HTML Citações

# Citações e HTML

## <addres>

Serve para informações do autor ou do texto da página como: endereços, site, email etc. Sempre que finalizar uma linha terá que usar a tag <br> para quebrar a linha.

## <bdo>

Serve para inverter o texto com o parâmetro dir=”” que pode ter os valores rtl e ltr

Links HTML

# Links marcadores

## Href=#000

Quando clicar no link você será mandado para a parte da página cujo id terá os valores após a cerquilha.

# Como navegar por links

## Saindo do diretório

Coloque ../ para sair do diretório

## Diretório raiz

Coloque “./” ou não

Imagens HTML

# Imagens

## Style=”float: ;”

O valor float é um código CSS, mas pode ser usado em HTML através do parâmetro style. O float serve para deixar a imagem em uma parte da tela como nas atribuições de valores: left e right

# Images map

## <map> e <area>

A tag <map> é utilizada após a <img> e serve para atribuir áreas pressionáveis utilizando a tag <área>.

Para utilizar estas tag é necessário que na tag <img> tenha o parâmetro ‘usemap=”#000”’

E na tag map um ‘name=”000”’

E para utilizar as tags <áreas> sera necessário fazer o seguinte

<map name="000">  
  <area shape="rect" coords="34,44,270,350" alt="Computer" href="computer.htm">  
  <area shape="rect" coords="290,172,333,250" alt="Phone" href="phone.htm">  
  <area shape="circle" coords="337,300,44" alt="Coffee" href="coffee.htm">  
</map>

## Shape e coords

O parametron ‘shape’ serve para dizer qual o tipo de formato e os valores poderm ser: rect = quadrado, circle = círculo, poly = forma a montar, default = ocupa a imagem inteira.

O parâmetro ‘coords’ serve para as coordenas referentes as image maps.

Para quadrados é necessário colocar os valores Y e X, tendo que colocar os valores de início sendo o Y partindo da distância até o topo e X da distância até a esquerda

Para círculo é necessário colocar os valores Y e X apenas uma vez para dizer o centro do círculo e após isto colocar o raio do círculo.

Para poly terá que colocar quantos valores Y e X até chegar o formato desejado e quando colocar o último valor, ele irá sozinho se ligar ao primeiro ponto.

# Background image

## Background-image

Background-image=[url:”(‘ ’)](url:(‘%20’))”

## Não repetição

Caso a imagem não ocupe a tela toda (não vai), deverá ser usado o código CSS background-repeat: no-repeat;

## Imagem ocupando tudo

Este código fará com que ocupe a tela, mas não de maneira muito certa. para isso será utilizado o código CSS background-size: cover;

Outra forma é background-size: 100%, 100%;

## Imagem sempre cobrindo a tela

Isso fará que talvez a imagem fique estática sempre te perseguindo, o código em CSS é background-attachment: fixed;

Tabelas

# Primeiros passos

## <table> <tr> <th> <td>

Para criar uma tabela devesse começar com a tag <table> e dentro delas terem as tags <tr> que serve para delimitar o tamanho da linha da tabela e dentro dela irá a tag <td> para criar as células e é dentro destas que será colocado os valores da tabela que podem ser: textos, imagens, listas, outras tabelas etc. Ainda existe a tag <th> para cabeçalhos, isto deixa a fonte em negrito.

## Outras tags

Temos o <caption> Que serve para a rubrica das tabelas

<colgroup> especifica um ou mais grupos de colunas em uma formatação de tabelas.

