

HTML (Linguagem de Marcação Hipertexto), é uma ferramenta para a criação de estrutura de elementos Web.

O HTML utiliza-se de **tags** (etiqueta), para informar ao navegador o que deve e como deve ser mostrado.

Estrutura Básica

A estrutura básica do HTML é composta de algumas tags, veja abaixo:

<!DOCTYPE html> diz que o tipo do documento é HTML.
<html lang="en"> informa o idioma da página.
<head> abertura da cabeça.
<meta charset="UTF-8"> aceita diversos tipos de caracteres
<meta name="viewport" content="width=device-width, initial-scale=1.0"> informa que vai ocupar toda a tela disponível.
<title> Nome da Página **</title>** Título da Página
</head> fechamento da cabeça.
<body> abertura do corpo.
</body> fechamento do corpo.
</html> informa o fim do código e da página.

Semântica

A semântica é importante no HTML5, pois ela ajuda o seu projeto a ter uma dinâmica e uma leitura melhor por parte do Navegador.

Um projeto em HTML sem tags semânticas, pode ocasionar dores de cabeça, pois não pode funcionar de maneira como você planejou.

As tags semânticas tem um significado por trás delas um sentido, é tudo que não tenha semântica, fica por parte de outras tecnologias como CSS.

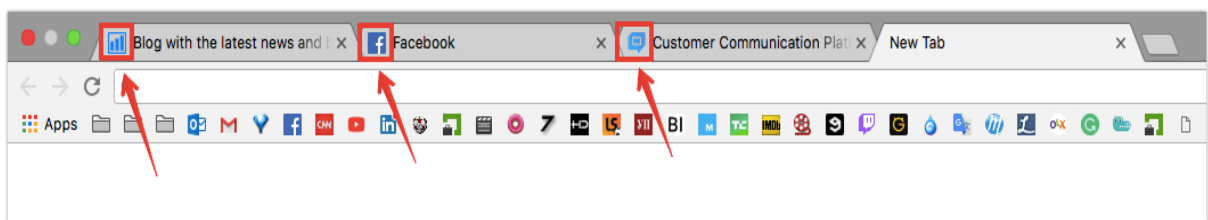
Imagem no HTML

Para colocar uma imagem no HTML, utiliza-se a tag

``, no **src** coloque o endereço da imagem e no **alt** coloque a descrição da imagem, para caso utilizem leitores de tela, para deficientes visuais.

Favicon

Um favicon é um ícone que aparece no título da página.



Para colocar um favicon, coloque `<link rel="shortcut icon" href="favicon.ico" type="image/x-icon">`, dentro da tag `<head>` do HTML, é dentro do **href** coloque o endereço do ícone.

Formatos de Imagem

Os formatos de imagens são importantes pois cada um tem suas características principais.

JPG / JPEG Permite uma compactação maior de pixels, trazendo uma boa qualidade. Também vale lembrar que pode ter seu tamanho alterado sem perda de qualidade.

Porém se for muito esticado, a imagem pode ter sua qualidade diminuída.

PNG Permite uma imagem com transparência, porém em comparação ao JPG o tamanho do arquivo PNG

geralmente e maior. E se por acaso a imagem não precisar de transparência, busque alternativas.

SVG É um formato Vetorial, uma imagem formada por códigos, esse formato não tem sua qualidade alterada ao mudar o tamanho, porém não é recomendada para imagens grandes, pois terá uma qualidade baixa.

WEBp Um formato que tem uma qualidade boa, suporta transparência e tem um arquivo pequeno, porém é um formato recente, e nem todos os navegadores suporta.

Observação: Ao colocar uma imagem, coloque em diversos formatos, pois assim se um não for compatível outras opções serão escolhidas.

Atenção: Ao criar um Site ou Página, atendesse ao tamanho das imagens, pois elas podem prejudicar o desempenho do seu projeto, causando lentidão e sofrendo rejeição das ferramentas de busca.

Vale lembrar também, que tome muito cuidado com os direitos autorais, pois podem causar um processo judicial, para você ou para seu cliente.

