## HTML

### Elemento

O Elemento principal do Html é a tag.

Exemplo: <p></p> <h1></h1>.

* A estrutura do padrão do Html:



### **Títulos**

São utilizados para ajudar no motor de busca do google, e organizar seu site;

É utilizado a partir do <h1></h1> ou aumentando o número

depois do “h”. “h2..h3..h6”



### **Parágrafos**

Utilizados para inserir textos maiores;  
Ou textos como conteúdo no seu site.

É utilizado a partir do <p></p>.



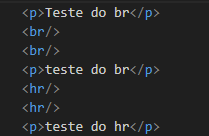
### Tags sem conteúdo

Elas possuem recursos específicos como:

<br/> ← utilziado para quebrar linha ;

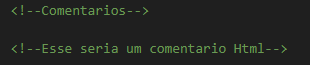
<hr/> ← Utilizado para fazer uma linha na horizontal;

Elas também são oque chamamos de self closing tags, onde a tag não possui uma outra como fechamento.



### Comentários no HTML

Utilizado para fazer anotações no código;



### **Atributos ”ex: Ancor”**

Podem ser utilizados para adicionar funcionalidades a tags.

como exemplo “href=” em ancor que tem o valor de um link

para algum site;

<a href=”https://www.google.com.br/”></a>



1. Âncora para uma nova página:

Para fazer que um link seja aberto em uma nova aba.

é utilizado o atributo target=”blank”;

1. Também temos uma outra função para utilizar o link   
   Podemos utilizar para emails como exemplo:  
   <a href=”<mailto:patrick@gmail.com>”>Envie um email</a>

Podemos utilizar um link para ligar:

<a href=”tel:999999999”>Ligue aqui</a>  
Ou para usar como atalhos dentro de uma pagina:

<a href=”Topo”></a>

<a href=”#Topo”>Ir para topo do site</a>



EXPLICAÇÃO: Atributos são todos que compõem uma TAG.

como o href ou target;

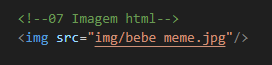
### Imagens no Html

Podemos inserir imagens no nosso site com a tag <img>;  
Caminho da imagem é colocado no atributo src;

Para fazer uma organização padrão podemos colocar dentro

de uma pasta chamada assets ou img;

Obs: a tag <img> é uma self closing tag;



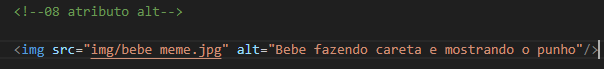
### Atributo alt

Nas tags de imagem temos um atributo chamado alt;

Ele é utilizado para descrever o'que tem na imagem;

É muito útil para alavancar o seu site;

Ex: pessoas que têm deficiência visual o site descreve para elas em forma de áudio.

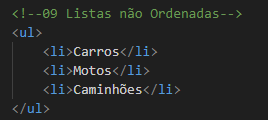


### Listas não ordenadas

Listas são importantes para organizarmos nosso site como

Ex: Criar um menu;   
As não ordenadas são criadas pela tag <ul>;

Cada item na lista é representado pela tag <li>;



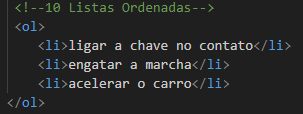
### **Listas ordenadas**

Listas ordenadas são importantes para organizarmos nosso

site como ex: Organizar um passo a passo;

As listas ordenadas são criadas pela tag <ol>;

E os itens continuam com a tag <li>;



### **Tabelas**

Nós usamos as tabelas para exibir dados que podem ser

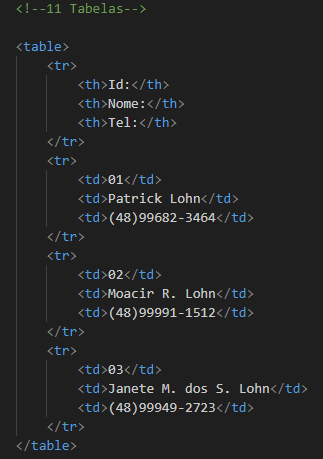
categorizados em colunas;

