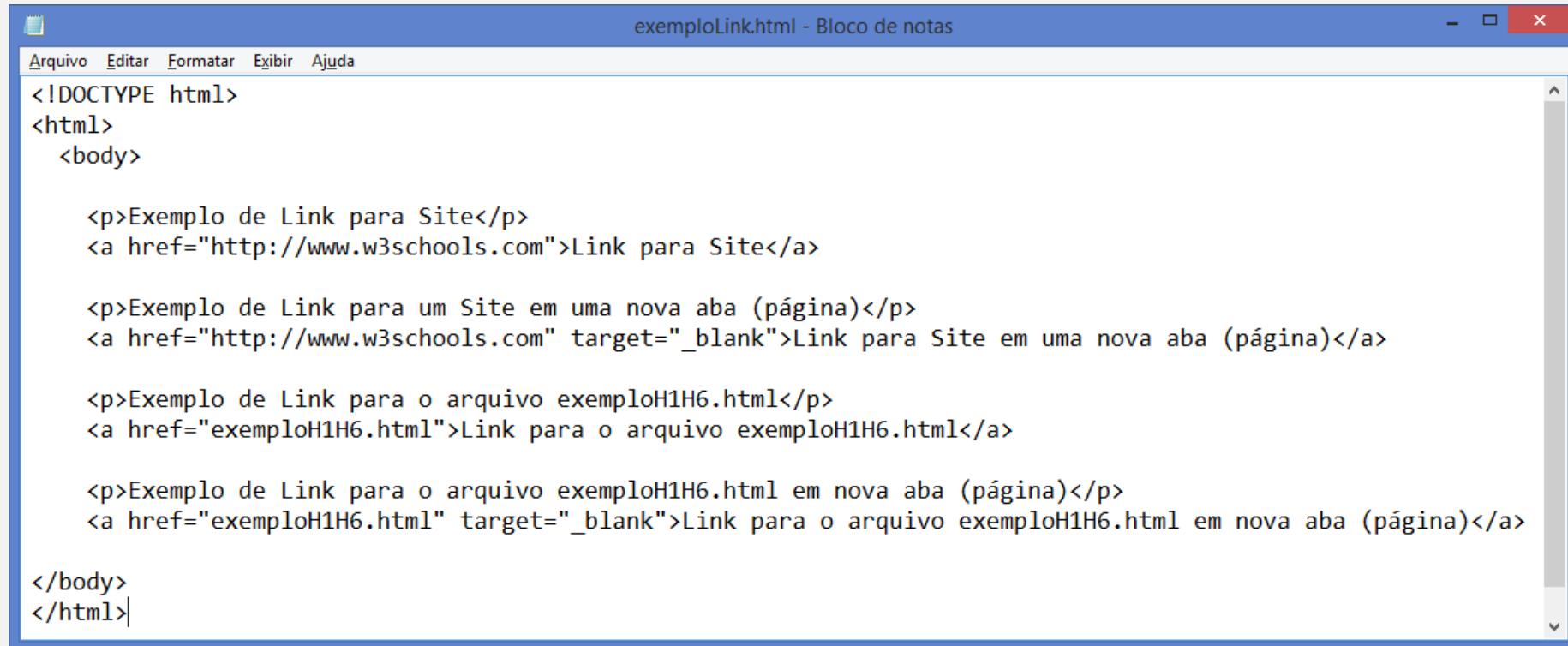
Tags de Formatação

```
exemploAula1.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="UTF-8">
5   <meta name="author" content="Rodrigo e Maromo">
6   <meta name="description" content="Aula de HTML">
7   <meta name="keywords" content="HTML,CSS,XML,JavaScript">
8   <meta name="viewport" content="width=device-width, initial-scale=1.0">
9 </head>
10 <body>
11   <em>Estilo Itálico</em> <br />
12   <strong>Estilo Negrito</strong> <br />
13   <u>Estilo sublinhado</u><br />
14   <mark>Estilo com marcação</mark> <br />
15   CBF - <dfn>Confederação Brasileira de Futebol</dfn> <br />
16   A <abbr title="Confederação Brasileira de Futebol">CBF</abbr> foi fundada em 20 de Agosto de 1914 <br />
17   Este código é importante para o nosso aprendizado<br />
18   <code>
19     public void teste(){ <br />
20       System.out.println("teste"); <br />
21     }<br />
22   </code>
23   O programa dará a seguinte saída:<br />
24   <samp>Tem certeza que deseja excluir?</samp> Clique Sim ou Não<br />
25   Digite em seu navegador: <kbd>www.uol.com.br</kbd> e depois clique em OK<br />
26   Devemos ficar atentos a variável <var> n </var> para não termos problemas com o código <br />
27   H<sub>3</sub>PO<sub>4</sub><br />
28   X<sup>2</sup>+2X<sup>3</sup>+2*Z<sup>4</sup>
29
30 </body>
31 </html>
```

Links em HTML

Links HTML são definidas com a tag **<a>**.

O destino do link é especificado no atributo **href**.

