

HTML

- **O que é e pra que serve**

- HTML significa HyperText Markup Language ou em português Linguagem de Marcação de Hipertexto
- É a linguagem utilizada para construir páginas web / websites
- Não é uma linguagem de programação e sim uma linguagem de marcação
- Serve pra definir a estrutura do conteúdo da página web
- Usando HTML é possível definir quais conteúdos serão mostrados e qual a sua disposição na tela
- Sempre que vemos uma página da internet estamos vendo uma página HTML

- **Como funciona o HTML**

- A linguagem HTML segue uma estrutura bem definida de TAGs
- As TAGs são utilizadas pra definir qual o conteúdo que será apresentado e como
- O código da página HTML é salvo em um arquivo com a extensão .html
- O navegador, quando abre este arquivo, irá ler o código, interpretá-lo e então renderizar o conteúdo de acordo com as marcações

- **O que é e como funciona uma TAG**

- Uma TAG HTML é uma marcação que define o tipo de conteúdo que vai ser apresentado na tela
- Cada TAG determina qual é o objetivo do conteúdo sendo apresentado e com isso determina também o seu formato e o seu comportamento

```
<h1>Título de nível 1</h1>
<h2>Título de nível 2</h2>
<h3>Título de nível 3</h3>
<h4>Título de nível 4</h4>
<h5>Título de nível 5</h5>
<h6>Título de nível 6</h6>
<hr />
<a href="http://globo.com" target="_blank">globo.com</a>
<br />
<p>
Um parágrafo com um texto qualquer
</p>
```

O código HTML acima será renderizado pelo navegador assim:

Título de nível 1

Título de nível 2

Título de nível 3

Título de nível 4

Título de nível 5

Título de nível 6

globo.com

Um parágrafo com um texto qualquer

- Toda TAG HTML sempre vai seguir o mesmo padrão

```
<nome-da-tag atributo="valor">conteúdo</nome-da-tag>
```

- Características de uma TAG
 - A TAG sempre vai ter um nome
 - Uma TAG tem uma outra TAG de fechamento correspondente
 - Porém, há algumas TAGs que não possuem um correspondente de fechamento ou tem "fechamento simples", por exemplo

 - Tudo que está entre a tag de abertura e a tag de fechamento é o conteúdo
 - Uma tag pode ter atributos que definem ou especificam o seu comportamento
 - Como exemplo a tag de ancora e o atributo target
 - Atributos tem um formato específico: nome="valor" e sempre são colocados na tag de abertura
- Essa estrutura de TAG com abertura, fechamento, atributos e conteúdo vamos chamar de elemento HTML
- **Hierarquia dos elementos**
 - Além de apenas textos o conteúdo de um elemento HTML pode ser composto por outros elementos
 - Dessa forma temos um elemento pai cujo conteúdo são outros elementos filhos, que por sua vez podem ou não ter também outros elementos filhos
 - Dessa forma criamos uma hierarquia de elementos da página

```
<div>
  <h3>Primeiro poste do blog</h3>
  <hr />
  <p>
    <h5>
      Texto
      <span>
        outro texto
      </span>
    </h5>
  </p>
</div>
```

```
</p>
<p>Primeiro parágrafo</p>
</div>
```