Exemplo: excel;

Utilizamos a tag <table> criando assim a tabela;

cada linha é utilizado a tag <tr> e os dados ficam em <td>;

os cabeçalho utilizamos a tag <th>;

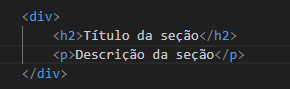


### Tag Div

A tag div é utilizada para criar divisões/seções no nosso site;

Como por exemplo cada tópico do nosso site utilizarmos uma

<div> para facilitar a organização e estilização.



## FORMULÁRIOS

### O que é um formulário ?

Forms tem o papel de receber dados do usuário e enviar para

um servidor;

Podemos validar os dados;

As tags mais utilizadas são: form, label e input;

A tag form cria o formulário e o delimita;

Label descreve os inputs;

E o input é a tag que inserimos dados, temos vários tipos:

number, email, text e outros.

### Criando o primeiro formulário

Para criar um formulário vamos precisar da tag form, que

encapsula todos os elementos do formulário;

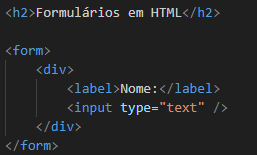
Dentro dela temos labels e inputs, mas podemos ter outras

tags como divs;

A tag de input tem um atributo chamado type, que é onde

definimos o propósito do input;

Um input do tipo text, recebe dados de texto;



### Atributos da tag form

A tag form tem dois atributos geralmente, que são:

action: o arquivo/página que os dados serão enviados:

method: GET ou POST, receber dado ou enviar dado;



### Atributo name

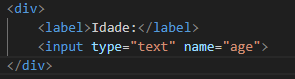
Utilizamos o atributo name para configurar os nossos inputs;

O valor é ligado ao propósito do input, a sua categoria;

Exemplo: Um input que recebe a idade de um usuário, pode

ter um name com o valor de age/idade;

Este atributo é utilizado para pegar o valor quando o form é enviado para o servidor;



### Atributo da label

A tag label tem um atributo ;

E nós o utilizamos para linkar com um input, o nome do atributo é for;

O valor deve ser o mesmo que o atributo name do atributo que corresponde aquela label;

Utilizamos for por propósitos semânticos;

Isso ajuda o nosso site a ser melhor rankeado no Google;



### Enviando dados de formulário

Podemos enviar os dados do form para o servidor através de

um botão de submit;

O botão também, é um input, porém mudamos o type para

submit;

Quando o usuário clicar no botão o processamento do form

acontece;

Os dados serão enviados ao servidor através de uma

requisição HTTP;

Aqui é onde precisamos de uma integração com o back-end,

para aproveitar os dados do formulários;



### Elemento select

A tag select tem as opções representadas por tags de option;

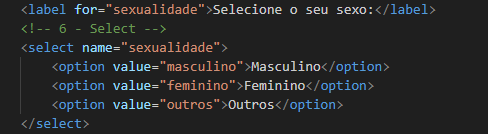
Select também tem um atributo name;

O value estará em cada uma das options, e é isso que

receberemos no lado do servidor;

Então temos duas tags para criar um elemento de seleção:

select e option;



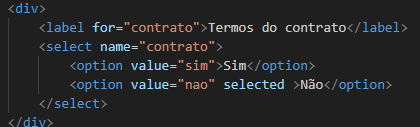
### **Atributo selected**

Podemos iniciar o nosso input de select com uma opção

selecionada;

Para isso está option precisa ter o atributo selected;

Abordagem interessante para quando temos uma opção muito provável;



### Múltiplas seleções

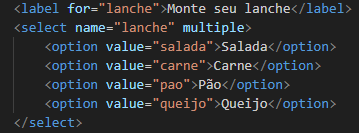
Podemos criar um select que nos permite mais de uma option

selecionada;

