



Universidade Federal de Santa Catarina
Curso de Tecnologias da Informação e Comunicação
Desenvolvimento de Sistemas Web
Professor: Adriano de Oliveira

Desenvolvimento de Sistemas Web

Prof.: Adriano de Oliveira
email: adriano.inovar@gmail.com
Turma: 05652 - 3-2020-2 e 5-1830-2

ESTRUTURA E SINTAXE HTML

HTML

- ✓ A **HTML** – HyperText Markup Language (Linguagem de Marcação de Hipertexto) é uma linguagem de marcação usada para a construção de páginas na web.
- ✓ O **W3C** é o responsável pela definição das especificações.

MOTOR DE RENDERIZAÇÃO

- ✓ Motor de renderização é um software que transforma conteúdo em linguagem de marcação e informações de formatação (como CSS) em um conteúdo formatado para ser exibido na tela.
 - Gecko – Firefox
 - Chrome – Chromium (Blink)
 - WebKit – Safari
 - Presto – Opera
 - Edge – EdgeHTML/Chromium

ESTRUTURA BÁSICA

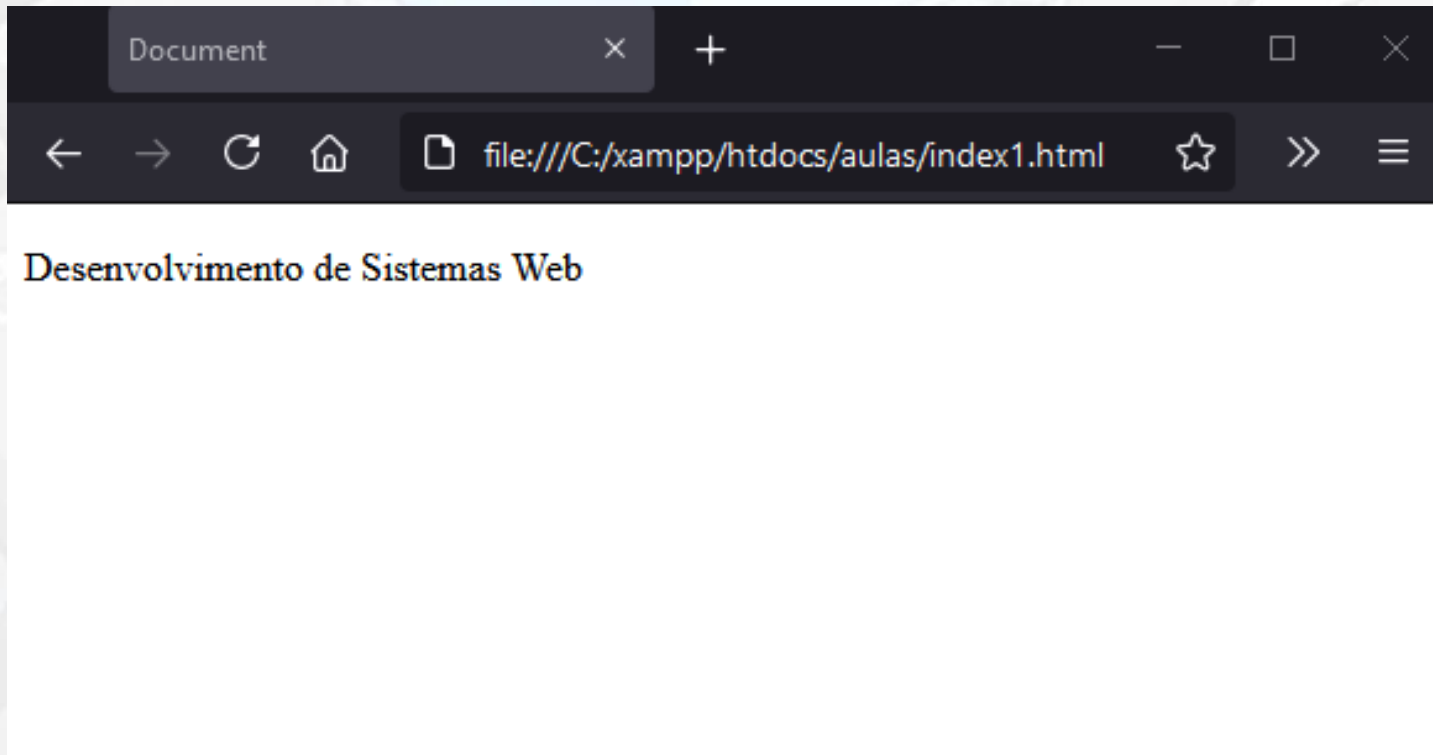
- ✓ Basicamente, um documento *HTML* é composto por elementos hierarquicamente organizados.
- ✓ Para inserir um elemento em um documento *HTML*, devemos utilizar as *tags* correspondentes a esse elemento. As *tags* são definidas com parênteses angulares (*< e >*).

ESTRUTURA BÁSICA

- ✓ Os elementos podem possuir atributos e conteúdo.
- ✓ Os atributos são formados por nome e valor.
- ✓ Normalmente, os valores dos atributos são definidos dentro de aspas dupla e o conteúdo dos elementos é um texto ou outros elementos.



ESTRUTURA BÁSICA



ESTRUTURA BÁSICA

<html>

- ✓ Os elementos *HTML*, com exceção do DOCTYPE, devem ser inseridos no conteúdo do elemento *html*. Esse elemento é aberto com a tag <html>, fechado com a tag </html> e deve conter exatamente um elemento *head* seguido de exatamente um elemento *body*.
- ✓ Recomenda-se a utilização do atributo *lang*, que indica a língua utilizada no documento HTML.

ESTRUTURA BÁSICA

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

ESTRUTURA BÁSICA

<head>

- ✓ A principal função do elemento *head* é agrupar informações sobre o documento HTML (meta informação).
- ✓ São exemplos de meta informações: o encoding, a taxa de atualização, o autor, a descrição e as palavras chaves do documento HTML.

ESTRUTURA BÁSICA

```
<head>
```

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

```
<meta http-equiv="refresh" content="30">
```

```
<meta name="author" content="Nome do Autor">
```

```
<meta name="description" content="Descrição da página">
```

```
<meta name="keywords" content="Palavras chave">
```

```
</head>
```


ESTRUTURA BÁSICA

- ✓ O elemento *head* é aberto e fechado pelas tags `<head>` e `</head>` respectivamente.
- ✓ O corpo do *head* deve conter uma ocorrência do elemento *title*, esse elemento define o título do documento *HTML*.

```
<head>
```

```
...
```

```
<title>Título da Página</title>
```

```
</head>
```


ESTRUTURA BÁSICA

<body>

- ✓ O conteúdo de uma página web deve ser definido no corpo do elemento *body*.
- ✓ Esse elemento é aberto pela tag **<body>** e fechado pela tag **</body>**.

```
<body>
```

```
    <h1>Desenvolvimento de Sistemas Web</h1>
```

```
</body>
```