<col> específicas propriedades de cada coluna em um <colgroup>

<thead> grupos de header contidos na tabela

<tbody> grupos de body contidos na tabela

<tfoot> grupo de footer contido na tabela

# Bordas

## Para fazer

Para fazer é como qualquer outra borda é só utilizar o CSS com o código

Border: 1px solid black;

## Bordas colapsadas

Algumas atribuições de bordas em tabelas criam duplas bordas e para evitar isto deve se usar o CSS com

Border-collapse: collapse;

# Tamanho em tabelas

## Tamanho da tabela

Para isso é necessário apenas utilizar formatações de tamanho na tag <table>

<table style=”width: 100%”>

## Tamanho de colunas

Para isso é necessário apenas utilizar formatações de tamanho nas tags <td> ou <th>

<table>

<tr>

<td style=”width:70%”

<td>

</tr>

</table>

## Tamanho de linhas

Para isso é necessário apenas utilizar formatações de tamanho na tag <tr>

<table>

<tr style=”width: 20px”>

<td>

</tr>

</table>

# Cabeçalhos em Tabelas

## Diferenciando do resto da tabela

Pode ser usado o comando text-align:”” para diferenciar o cabeçalho do resto da tabela.

## Cabeçalhos com múltiplas colunas

Para que uma célula do cabeçalho ocupe mais de 1 coluna use este comando:

<th colspan=”2”>

# Padding e Spacing

## Padding

As células de uma tabela já vêm com o padding com 0px de padrão.

## Spacing

As tabelas possuem espaços entre as células, que vem por padrão 2px de espaço, para mudar o espaço é necessário usar o CSS border-spacing

# Alcance das colunas e linhas

## Alcance das colunas

Para editar o alcance das colunas se utiliza o parâmetro colspan=””

## Alcance das linhas

Para editar o alcance das linhas se utiliza o parâmetro rowspan=””

# Estilizando as tabelas

## Estilo Zebra Horizontal

Para fazer o estilo zebra é necessário utilizar uma pseudoclasse :nth-child(even)

Exp:

tr:nth-child(even) {  
  background-color: #D6EEEE;  
}

Você também pode mudar a ordem de células modificando a opções em parêntese

(odd) pinta as células pares

(even) pinta as células ímpares

## Estilo Zebra Vertical

Para fazer este estilo é necessário que ao invés de editar a tag <tr> se estilize as tags <th> e <td> como a mesma pseudoclasse

td:nth-child(even), th:nth-child(even) {  
  background-color: #D6EEEE;  
}

## Bordas personalizadas

Pode ser utilizado o CSS para criar uma borda e indicar para qual direção ela se encontrará

tr {  
  border-bottom: 1px solid #ddd;  
}

## Tabelas interativas

Você poder fazer o uso de pseudoclasses como :hover para estilizar mais sua tabela.

# Grupos de Colunas em Tabelas

## <Colgroup> e <col>

O <colgroup> é utilizado para poder ser colocado a ou as <col> para editação de tabela. Um adendo, a tag <colgroup> deve ser colocada antes de todas as tag de uma tabela, mas deve ser filha de uma <table>

Span é um parâmetro de <col> e é nele que se deve colocar a quantidade de colunas que serão estilizadas.

## Limitação

As tags de <colgroup> não possui uma grande rede de estilos em CSS para estilizar, os que surtem efeitos estão listados abaixo

[width](https://www.w3schools.com/cssref/css3_pr_dim_width.asp) property  
[visibility](https://www.w3schools.com/cssref/css3_pr_dim_width.asp) property  
[background](https://www.w3schools.com/cssref/css3_pr_background.asp) properties  
[border](https://www.w3schools.com/cssref/css3_pr_border.asp) properties

## Escondendo a ou as colunas

Para isso é necessário utilizar o CSS visibility: colapse; para que possa esconder a fonte

## Scope

Scope é um atributo dos elementos th e td e serve para dizer qual é o seu scopo/alcance na tabela. Os valores que podem ser colocador são: colgroup, col, rowgroup e row

<head>

# <meta data>

## <meta>

A tag <meta> tem três parâmetros name, http-equiv e content que podem ter os seguintes valores

<meta charset="UTF-8">

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

<meta name="description" content="Free Web tutorials">

<meta name="author" content="John Doe">

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

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

A http-equiv=”refresh” se referi a atualização da sua página e o contet ao tempo em segundos

O último serve para configurar o tamanho da tela o primerio width=device-width serve para o tamanho da tela dependendo do dispositivo. initial-scale=1.0 esta parte define o zoom quando a página carregar pela primeira vez em seu browser

## <base>

Este elemento serve para especificar as bases de URL e parâmetros target para todas URLs relativas da sua página.