- **Estrutura básica de uma página**

- Todo documento HTML deve seguir esta estrutura básica

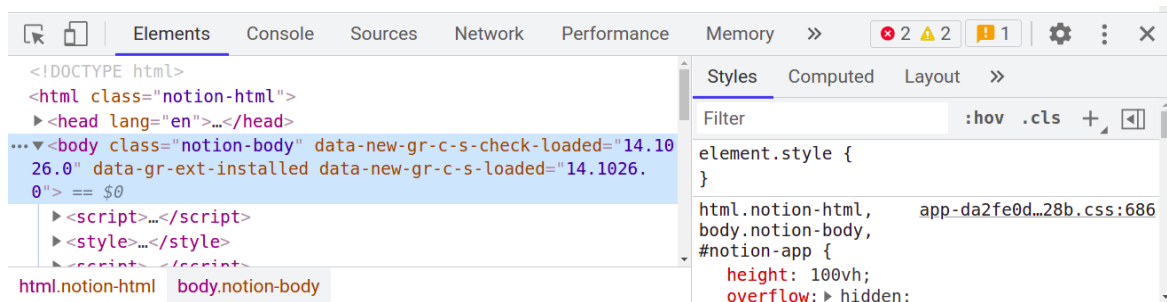
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Título da página</title>
</head>
<body>
  <h1>Conteúdo da página</h1>
</body>
</html>
```

- A primeira linha, <!DOCTYPE html> informa ao navegador que este documento se trata de um documento HTML
- O elemento <html></html> é o primeiro nível no documento HTML e todo conteúdo da página deve estar dentro dessa tag. O atributo lang informa ao navegador qual é a linguagem do conteúdo da página.
- O elemento <head></head> tem o objetivo de definir metadados da página html. O conteúdo deste elemento não é visto pelo usuário. Exemplo:
 - Metadados são informações a respeito da página, como por exemplo, o autor da página, descrição do conteúdo da página, etc.
 - Por exemplo a tag meta charset="UTF-8" define qual é charset (conjunto de caracteres) em que o conteúdo foi escrito e deve ser renderizado
 - A meta viewport informa ao navegador como esta página deve se comportar em outros dispositivos. Esta meta tag é necessário para que a página seja responsiva.
 - A tag title determina o título da página
 - Outras meta tags:

```
<!-- Descreve em poucas palavras o conteúdo da página-->
<meta name="description" content="Descrição do conteúdo">
<!-- Palavras chave a respeito do conteúdo da página -->
<meta name="keywords" content="HTML, CSS, JavaScript">
<!-- Pessoa ou empresa autor da página -->
<meta name="author" content="Fulano de Tal">
```

- A tag body determina o corpo da página html e tudo que estiver dentro deste elemento será apresentado para o usuário.
- Toda página HTML deve, obrigatoriamente, seguir esta estrutura básica pra ser considerada uma página HTML válida.
- **DevTools do navegador**
 - Uma ferramenta que todo navegador disponibiliza
 - Utilizado por programadores para inspecionar o conteúdo da página e o seu comportamento
 - Funcionalidades:
 - Inspecionar o HTML renderizado na página

- Avalizar e alterar o CSS aplicado a cada elemento
- Visualizar o código fonte da página, tanto HTML e CSS quanto JavaScript
- Exemplo do DevTools do Navegador Google Chrome:



• Elementos de texto

- Elementos de títulos, de h1 a h6
- Elemento de parágrafo, representado pela tag `p`
- `br` e `hr`. Não são exatamente de textos mas tem utilidade junto os elementos de texto

• Elementos de formatação

- Elementos de formatação, como o nome sugere, tem o objetivo de dar algum formato específico ao texto ou a alguma parte da página.
- `` - Texto negrito
- `` - Texto importante, na prática tem o mesmo efeito de negrito mas com mais semântica.
- `<i>` - Texto itálico
- `` - Texto destacado, na prática tem o mesmo efeito do itálico mas com mais semântica.
- `<mark>` - Texto marcado em amarelo
- `<small>` - Texto com tamanho reduzido
- `` - Texto removido, taxado, com um rico
- `<ins>` - Texto inserido, sublinhado
- `<sub>` - Texto com tamanho reduzido e alinhado abaixo
- `<sup>` - Texto com tamanho reduzido e alinhado acima

• Comentários

- Comentários tem como objetivo documentar alguma parte do código
- Tudo que estiver comentado não vai ser interpretado pelo navegador
- Tudo que estiver comentado não vai ser apresentado na tela para o usuário
- Tags de comentário: `<!--` e `-->`
- Tudo que estiver dentro destas tags é considerado um comentário

• Exercício 1

- Construir uma página HTML cujo conteúdo seja um post de blog
 - No cabeçalho do documento HTML (head)

- Deve conter um título
- Deve conter algumas meta tags a respeito do artigo do post, como autor, descrição e keywords
- No corpo do documento HTML (body)
 - Deve conter o título do artigo apresentado com uma tag h1
 - Deve conter o nome do autor e a data de publicação
 - O post deve ter pelo menos 4 seções
 - Cada nova seção deve ser identificada com um título usando a tag h3
 - Cada seção deve ter pelo menos 2 parágrafos de texto
 - Deve ser usado as tags html de formação de texto em algumas partes do texto
 - Adicione alguns comentários no documento HTML
 - O texto não precisa ser original e pode ser usado "lorem ipsum"
- **Elementos de lista**
 - Em HTML há três tipos de listas que podemos contruir
 - Lista ordenada, lista não ordenada e lista de definição
 - A lista ordenada, como o nome sugere, é uma lista que possui uma ordem e é apresentada com uma sequência numérica
 - A lista ordenada é representada pela tag **ol**, *ordered list*
 - Cada item da lista é representado pela tag **li**, *list item*