Ferramentas para conseguir Imagens

Pixabay e Pexels são Sites que oferecem imagens sem direito autoral.

Ferramentas para editar Imagens

GIMP e Photopea pode ser usado para editar imagens de graça.

Imagem Dinâmica

Torna uma imagem dinâmica e fundamental, pois ela pode ser vista em diversos dispositivos.

Para torna uma imagem dinâmica, e necessário ter outras imagens de tamanho diferentes.

Crie uma tag `<Picture>` e dentro coloque `` com a imagem de tamanho padrão. Em seguida coloque

`<source média="(min-width:)" srcset="" type="image/">`, acima da imagem padrão. No `srcset` coloque o endereço da imagem menor, no `type` o formato da imagem, e no `min-width` coloque o tamanho mínimo que a tela vai ter até trocar pela imagem menor.

Se precisar, coloque essa configuração mais de uma vez para colocar outros tamanhos de imagem.

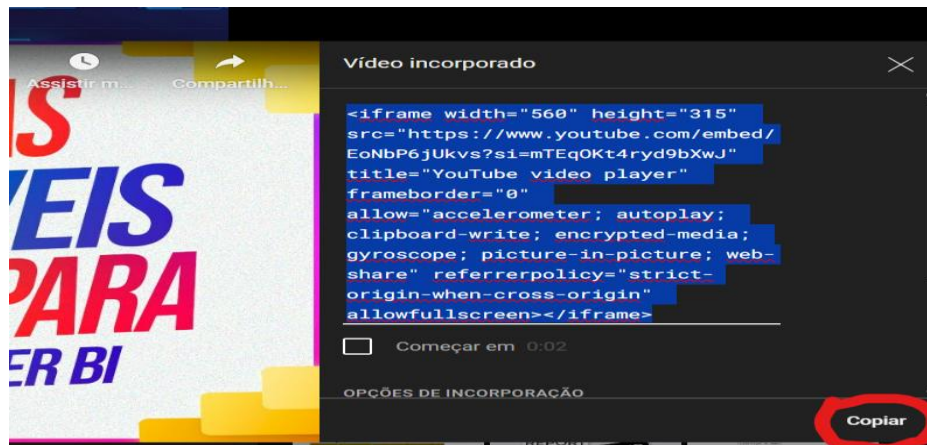
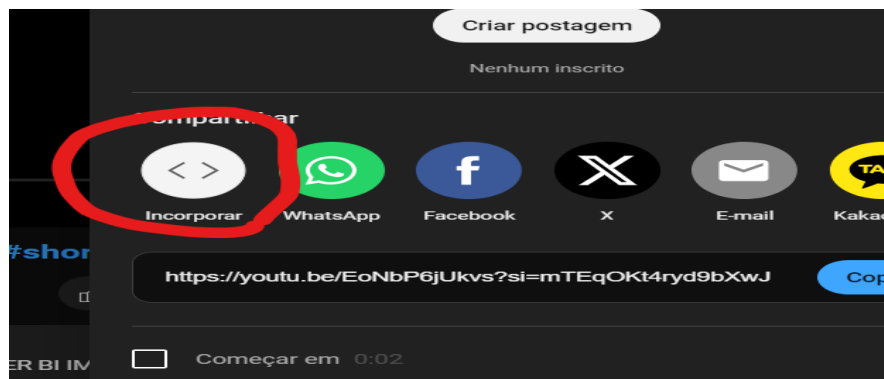
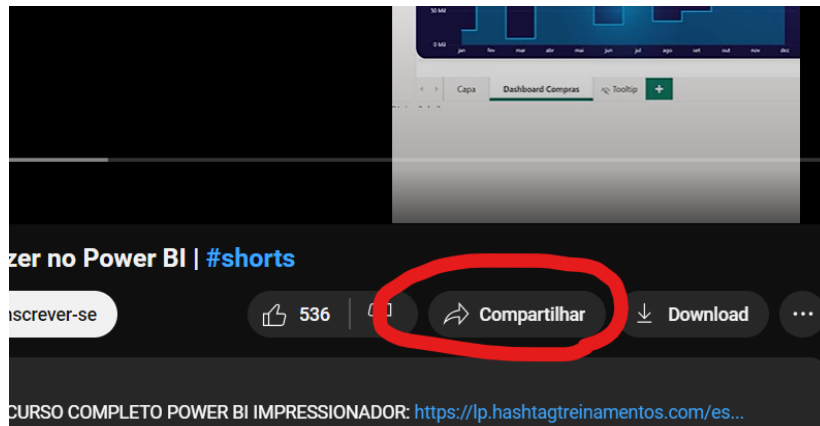
Vídeo no HTML