ESTRUTURA BÁSICA

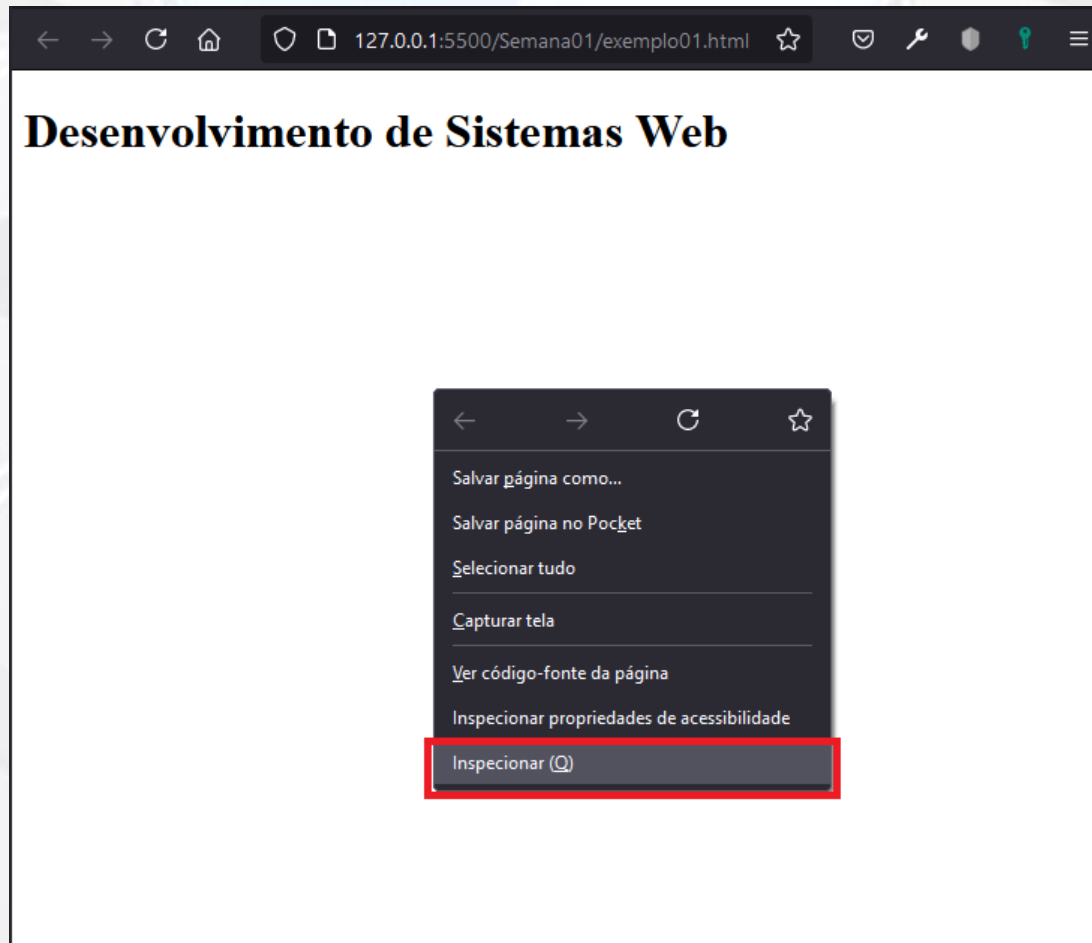
✓ Comentários

- Podemos adicionar *comentários* em um documento HTML dentro das tags `<!-- e -->`.
- Os comentários são ignorados pelos navegadores.

