Fundamentos de HTML

Fundamentos da Web

Objetivos

1. Como o Browser lê arquivos HTML
2. Sintaxe de um elemento HTML; abrindo e fechando tags, elementos, atributos, conteúdos
3. Aprendendo estrutura de um documento HTML,
   - Elementos <html> <head> <title> & <body>
4. Aprendendo metadados de documentos:
   - Elementos <link>, <meta>, <style>

1. Como o Browser lê **Páginas Web?**

https://developer.mozilla.org/en-US/docs/Web/Performance/How_browsers_work#parsing

Lendo arquivos HTML

PARSING

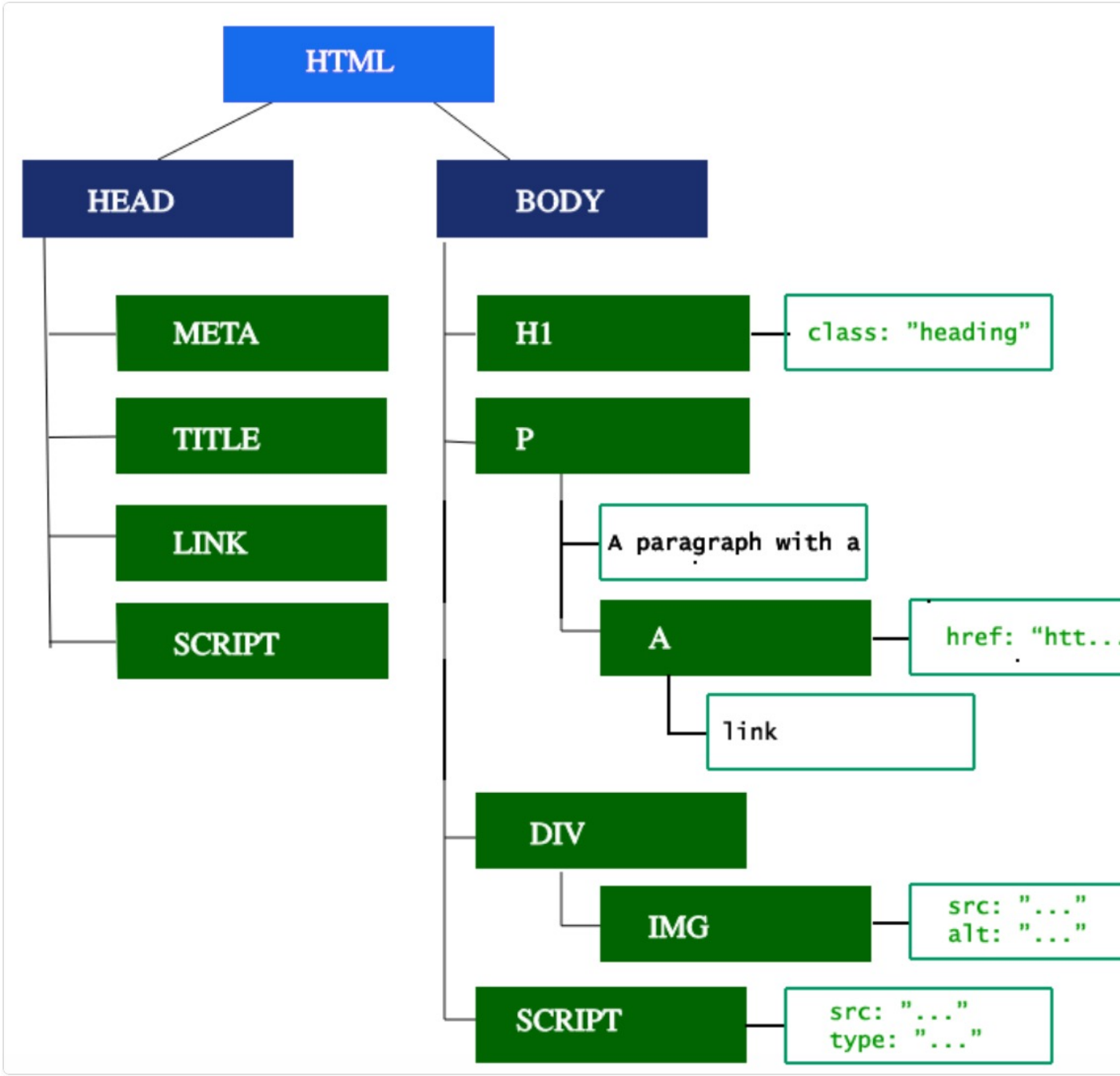
- Processo disparado quando dados chegam ao navegador
  - O Browser analisa e transforma o arquivo de texto no DOM e CSSOM para renderizar a página

DOM e CSSOOM PARSING

- DOM – Document Object Model
  - árvore de conteúdos da página
- CSSOM – CSS Object Model
  - representação (processável, manipulável) do CSS.
  - Permite que a estilização atribuída ao documento possa ser lida e manipulada via script