Vídeo Interno

Para colocar um vídeo interno no HTML, coloque `<video src=""></video>`, no `src` coloque o endereço do vídeo. As vezes e necessário colocar o elemento `controls`.

Vídeo Externo

Para colocar um vídeo externo, por exemplo do Youtube, você pode incorporar ele.



```
<iframe width="560" height="315" src="https://www.youtube.com/embed/EoNbP6jUkvs?si=mTEqOKt4ryd9bXwJ"
title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write;
encrypted-media; gyroscope; picture-in-picture; web-share"
referrerpolicy="strict-origin-when-cross-origin" allowfullscreen></iframe>
```

Formatos de Vídeo

MP4: Um padrão internacional que funciona com vários dispositivos. É mais compactado e menor em tamanho do que o MOV, e é recomendado por muitas plataformas de streaming.

MOV: Geralmente tem qualidade mais alta e é maior em tamanho do que o MP4.

3GP: Compacta os vídeos em arquivos pequenos, o que é ideal para dispositivos móveis. No entanto, a compressão pode fazer com que a qualidade do vídeo fique mais baixa.

OGG: é um arquivo recipiente orientado a stream, que é muito usado na internet para carregamentos de vídeos e rádios ao vivo

Atenção: Cuidado com o tamanho e os direitos autorais.

Observação: Ao colocar um vídeo, coloque em diversos formatos, pois assim o navegador terá outras opções.

Áudio em HTML

Para colocar um áudio, coloque `<áudio src = "">`, no `src` coloque o endereço do áudio.

Formatos de Áudio

MP3: É mais popular e é uma boa opção para quem não se importa tanto com a fidelidade do áudio.

WAV: É mais indicado para quem quer extrair o máximo de cada faixa, porém o arquivo é maior que o mp3.

OGG

Atenção: Cuidado com o tamanho, direitos autorais e colocar em diversos formatos, caso os formatos não forem compatíveis, coloque um parágrafo para Download.

Título e Hierarquia

No HTML, existe uma hierarquia nos títulos, onde o título mais importante começa com h1 e o menor com h6. O h1 sendo o principal, não é necessário ter apenas um, poderá ter mais de um h1 no seu projeto.

Tags para Texto

<p> - Parágrafo (Serve para colocar texto)

**** - Negrito (Serve para dar destaque)

**** - Itálico (Serve para dar ênfase)

<Mark> - Marcação de Texto (Teste de **Texto**)

<Small> - Torna o texto menor (Teste de texto)

**** - Efeito de Texto deletado (~~Teste de Texto~~)

<Ins> - Efeito de Texto Inserido (Teste de Texto)

<Sup> - Texto Sobrescrito (Texto de ^{teste})

<Sub> - Texto Subscrito (Texto de _{Teste})

<Code> - Texto Mono – espaçado (Torna o texto com tamanho igual entre as letras)

<pre> - Mostra o Texto pré-definido (Mostra o texto como ele está escrito no código)

<q> - Citações simples (serve para colocar uma frase que foi dita por outra pessoa)

<blockquote> - Citações em Bloco (serve para colocar citações maiores retiradas de um livro, poema ou ditas por uma pessoa) **obs:** As vezes é acompanhado de outra tag o **cite** que indica para o mecanismo de busca a origem da citação.

Lista no HTML

No HTML as listas servem para organizar elementos, existe dois tipos de listas, as ordenadas **** e as não ordenadas ****.