A screenshot of a Notepad window titled "exemploLink.html - Bloco de notas". The window has a menu bar with "Arquivo", "Editar", "Formatar", "Exibir", and "Ajuda". The text area contains the following HTML code:

```
<!DOCTYPE html>
<html>
  <body>

    <p>Exemplo de Link para Site</p>
    <a href="http://www.w3schools.com">Link para Site</a>

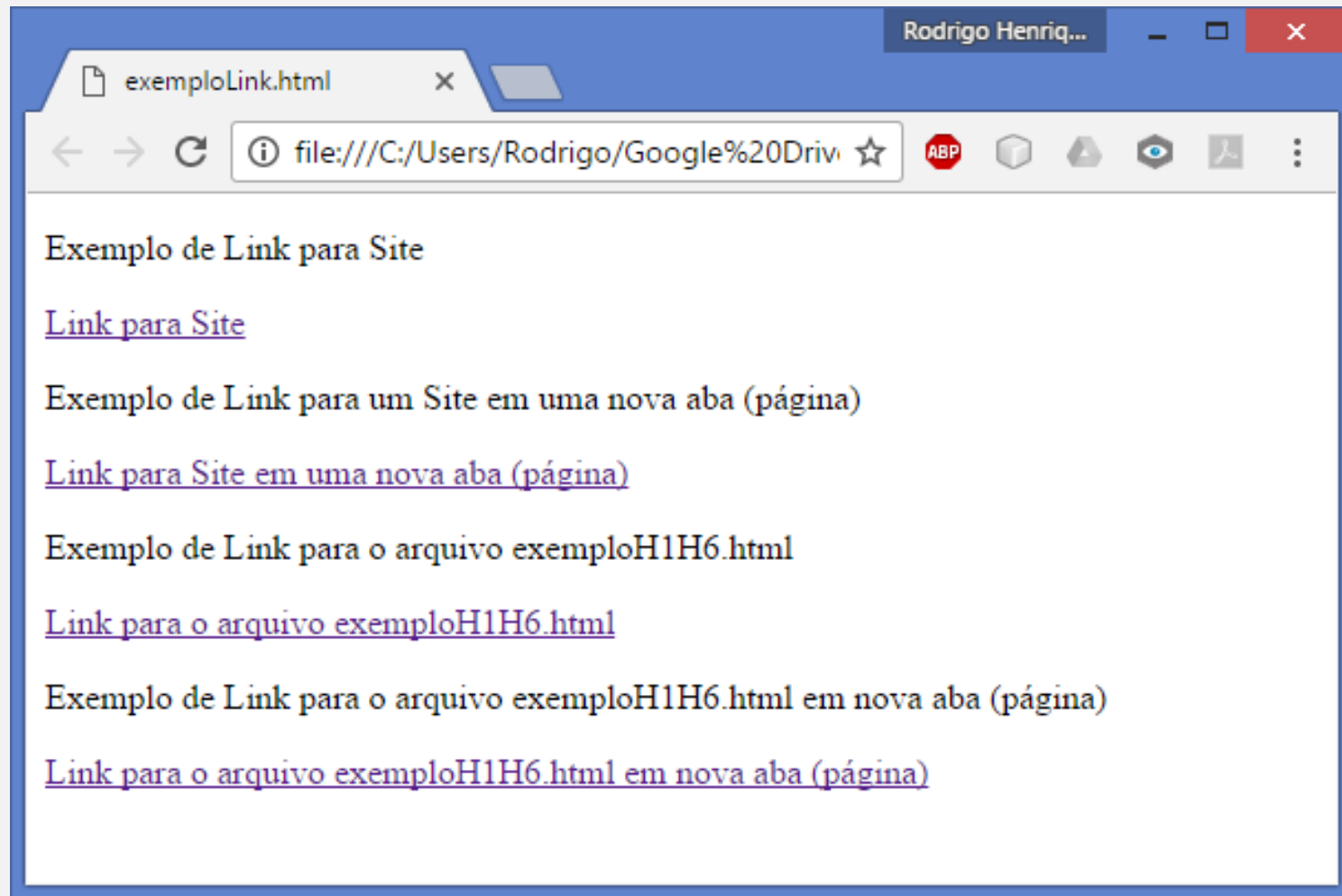
    <p>Exemplo de Link para um Site em uma nova aba (página)</p>
    <a href="http://www.w3schools.com" target="_blank">Link para Site em uma nova aba (página)</a>

    <p>Exemplo de Link para o arquivo exemploH1H6.html</p>
    <a href="exemploH1H6.html">Link para o arquivo exemploH1H6.html</a>

    <p>Exemplo de Link para o arquivo exemploH1H6.html em nova aba (página)</p>
    <a href="exemploH1H6.html" target="_blank">Link para o arquivo exemploH1H6.html em nova aba (página)</a>

  </body>
</html>
```

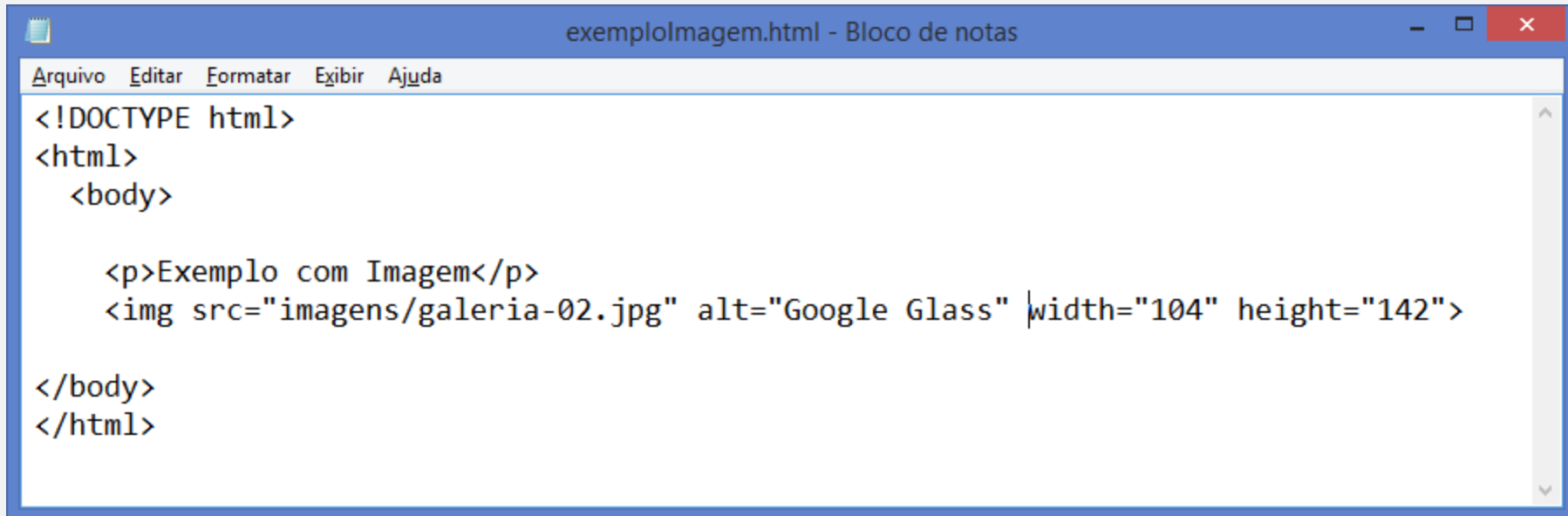
Links em HTML



Imagens em HTML

Imagens HTML são definidos com a tag ****.