Interessante quando queremos aceitar um ou mais dados;

Por exemplo: opcionais de um carro;

Precisamos apenas inserir o atributo multiple na tag select;



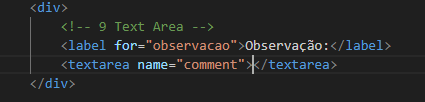
### **Elemento de textarea**

A tag textarea é semelhante ao input text;

Utilizamos para textos maiores;

Por exemplo: Complemento,observação,bio instagram.

Isso permite um input maior para digitar o texto desejado;

  
Obs: Podemos usar a regra do Css (Resize:none) para deixar ele bloqueado com as dimensões bloqueadas.

### **Fieldset e legend**

Fieldset é uma tag para agrupar inputs;

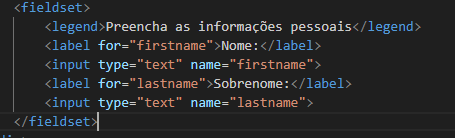
E legend é como uma label, que descreve os inputs

agrupados;

Utilizamos esta tag para conectar dois ou mais inputs que

tenham o mesmo sentido;

Por exemplo; nome e sobrenome;



### **Datalist**

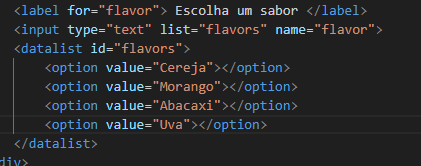
Datalist é como um select, porém com um autocomplete;

Podemos pesquisar por possíveis valores para preencher o

input;

Ou selecionar alguma opção por meio de uma lista;

As opções são linkadas por um atributo chamado list;



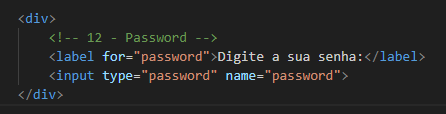
### **Input para senha**

Se nós estamos esperando uma senha do usuário, podemos

utilizar a tag input;

Porém no atributo “Type” colocaremos password;

E então o texto passa a ser exibido com \*, para mascarar os dígitos;



### **Reiniciando o form**

Podemos reiniciar todos os campos do form;

Isso é feito através de um input do tipo reset;

Ele é um botão que ao ser clicado limpa o form;



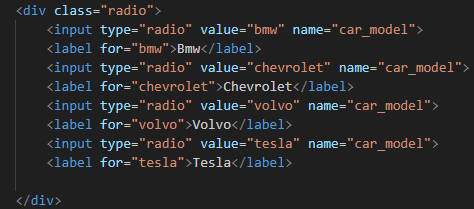
### **Input radio**

Este input é utilizado para selecionar apenas uma opção de várias possibilidades;

Por exemplo: A marca de carro que estamos comprando;

Não podemos escolher dois, então há a necessidade da

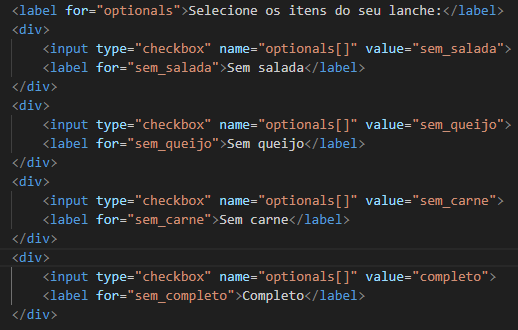
decisão entre uma das opções;



### Input checkbox

O checkbox é similar ao radio;   
Temos que selecionar uma ou mais opções, e também cancelar a seleção de uma opção;

Por exemplo: Os opcionais de uma lanche;



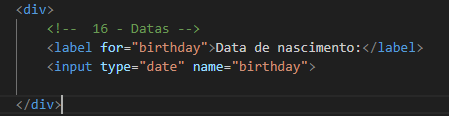
### Input de data

O input do tipo date é utilizado para selecionar uma data;