Diferença:

Lista ordenada

1. Lista 1
2. Lista 2
3. Lista 3
4. Lista 4

Não ordenada

- Lista 1
- Lista 2
- Lista 3
- Lista 4

Coloque **** para colocar os elementos dentro das listas. E para mudar os ícones de marcação da lista coloque **type**.

Lista de Definição

Lista de definições servem para dar informações sobre os termos que estão sendo abordados, como se fossem um dicionário.

`<dl>` - Engloba as tags.

`<dt>` - Termo de que vai receber uma definição.

`<dd>` - Termo de definição.

Atributos Details

O atributo `details` no HTML é utilizado para criar um widget interativo que pode ser expandido ou recolhido pelo usuário. Ele é muito útil para apresentar informações adicionais, como FAQs, descrições detalhadas ou conteúdos que não precisam ser exibidos de imediato.

Atributo Summary

Serve para fazer um resumo, e utilizado nas `table` e no `details`.

Links no HTML

Links no HTML são fundamentais, pois elas podem trazer dinâmica para seu projeto, podendo fazer ligações entre outras páginas e Sites.

Coloque a tag `< a href="">` e dentro do href, coloque o endereço do link, que você quer ligar no seu projeto.

Você também pode usar a tag **Target** com o valor **_blank** para informa que o link vai abrir em uma outra janela.

E a tag **rel** serve para indicar a natureza do link, se ele vai ser externo ou interno.

Caso queira fazer um link interno, basta colocar a pasta e o arquivo que você quer, no **href**, e se precisar voltar uma pasta, coloque dois pontos e uma barra **../**

Download com HTML

Basta fazer o **link** diretamente para o arquivo que se deseja efetuar o download e adicionar o atributo **download** com o valor configurado para o nome do arquivo a ser baixado e o atributo **type** para indicar ao navegador que tipo de arquivo está sendo baixado.

Exemplo:

```
<a href="arquivos/meulivro.pdf" download="meulivro.pdf" type="application/pdf">  
|   Baixe aqui o PDF do meu livro  
</a>
```

Recomendação: **Media Type** *é uma ferramenta que mostra uma lista, que pode te ajudar a conseguir diversos tipos de* **Type**.

Tabela

Para criar uma tabela, crie uma tag **Table** e englobe outras tags.

tabelas pequenas

TABLE ROW **<tr>** – Linha da tabela.

TABLE HEADER **<th>** – Cabeçalho da tabela.

TABLE DATA **<td>** – Dados da tabela (conteúdo de dentro).

Para Tabelas Grandes

Thead – para criar cabeçalhos de tabela

Tbody – para criar o corpo da tabela

Tfoot – para criar o rodapé da tabela

Em tabelas Grandes também pode usar Tags da tabela pequena como **<tr>**, **<th>** e **<td>**

Observação: ao colocar os dados preste atenção, pois se for colocado de forma errada, o dado vai aparecer ou encima ou em baixo.

Legendas na Tabela

Coloque a tag **Caption** para coloca uma legenda na Tabela (sempre ficara em cima da tabela)

Escopo de Tabela

O escopo de tabela serve para informar para o navegador como as informações se encontra, se é em linha **row** em um grupo de linhas **rowgroup**, ou se as informações estão em uma coluna **col** ou em grupo **colgroup**. Para colocar um escopo coloque a tag **scope**.

Alinhamento de Tabela

Text – Align: Alinhamento Horizontal (left, center e right)

Vertical – Align: Alinhamento Vertical (top, Middle e bottom)

Efeito Zebrado na Tabela

O efeito zebrado ajuda na leitura, e para colocar utiliza-se a tag **tbody tr: nth – child (2n)** os valores odd (ímpar) e even (par) também podem ser usados.

Logo após coloque os valores de cor de fundo.

Exemplo:

```
tbody > tr:nth-child(2n){  
    background-color: gray;  
}
```

Cabeçalho fixo

Para tornar o cabeçalho fixo, coloque a propriedade **position: relative** na **table**, crie um elemento no CSS com **tbody > tr >**

th com `position: sticky`, `top: -1px` e `background-color` da mesma cor do cabeçalho.

Mesclagem de Célula

Mesclar uma célula e quando ela ocupa o lugar de duas ou mais células tanto verticalmente quanto horizontalmente.