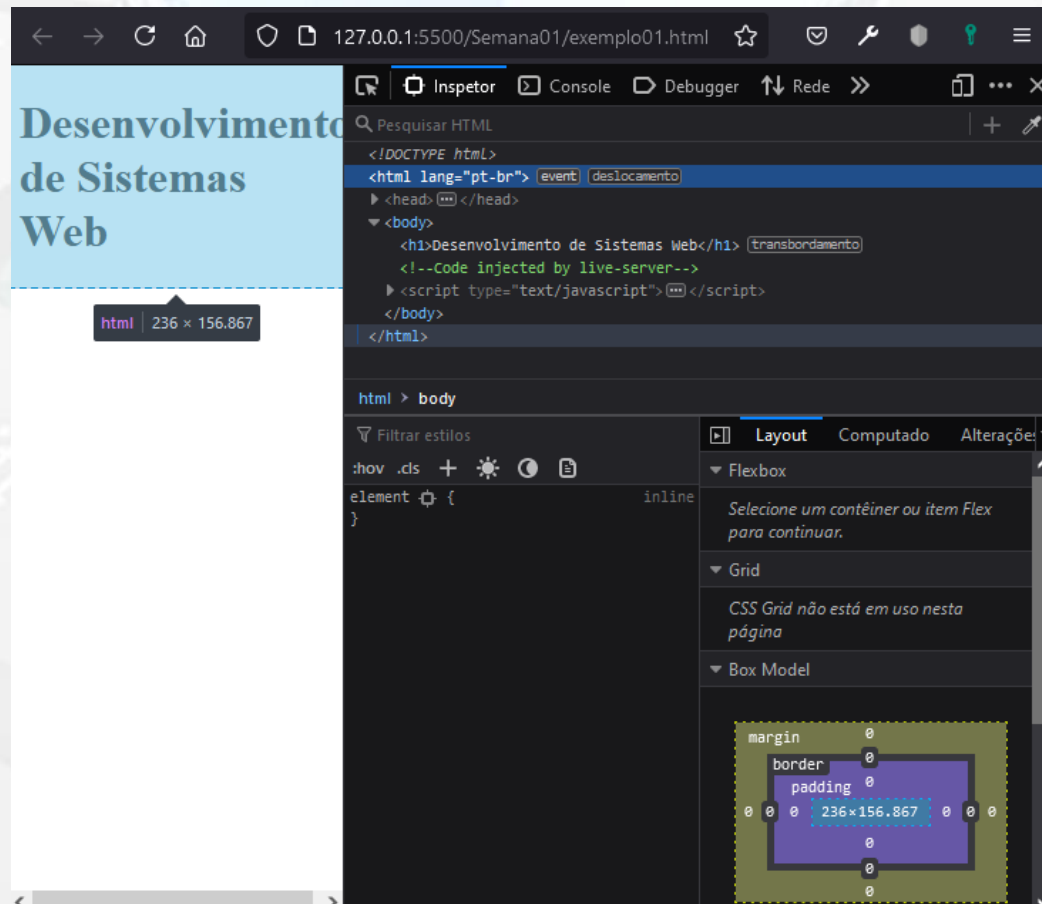
ESTRUTURA BÁSICA

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

FERRAMENTAS DE DESENVOLVIMENTO WEB



FERRAMENTAS DE DESENVOLVIMENTO WEB



TÍTULOS

- ✓ Uma página web pode conter uma ***hierarquia de títulos*** para estabelecer uma divisão do seu conteúdo.
- ✓ Para inserir ***títulos*** em uma página web, devemos utilizar os elementos ***h1, h2, h3, h4, h5 e h6***.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Exemplo de Títulos</title>
</head>
<body>
  <h1>Título 1</h1>
  <h2>Título 2</h2>
  <h3>Título 3</h3>
  <h4>Título 4</h4>
  <h5>Título 5</h5>
  <h6>Título 6</h6>
</body>
</html>
```



Título 1

Título 2

Título 3

Título 4

Título 5

Título 6

TÍTULOS

- ✓ ***Títulos*** não são exibidos lado a lado e sim um embaixo do outro.
- ✓ Por padrão ***títulos*** são exibidos como blocos e ocupam todo o espaço horizontal.
- ✓ Esse é o comportamento padrão dos ***block-level elements***.

PARÁGRAFOS

- ✓ Os *parágrafos* de uma página web são definidos através do elemento *p*.
- ✓ Assim como os títulos, normalmente, os navegadores definem os parágrafos como *block-level elements*.

PARÁGRAFOS

- ✓ Caso seja necessário forçar uma quebra de linha entre duas palavras contidas em um *parágrafo*, podemos utilizar o elemento ***br***.

TEXTO PRÉ-FORMATADO

- ✓ É possível inserir *texto formatado* com quantos espaços e quebras de linha desejarmos através do elemento *pre*.
- ✓ O *texto* contido no conteúdo de um elemento *pre* é exibido com todos os espaços e quebras de linha inseridos no documento HTML.

CÓDIGO

- ✓ Para adicionar *códigos* em um documento **HTML**, devemos utilizar o elemento **code**.
- ✓ Normalmente, esse elemento é utilizado para definir *códigos* escritos em alguma linguagem de programação.
- ✓ Os *códigos* como *inline-level elements*, e ocupam horizontalmente somente o espaço necessário.

ELEMENTO *i*, **b**, sub, sup

- ✓ O elemento *i* exibem o seu conteúdo em *itálico*.
- ✓ O elemento **b** exibem o conteúdo em **negrito**.
- ✓ Podemos definir *textos subscrito* ou *sobrescrito* com os elementos *sub* e *sup* respectivamente

ELEMENTO `hr`

- ✓ O elemento HTML `<hr>` representa uma quebra temática entre elementos de nível de parágrafo.
- ✓ Nas versões anteriores do HTML, representava uma linha horizontal.
- ✓ Pode continuar sendo exibida como uma linha horizontal nos navegadores, mas agora está definida em termos semânticos, em vez de termos de apresentação.

LISTAS

- ✓ A linguagem *HTML* define três tipos distintos de *listas*.
 - Lista de descrições
 - Lista com ordem
 - Lista sem ordem

LISTA DE DESCRIÇÕES

- ✓ Para criar uma *lista de descrições*, devemos utilizar o elemento *dl*.
- ✓ Os termos ou nomes são definidos com o elemento *dt*.
- ✓ As descrições são definidas com o elemento *dd*.

LISTA COM ORDEM

- ✓ Para criar uma *lista com ordem*, devemos utilizar o elemento *ol*.
- ✓ Os itens de uma lista com ordem são definidos com o elemento *li*.

LISTA SEM ORDEM

- ✓ Para criar uma *lista sem ordem*, devemos utilizar o elemento *ul*.
- ✓ Os itens de uma *lista sem ordem* são definidos com o elemento *li*.

LINKS

- ✓ Normalmente, um site é formado por um conjunto de páginas que estão interligadas de alguma forma.
- ✓ Para permitir que um usuário navegue de uma página para outra, devemos utilizar os *links*.

LINKS

- ✓ Para criarmos um *link*, devemos utilizar o elemento **a**.
- ✓ Esse elemento possui um atributo chamado *href*.
- ✓ O valor desse atributo indica o caminho relativo ou absoluto de uma outra página.

ÂNCORAS

- ✓ Podemos criar um *link* para uma determinada seção de um documento HTML.
- ✓ Esse recurso é chamado de *ancoragem*.
- ✓ Identificamos o atributo **id** de um elemento e criamos os links utilizando os identificadores #.

IMAGENS

- ✓ Podemos adicionar *imagens* em documento *HTML* com o elemento *img*.
- ✓ Esse elemento possui um atributo chamado *src*, que indica o caminho absoluto ou relativo da imagem.
- ✓ O atributo *alt* define um texto alternativo para a imagem.

TABELAS

- ✓ O elemento ***table*** permite apresentar dados de forma tabular.
- ✓ As linhas de uma ***tabela*** são definidas com o elemento ***tr***.
- ✓ As células de títulos com o elemento ***th***.
- ✓ As células de dados com o elemento ***td***.

TABELAS

- ✓ Os elementos *th* e *td* possuem um atributo chamado *colspan* e outro chamado *rowspan*.
- ✓ O *colspan* define quantas colunas uma célula deve ocupar.
- ✓ O *rowspan* define quantas linhas uma célula deve ocupar.

CABEÇALHO, CORPO E RODAPÉ

- ✓ Para melhorar a semântica das *tabelas*, podemos definir explicitamente o *cabeçalho*, *o corpo e o rodapé* de uma *tabela* através dos elementos *thead*, *tbody* e *tfoot* respectivamente.
- ✓ Facilita a aplicação de estilos **CSS**.

TÍTULOS DE TABELAS

- ✓ Também podemos definir um *título* para uma tabela através do elemento *caption*.
- ✓ Esse elemento deve ser o primeiro filho do elemento *table*.



Universidade Federal de Santa Catarina
Curso de Tecnologias da Informação e Comunicação
Desenvolvimento de Sistemas Web
Professor: Adriano de Oliveira

TRABALHANDO COM FORMULÁRIOS

TAG <form>

- ✓ A tag <form> é usada para criar um *form HTML* para entrada de dados de usuários. O elemento <form> pode conter um ou mais dos seguintes elementos:

<input>

<textarea>

<button>

<select>

<option>

<optgroup>

<fieldset>

<label>

ATRIBUTOS DA TAG <form>

| Atributo | Valor | Descrição |
|----------------|--|--|
| accept-charset | <i>Character_set</i>
accept-charset="UTF-8" | Especifica a codificação de caracteres que devem ser utilizados para o envio do formulário |
| action | <i>URL</i>
action="demo_form.asp" | Especifica para onde (URL) enviar o formulário de dados |
| autocomplete | on
off
autocomplete="on" | Especifica se o autocomplete está ligado ou desligado |

ATRIBUTOS DA TAG <form>

| Atributo | Valor | Descrição |
|----------|---|---|
| enctype | application/x-www-form-urlencoded
multipart/form-data
text/plain
enctype="multipart/form-data" | Especifica como os dados dos formulários devem ser codificados quando enviados para o servidor (somente para o method = "post") |
| method | get
post
method="get" | Especifica o método HTTP para o envio de formulário de dados |
| name | text
name="myForm" | Especifica um nome para o formulário |

ATRIBUTOS DA TAG <form>

| Atributo | Valor | Descrição |
|-------------------|--|---|
| novalidate | novalidate
novalidate | Especifica que o formulário não deve ser validado quando submetido |
| target | _blank
_self
_parent
_top
target="_blank" | Especifica onde será exibido a resposta após o envio do formulário |

TAG `<input>`

- ✓ A tag `<input>` especifica um campo de entrada onde o usuário pode inserir dados.
- ✓ Um campo de entrada pode variar de muitas formas, dependendo do atributo *type*.

ATRIBUTOS DA TAG `<input>`

| Atributo | Valor | Descrição |
|---------------------|---|--|
| <code>accept</code> | <code>file_extension</code>
<code>audio/*</code>
<code>video/*</code>
<code>image/*</code>
<code>media_type</code>
<code>accept="image/*"</code> | Especifica os tipos de arquivos que o servidor aceita (somente para <code>type = "file"</code>) |
| <code>alt</code> | <code>text</code>
<code>alt="Submit"</code> | Especifica um texto alternativo para as imagens (somente para <code>type = "imagem"</code>) |