```
<h3>Lista ordenada</h3>
<hr />
<ol>
  <li>Primeiro item da lista</li>
  <li>Segundo item da lista</li>
  <li>Terceiro item da lista</li>
  <li>Quarto item da lista</li>
  <li>Último item da lista</li>
</ol>
```

- A lista não ordenada não possui ordem e a apresentação é com "bullets" ou pontos
 - A lista não ordenada é representada pela tag **ul**, *unordered list*
 - Cada item da lista é representado pela tag **li**, *list item*

```
<h3>Lista não ordenada</h3>
<hr />
<ul>
  <li>Item número um</li>
  <li>Item número dois</li>
  <li>Item número três</li>
  <li>Item número quatro</li>
  <li>Último item da lista</li>
</ul>
```

- A lista de definição é caracterizada por um item que contém uma descrição para cada item, ou seja, uma par de chave e valor para item e descrição
 - A lista de definição é representada pela tag **dl**, *definition list*
 - Cada item da lista é representada por um par de outros itens, **dt** e **dd**, *definition term* e *definition description* respectivamente

```

<h3>Lista de definição</h3>
<hr />
<dt>
  <dt>globo.com</dt>
  <dd>Site de notícias do brasil e do mundo</dd>

  <dt>github.com</dt>
  <dd>Maior repositório de código aberto do mundo</dd>

  <dt>midium.com</dt>
  <dd>Plataforma para posts técnicos</dd>
</dt>

```

• Elementos de bloco VS. Elementos de linha

- Cada elemento HTML que colocamos na página vai ocupar um determinado espaço
- O espaço que será ocupado pelo elemento é definido pelo tipo do elemento
- Se for um elemento de bloco, então este elemento irá ocupar toda a linha da página
- Já um elemento de linha irá ocupar apenas o espaço necessário para comportar o seu conteúdo
- Exemplo de elementos de bloco:
 - H1 ... H6
 - p
 - div
- Exemplo de elementos de linha:
 - a
 - elementos de formatação
 - span

• Elementos semânticos

- Semântica tem a ver com o significado de uma palavra, ou seja, ao ler você já consegue identificar o que significa
- Algumas tags html não possuem nenhuma semântica, ou seja, não é possível entender o significado apenas ao ler o nome da tag.
- Por exemplo a tag **div**, não diz nada sobre o seu objetivo na página e tão pouco deixa claro que tipo de conteúdo que devemos esperar dentro de um elemento **div**.
- Pra fazer com que o documento HTML faça mais sentido pra quem está lendo (programador) e também pra fazer um vínculo entre a estrutura do documento e o conteúdo que está sendo apresentado existem as tag ou elementos semânticos em HTML
- O objetivo das tags semânticas é descrever o significado do conteúdo presente no documento HTML tornando o conteúdo mais claro tanto para programadores quanto para os navegadores que vão exibir essa informação e também para outros "motores" que vão ler e processar essa informação
- Tags semânticas e seu objetivo:
 - Header: Define um cabeçalho para a página ou para uma seção
 - Section: Define uma seção da página
 - Article: Define um artigo ou conteúdo de texto extenso
 - Nav: Define uma seção de navegação como, por exemplo, um menu

- **Aside:** Utilizado para definir uma barra lateral para a página
 - **Main:** Utilizado para representar o conteúdo principal da página
 - **Figure:** Utilizado para apresentar uma figura que pode ser uma imagem, um gráfico, um trecho de código, etc. Uma figura geralmente é acompanhada por uma legenda que é representada pela tag `figcaption`
 - **Footer:** Define o rodapé da página ou de uma seção
 - **a, ancora:** Define um link para outra página ou para outra seção
 - **em:** Utilizado pra enfatizar algum texto. Carrega mais semântica que a tag `i`
 - **strong:** Utilizado para marcar algum texto como importante. Carrega mais semântica que a tag `b`
 - **Time:** Utilizado para apresentar uma hora para o usuário
- **Exercício 2**
 - Melhorar a página de blog que fizemos no exercício 1:
 - Adicionar um cabeçalho a página (usando a tag `header`) com o título do seu blog
 - Adicionar ao cabeçalho um menu para o seu blog, com os itens (usando a tag `nav`):
 - Início
 - Postagens
 - Sobre mim
 - Contato
 - Utilizar as tags semânticas ao longo do conteúdo do post do blog (usando as tags `main`, `section`, `article` e outras que forem necessárias)
 - Logo após o conteúdo do post do blog, crie uma lista "lateral" com "Outros posts" (usando a tag `aside`. Ainda não ficar "do lado" e sim em baixo) com links para outros posts do blog.
 - Adicionar um rodapé ao final da página (usando a tag `footer`) contendo informações de um endereço fictício da empresa do blog e um email, também fictício, para contato.
 - Extra:
 - Crie as outras páginas do site (Sobre mim e Contato) e adicione links a estas páginas no menu da página inicial.
 - As outras páginas terão o conteúdo diferente, porém o cabeçalho e o rodapé devem ser iguais entre as páginas.
 - Crie pelo menos mais uma página com um outro post de blog e coloque um link para acessar esta página na lista de "outros posts".
 - **Elementos de mídia (audio, video e imagem)**
 - Imagem
 - Representada pela tag `img` e é utilizada para apresentar uma imagem para o usuário
 - Os principais atributos da tag imagem são:
 - `src`: define onde está a imagem que deve ser mostrada
 - `alt`: define um texto alternativo a ser apresentado quando não for possível mostrar a imagem
 - `height`: define a altura da imagem
 - `width`: define a largura da imagem