O arquivo de origem (**src**), texto alternativo (**alt**), largura e altura são fornecidos como atributos:

A screenshot of a Notepad window titled "exemploImagem.html - Bloco de notas". The window has a menu bar with "Arquivo", "Editar", "Formatar", "Exibir", and "Ajuda". The text area contains the following HTML code:

```
<!DOCTYPE html>
<html>
  <body>

    <p>Exemplo com Imagem</p>
    

  </body>
</html>
```

Os atributos src, width, height e alt

A URL de um conteúdo incorporável é com o atributo **src**, a largura da imagem com o atributo **width** e altura com o atributo **height**.

O atributo **src** pode ser usado com as tags **<audio>**, **<embed>**, **<iframe>**, ****, **<input>**, **<script>**, **<source>**, **<track>**, **<video>**.

Os atributos **width** e **height** pode ser usado com as tags **<canvas>**, **<embed>**, **<iframe>**, ****, **<input>**, **<object>**, **<video>**.

O atributo **alt** especifica um texto alternativo para ser usado, quando uma imagem não poder ser exibida. O atributo alt pode ser usado com as tags **<applet>**, **<area>**, ****, **<input>**.

O valor do atributo podem ser lidos pelos leitores de tela. Desta forma, alguém "escuta" a página da web, por exemplo, uma pessoa cega, pode "ouvir" o elemento.

A tag <nav>

Esta tag permite ao desenvolvedor criar uma área ou seção específica para menus e links para conteúdos externos e internos. Nem todos os links precisam estar na influência da tag <nav>, porém é importante utilizar essa tag quando houver algum bloco de links na página HTML.



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Aula de HTML</title>
5     <meta charset="utf-8">
6     <meta name="author" content="Rodrigo e Maromo">
7     <meta name="description" content="Aula de HTML">
8     <meta name="keywords" content="HTML,CSS,XML,JavaScript">
9     <meta name="viewport" content="width=device-width, initial-scale=1.0">
10  </head>
11  <body>
12    <nav>
13      <a href="/html/">HTML</a> |
14      <a href="/css/">CSS</a> |
15      <a href="/js/">JavaScript</a> |
16      <a href="/jquery/">jQuery</a>
17    </nav>
18  </body>
19 </html>
```

As tags `<article>`, `<aside>` e `<header>`

A tag **`<article>`** define um artigo independente do conteúdo geral. Um artigo deve estar independente e deve ser compartilhado independentemente do resto do local onde ele foi criado. Quando os elementos dos artigos são encaixados, os elementos dos artigos inferiores representam artigos que são em princípio relacionados com os conteúdos do artigo exterior e podem ser utilizados em blog, fóruns, comentários e notícias que aproveitam o potencial do elemento **`<article>`**.

Também pode se usar a tag **`<aside>`**, que define algum conteúdo, como um banner lateral ou um auxiliador na utilização da tag **`<article>`**.

A tag **`<header>`** define o cabeçalho para um artigo.

As tags <article> e <header>

```
exemploAula2.html x
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Aula de HTML</title>
5      <meta charset="utf-8">
6      <meta name="author" content="Rodrigo e Maromo">
7      <meta name="description" content="Aula de HTML">
8      <meta name="keywords" content="HTML,CSS,XML,JavaScript">
9      <meta name="viewport" content="width=device-width, initial-scale=1.0">
10 </head>
11 <body>
12 <nav>
13     <a href="/html/">HTML</a> |
14     <a href="/css/">CSS</a> |
15     <a href="/js/">JavaScript</a> |
16     <a href="/jquery/">jQuery</a>
17 </nav>
18 <article>
19     <header>
20         <h1>Mais importante</h1>
21         <h3>Menos importante</h3>
22         <p>Lorem Ipsum dolor set amet...</p>
23     </header>
24 </article>
25 <aside>
26     Como proteger seu site? Saiba mais em nosso site.
27 </aside>
28 </body>
29 </html>
```

A tag <hr>

A tag <hr> cria uma linha horizontal no documento, definindo uma mudança temática. Geralmente é utilizada para dividir conteúdo.

```
25 <aside>
26     Como proteger seu site? Saiba mais em nosso site.
27 </aside>
28
29 <h1>HTML</h1>
30 <p>HTML (abreviação para a expressão inglesa HyperText Markup Language, que significa Linguagem de Marcação de Hipertexto)
31     é uma linguagem de marcação utilizada na construção de páginas na Web.</p>
32 <hr>
33
34 <h1>CSS</h1>
35 <p>Cascading Style Sheets (CSS) é uma linguagem de folhas de estilo utilizada para definir a apresentação de documentos escritos em
36     uma linguagem de marcação, como HTML ou XML. O seu principal benefício é a separação entre o formato e o conteúdo de um documento.</p>
37
38 </body>
39 </html>
40
```

A tag <address>

Esta tag define as informações de contato do autor ou proprietário de um documento ou um artigo. É importante destacar que se o elemento **<address>** estiver dentro do elemento **<body>** as informações do contato serão para o documento, porém, se o elemento estiver dentro de um elemento **<article>**, as informações de contato serão para esse artigo. O texto dentro da tag **<address>** normalmente ficará em itálico.