ATRIBUTOS DA TAG `<input>`

| Atributo | Valor | Descrição |
|---------------------------|---|---|
| <code>autocomplete</code> | <code>on</code>
<code>off</code>
<code>autocomplete="on"</code> | Especifica se um elemento <code><input></code> tem o autocomplete ativado |
| <code>autofocus</code> | <code>autofocus</code>
<code>autofocus</code> | Especifica se um elemento <code><input></code> deve obter automaticamente o foco quando a página é carregada |
| <code>checked</code> | <code>checked</code>
<code>checked</code> | Especifica que um elemento <code><input></code> devem ser pré-selecionado quando a página é carregada (para <code>type = "checkbox"</code> ou <code>type = "radio"</code>) |

ATRIBUTOS DA TAG <input>

| Atributo | Valor | Descrição |
|------------|--------------------------------|--|
| disabled | disabled
disabled | Especifica que um elemento <input> deve ser desativado |
| form | <i>form_id</i>
form="form1" | Especifica uma ou mais forms a qual o elemento <input> pertence |
| formaction | <i>URL</i> | Especifica a URL do arquivo que irá processar o controle de entrada quando o form é enviado (para type = "submit" e type = "imagem") |

ATRIBUTOS DA TAG `<input>`

| Atributo | Valor | Descrição |
|--------------------|---|--|
| formenctype | <code>application/x-www-form-urlencoded</code>
<code>multipart/form-data</code>
<code>text/plain</code>
<code>formenctype="multipart/form-data"</code> | Especifica como os dados dos forms devem ser codificados quando enviá-lo para o servidor (para <code>type = "submit"</code> e <code>type = "imagem"</code>) |
| formmethod | <code>get</code>
<code>post</code>
<code>formmethod="post"</code> | Define o método HTTP para o envio de dados para a URL do atributo <code>action</code> (para <code>type = "submit"</code> e <code>type = "imagem"</code>) |

ATRIBUTOS DA TAG <input>

| Atributo | Valor | Descrição |
|----------------|--|---|
| formnovalidate | formnovalidate
formnovalidate="formnovalidat
e" | Define que os elementos de formulário não deve ser validado quando submetidas |
| formtarget | _blank
_self
_parent
_top
framename
formtarget="_blank" | Especifica onde exibir a resposta que é recebida após enviar o form(para type = "submit" e type = "imagem") |

ATRIBUTOS DA TAG `<input>`

| Atributo | Valor | Descrição |
|----------|------------------------------------|--|
| height | pixels
height="48" | Especifica a altura de um elemento <code><input></code> (somente para type = "imagem") |
| list | datalist_id
list="browsers" | Refere-se a um elemento <code><datalist></code> que contém as opções pré-definidas para um elemento <code><input></code> |
| max | number
date
max="1979-12-31" | Especifica o valor máximo de um elemento <code><input></code> |

ATRIBUTOS DA TAG <input>

| Atributo | Valor | Descrição |
|-----------|------------------------------------|--|
| maxlength | number
maxlength="10" | Especifica o número máximo de caracteres permitidos em um elemento <input> |
| min | number
date
min="2000-01-02" | Especifica um valor mínimo para um elemento <input> |
| multiple | multiple
multiple | Especifica que um usuário pode inserir mais de um valor em um elemento <input> |

ATRIBUTOS DA TAG <input>

| Atributo | Valor | Descrição |
|-------------|----------------------------------|--|
| name | text
name="fullname" | Especifica o nome de um elemento <input> |
| pattern | regexp
pattern="[A-Za-z]{3}" | Especifica uma expressão regular que o elemento <input> é comparado |
| placeholder | text
placeholder="First name" | Especifica uma breve dica que descreve o valor esperado de um elemento <input> |

ATRIBUTOS DA TAG <input>

| Atributo | Valor | Descrição |
|----------|----------------------|---|
| readonly | readonly
readonly | Especifica que um campo de entrada é somente leitura |
| required | required
required | Especifica que um campo de entrada deve ser preenchido antes de enviar o formulário |
| size | number
size="35" | Especifica a largura, em caracteres, de um elemento <input> |

ATRIBUTOS DA TAG <input>

| Atributo | Valor | Descrição |
|----------|-------------------------|--|
| src | URL
src="submit.gif" | Especifica o URL da imagem para usar como um botão de envio (somente para type = "imagem") |
| step | number
step="3" | Especifica os intervalos de número para um campo de entrada |
| value | text
value="John" | Especifica o valor de um elemento <input> |
| width | pixels
width="48" | Especifica a largura de um elemento <input> (somente para type = "imagem") |

ATRIBUTOS DA TAG `<input>`

| Atributo | Valor | Descrição |
|-------------------|---|--|
| <code>type</code> | <code>button</code> / <code>checkbox</code> / <code>color</code>
<code>date</code> / <code>datetime</code>
<code>datetime-local</code>
<code>email</code> / <code>file</code> / <code>hidden</code>
<code>image</code> / <code>month</code> / <code>number</code>
<code>password</code> / <code>radio</code> / <code>range</code>
<code>reset</code> / <code>search</code> / <code>submit</code>
<code>tel</code> / <code>text</code> / <code>time</code> / <code>url</code> / <code>week</code>
<code>type="text"</code> | Especifica o tipo do elemento <code><input></code> |

ATRIBUTO type

| Valor | Descrição |
|----------------|---|
| button | Define um botão (usado principalmente com JavaScript para ativar um script) |
| checkbox | Define uma caixa de seleção |
| color | Define um seletor de cores |
| date | Define um controle de data (ano, mês e dia) |
| datetime | Define um controle de data e hora (ano, mês, dia, hora, minuto, segundo, e milissegundo (com base no fuso horário UTC)) |
| datetime-local | Define um controle de data e hora (ano, mês, dia, hora, minuto, segundo, e milissegundo (sem fuso horário)) |

ATRIBUTO type

| Valor | Descrição |
|-----------------|---|
| email | Define um campo para um endereço de e-mail |
| file | Define um campo para seleção de arquivo e um botão "Browse ..." (para uploads de arquivos) |
| hidden | Define um campo de entrada oculto |
| image | Define uma imagem como um botão enviar |
| month | Define um controle de mês e ano |
| number | Define um campo para inserir um número |
| password | Define um campo de senha (caracteres são mascarados) |

ATRIBUTO type

| Valor | Descrição |
|---------------|--|
| radio | Define um botão de opção |
| range | Define um controle para a entrada de um número (como um controle deslizante) |
| reset | Define um botão de reset (reseta todos os valores do formulário para os valores padrão) |
| search | Define um campo de texto para digitar uma seqüência de pesquisa |
| submit | Define um botão de envio |
| tel | Define um campo para digitar um número de telefone |

ATRIBUTO type

| Valor | Descrição |
|--------------|---|
| text | Define um campo de texto de linha única (largura padrão é de 20 caracteres), é o padrão para <input> |
| time | Define um controle para a entrada de hora |
| url | Define um campo para digitar uma URL |
| week | Define um controle para escolher semana |

TAG `<textarea>`

- ✓ A tag `<textarea>` define um controle de entrada de texto de várias linhas.
- ✓ Uma *área de texto* pode conter um número ilimitado de caracteres.
- ✓ O tamanho de uma *área de texto* pode ser especificado pelos atributos *cols e rows*, ou ainda, através de *CSS* pelas propriedades *height e width*.