Temos um calendário que nos auxilia para datas passadas ou

futuras;

Podemos também preencher o valor digitado;

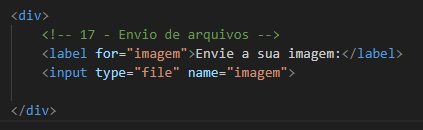


### **Input para arquivos**

O input de arquivos pode ser criado com o type igual a file;

Assim podemos enviar um arquivo ao servidor;

Por exemplo: imagens ou pdfs;



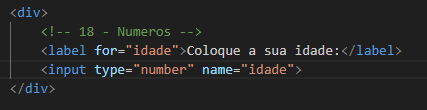
### **Input para números**

Configurando o type para number temos um input que só

aceita dígitos;

Este input possui setas, que nos permite alterar o número

através de cliques;

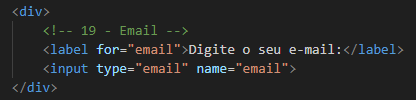


### **Input para e-mails**

O input de e-mail é similar ao de texto;

Porém quando enviamos o formulário temos uma validação;

Que checa se o texto está no padrão de email.

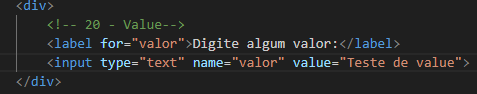


### **Atributo value**

Com o value podemos definir um valor ao input alvo;

Como se o usuário já tivesse preenchido algo;

Muito utilizado quando temos um valor padrão;



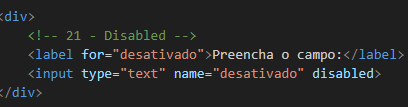
### **Atributo disabled**

O atributo disabled é utilizado para bloquear um input;

Então não podemos mais digitar neste input;

Útil quando não queremos que o usuário preencha um

determinado input;



### **Atributo placeholder**

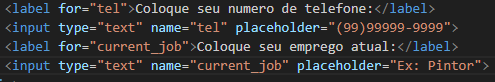
Como atributo placeholder podemos adicionar dicas para os

usuários do sistema;

Ela será exibida no próprio input;

Ao começarmos a preencher com algum valor, a dica some e

o nosso valor que fica sendo exibido;



### 

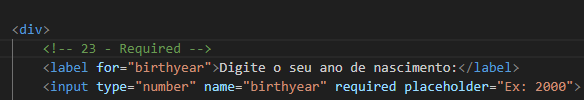
### **Atributo required**

O atributo required força o preenchimento de algum campo;

Se tentarmos enviar o form sem um valor no campo com

required, receberemos um alerta da página;

Isto é um tipo de validação HTML;



## HTML SEMÂNTICO

### O que é Html semântico ?

* Tags de html que dão significado ao elemento;
* Isso ajuda a estruturar melhor o site;
* Contribui a SEO da nossa página;
* Exemplos de elementos não semânticos: div, span;
* Exemplos de elementos semânticos: main, section, article;

### Estrutura da página com semântica:

* Ao adicionar elementos semânticos podemos estruturar nossa página;
* Cada seção do site tem uma tag que faz sentido ao elemento;
* Isso até facilita a manutenção do nosso projeto;

<header>

<nav>

<section>

<aside>

<article>

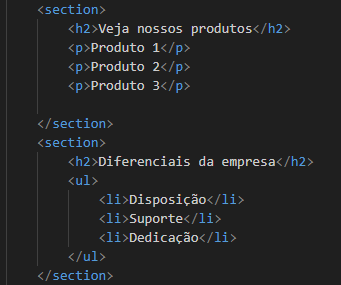
<footer>

### Section:

A tag Section define uma seção no documento;

Geralmente um agrupamento por categorias, exemplo: seção dos produtos, cessão de contrato;