```
41 <address>
42 Desenvolvido por <abbr title="rodrigo.martins45@etec.sp.gov.br"><u>Rodrigo Henrique Martins</u></abbr><br>
43 Visite-nos<br>
44 www.meusite.com.br<br>
45 Mogi Mirim - SP
46 </address>
```

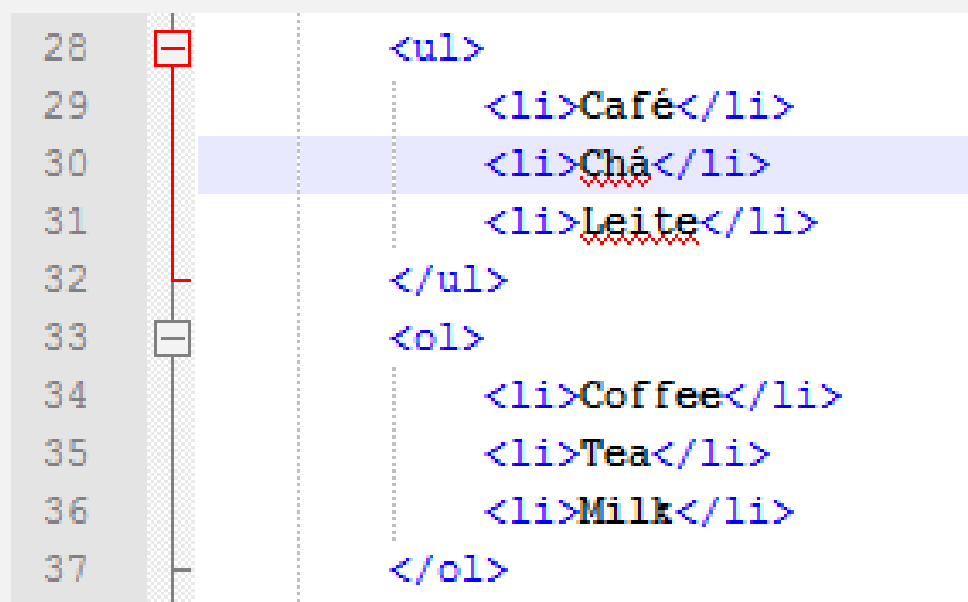
A tag <footer>

Esta tag define o rodapé de um documento ou uma seção de sua página. Ao desenvolver uma página é importante separar o espaço do rodapé, onde normalmente contém o nome do autor do documento, copyright.

```
45  <footer>
46      <p>Postado por: Rodrigo Henrique Martins</p>
47      <p>Contato: <a href="mailto:rodrigo.martins45@etec.sp.gov.br">
48          rodrigo.martins45@etec.sp.gov.br</a>.</p>
49  </footer>
50
```

As tags , e




- **** Lista não ordenada, ou seja, apresentam os itens em lista sem numerá-los. Sempre utilize a tag **** em conjunto com a tag **** para criar a lista não ordenada.
- **** Listas ordenadas, ou seja, apresentam os itens com números na sequência. Sempre utilize a tag **** em conjunto com a tag **** para criar a lista ordenada.



```
28  <ul>
29      <li>Café</li>
30      <li>Chá</li>
31      <li>Leite</li>
32  </ul>
33  <ol>
34      <li>Coffee</li>
35      <li>Tea</li>
36      <li>Milk</li>
37  </ol>
```

O atributo start

Indica onde deverá iniciar a sequencia dos itens. Esse atributo só é utilizado com a tag ****.

28		<pre></pre>
29		<pre> Café</pre>
30		<pre> Chá</pre>
31		<pre> Leite</pre>
32		<pre></pre>
33		<pre></pre>
34		<pre> Café</pre>
35		<pre> Chá</pre>
36		<pre> Leite</pre>
37		<pre></pre>
38		<pre><ol start="50"></pre>
39		<pre> Café</pre>
40		<pre> Chá</pre>
41		<pre> Leite</pre>
42		<pre></pre>

O atributo reversed

Indica se a lista dever ser mostrada em ordem decrescente em vez de crescente. Esse atributo só é utilizado com a tag ****.

```
28  <ul>
29      <li>Café</li>
30      <li>Chá</li>
31      <li>Leite</li>
32  </ul>
33  <ol reversed>
34      <li>Café</li>
35      <li>Chá</li>
36      <li>Leite</li>
37  </ol>
38  <ol start="50">
39      <li>Café</li>
40      <li>Chá</li>
41      <li>Leite</li>
42  </ol>
```

As tags <dl>, <dt> e <dd>

- **<dl>** Lista de definição, ou seja não trabalha com marcador, porém organiza definições para cada item. A Tag define um item em uma lista de definições e é usado em conjunto com a tag **<dt>** que defini um item da lista, e **<dd>** que descreve os itens da lista.

```
44  <dl>
45      <dt><strong>Café</strong></dt>
46          <dd>Quente</dd>
47          <dd>Morno</dd>
48          <dd>Fio</dd>
49      <dt><strong>Leite</strong></dt>
50          <dd>Quente</dd>
51          <dd>Morno</dd>
52          <dd>Fio</dd>
53  </dl>
```

As tags <audio> e <video>

- A tag <audio> define o som, como música ou outros fluxos de áudio. Atualmente, existem 3 formatos de arquivos suportados para a tag <audio>: MP3, WAV e Ogg.
- A tag <video> especifica de vídeo, como um clipe de filme ou de outros fluxos de vídeo. Atualmente, existem 3 formatos de vídeo suportados para o elemento <video>: MP4, WebM, e Ogg.