```

```

- Uma imagem pode também ser utilizada como link

```
<a href="http://github.com" target="_blank">  
    
</a>
```

- E pode também ser apresentada como um elemento `figure`

```
<figure>  
    
  <figcaption>  
    Imagem de um abacate cortado ao meio  
  </figcaption>  
</figure>
```

- Video

- A tag `video` é utilizada para apresentar videos para o usuário
- Os principais atributos da tag video são
 - `height`: define a altura do video
 - `width`: define a largura do video
 - `controls`: define se os controles (play, pause, barra, volume e etc) estarão visíveis para o usuário
 - `autoplay`: determina se o video será iniciado automaticamente ao carregar a página
 - `muted`: determina se o video estará mutado ou não por padrão

```
<video height="200px" width="200" src="../assets/theoffice.mp4" controls muted></video>
```

- Para definir qual video será apresentado é utilizado a tag `source` dentro da tag `video`
 - A tag `source` deve ter os seguintes atributos:
 - `src`: determina o caminho onde está o video a ser exibido
 - `type`: informa ao navegador qual é o formato do video
 - Dentro da tag video pode haver várias tags `source` com vários formatos de video
 - O navegador vai apresentar o primeiro video cujo formato é conhecido pelo mesmo

- Audio

- A tag `audio` é muito parecida com a tag video
- A diferença é que, obviamente, é utilizada pra apresentar um audio em vez de video
- Não é necessário definir altura e largura. O player tem um formato e tamanho padrão
- O audio a ser tocado também é definido pelas tags `source`, com os mesmos atributos
- A tag `audio` também tem os mesmos atributos da tag video (exceto `height` e `width`)

```
<audio src="../assets/theoffice.m4a" controls></audio>
```


- **iframe**
 - É possível também "embarcar" outra página HTML dentro de uma página
 - Fazemos isso com a tag `iframe`
 - Os seus principais atributos são:
 - `src`: qual página HTML será renderizada
 - `title`: título desta página
 - `height`: altura da página
 - `width`: largura da página
 - Exemplo: colocar um video do youtube na página