ATRIBUTOS DA TAG <textarea>

| Atributo | Valor | Descrição |
|-----------|----------------------------------|--|
| autofocus | autofocus
autofocus | Especifica que uma área de texto deve obter automaticamente o foco quando a página é carregada |
| cols | number
cols="50" | Especifica a largura visível de uma área de texto |
| disabled | disabled
disabled | Especifica que uma área de texto deve ser desativado |
| form | <i>form_id</i>
form="usrform" | Especifica um ou mais forms a qual a área de texto pertence |

ATRIBUTOS DA TAG <textarea>

| Atributo | Valor | Descrição |
|-------------|----------------------------------|---|
| maxlength | number
maxlength="50" | Especifica o número máximo de caracteres permitidos na área de texto |
| name | text
name="fullname" | Especifica o nome de um elemento <textarea> |
| placeholder | text
placeholder="First name" | Especifica uma breve dica que descreve o valor esperado de um elemento <textarea> |
| required | required
required | Especifica que uma área de texto deve ser preenchida |

ATRIBUTOS DA TAG <textarea>

| Atributo | Valor | Descrição |
|-------------|--|--|
| rows | number
<code>cols="50"</code> | Especifica o número de linhas visíveis numa área de texto |
| wrap | hard
soft
off
<code>wrap="hard"</code> | Especifica como o texto em uma área de texto deve ser ajustada quando apresentados um em form |

TAG **<button>**

- ✓ A tag **<button>** define um botão.
- ✓ Dentro de um elemento **<button>** você pode colocar conteúdo, como texto ou imagens. Esta é a diferença entre este elemento e botões criados com o elemento **<input>**.

ATRIBUTOS DA TAG <button>

| Atributo | Valor | Descrição |
|------------|---------------------------|--|
| autofocus | autofocus
autofocus | Especifica que um botão deve obter automaticamente o foco quando a página é carregada |
| disabled | disabled
disabled | Especifica que um botão deve ser desativado |
| form | form_id
form="usrform" | Especifica um ou mais forms a qual o botão pertence |
| formaction | URL | Especifica a URL do arquivo que irá processar o controle de entrada quando o form é enviado (para type = "submit") |

ATRIBUTOS DA TAG <button>

| Atributo | Valor | Descrição |
|-------------|---|---|
| formenctype | application/x-www-form-urlencoded
multipart/form-data
text/plain
formenctype="multipart/form-data" | Especifica como os dados dos forms devem ser codificados quando enviá-lo para o servidor (para type = "submit") |
| formmethod | get
post
formmethod="post" | Define o método HTTP para o envio de dados para a URL do atributo action (para type = "submit") |

ATRIBUTOS DA TAG <button>

| Atributo | Valor | Descrição |
|----------|--|--|
| name | name
name="subject" | Especifica um nome para o botão |
| type | button
reset
submit
type="submit" | Especifica o tipo de botão |
| value | text
value="fav_HTML" | Especifica um valor inicial para o botão |

ATRIBUTOS type

| Valor | Descrição |
|---------------|---|
| button | É um botão que pode ser clicado |
| reset | É um botão de reset (zera os dados do formulário para seus valores iniciais) |
| submit | É um botão de envio (envia os dados do formulário) |

TAG `<select>`

- ✓ A tag `<select>` é usado para criar uma *lista drop-down*.
- ✓ A tag `<option>` dentro do elemento `<select>` definir as opções disponíveis na *lista*.

ATRIBUTOS DA TAG <select>

| Atributo | Valor | Descrição |
|------------------|---|--|
| autofocus | autofocus
autofocus | Especifica que um a lista drop-down deve obter automaticamente o foco quando a página é carregada |
| disabled | disabled
disabled | Especifica que um lista drop-down deve ser desativada |
| form | form_id
form="usrform" | Especifica um ou mais forms a qual a lista drop-down pertence |

ATRIBUTOS DA TAG <select>

| Atributo | Valor | Descrição |
|-----------------|------------------------------|--|
| multiple | multiple
multiple | Especifica que várias opções podem ser selecionadas |
| name | name
name="cars" | Especifica um nome para a lista drop-down |
| required | required
required | Especifica que uma lista drop-down deve ser preenchida |
| size | number
size="3" | Define o número de opções visíveis em uma lista drop-down |

TAG `<option>`

- ✓ A tag `<option>` define uma opção em uma *lista de seleção*.
- ✓ Os elementos `<option>` vão dentro de um elemento `<select>`.
- ✓ Se você tem uma longa lista de opções, você pode agrupá-las com a tag `<optgroup>`.

ATRIBUTOS DA TAG <option>

| Atributo | Valor | Descrição |
|----------|-----------------------|---|
| disabled | disabled
disabled | Especifica que uma opção deve ser desativada |
| label | text
label="Volvo" | Especifica um rótulo mais curto para uma opção |
| selected | selected
selected | Especifica que uma opção deve ser pré-selecionado quando a página é carregada |
| value | text
value="volvo" | Especifica o valor a ser enviado para um servidor |

TAG **<fieldset>**

- ✓ A tag **<fieldset>** é utilizado para agrupar elementos relacionados em um formulário.
- ✓ A tag **<fieldset>** desenha uma caixa em torno dos elementos relacionados.
- ✓ A tag **<legend>** define uma legenda para um elemento **<fieldset>**.

ATRIBUTOS DA TAG `<fieldset>`

| Atributo | Valor | Descrição |
|-----------------------|---|---|
| <code>disabled</code> | <code>disabled</code>
<code>disabled</code> | Especifica que um grupo de elementos de formulário relacionados deve ser desativado |
| <code>form</code> | <code>form_id</code>
<code>form="usrform"</code> | Especifica uma ou mais forms ao qual o fieldset pertence |
| <code>name</code> | <code>name</code>
<code>name="cars"</code> | Especifica um nome para um fieldset |

TAG <div>

- ✓ Utilizado para dividir blocos de páginas.
- ✓ Pode criar sessões, e dentro delas ter imagens, links, textos e o que mais você quiser.
- ✓ *div é block-level elements*

TAG ``

- ✓ Utilizado para dividir blocos de páginas.
- ✓ Pode criar sessões, e dentro delas ter imagens, links, textos e o que mais você quiser.
- ✓ ***`span` é inline-level elements***

Containers HTML5

`<header>`

`<nav>`

`<section>`

`<aside>`

`<header>`

`<article>`

`<footer>`

`<footer>`

TAG <header>

- ✓ O elemento header representa uma área de introdução.
- ✓ Pode ser utilizado para agrupar índices de conteúdo, campos de busca, cabeçalho do site com título ou logo.

TAG <section>

- ✓ A tag section define uma nova seção do site, uma área genérica do site.
- ✓ Por exemplo a home pode ser dividida em destaque, novidades, chamada para conteúdo completo.

TAG <footer>

- ✓ A tag footer representa o rodapé do site, ou de uma seção.

TAG <nav>

- ✓ O elemento nav é utilizado para representar uma seção da página que contém links para outras partes do site.
- ✓ Este elemento deve ser utilizado para grupos de links importantes do site tipicamente menus de navegação.

TAG <article>

- ✓ O elemento `article` é utilizado para representar o conteúdo do site propriamente dito, como posts, artigos e outros textos em geral.

TAG <aside>

- ✓ A tag `aside` representa um bloco de conteúdo que referencia o conteúdo que envolve o elemento `aside`.

ELEMENTOS FIGURE E FIGCAPTION

- ✓ Quando adicionamos uma imagem em nosso documento HTML podemos envolvê-lo e dar uma descrição.
- ✓ A ideia desses dois novos elementos é realmente criar um vínculo semântico entre a imagem e o texto.

ELEMENTOS FIGURE E FIGCAPTION