Para isso use a propriedade `colspan=""` nas aspas coloque o número de colunas que você quer ocupe para mescla colunas (horizontalmente), e para as linhas coloque `rowspan` e o número de linhas para mescla (verticalmente).

Agrupando Colunas

É possível agrupar colunas e colocar estilo de forma individual, basta colocar as colunas com o escopo `col`, e criar um elemento `colgroup` e dentro colocar a quantidade de colunas com as propriedades `col` com uma `class` em cada.

Assim você pode usar o CSS, com um seletor para dar estilo de forma individual em cada coluna.

```

        .col1{
            background-color: gray;
        }
    </style>
</head>
<body>
    <h1>Tornando Uma tabela Responsiva</h1>
    <div id="tabela">
        <table>
            <colgroup>
                <col class="col1">
            </colgroup>
            <thead>
                <caption>Tabela de Teste</caption>
                <th scope="col">Produtos</th>
            </thead>
            <tbody>
                <tr>
                    <td>
                </td>
            </tr>
        </table>
    </div>

```

Outra dica é usar o elemento **span** com um número, que indica quantas colunas você quer dar estilo de uma vez só. Sem precisar criar diversas propriedades **col**.

```

<colgroup>
    <col class="col1">
    <col class="col2" span="2">
</colgroup>

```

Tabelas Responsivas

Para fazer com que uma tabela fique fácil de se ver em tela pequenas e necessário colocar algumas propriedades, como colocar a tabela dentro de uma **div** e um seletor, logo depois no CSS coloque a propriedade **overflow - x** e escolha **auto**.

Exemplo:

```
<div id = "tabela">
```

```
<table> </table>
```

```
</div>
```

O que é Overflow?

O overflow é uma propriedade do CSS utilizada para tratar conteúdos que ultrapassam os limites de um elemento.

iframe

O iframe é uma tag ou etiqueta em um documento HTML que viabiliza a inclusão de outro documento HTML dentro do primeiro, podendo criar uma seção de página.

Tamanho

*Coloque a tag **iframe**, e o endereço no **src**. O tamanho padrão do iframe é 300 de largura por 150 de altura, você pode mudar a configurações de altura e largura, com **height** e **width**. (Pode mudar o tamanho no CSS, ela vai prevalecer sobre as demais).*

***Scrolling** é um parâmetro que você escolher se o conteúdo vai ou não, ter a opção de rolagem de tela. (alguns casos os navegadores não funcionam esse parâmetro).*

Iframe Local

*Para colocar um iframe local e simples basta colocar o nome do arquivo no **src** normalmente, se por acaso o arquivo estiver em uma outra pasta, coloque o nome da pasta, barra o nome do arquivo.*

Abrindo Links no iframe

*E possível fazer o conteúdo dos links abrirem no iframe, basta colocar o parâmetro **name** e colocar um nome, e logo em seguida colocar o **target** dos links, como o nome que colocou no **name**.*

Cuidados com Iframe

Ao colocar um iframe em seu projeto, tenha em mente que é necessário presta atenção em alguns pontos.

Como Mecanismo de Busca, que podem ter dificuldade em indexa (organizar), prejudicando a posição do seu projeto em relação a lista de busca na web.

Problema na Navegação, alguns Navegadores podem ter dificuldade em mostrar e como deve agir em relação ao iframe.

Segurança, pois se o site que for colocado para abrir no iframe, estiver infectado, poderá passar para o seu projeto.

Responsividade, se colocar um projeto de terceiros e ele estiver com problemas de responsividade, pode dar um efeito de quebrado no seu projeto.

Acessibilidade, Ferramentas de leitura de tela, para deficientes visuais, podem ter dificuldade para entender a parte do iframe.

Tornando o Iframe Seguro

Coloque sandbox para criar uma proteção contra roubos de dados, para uma proteção máxima coloque sandbox = sandbox, já para configurações detalhadas pode

ser colocar outras propriedades, como aceitar apenas formulários, scripts e arquivos de mesma origem.