```
14 <audio controls>
15   <source src="media/2009-lovers-carvings-bibio.mp3" type="audio/mpeg">
16   Your browser does not support the <code>audio</code> tag.
17 </audio>
18
19 <video width="320" height="240" controls>
20   <source src="media/getting-started.mp4" type="video/mp4">
21   Your browser does not support the <code>video</code> tag.
22 </video>
23 </body>
```

Os atributos das tags <audio> e <video>

- **controls**: Mostra os controles padrão para o áudio na página.
- **autoplay**: Faz com que o áudio reproduza automaticamente.
- **loop**: Faz com que o áudio repita automaticamente.
- O atributo **preload** é usado em elementos <audio> para carregar arquivos grandes. Ele pode assumir 3 valores:
 - **"none"** não carrega o arquivo
 - **"auto"** carrega o arquivo
 - **"metadata"** carrega apenas os meta dados do arquivo
- Vários arquivos podem ser especificados utilizando o elemento **<source>** para disponibilizar vídeo ou áudio codificados em formatos diferentes para navegadores diferentes.

O atributo da tag <video>

poster: Especifica uma imagem a ser mostrada enquanto o vídeo está sendo baixado, ou até que o usuário acesse o botão de reprodução.

height: Define a altura do player de vídeo.

width: Define a largura do player de vídeo

A tag <table>

A tag <table> define uma tabela HTML.

Uma tabela HTML consiste no elemento <table> e um ou mais <tr> , <th> , e <td> elementos.

O <tr> elemento define uma linha da tabela, o elemento <th> define um cabeçalho da tabela, e o elemento <td> define uma célula da tabela.

Uma tabela HTML mais complexa pode também incluir <caption>, <col>, <colgroup>, <thead>, <tfoot> e <tbody> elementos.

A tag <table>

```
7  [-] <html>
8  [-]   <head>
9      <title>Tabelas</title>
10     <meta charset="UTF-8">
11     <meta name="viewport" content="width=device-width, initial-scale=1.0">
12     <style>
13     [-]     table, th, td {
14         border: 1px solid black;
15     }
16     </style>
17 </head>
```

A tag <table>

```
18 <body>
19   <h4>Tabela Horizontal:</h4>
20   <table>
21     <tr>
22       <th>Nome</th>
23       <th>Telefone</th>
24     </tr>
25     <tr>
26       <td>Rodrigo Martins</td>
27       <td>9999-9999</td>
28     </tr>
29   </table>
30   <br>
```


A tag <table>

```
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
```

```
<h4>Tabela Vertical:</h4>
<table>
  <tr>
    <th>Primeiro Nome:</th>
    <td>Rodrigo</td>
    <td>Marcos</td>
    <td>Janete</td>
    <td>Marli</td>

  </tr>
  <tr>
    <th>Telefone:</th>
    <td>9999-9999</td>
    <td>7777-7777</td>
    <td>5555-5555</td>
    <td>4444-7777</td>

  </tr>
</table>
```

A tag <table>

50		
51		
52		<code><table></code>
53		<code> <h4>Tabela com Legenda:</h4></code>
54		<code> <caption>Legenda</caption></code>
55		<code> <tr></code>
56		<code><th>Mês</th></code>
57		<code><th>Gastos</th></code>
58		<code> </tr></code>
59		<code> <tr></code>
60		<code><td>Janeiro</td></code>
61		<code><td>R\$100</td></code>
62		<code> </tr></code>
63		<code> <tr></code>
64		<code><td>Fevereiro</td></code>
65		<code><td>R\$50</td></code>
66		<code> </tr></code>
67		<code></table></code> <code>
</code>

A tag <table>

```
69 <table>
70   <h4>Elementos(tag) dentro das Tabelas:</h4>
71   <tr>
72     <td>
73       <p>Parágrafo 1</p>
74       <p>Paragrafo 2</p>
75     </td>
76     <td>Tabela dentro da tabela:
77       <table>
78         <tr>
79           <td>A</td>
80           <td>B</td>
81         </tr>
82         <tr>
83           <td>C</td>
84           <td>D</td>
85         </tr>
86       </table>
87     </td>
88   </tr>
```

A tag <table>

```
89 <tr>
90 <td>Lista não ordenada dentro da tabela
91 <ul>
92 <li>maças</li>
93 <li>bananas</li>
94 <li>abacaxi</li>
95 </ul>
96 </td>
97 <td>Olá</td>
98 </tr>
99 </table>
100
101 <h4>Célula que abrange duas colunas</h4>
102 <table>
103 <tr>
104 <th>Nome</th>
105 <th colspan="2">Telefone</th>
106 </tr>
107 <tr>
108 <td>Rodrigo Martins</td>
109 <td>88888-8888</td>
110 <td>99999-9999</td>
111 </tr>
112 </table>
113 <br>
```

A tag <table>

```
114
115      <h4>Célula que abrange duas linhas</h4>
116      <table>
117          <tr>
118              <th>Primeiro Nome:</th>
119              <td>Rodrigo</td>
120          </tr>
121          <tr>
122              <th rowspan="2">Telefone:</th>
123              <td>88888-8888</td>
124          </tr>
125          <tr>
126              <td>99999-9999</td>
127          </tr>
128      </table>
129
130  </body>
131 </html>
132
```

A tag <col> e <colgroup>

A tag **<colgroup>** especifica um grupo de uma ou mais colunas de uma tabela para formatação. A tag **<colgroup>** é útil para aplicação de estilos de colunas inteiras, em vez de repetir os estilos para cada célula, para cada linha.

A tag **<col>** especifica as propriedades da coluna para cada coluna dentro de um **<colgroup>**.

```
131 <table>
132   <colgroup>
133     <col span="2" style="background-color:red">
134     <col style="background-color:yellow">
135   </colgroup>
136   <tr>
137     <th>ISBN</th>
138     <th>Título</th>
139     <th>Preço</th>
140   </tr>
141   <tr>
142     <td>3476896</td>
143     <td>HTML5</td>
144     <td>R$53,00</td>
145   </tr>
146 </table>
```

Formulários

Um formulário é um componente de uma página web que possui controles, como campos de texto, botões, caixas de seleção ou de combinação, listas etc., para permitir a interação do usuário com o site.

Criação de Formulários

Os formulários podem ser criados por meio do elemento `<form>`. No interior desse elemento pode ser utilizado um conjunto de outros elementos para gerar os controles (legendas, campos, botões e agrupamentos) do formulário. Os elementos para criar os controles do formulário são `<input>`, `<textarea>`, `<legend>`, `<label>`, `<fieldset>`, `<datalist>`, `<keygen>`, `<output>`, `<select>`, `<option>` e `<optgroup>`.

A tag <form>

Todo formulário começa com o elemento <form>. Em seu interior, são colocados os campos, na maioria das vezes representados pelo elemento <input> que, se não tiver atributos, por padrão, exibe na tela um campo de texto. Para identificar um controle, como, por exemplo, um campo texto criado pelo elemento <input>, utiliza-se o elemento <label>.

Cada parte de um formulário é considerado um parágrafo, e é normalmente separado do resto do formulário por meio do elemento <p>.

A tag<form> - Exemplo

```
<html>
  <head>
    <title>Exemplo de formulário</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <form>
      <p>
        <label>
          Nome:<input>
        </label>
      </p>
    </form>
  </body>
</html>
```

Nome:

A tag <input>

O elemento <input> gera um campo em que o usuário pode selecionar ou entrar com dados de várias maneiras. O que define como será a entrada é o atributo type.