## Meta Social

São meta-dados que são criados por terceiros para utilizar em seu site. Todos metas social tem o atributo property na tag.

# Charset

O conjunto de caracteres ASCII

ASCII usa os valores de 0 a 31 (e 127) para caracteres de controle.

ASCII usa os valores de 32 a 126 para letras, dígitos e símbolos.

ASCII não usa os valores de 128 a 255.

O conjunto de caracteres ANSI (Windows-1252)

ANSI é idêntico ao ASCII para os valores de 0 a 127.

ANSI possui um conjunto proprietário de caracteres para os valores de 128 a 159.

ANSI é idêntico ao UTF-8 para os valores de 160 a 255.

O Conjunto de Caracteres ISO-8859-1

ISO-8859-1 é idêntico ao ASCII para os valores de 0 a 127.

ISO-8859-1 não usa os valores de 128 a 159.

ISO-8859-1 é idêntico ao UTF-8 para os valores de 160 a 255.

O Conjunto de Caracteres UTF-8

UTF-8 é idêntico ao ASCII para os valores de 0 a 127.

UTF-8 não usa os valores de 128 a 159.

UTF-8 é idêntico ao ANSI e 8859-1 para os valores de 160 a 255.

UTF-8 continua a partir do valor 256 com mais de 10.000 caracteres diferentes.

Para uma visão mais detalhada, estude nossa Referência de conjunto de caracteres HTML completa.

Computer Code

# Semântica de códigos

## <code>

Com uma fonte mono espaçada serve para escrever códigos no texto de maneira mais clara ao leitor

## <kbd>

Com uma fonte mono espaçada serve para escrever comandos input

## <samp>

Serve para escrever comando de Output

## <var>

Serve para escrever variáveis

Tags Semânticas

# Figures

## <figure>

São tags autocontidas de conteúdo e serve para colocar fotos, ilustrações, diagramas, lista de código etc.

## <figurecaption>

São tags que podem ser filhas ou não da tag <figure> e serve para denominar a rubrica da imagem.

# Sumários

## <details>

Serve para criar ferramentas de interação e detalhamento da sua página

## <summary>

Serve para criar sumários. É usada dentro de detalhes para criar os sumários

# Tempo

Formulários HTML

# Formulários introdução

## <label>

O elemento label é usado como identificador do input, sendo responsável por nomear o que aquele input fará.

O label possui um parâmetro chamado for este parâmetro deve ter o valor igual ao id do input. Isso não significa que o label pode ser usado fora do aninhamento do elemento form.

<form>  
  <label for="fname">First name:</label><br>  
  <input type="text" id="fname" name="fname"><br>  
  <label for="lname">Last name:</label><br>  
  <input type="text" id="lname" name="lname">  
</form>

Algo a salientar é que o tipo texto tem como tamanho do campo padrão de 20 carácteres.

# Tipos de input

## Radio

O tipo radio serve para inputs de escolha de opções, sendo que o usuário só poderá marcar uma opção.

## Checkbox

O tipo checkbox serve para inputs de escolha de opções, sendo que o usuário poderá marcar nenhuma ou todas as opções.

## Password

Funciona de maneira semelhante ao *text,* mas censura o que é escrito no campo de texto.

## Submit button

Este tipo de input serve para enviar o que foi digitado no seu formulário para o manipulador de formulário. Existe um form handler padrão, mas não é eficiente e recomendo criar o próprio para teste, caso saiba PHP ou javascript.

## Reset

Este tipo de input serve para restaurar as edições feitas pelo usuário no formulário para o padrão da programação.

## Button

Funciona de maneira igual ao elemento button.