```
<figure>
  
  <figcaption>Este texto tem uma relação semântica com a imagem
    <br />
    e será apresentado pelo navegador</figcaption>
</figure>
```

- ✓ Podemos ainda utilizar o elemento figure e figcaption para várias imagens, como o exemplo abaixo:

```
<figure>
  
  
  
  <figcaption>Este texto tem uma relação semântica com a imagem
    <br />
    e será apresentado pelo navegador</figcaption>
</figure>
```

VALIDAÇÕES EM FORMULÁRIOS

```
<form>
  <fieldset>
    <legend>Contate-nos</legend>
    <ul>
      <li>
        <label for="email">E-mail</label>
        <input required="required" type="email" id="email" name="email" />
      </li>
      <li>
        <label for="tel">Telephone</label>
        <input required="required" type="tel" id="tel" name="tel" />
      </li>
      <li>
        <label for="end">Endereço</label>
        <input required="required" type="text" id="end" name="tel" />
      </li>
      <li>
        <label for="url">Endereço Eletrônico</label>
        <input required="required" type="url" id="url" name="url" />
      </li>
    </ul>
    <input type="submit" value="Submit this" />
  </fieldset>
</form>
```

TIPO “email”

- ✓ Quando utilizamos no atributo type o valor “email” estamos indicando ao navegador que aquela informação deverá parecer com um e-mail válido.
- ✓ A validação se dá apenas sobre a informação e não sobre a existência da conta de e-mail.

TIPO “tel”

- ✓ Quando utilizamos no atributo type o valor “tel” estamos indicando ao navegador que aquela informação deverá parecer com um telefone válido.
- ✓ Isso quer dizer que deverá ser aceito qualquer tipo de caracteres, não somente números. Por exemplo o caractere ‘+’.

TIPO “url”

- ✓ Quando utilizamos no atributo type o valor “url” estamos indicando ao navegador que aquela informação deverá parecer com uma URL válida.
- ✓ A restrição será apenas sobre a informação e não sobre a existência da URL.

FORMULÁRIO DE PESQUISA

- ✓ Para um formulário de pesquisa usamos o `type="search"`.
- ✓ Isso não concederá nenhum efeito ou privilégio a mais ao seu formulário, apenas permitirá que o navegadores saibam que o campo é de pesquisa.

FORMULÁRIO DE PESQUISA

```
<form>
  <label for="pesq">Pesquisar:</label>
  <input required="required" type="search" id="pesq" name="pesq" />
  <input type="submit" value="Agora!" />
</form>
```

TIPO “datetime” e “datetime-local”

Data/Hora: : UTC

| Outubro 2012 | | | | | | |
|--------------|-----|-----|-----|-----|-----|-----|
| Seg | Ter | Qua | Qui | Sex | Sáb | Dom |
| 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 |
| 5 | 6 | 7 | 8 | 9 | 10 | 11 |

Hoje

Tipo de entrada “datetime”

Data/Hora: :

| Outubro 2012 | | | | | | |
|--------------|-----|-----|-----|-----|-----|-----|
| Seg | Ter | Qua | Qui | Sex | Sáb | Dom |
| 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 |
| 5 | 6 | 7 | 8 | 9 | 10 | 11 |

Hoje

Tipo de entrada “datetime-local”

TIPO “date” e “time”

Data:

| Seg | Ter | Qua | Qui | Sex | Sáb | Dom |
|-----|-----|-----|-----|-----|-----|-----|
| 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 |
| 5 | 6 | 7 | 8 | 9 | 10 | 11 |

Hoje

Tipo de entrada "date"

Hora:

Tipo de entrada "time"

TIPO “week”

Semana: 2012-W40 ▾

| Setembro | | | | | | | |
|----------|-----|-----|-----|-----|-----|-----|-----|
| | Seg | Ter | Qua | Qui | Sex | Sáb | Dom |
| 35 | 27 | 28 | 29 | 30 | 31 | 1 | 2 |
| 36 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 37 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 38 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 39 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 40 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

Hoje

Tipo de entrada "week"

TIPO “month”

Mês: 2012-10 ▼

| | | | | | | | | |
|------|---------|-----|-----|-----|-----|-----|------|-------------|
| ◀ | Outubro | | | | | ▶ | 2012 | ⬆
⬇
⬆ |
| Seg | Ter | Qua | Qui | Sex | Sáb | Dom | | |
| 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 | | |
| 5 | 6 | 7 | 8 | 9 | 10 | 11 | | |
| Hoje | | | | | | | | |

Tipo de entrada "month"

SELETOR DE NÚMERO

```
<label for="numero">Valor:</label>  
<input min="0" max="10" step="0.5" required="required" type="number" id="numero"  
name="numero" />
```

Valor:

! O valor deve ser menor ou igual a 10.

Navegar restringindo corretamente o valor máximo configurado

SLIDER

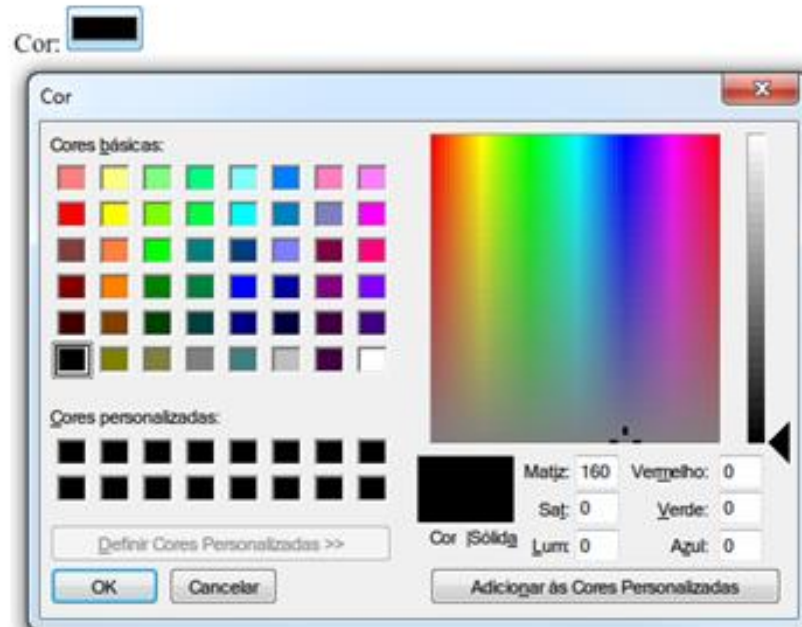
```
<label for="range">Potência</label>  
<input min="0" max="10" step="0.5" required="required" type="range" id="range"  
name="range" />
```

Potência 

Controle deslizante no Google Chrome

SELETOR DE COR

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



ELEMENTO OUTPUT