*A propriedade **referrerpolicy** serve para controlar a política de coleta de informações que o site vai colher, se colocar o valor **no - referrer** vai bloquear qualquer tipo de coleta de dados.*

Iframe com código HTML

*E possível colocar uma configuração dentro do iframe, com a propriedade **srcdoc** onde se coloca o código HTML e aparece no iframe.*

Formulário

*Os formulários servem para coletar informações, para colocar um formulário em um arquivo HTML, coloque a tag **<form>** que vai englobar as outras tag, como **input** que serve para coletar dados de cadastro, como por exemplo nomes, senhas, datas etc.*

Label

*Serve para colocar um vínculo (etiqueta) entre os componentes do formulário, ajudando os navegadores a saberem seus significados. Para isso coloque a tag **label** e coloque o nome do id dentro do atributo **for**, para que assim o HTML e os Navegadores saibam quais elementos estão interligados.*

Métodos GET e POST

Existe dois métodos de enviar os dados de um formulário.

Método GET: *Esse método é o padrão, ao enviar dados do formulário, é possível ver os dados obtidos através da URL.*

Os pontos Negativos são a exposição dos dados (principalmente se forem dados sensíveis como senha, documentos e dados pessoais), o limite de caracteres que eles suportam (3 mil caracteres/bites) e não é possível enviar arquivos através desse método.

Método POST: Esse método esconde dados, sem precisar aparecer na URL, é possível enviar arquivos e eles suportam mais de 3 mil bites.

Porém não significa que é o método mais seguro, pois eles não criptografam os dados, para isso é necessário outras tecnologias de HTTPs.

Método required

Esse método permite continuar apenas quando os campos forem preenchidos. Caso contrário pedirá para preencher os campos.

Método placeholder

Método que coloca caracteres e números dentro dos campos a serem preenchidos.

Autocomplete

O autocomplete é um recurso que pode ser colocado nos formulários, para dar sugestões de dados que já foram usados para autocompletar os campos.

O autocomplete pode servir para autocompletar textos e senhas, colocando o autocomplete nos inputs de senha e textos. Coloque o autocomplete e escolha a opção que vale o tipo de dado que você vai mexer.

Tipo Number

O tipo **number** só aceita números e não caracteres. Pode colocar a propriedade **step** que pode escolher como os números vão aumentar.

Os atributos **Min** e **Max** servem para colocar o mínimo de números aceitos e o máximo.

A propriedade **Value** serve para colocar um valor pré-definido no campo de dados. Obs: se a opção **placeholder** estiver ativada, ela ficará desativada automaticamente com o **Value** ativado.

Cuidados com Compatibilidade

Alguns navegadores podem não ser compatíveis com os atributos que você colocou no código, trazendo um resultado não esperado e mostrando versões diferentes do seu projeto.

Para que isso não ocorra, é necessário fazer um teste preliminar.

Input

Serve para coletar dados, usados em campos de preenchimento.

Atributo pattern

Esse atributo serve para especificar como vai ser o formato dos dados apresentado, dentro do input, se os dados apresentados não forem os específicos, não serão aceitos.

Atenção:

Estudar (RegEx) expressões regulares ou Regular Expression.

Fieldset

Serve para agrupar campos de dados semelhantes, como dados pessoais, dados bancários etc. É possível colocar um título nesse grupo com a tag **legend**.

Diferença entre Radio e Checkbox

A diferença do rádio para o Checkbox é que o Radio só pode ter uma escolha enquanto o Checkbox pode ter várias.

O Radio também precisa que coloque o **name** com os mesmos atributos, caso contrário não irá funcionar.

Método File

O método te dá a possibilidade de enviar arquivos de uma máquina para um formulário.

Para esse método é necessário colocar o atributo **POST**

Select

O método select serve para criar uma caixa de seleção, onde você pode colocar as opções de seleção, com a tag `option` e criar diversos grupos de opções, com `optgroup`.

Datalist

O datalist serve para criar uma lista com opções de dados.

Output

O Output serve para fazer saídas de dados.