## Color

Inputs com esse valor apresenta ao usuário a opções de escolher cores (muito parecido com o do visual studio). Você pode atribuir uma cor inicial colocando um código RGB ou hexadecimal no parâmetro value ou se pode utilizar as tags datalist e option para criar opções neste tipo de input.

O input básico não faz nada, então será necessário utilizar uma linguagem de programação para dar uma função a este tipo de input.

## Date

Este tipo de input é parecido com o valor *text,* mas serve para colocar datas, dando ao usuário um pequeno calendário atual e lhe dá a opção de editar o espaço com a data que quiser.

## Datetime-local

Muito parecido com o valor *date,* mas pode colocar o horário também.

## Email

Parecido com o tipo valor *text,* mas serve para colocar e-mails. Dependendo do tipo de email e do browser o valor pode ser validado quando enviado.

## File

Cria um botão para poder carregar arquivos.

Inputs do tipo file tem um atributo chamado accept que serve para dizer quais tipos de arquivos são aceitos. Neste atributo para selecionar diversas extensões, mas um tipo de arquivo você deve colocar como valor algo assim:

Accept=”image/\*”, accept=”video/\*”

## Hidden

São tipos de inputs que são omitidos ao usuário e não podem ser editados por ele, mas podem ser validados e enviados junto com o resto do formulário.

## Month

São parecidos com o valor *date,* mas serve somente para ser colocados meses e anos.

Pode mudar a forma de exibição dependendo do navegador

## Number

Cria um campo onde se pode colocar valores numéricos. Com os parâmetros min e maxmostra ao usuário quais os valores que ele pode colocar.

## Range

Cria um pequeno campo para colocar valores numéricos (parecido com o slider control). Ele vem por padrão com os valores de 0 a 100, porém estes valores podem ser mudados com os parâmetros min e max.

## Search

Cria um campo muito parecido com o tipo *text,* mas serve para fazer pesquisas

Pode ser usado os elementos datalist com option, pattern e aria-label que serve para acessibilidade

## Tel

Cria um campo que serve para colocar números de telefone.

## Time

Cria um campo parecido com o de valor *date*, mas serve somente para horários e não pode ser um valor 0. Não roda em safari e internet explorer

## Url

Cria um campo de texto para ser colocado endereços de Url.

Dependendo do navegador é validado e enviado automaticamente.

## Week

Cria um campo parecido com o de valor *date*, mas só pode ser colocados valores de semana e ano.

Dependendo do navegador cria-se um campo para este elemento.

# Atributos de formulário HTML

## name

O parâmetro name está presente em todos os campos de input junto do value, mas diferente deste o name é omitido e não é enviado ao form handler

## Action

Este atributo é colocado no elemento <form> e serve para direcionar o envio do formulário a outro arquivo manipulador de formulário, pode ser um arquivo seu feito em PHP por exemplo.

## Target

Este parâmetro funciona igual ao dos links, mas possui outro tipo de valor o *framename* para exibir um iframe nomeado. Também e posto no elemento <form>

## Method

Este parâmetro também é colocado no elemento <form> e serve para dizer qual o tipo de envio dos dados do formulário, se será pela URL com o valor *get e* HTTP com o valor *post*.

## Autocomplete

Serve para completar os dados não especificados no formulário. Pode possuir os valores *on e off*

## Novalidate

É um atributo booleano que quando presente não deixaram o envio dos dados do formulário. Este atributo não possui valor.

## Accept-charset

É um atributo de caractere, é praticamente o mesmo do elemento <meta> utilizado no <head>

## Enctype

É um atributo de codificação que é usado junto do method=”post” e serve para dizer qual o tipo de codificação.

Alguns valores são

application/x-www-form-urlencoded

Predefinição. Todos os caracteres são codificados antes do envio (os espaços são convertidos em símbolos "+" e os caracteres especiais são convertidos em valores ASCII HEX)

multipart/form-data

Este valor é necessário se o usuário for enviar um arquivo através do formulário

text/plain  
Envia dados sem nenhuma codificação. Não recomendado

## Rel

Serve para relações entre documentos atuais e documentos linkados.