```
<label for="range">Potência</label>  
<input min="0" max="10" step="0.5" required="required" type="range" id="range"  
name="range" />  
<output onforminput="value=range.value"></output>
```

Potência  5.5

Exemplo de utilização do elemento output

AUTOCOMPLETE COM LIST E DATALIST

```
<label for="time">Qual é o seu time favorito?</label>
<input list="time-opcoes" type="text" id="time" name="time" />
<datalist id="time-opcoes">
  <option label="São Paulo" value="São Paulo" />
  <option label="Corinthians" value="Corinthians" />
  <option label="Palmeiras" value="Palmeiras" />
  <option label="Santos" value="Santos" />
</datalist>
```

Lista para completar criada para uma entrada de dados

São Paulo

Santos

ELEMENTO DE PROGRESSÃO

```
<p>Estamos executando uma atividade, por favor espere.</p>  
<progress value="390" max="500"><span>78</span>% completado</progress>
```

Estamos executando uma atividade, por favor espere.



*Utilizando o elemento progress para indicar um
porcentagem de uma tarefa*

VÍDEOS NOS DOCUMENTOS

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Galeria de Vídeos</title>
  <meta charset="utf-8" />
</head>
<body>
  <video src="video1.mp4"></video>
</body>
</html>
```

VÍDEOS NOS DOCUMENTOS

```
<video width="640" height="480" controls="controls">  
  <source src="video1.mp4" type="video/mp4" />  
  <source src="video1.webm" type="video/webm" />  
  <source src="video1.ogv" type="video/ogg" />  
</video>
```

ATRIBUTO “poster”

```
<video width="640" height="480" controls="controls" poster="img\poster.png">  
  <source src="video1.mp4" type="video/mp4" />  
  <source src="video1.webm" type="video/webm" />  
  <source src="video1.ogv" type="video/ogg" />  
</video>
```


ÁUDIOS NOS DOCUMENTOS

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Galeria de Musicas</title>
  <meta charset="utf-8" />
</head>
<body>
  <audio src="musical.mp3" controls="controls" />
</body>
</html>
```

ÁUDIOS NOS DOCUMENTOS

```
<audio controls="controls">  
  <source src="musical.mp3" type="audio/mp3" />  
  <source src="musical.ogg" type="audio/ogg" />  
</audio>
```

TRABALHANDO COM CANVAS

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Galeria de Musicas</title>
  <meta charset="utf-8" />
  <style>
    canvas {
      border: 1px solid #000;
    }
  </style>
</head>
<body>
  <!-- Adicionando o elemento canvas -->
  <canvas id="canvas" width="640" height="480"></canvas>
</body>
</html>
```

```
// Recupera o elemento canvas e o contexto 2D
var canvas = document.getElementById('canvas').getContext('2d');
// Altera a distância da sombra no eixo X
canvas.shadowOffsetX = 10;
// Altera a distância da sombra no eixo Y
canvas.shadowOffsetY = 10;
// Define o grau de desfocagem
canvas.shadowBlur = 10;
// Define a cor da sombra
canvas.shadowColor = 'rgba(200, 0, 200, .3)';
// Define o estilo (cor) do preenchimento
canvas.fillStyle = 'rgba(200, 0, 200, 1)';
// Preenche o quadrado 100 x 100 no ponto 0, 0
canvas.fillRect(0, 0, 100, 100);
```



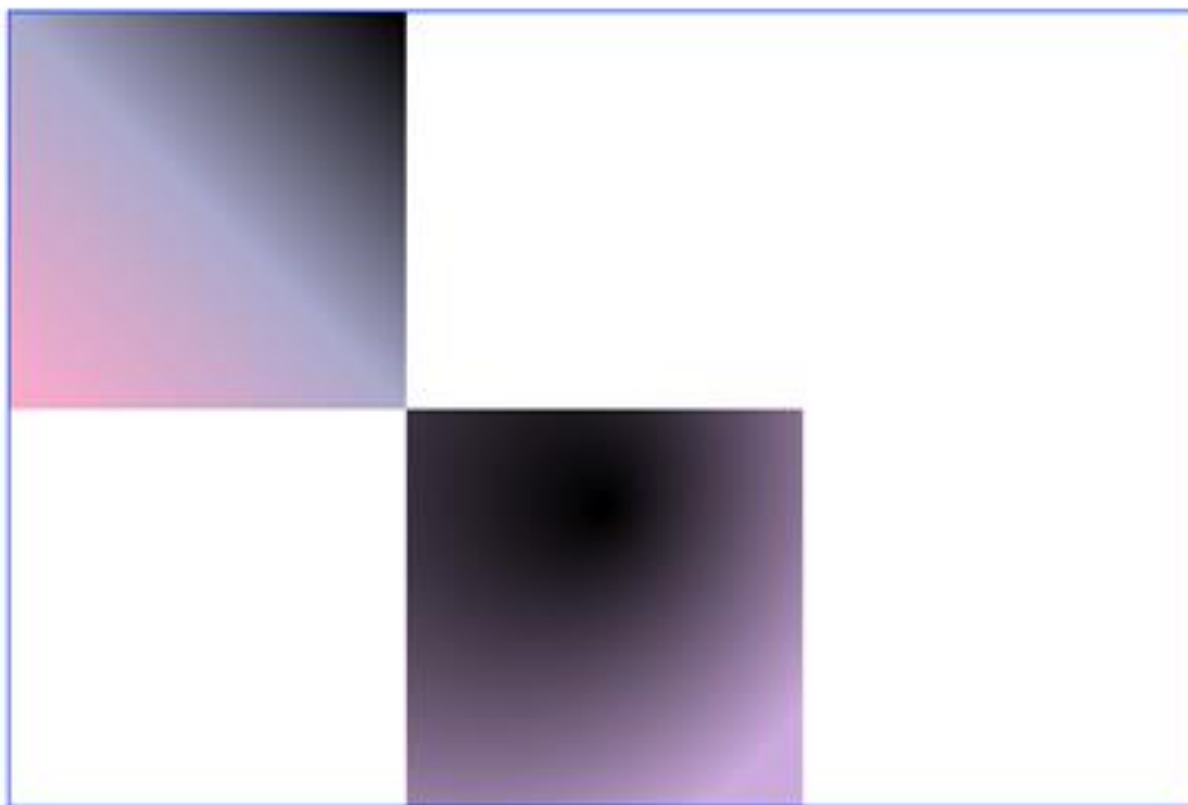
```
// Recupera o elemento canvas e o contexto 2D
var canvas = document.getElementById('canvas').getContext('2d');
// Altera a distância da sombra no eixo X
canvas.shadowOffsetX = 10;
// Altera a distância da sombra no eixo Y
canvas.shadowOffsetY = 10;
// Define o grau de desfocagem
canvas.shadowBlur = 10;
// Define a cor da sombra
canvas.shadowColor = 'rgba(200, 0, 200, .3)';
// Define o estilo (cor) do preenchimento
canvas.fillStyle = 'rgba(200, 0, 200, 1)';
// Define o estilo do pincel
canvas.strokeStyle = '#09c';
// Define a largura da linha
canvas.lineWidth = 5;
// Cria um retângulo preenchido 100 x 100 na posição 0, 0
canvas.fillRect(0, 0, 100, 100);
// Limpa a área de 50 x 50 na posição 25, 25
canvas.clearRect(25, 25, 50, 50);
// Desenha um retângulo vazado de 50 x 50 na posição 25, 25
canvas.strokeRect(25, 25, 50, 50);
```