```
<form>
  <p>
    <label>
      Nome:<input type="text">
    </label>
  </p>
  <p>
    <label>
      Senha:<input type="password">
    </label>
  </p>
</form>
```

Nome:

Senha:

Campo do tipo telefone

Para definir um campo de entrada de telefone, é necessário que o elemento `<input>` possua o atributo `type` com valor “tel”.

```
<form>
  <p>
    <label>
      Telefone:<input type="tel" required>
    </label>
  </p>
```


Telefone:

Campo do tipo email

Para definir um campo de entrada de email, é necessário que o elemento `<input>` possua o atributo `type` com valor "email".

```
<form method="post">
  <p>
    <label>
      Email:<input type="email">
    </label>
  </p><br />
  <p>
    <input type="submit" value="ENVIAR">
  </p>
</form>
```

Email:


 Inclua um "@" no endereço de e-mail. "rodrigo" está com um "@" faltando.

Campo do tipo url

Para definir um campo de entrada de endereço web, é necessário que o elemento `<input>` possua o atributo `type` com valor "url".

```
<form method="post">
  <p>
    <label>
      Site:<input type="url">
    </label>
  </p><br />
  <p>
    <input type="submit" value="ENVIAR">
  </p>
</form>
```

Site:

 Insira um URL.

Campo do tipo number

O campo “number” é utilizado nos elementos <input> para entrada de valores numéricos. É possível definir restrições de entrada para esse tipo de campo.

```
<form method="post">
  <p>
    <label>
      Nota:<input type="number" name="nota"
                  min="0" max="10" step="0.5">
    </label>
  <p>
    <input type="submit" value="ENVIAR">
  </p>
</form>
```

Nota: 1,5

ENVIAR

Nota: 1,15

ENVIAR

! Insira um valor válido. Os dois valores válidos mais próximos são 1 e 1,5.

Campo do tipo range

O tipo “range” é usado para campos de entrada que devem receber um valor numérico contido em uma faixa de valores numéricos.

```
<form method="post">
  <p>
    <label>
      Volume:<input type="range" name="volume"
                    min="0" max="10" step="2">
    </label>
  <p>
    <input type="submit" value="ENVIAR">
  </p>
</form>
```

Volume:

ENVIAR

Campos para entrada de data e hora

Date – Permite selecionar o dia, mês e ano

```
<form method="post">
  <p>
    <label>
      Data:<input type="date" name="data">
    </label>
  <p>
    <input type="submit" value="ENVIAR">
  </p>
</form>
```

Data: 03 / 05 / 2016

ENVIAR

Campos para entrada de data e hora

Month – Permite selecionar mês e ano

```
<form method="post">  
  <p>  
    <label>  
      Mês: <input type="month" name="mes">  
    </label>  
  <p>  
    <input type="submit" value="ENVIAR">  
  </p>  
</form>
```

Mês: maio de 2016

ENVIAR

Campos para entrada de data e hora

Week – Seleciona dia da semana e ano

```
<form method="post">  
  <p>  
    <label>  
      Semana:<input type="week" name="semana">  
    </label>  
  <p>  
    <input type="submit" value="ENVIAR">  
  </p>  
</form>
```

Semana: Semana 04, 2016

ENVIAR

Campos para entrada de data e hora

Time – Permite selecionar hora e minuto

```
<form method="post">
  <p>
    <label>
      Hora:<input type="time" name="hora">
    </label>
  <p>
    <input type="submit" value="ENVIAR">
  </p>
</form>
```

Hora:

ENVIAR

Campos para entrada de data e hora

datetime-local – Permite selecionar dia, mês, ano, hora e minuto.

```
<form method="post">
  <p>
    <label>
      Data e Hora:<input type="datetime-local" name="datahora">
    </label>
  <p>
    <input type="submit" value="ENVIAR">
  </p>
</form>
```

Data e Hora:

ENVIAR

Validação de entrada de valores de data e hora

Se for inserido um valor não válido em campos que manipulam data e hora, será exibida a mensagem indicando que o valor é inválido.

```
<form method="post">
  <p>
    <label>
      Tempo de entrega do Produto:<input type="time" min="19:00"
                                         max="23:00" step="300" name="hora">
    </label>
  <p>
    <input type="submit" value="ENVIAR">
  </p>
</form>
```

Tempo de entrega do Produto:

19:02 x

ENVIAR



Insira um valor válido. Os dois valores válidos mais próximos são 19:00 e 19:05.

Validação de entrada de valores de data e hora (cont.)

```
<form method="post">
  <p>
    <label>
      Data:<input type="date"
        min="2016-01-01" max="2016-12-31" value="2016-05-03">
    </label>
  <p>
    <input type="submit" value="ENVIAR">
  </p>
</form>
```

Data: 03/05/2016 x ▾

ENV maio de 2016 ▾ < • >

dom	seg	ter	qua	qui	sex	sáb
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

Campo do tipo hidden

O elemento `<input>` com o atributo `type` definido como “hidden” indica que o campo não aparece na tela para o usuário. Esse tipo de campo é utilizado para transportar valores do cliente para o servidor para a identificação do formulário, ou outros valores que a aplicação que executa no servidor precisa para executar algum procedimento.

Campo do tipo hidden (cont)

Quando o servidor receber o formulário, a aplicação pode capturar o conteúdo do atributo value por meio do conteúdo do atributo name e identificar que o formulário é de contato.