## Readonly

Este parâmetro não precisa de atribuição de valores. Ele faz com que o input só possa ser lido, não editado

## Disabled

Os inputs com este parâmetro serão inclicavéis e não serão enviados junto do formulário. Basicamente desativa o input.

## Size

Este parâmetro serve para editar o tamanho da largura e quantidade de carácteres.

Os tipos de inputs que aceitam size são: text, search, tel, url, email, and password.

## Maxlength

Serve para delimitar o size porque o size serve somente de maneira estética, já o maxlength serve para delimitar o número máximo de carácteres

## Min e max

Serva para delimitar o mínimo e o máximo dos valores que o input pode receber ou entregar ao usuário. Segue um exemplo:

<form>  
  <label for="datemax">Enter a date before 1980-01-01:</label>  
  <input type="date" id="datemax" name="datemax" max="1979-12-31"><br><br>  
  
  <label for="datemin">Enter a date after 2000-01-01:</label>  
  <input type="date" id="datemin" name="datemin" min="2000-01-02"><br><br>  
  
  <label for="quantity">Quantity (between 1 and 5):</label>  
  <input type="number" id="quantity" name="quantity" min="1" max="5">  
</form>

Os seguintes tipos aceitam o min e max: number, range, date, datetime-local, month, time and week.

## Multiple

Serve para aumentar a quantidade de valores de entrada. Não é necessário atribuir valores a este parâmetro

Os seguintes tipos aceitam o multiple: email e file.

## Pattern

Serve para padronizar as entradas de dados, sendo os valores entre parênteses os valores limitantes (máximo e mínimo) e o entre chaves a quantidade de valores que aquela parte em específica ira pegar. Segue o exemplo abaixo

<form>  
  <label for="phone">Enter a phone number:</label>  
  <input type="tel" id="phone" name="phone"  
  placeholder="123-45-678"  
  pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}">  
</form>

Os seguintes tipos aceitam pattern: text, date, search, url, tel, email, and password.

## Placeholder

Serve para atribuir uma pequena dica ai input, mas onde serão colocados os dados.

Os tipos que aceitam o placeholder são: text, search, url, tel, email, and password.

## Required

Serve para que o campo de input seja obrigatoriamente preenchido antes que o formulário seja enviado.

Não é necessário atribuir valores a este parâmetro. Os seguintes types de inputs aceitam este atributo: text, search, url, tel, email, password, date pickers, number, checkbox, radio, and file.

## Step

Serve para criar intervalos com o valor deste parâmetro, como por exemplo pular números de 3 em 3 em um tipo number.

Os seguintes tipos aceitam: number, range, date, datetime-local, month, time and week.

## Autofocos

Serve para que quando uma página seja carregada o input com este parâmetro receba o foco.

## Height and width

Serve para delimitar o espaço de um input tipo image

## List

É um parâmetro para o elemento datalist, mas deve ser colocado em uma tag input como se fosse um parâmetro id.

# Tags de um formulário

## <select> e <option>

O elemento select serve para criar listas suspensas e o elemento option para definir os valores da lista.

A ainda os parâmetros selected para marcar um item da lista, size que é usado para aumentar o número de opções mostradas por padrão e multiple para permitir múltiplas escolhas da lista. Os elementos label também podem ser usados no select.

## <textarea>

Textarea é um elemento para criar áreas de texto grande com seu tamanho podendo ser delimitado com os parâmetros cols e rows ou utilizando CSS com width e height.

## <button>

Este elemento serve para criar butões clicáveis. Segue o exemplo abaixo

<button type="button" onclick="alert('Hello World!')">Click Me!</button>

O button tem um atributo chamado form que deve ter o mesmo valor do ID na tag <form>, para assim enviar o formulário de mesmo valor.

## <fieldset> e <legend>

O elemento fieldset serve para criar uma área entorno de um input ou mais na sua página, como se separasse em partes seu formulário. legend é a rubrica de fieldset.