Mais um quadrado desenhado em 2D

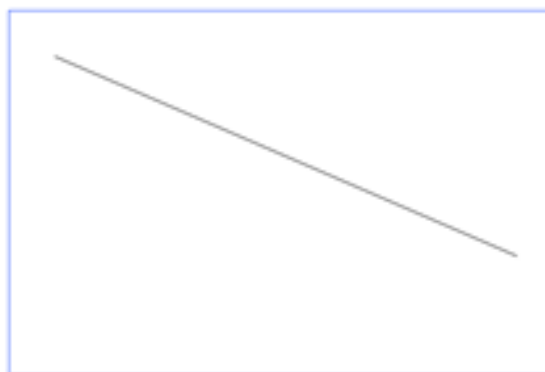
GRADIENTE COM FORMAS

```
<canvas id="canvas" width="600" height="400">O elemento canvas de HTML5 não é
suportado
    neste browser. Utilize um navegador mais moderno!
</canvas>
<script type="text/javascript">
    var canvas = document.getElementById('canvas').getContext('2d');
    var grd = canvas.createLinearGradient(0, 200, 200, 0);
    grd.addColorStop(0, '#fac');
    grd.addColorStop(.5, '#aac');
    grd.addColorStop(1, '#000');
    canvas.fillStyle = grd;
    canvas.strokeStyle = '#09c';
    canvas.lineWidth = 5;
    canvas.fillRect(0, 0, 200, 200);
    canvas.closePath();
    var grd = canvas.createRadialGradient(300, 250, 2, 200, 200, 250);
    grd.addColorStop(0, '#000');
    grd.addColorStop(1, '#cad');
    canvas.fillStyle = grd;
    canvas.fillRect(200, 200, 200, 200);
    canvas.closePath();
</script>
```



Dois quadrados preenchidos com gradientes.

```
<canvas id="canvas" width="600" height="400">O elemento canvas de HTML5 não é  
suportado  
    neste browser. Utilize um navegador mais moderno!  
</canvas>  
<script type="text/javascript">  
    var canvas = document.getElementById('canvas').getContext('2d');  
    canvas.moveTo(50, 50);  
    canvas.lineTo(560, 270);  
    canvas.stroke();  
</script>
```



*Desenhamos a linha utilizando os
métodos discutidos*

DESENHANDO LINHAS E CAMINHOS

```
<canvas id="canvas" width="600" height="400">O elemento canvas de HTML5 não é
suportado
    neste browser. Utilize um navegador mais moderno!
</canvas>
<script type="text/javascript">
    var canvas = document.getElementById('canvas').getContext('2d');
    canvas.beginPath();
    canvas.moveTo(20, 20);
    canvas.lineTo(530, 370);
    canvas.lineTo(20, 370);
    canvas.lineTo(20, 20);
    canvas.closePath();
    canvas.strokeStyle = '#000';
    canvas.lineWidth = 3;
    canvas.fillStyle = '#ccc';
    canvas.fill();
    canvas.stroke();
</script>
```



```
<canvas id="canvas" width="600" height="400">O seu navegador não suporta o
elemento canvas, por favor troque!
<script type="text/javascript">
    var canvas;
    var context;

    canvas = document.getElementById('canvas');
    context = canvas.getContext('2d');

    function desenhaPoligono(numeroLados, radius) {
        var xCenter = parseInt(canvas.width / 2);
        var yCenter = parseInt(canvas.height / 2);
        context.clearRect(0, 0, canvas.width, canvas.height);

        context.beginPath();
        var xPos = xCenter + radius * Math.cos(2 * Math.PI * 0 / numeroLados);
        var yPos = yCenter + radius * Math.sin(2 * Math.PI * 0 / numeroLados);
        context.moveTo(xPos, yPos);
        for (i = 1; i <= numeroLados; i++) {
            xPos = xCenter + radius * Math.cos(2 * Math.PI * i / numeroLados);
            yPos = yCenter + radius * Math.sin(2 * Math.PI * i / numeroLados);
            context.lineTo(xPos, yPos);
        }

        context.closePath();

        context.lineWidth = 30;
        context.lineJoin = 'round';
        context.stroke();

        context.fillStyle = '#0CF';
        context.fill();
    }

    desenhaPoligono(12, 150);
</script>
```

```
<canvas id="canvas" width="600" height="400">O seu navegador não suporta o  
elemento canvas, por favor troque!  
<script type="text/javascript">  
  var canvas = document.getElementById('canvas').getContext('2d');  
  canvas.arc(100, 100, 40, 0, 2 * Math.PI, true);  
  canvas.strokeStyle = '#000';  
  canvas.lineWidth = 5;  
  canvas.stroke();  
</script>
```



*Figura 7*Desenhando um
circulo



*Figura 6*Desenhando
apenas um arco

```
<canvas id="canvas" width="600" height="400">O seu navegador não suporta o  
elemento canvas, por favor troque!  
</canvas>  
<script type="text/javascript">  
    var canvas = document.getElementById('canvas').getContext('2d');  
    // Define a fonte que será utilizada  
    canvas.font = 'bold 48px Tahoma';  
    // Cor de preenchimento  
    canvas.fillStyle = '#CAC';  
    // Desenha o texto preenchido na coordenada (10, 100)  
    canvas.fillText('Escrevendo', 10, 100);  
    // Define o estilo do pincel  
    canvas.strokeStyle = '#000';  
    // Define a largura da linha  
    canvas.lineWidth = 3;  
    // Define o estilo de preenchimento  
    canvas.fillStyle = '#DED';  
    // Define o alinhamento do texto  
    canvas.textAlign = 'center';  
    // Desenha o texto preenchido na coordenada (320, 200)  
    canvas.fillText('com Canvas', 320, 200);  
    // Desenha o texto sem preenchimento na coordenada (320, 200)  
    canvas.strokeText('com Canvas', 320, 200);  
</script>
```

Escrevendo com Canvas

Manipulação de texto com Canvas

Geolocalização

- ✓ A API de geolocalização permite que você compartilhe sua localização em sites confiáveis.
- ✓ A latitude e longitude são disponibilizadas na página via JavaScript, que por sua vez pode ser enviado a um servidor web e fazer coisas como encontrar locais ao seu redor ou mostrar sua posição em um mapa.

Drag and Drop

- ✓ No contexto de aplicações web, drag and drop é o recurso que permite ao usuário selecionar determinado elemento de uma página e arrastá-lo para outra posição na mesma página.
- ✓ Este recurso é importante pois permite alto nível de personalização do ambiente operacional por parte do usuário, característica desejável pela maioria deles.

Drag and Drop

- ✓ Para tornar um elemento “arrastável”, basta colocar o atributo “draggable” para true.

```

```

Fonte W3schools:

✓ [HTML Tutorial \(w3schools.com\)](https://www.w3schools.com/html/)