Provavelmente um elemento que será utilizado muitas vezes em um projeto;

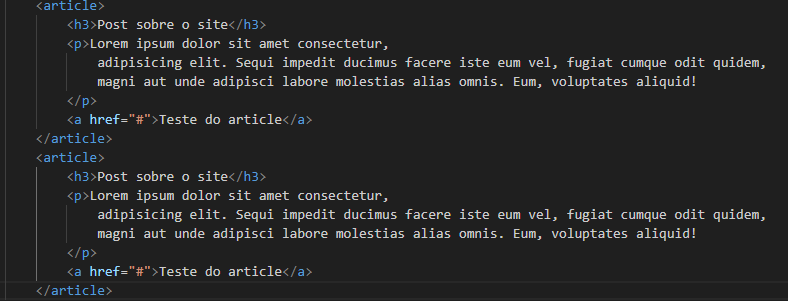


### Article:

A tag article é utilizada para elementos informativos;

Podemos aplicar em : post de blog, comentários, card de produtos;

O conteúdo é individual, não fazendo relação com outro elemento ou article;

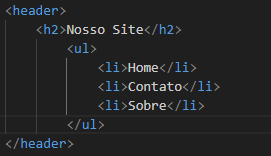


### Header:

A tag de Header é utilizada para um conteúdo de introdução ou links de navegação;

Geralmente tem elementos como: headings,logo, informação do autor do post/página;

Podemos ter mais de um header por página;

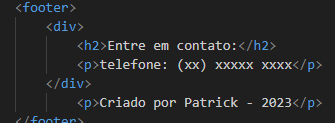


### Footer:

A tag footer é utilizado como rodapé de página, última seção do site;

Mas também pode ser utilizada em outras tags, que precisem deste último elemento;

Geralmente contém: informações sobre a empresa, copyright, dados de contato,sitemap;



### Nav:

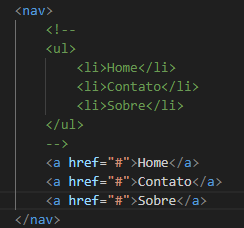
A tag nav é utilizada para demarcar blocos de navegação;

Ou seja, é esperado que haja links dentro desta tag;

E nem todo link precisa estar numa nav;

Costumeiramente são blocos grandes de link,

exemplo: barra de navegação principal;

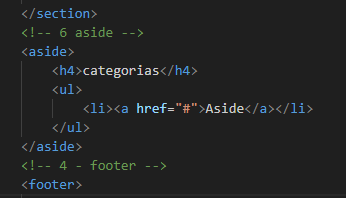


### Aside:

A tag aside é utilizada para criar conteúdos ao lado do contúdo principal;

O que fica dentro de aside precisa estar indiretamente relacionado ao conteúdo principal;

Exemplo; Temos uma lista de produtos sendo exibida, em aside temos os filtros e ordenação;



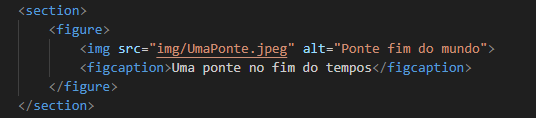
### Figure e figcaption:

As tags figure e figcaption são utilizadas em conjunto;

Figure é o elemento pai, e sua função é exibir alguma imagem na página;

A imagem é inserida pela tag img;

e figcaption serve de legenda para a imagem;

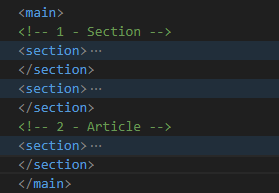


### Main:

A tag main é a que contém o conteúdo principal da página;

Não podemos utilizar mais de uma tag main por página;

o conteúdo de main deve ser único, e não repetido ao longo da página;



### Mark:

A tag mark deve conter um texto dentro;

Ela define um conteúdo que precisa estar em evidência;

Gerando uma importância maior para o mesmo;