```
<label>  
  <input type="hidden" name="formulario" value="contato">  
</label>
```

Botões dos tipo image

Quando o elemento `<input>` possui o atributo `type` configurado com o valor “image”, o botão pode ser uma imagem que será utilizada para submeter o formulário .

```
<form name="formulario" action="teste.jsp" method="post">
  <p>
    <label>
      Nome:<input type="text" name="pname">
    </label>
  </p>
  <p>
    <label>
      Sobrenome:<input type="text" name="sname">
    </label>
  </p>
  <p>
    <input type="image" src="botao.jpg" alt="botao">
  </p>
</form>
```

Nome:

Sobrenome:



Botão do tipo radio

Campos radio (radio buttons) servem para permitir que o usuário selecione uma (e somente uma) dentre muitas opções de escolha disponibilizadas em um formulário. Campos radio são criados pelo elemento `<input>` contendo o atributo `type` com o valor "radio".

```
<body>
  <fieldset>
    <legend>Estado Civil:</legend>
    <p>
      <label>
        <input type="radio" name="estadoCivil" value="Casado">Casado
      </label>
    </p>
    <p>
      <label>
        <input type="radio" name="estadoCivil" value="Solteiro">Solteiro
      </label>
    </p>
    <p>
      <label>
        <input type="radio" name="estadoCivil" value="Divorciado">Divorciado
      </label>
    </p>
  </fieldset>
</body>
```

Estado Civil:

☐ Casado

☐ Solteiro

☐ Divorciado

Botão do tipo checkbox

Para permitir que o usuário selecione várias das muitas opções disponíveis em uma sequência de itens, pode-se utilizar elementos `<input>` com o atributo `type` contendo o valor “checkbox”.

```
<body>
  <fieldset>
    <legend> Produtos </legend>
    <p>
      <label>
        <input type="checkbox" name="computador" value="Computador">Computador
      </label>
    </p>
    <p>
      <label>
        <input type="checkbox" name="impressora" value="Impressora">Impressora
      </label>
    </p>
    <p>
      <label>
        <input type="checkbox" name="monitor" value="Monitor">Monitor
      </label>
    </p>
  </fieldset>
</body>
```

Produtos

☐ Computador

☐ Impressora

☐ Monitor

Campo do tipo search

O tipo search é usado para campos utilizados em pesquisas, como uma busca no site, ou de pesquisa nos buscadores web.

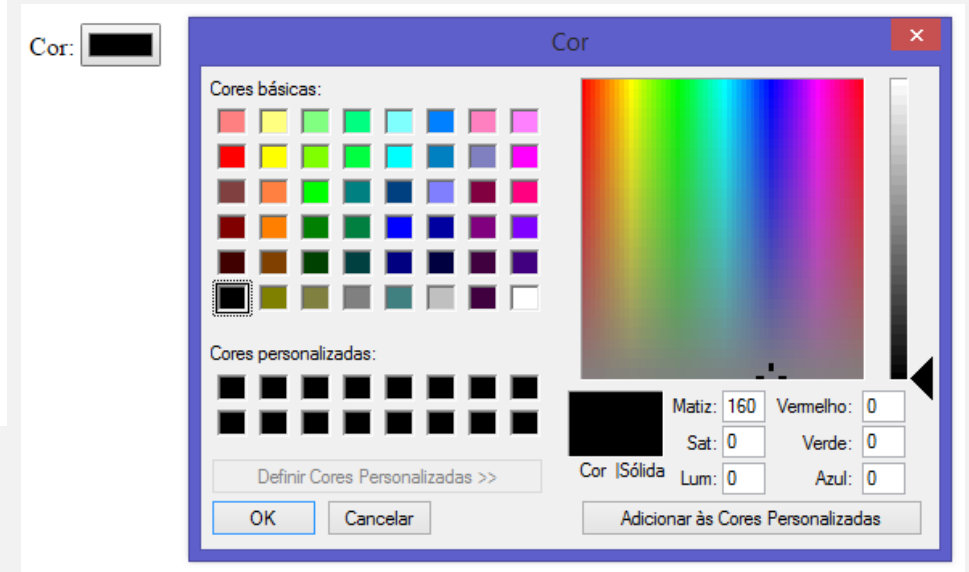
```
<p>  
  <label>  
    O que deseja pesquisar?  
    <input type="search" name="pesquisa">  
  </label>  
</p>
```

O que deseja pesquisar?

Campo do tipo color

Campos do tipo color são usados para entrada de cores. Nos campos para seleção de cores o atributo type deve receber o valor “color”.

```
<p>  
  <label>  
    Cor:  
    <input type="color" name="cor">  
  </label>  
</p>
```



O elemento <button>

O elemento <button> é utilizado para criar botões independentes ou não dos formulários. No interior do elemento <button> você pode colocar conteúdo, como texto ou imagens. Esta é a grande diferença entre o elemento <button> e o elemento <input> com o atributo type igual a "button".

```
<body>
  <p>
    <button type="button">
      Clique aqui
    </button>
  </p>
  <p>
    <button type="button">
      
    </button>
  </p>
</body>
```



Clique aqui



ENTRAR

O elemento <button> (cont)

```
<form name="formulario" action="teste.php" method="post">
  <p>
    <label>
      Nome:<input type="text" name="pnome">
    </label>
  </p>
  <p>
    <button type="button" onclick="alert('Botão Clicado.')">
      Clique aqui
    </button>
    <button type="submit">
      Enviar
    </button>
    <button type="reset">
      Limpar
    </button>
  </p>
</form>
```

Nome:

Clique aqui

Enviar

Limpar

localhost:8383 diz:

Botão Clicado.

☐ Impedir que esta página crie caixas de diálogo adicionais.

OK

O elemento <textarea>

Campos criados com o elemento <textarea> permitem a digitação de um texto com múltiplas linhas.

```
<p>
  <label>
    Deixe sua sugestão:<br />
    <textarea cols="30" rows="5"></textarea>
  </label>
</p>
```

Deixe sua sugestão:

Os dados do produto
disponíveis nessa página não
são suficientes.

O elemento <datalist>

O elemento <datalist> especifica uma lista oculta de opções sugeridas (não obrigatórias) para um campo de entrada. A lista é criada com elementos <option> dentro do <datalist>.