```
<iframe width="560" height="315" src="https://www.youtube.com/embed/0Kvw2BPKjz0" frameborder="0"></iframe>
```

- **Formulários**

- Um formulário é utilizado para que o usuário possa informar alguns dados através da página web para que sejam processados pelo sistema
- Elementos de um formulário
 - `input`: Campo de entrada de dados em diversos formatos ou tipos. É o elemento mais utilizado em um formulário e a forma que é apresentado muda de acordo com o valor do atributo `type`.
 - `label`: Representa a descrição de um campo. O atributo `for` do elemento label deve ter o mesmo valor do atributo `id` do campo de input que o label está representado para vinculá-los.
 - `select`: Utilizado para apresentar um campo "dropdown" com várias opções de seleção. As opções do campo select são definidas através de elementos `option` dentro do elemento `select`. O primeiro elemento `option` é selecionado por padrão
 - `textarea`: Campo de entrada de texto com múltiplas linhas.
 - `button`: Botão de ação em um formulário
 - `fieldset`: Elemento utilizado para agrupar campos relacionados entre si
 - `legend`: Utilizado para dar nome a seção criada pelo elemento fieldset
 - `datalist`: Especifica uma lista de opções pré-definidas para um campo de input
- Tipos de inputs
 - `button`
 - `checkbox`
 - `color`
 - `date`
 - `datetime-local`
 - `email`
 - `file`
 - `hidden`
 - `image`
 - `month`

- `number`
- `password`
- `radio`
- `range`
- `reset`
- `search`
- `submit`
- `tel`
- `text`
- `time`
- `url`
- `week`
- **Exercício 3**
 - Criar uma página HTML que contenha todos os elementos input com os tipos listados acima.
 - O objetivo é para que você veja cada um dos elementos de input, como é apresentado na tela e como se comporta ao interagir com este campo de input.
 - Depois reflita sobre os campos de inputs que você viu e pense qual destes campos de input você mais gostou ou mais te surpreendeu.
- Elemento button
 - Apresenta um botão que ao ser clicado irá executar alguma ação
 - Deve ser definido o atributo `type` para o botão, sendo que o tipo padrão é `submit` (submeter ou enviar o formulário)
 - Os tipos para botões são:
 - `Submit`: Ação que é executada quando o formulário foi preenchido e agora precisa ser enviado ou submetido.
 - `Reset`: Esta ação quando executada faz com que os valores dos campos do formulário voltem para o valor inicial. O valor inicial de um campo é definido pelo atributo `value`
 - `Button`: Este tipo de botão representa uma ação que não é padrão e que será executada quando o usuário clicar no botão
- Elemento form
 - O elemento `form` é a tag semântica para agrupar elementos de `input` de dados
 - Principais atributos do elemento form
 - `action`: Define qual a ação que será executada quando o usuário submeter o formulário. Geralmente quando o usuário clicar para submeter os dados do formulário serão enviados para o servidor tratá-los e responder de acordo.
 - `target`: Define onde será exibido a resposta do servidor após processar os dados. Tem as mesmas opções do atributo `target` do elemento `a` (âncora) e o valor padrão é `_self`, ou seja, o resultado será exibido na mesma página
 - `method`: Define qual método http será usado ao enviar os dados para o servidor. O método padrão é GET

- **autocomplete** : Determina se o formulário pode ou não ser preenchido automaticamente com os dados salvos pelo navegador. Os valores para esse atributo são "on" e "off": ligado e desligado
- **novalidate** : Este atributo é booleano, ou seja, não precisa informar um valor. Se o atributo está presente então esta "opção" está ligada. Quando presente informa ao navegador para não validar os dados do formulário quando for submetido
- **name** : Especifica o nome para o formulário. Esta informação é bem útil quando há mais de um formulário na página e também quando precisamos acessar o formulário através de código JavaScript (Veremos mais a frente)
- Principais atributos dos elementos de **input**
 - **value** : Especifica um valor inicial para o campo de **input**
 - **readonly** : informa que o campo é apenas leitura e o usuário não consegue alterar o valor. Porém o usuário consegue "navegar" pelo campo, selecionar o valor e copiar o valor