Colocando Javascript nos formulários

Para colocar interatividade nos formulários do HTML, pode ser usado a tag `oninput`, logo em seguida colocar sinal de igual (=), coloque a identificação de output (saída) com `innerHTML` que indica o endereço está dentro da HTML, coloque outro sinal de igual, e em seguida pode ser colocado o valor `Number` para informar que os dados são números.

Exemplo:

`oninput="isoma.innerHTML=Number(in1.value) + Number(in2.value)"` serve para somar dois números em dois campos de anotação.

Também é possível colocar um parâmetro único e uma única declaração, como a `function` do Javascript, basta colocar o atributo `calc` e o nome que vai dar a ela, exemplo `calcidade` para calcular a idade. Em seguida crie um script no HTML, e coloque uma função `(function)` com o atributo

calclidade. Em seguida crie uma variável com o nome atual e informe que o valor pegar toda a informação do campo de dados. Depois coloque o atributo do **output** com innerHTML e que vai receber valor da variável chamada atual e o atributo do campo de dados.

Exemplo:

```
<script>

    function calclidade(){

        let atual = new Date().getFullYear()

        idad.innerHTML = Number(atual) -
        Number(lano.value)

    }

</script>
```

Média Queries

Média Query são recursos do CSS, que permite adaptar a apresentação de páginas web, para diversos dispositivos e telas.

Os medias queries são feitos de **Media Type** específica para qual o tipo de Média que vai apresentar e **Média Features** que é o tipo de característica da média, como a página vai ser apresentada com uma determinada propriedade.

Chamadas Média Queries

As chamadas de Média Queries são tags que servem para indicar para o navegador, quando e como usar as propriedades, para apresentar de forma adequada e descrita. Para fazer uma chamada basta colocar a propriedade **média =** e colocar os valores.

Medias Type

As **médias type** serve para criar arquivos CSS, que vão criar diferentes versões do arquivo.

Para isso, coloque os valores, como **screen** para configurar a versão de tela, **print** para configurar a versão que vai ser usada para imprimir e **all** para configurar todos.

EXEMPLO: **média = screen** ou **média = print**

Medias Features

Os médias features são as características de telas.

Para colocar o média feature, coloque a tag **media =** e dentro coloque dois parênteses **()**, dentro dos parênteses coloque os atributos que deseja utilizar, como **orientation : Portrait**, para versão em pé ou **orientation : landscape** para versão paisagem.

EXEMPLO: **media = screen and (orientation: Portrait)**

Declarações Diferentes

É possível fazer diversas chamadas de média queries com declarações diferentes, e possível fazer por links css como por exemplo:

```
<link rel="stylesheet" href="estilo/style.css" média="all">
```

```
<link rel="stylesheet" href="estilo/retrato.css"  
media="screen and (orientation:portrait)">
```

```
<link rel="stylesheet" href="estilo/paisagem.css"  
media="screen and (orientation:landscape)">
```

Ou colocando uma declaração em um arquivo do CSS chamado **@media**, sendo possível colocar atributos como screen, print, all e colocando o média feature. É possível colocar essas declarações tanto no CSS interno quanto no externo, tornando possível criar apenas um arquivo CSS para diversas configurações.

Mobile First

O mobile first é um conceito onde ao criar um site, primeiro se cria a versão móvel (para celular) e depois vai se acrescentando detalhes para mostrar para outras mídias. É recomendável seguir esse conceito, pois ferramentas de busca como Google, vai dar prioridade para páginas que atente a esses parâmetros.

Device breakpoints

Serve para indicar os tamanhos de telas que serão usadas para mostrar seus projetos. É usado para indicar as média queries onde e quando mudar os tamanhos, dos layouts.

Média Queries com outros tamanhos

Para mudar o tamanho em outros dispositivos crie outro arquivo CSS e colocar os Device Breakpoints do dispositivo

desejado. Faça uma configuração com JavaScript, para monitorar o tamanho e coloque onresize com um nome, e faça as seguintes configurações.

```
function valor do onresize (){  
    if (window.innerWidth >= tamanho do device  
breakpoints) {  
        menu.style.display = 'block'  
    }else {  
        menu.style.display = 'none'  
    }  
}
```