## <datalist> e <option>

O elemento datalist funciona parecido ao elemento select, mas este tem uma integração com tags inputs. Através do atributo id no datalist e list em um input é possível interligar estes elementos, fazendo com que o que está escrito nas tags options seja colocado como uma opção no input com mesmo valor nos atributos list e id.

## <output>

Serve para demonstrar os dados transformados por um script por exemplo. Segue um exemplo (precisará ter conhecimentos de javascript).

<form action="/action\_page.php"  
  oninput="x.value=parseInt(a.value)+parseInt(b.value)">  
  0  
  <input type="range"  id="a" name="a" value="50">  
  100 +  
  <input type="number" id="b" name="b" value="50">  
  =  
  <output name="x" for="a b"></output>  
  <br><br>  
  <input type="submit">  
</form>

## <optgroup>

Este elemento serve para nomear partes da lista do elemento select. Funciona de forma parecida ao elemento label ao ponto que o parâmetro de valor se chama label, mas optgroup serve para nomear os itens da lista e não a lista como o elemento label. Não é necessário atribuição de id para linkar, pois funciona com o modelo de aninhamento.

# Atributos de formulário para input

## Formaction

Funciona de maneira muito parecida com o action, mas para inputs

<input type="submit" formaction="/action\_page2.php" value="Submit as Admin">

## Formenctype

Serve para denominar o tipo de codificação.

Os seguintes tipos aceitam: image e submit

## Formmethod

Funciona da mesma forma do parâmetro method, porém para inputs

Os seguintes tipos aceitam: image e submit

## Formtarget

Funciona da mesma forma do que o parâmetro target, mas para inputs

Os seguintes tipos aceitam: image e submit

## Formnovalidate

Serve para dizer que o input não será validado quando o formulário for enviado. Precisa receber o valor formnovalidate="formnovalidate"

Os seguintes tipos aceitam: submit

## Novalidate

Colocado no elemento form, faz com que os dados não sejam validados quando enviados. Não é necessário atribuição de valores.

Trabalhando com elementos

# Lista de descrição

## <DL>

Abre a lista

## <DT>

Título da lista

## <DD>

Descrição da lista

Box-Model

## Tipos de caixa

**Box-level**: Ela sempre que uma caixa for começar ela vai pular uma linha e ocupará a largura por completo e vai pular para a próxima linha para continuar o conteúdo.

Exp: <div>, <h1>/<h6>, <p>, <main>, <header>, <nav>, <article>, <aside>, <footer> etc

**Inline-level**: Ela vai pular para o lado e vai desenhar a caixa do tamanho para conter o conteúdo e não quebra a linha e dá continuidade na mesma linha.

Exp: <span>. <a>, <code>, <small>, <strong>, <em>, <label> etc.

# Grouping Tags

Header: Cabeçalho

Nav: Menus ou links de navegação

Main: Conteudo principal

Footer: Rodapé

Article: elemento especifica conteúdo independente e autocontido

Section: elemento define a seção em um documento como: capítulos, introduções etc.

Aside: elemento define algum conteúdo além do conteúdo em que é colocado (como uma barra lateral).

Figure: tag específica conteúdo independente, como ilustrações, diagramas, fotos, listagens de código etc.

Figutecaption: a tag define uma legenda para um <figure>

<Iframe>

# Como usar

## Atributos

Height, width, frameborder, src, allow, allowfullscreen.

O atributo allow serve para colocar atributo na parte da página colocada, como autoplay para vídeos trazidos com iframes.

<svg>

# O que são

## O que são

São imagens vetorizadas, ou seja, não trabalham com pixels, utilizam URL’s para utilizar as tags, são imagens pesadas e algumas versões de navegadores antigas não tem suporte a svg.

## Como utilizar

Existem atributos internos do svg para criar formas e desenhos, mas é inviável sua utilização por isto se utiliza programas para criar estas imagens, um exemplo é o figma.