 - **disabled** : Informa que o campo está desabilitado e o usuário não consegue interagir com o campo. O valor de um campo desabilitado não será enviado para o servidor quando o formulário for submetido.
 - **maxlength** : Determina a quantidade máxima de caracteres que o usuário pode inserir no campo de input. Quando o limite for atingido o usuário simplesmente não consegue colocar mais texto no campo
 - **min** e **max** : Determinam o valor mínimo e máximo para um campo de **input**.
 - Estes atributos funcionam para os seguintes tipos de input: number, range, date, datetime-local, month, time e week
 - Se usar os atributos min e max em conjunto é possível criar um range de valores válidos
 - **multiple** : Este atributo funciona para os tipos de input **file** e email e determinam se o campo pode ou não aceitar mais de um valor ou arquivo no caso de input file.
 - **placeholder** : Apresenta para o usuário uma descrição breve do objetivo do campo ou da informação esperada.
 - Este atributo funciona para o campos de input do tipo: text, search, url, tel, email e password
 - **required** : Determina se o campo é de preenchimento obrigatório ou não
 - **autofocus** : Determina que o campo deve receber foco automaticamente quando a página é carregada
 - **autocomplete** : Especifica se o campo pode ou não ser preenchido automaticamente pelo navegador. Os valores possíveis para esse atributo são "on" e "off", ligado ou desligado.
 - Um exemplo de uso desse atributo "desligado" é para campos de "repetir a senha" ou "repetir o email"
- **Exercício 4**
 - Imagine que você está implementando um sistema para uma loja que vende de tudo e você precisa criar uma página de cadastro de clientes.
 - Crie uma página de formulário para cadastrar um novo cliente. Que deve atender aos seguintes requisitos:
 - Deve ter um título
 - Deve ter uma seção para cadastrar os dados pessoais do cliente (fieldset)
 - identificador (hidden)

- Nome (required)
- Sobrenome
- Data de nascimento (date, required)
- Idade (numero, min 18 anos)
- Sexo (radio, required)
- Adicionar uma foto de perfil (file)
- Deve ter uma seção para cadastrar as informações o endereço do cliente (fieldset)
 - Endereço (required)
 - Número da casa (numero)
 - Complemento
 - Cidade (select, required)
 - Estado (campo com um valor inicial e readonly)
- Deve ter uma seção para cadastrar as informações de contato com o cliente (fieldset)
 - telefone (tel, required)
 - email (email)
 - site pessoal (url)
 - Escolher as formar de receber informações de promoção (multiselect)
- Deve ter uma seção para cadastrar as informações de login do usuário
 - nome do usuário
 - senha
 - repetir a senha
 - Checkbox para aceitar os termos de uso (required)
- Deve ter uma seção para que o usuário informe algumas opções de preferencias:
 - Cor favorita
 - Quais tipos de itens mais gosta de comprar (varias opções com checkbox)
 - Qual mes do ano que mais gosta de comprar
 - Algumas outras opções
- O formulário precisa ter as ações de "Salvar" (submit) e "Limpar" (reset)
- Quando acessar a página o cursor deve ser posicionado no campo "Nome" (autofocus)
- Quando o usuário clicar em "Salvar" deve ser redirecionado para uma outra página de "Sucesso".
- **Tabela**
 - Tabelas são utilizadas para apresentar dados e informações de forma organizada em linhas e colunas
 - Em HTML utilizamos o elemento `table` para construir tabelas
 - Dentro do elemento `table` para definir uma linha da tabela é utilizado a tag `tr` (table row)
 - Dentro de uma linha, cada coluna ou célula é definida utilizando a tag `td` (table data)
 - Se as colunas ou células forem de uma linha que representa um cabeçalho, então é utilizado a tag `th` (table head)

- Por padrão o texto dentro de um elemento `th` é apresentado em negrito e centralizado
- Por padrão o texto dentro de um elemento `td` é apresentado alinhado a esquerda
- Um elemento `td`, ou célula da coluna, pode conter qualquer outro elemento html como textos, imagens, listas, outras tabelas e etc.
- Atributos para expandir linhas e colunas:
 - `colspan`: Faz com que a coluna da tabela ocupe o espaço de x colunas, por exemplo `colspan="2"`
 - `rowspan`: Faz com que a linha se estenda por x linhas, por exemplo `rowspan="2"`
- **Exercício 5**