```
<p>
  <label>
    Site:<input type="url" list="sites" name="link">
  </label>
</p>
<datalist id="sites">
  <option label="Ig"
    value="http://www.ig.com.br">
</option>
  <option label="Google"
    value="http://www.google.com">
</option>
  <option label="Microsoft"
    value="http://www.microsoft.com.br">
</option>
</datalist>
```

Site:

- http://www.ig.com.br Ig
- http://www.google.com Google
- http://www.microsoft.com.br Microsoft

O elemento <output>

O elemento <output> é usado para diferentes tipos de saída, como cálculos ou saída do script.

```
<html>
  <head>
    <title>OUTPUT</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <p>Multiplicação de dois valores:</p>
    <form onsubmit="return false"
      oninput="resultado.value=valor1.valueAsNumber*valor2.valueAsNumber">
      <input name="valor1" type="number">*
      <input name="valor2" type="number">=
      <output name="resultado"></output>
    </form>
  </body>
</html>
```

Multiplicação de dois valores:

3 * 10 = 30

O elemento <select>

O elemento <select> é usado para criar uma lista drop-down. Os <option> dentro do elemento <select> definir as opções disponíveis na lista.

```
14      Marca de Carro:  
15      <select>  
16          <option value="volvo">Volvo</option>  
17          <option value="saab">Jeep</option>  
18          <option value="mercedes">Mercedes</option>  
19          <option value="audi">Audi</option>  
20      </select>
```

Os atributos da tag <select>

autofocus: Especifica que a lista suspensa deve automaticamente obter foco quando a página é carregada.

disabled: O atributo disabled é um atributo booleano. Quando presente, ele especifica que a lista drop-down deve ser desativado.

multiple: O atributo múltiplo é um atributo booleano. Quando presente, ele especifica que várias opções podem ser selecionadas de cada vez.

name: O atributo name especifica o nome de uma lista drop-down. O atributo nome é usado para fazer referência a elementos em um JavaScript, ou fazer referência a dados do formulário depois que um formulário é enviado.

required: O atributo obrigatório é um atributo booleano. Quando presente, ele especifica o usuário é obrigado a selecionar um valor antes de enviar o formulário.

size: O atributo de tamanho especifica o número de opções visíveis em uma lista drop-down. Se o valor do atributo de tamanho é maior do que 1, mas menor do que o número total de opções na lista, o navegador irá adicionar uma barra de rolagem para indicar que existem mais opções para ver.

Os atributos da tag <select>

14

15

16

17

18

19

20

21



Marca de Carro:

```
<select>
```

```
    <option value="volvo">Volvo</option>
```

```
    <option value="saab">Jeep</option>
```

```
    <option value="mercedes">Mercedes</option>
```

```
    <option value="audi">Audi</option>
```

```
</select>
```

```
<br>
```

Os atributos da tag <select>

22
23
24
25
26
27
28
29
30
31



Exemplo com autofocus e size


```
<select autofocus size="2">
```

```
    <option value="volvo">Volvo</option>
```

```
    <option value="saab">Jeep</option>
```

```
    <option value="opel">Opel</option>
```

```
    <option value="audi">Audi</option>
```

```
</select>
```

```
<br>
```

Os atributos da tag <select>

```
32      Exemplo com disabled
33      <select disabled>
34          <option value="volvo">Volvo</option>
35          <option value="saab">Jeep</option>
36          <option value="mercedes">Mercedes</option>
37          <option value="audi">Audi</option>
38      </select>
39      <br>
40
41      Exemplo com multiple, name e required<br>
42      <select name="carros" multiple required>
43          <option value="volvo">Volvo</option>
44          <option value="saab">Jeep</option>
45          <option value="opel">Opel</option>
46          <option value="audi">Audi</option>
47      </select>
48  </body>
49  </html>
```


Outros atributos da tag <form>

O atributo **action** especifica para onde enviar o formulário de dados quando um formulário é enviado.

O atributo **accept-charset** especifica as codificações de caracteres que estão a ser utilizados para o envio do formulário.

- UTF-8: Codificação de caracteres para Unicode.
- ISO-8859-1: Codificação de caracteres para o alfabeto latino

```
49  <form action="cadastrar.php" method="get" accept-charset="ISO-8859-1">
50      Primeiro nome: <input type="text" name="pNome"><br>
51      Último nome: <input type="text" name="uNome"><br>
52      <input type="submit" value="ENVIAR">
53  </form>
```

Outros atributos da tag <form>

O atributo `method` especifica como enviar os dados (o formulário de dados são enviados para a página especificada no atributo `action`). O formulário de dados podem ser enviados como variáveis de URL (com `method = "get"`) ou como pós HTTP transação (com `method = "post"`).

Notas sobre GET:

- Acrescenta form-data para o URL em pares nome / valor
- O comprimento de um URL é limitado (cerca de 3000 caracteres)
- Nunca use GET para enviar dados que requer extrema segurança(Será visível na URL)
- Útil para envios de formulários onde um usuário deseja marcar o resultado
- GET é melhor para dados não-seguros, como sequências de consulta no Google

Notas sobre POST:

- Acrescenta-forma de dados no interior do corpo do pedido HTTP (dados não é mostrada é em URL)
- Não tem limitações de tamanho envios de formulários com POST não podem ser marcadas

Outros atributos da tag <form>

O atributo **autocomplete** especifica se um formulário deve ter autocomplete ligado ou desligado.

Quando o preenchimento automático está ligado, o navegador irá completar automaticamente com base nos valores que o usuário tenha entrado antes.

O valor padrão é on.

```
<form action="cadastrar.php" name="frmCadastrar" method="get"
      autocomplete="off" accept-charset="ISO-8859-1">
  Primeiro nome: <input type="text" name="pNome"><br>
  Último nome: <input type="text" name="uNome"><br>
  <input type="submit" value="ENVIAR">
</form>
```