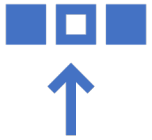
Construindo a árvore DOM

1. Processamento do HTML e construção da árvore DOM
   - Tokenização e construção da árvore
     - Tokenização: identificação das tags e atributos e processamento
   - A árvore DOM descreve o conteúdo do documento



Observações

- A Árvore representa as relações e hierarquias entre diferentes tags
  - Tags aninhadas são nós filhos
- Quanto maior o número de nós, mais demorado para carregar a página... CUIDADO
- Quando o parser encontra um recurso não-bloqueante, (img, p.ex.), o browser requisita o recurso e segue parseando
- Scripts sem atributo **`async`** ou **`defer`** bloqueiam a renderização e pausam o parsing
- Muitos scripts podem retardar o carregamento de páginas



Scanner de Pré-carregamento (Preload scanner)

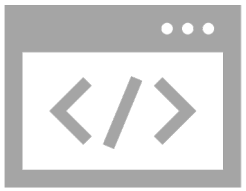
- Enquanto o navegador constrói a árvore DOM, este processo ocupa a thread principal.
- Analisa o conteúdo disponível e solicitará recursos de alta prioridade, como CSS, JavaScript e fontes da web.
- Busca um recurso antes mesmo do Parser requisitar
- Recupera recursos em segundo plano para otimizar a carga da página
  - Reduzem os bloqueios.

2. Construindo o CSSOM

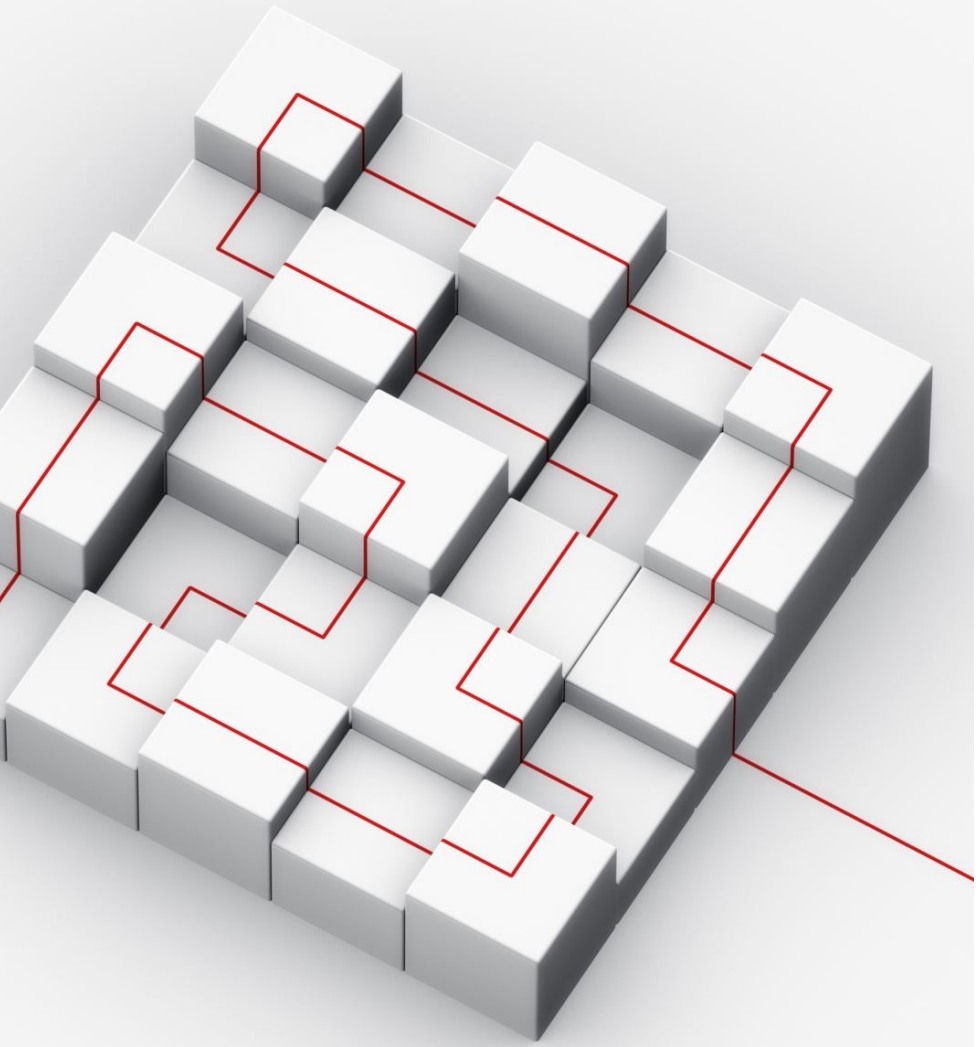
- O CSS object model é similar ao DOM – é uma árvore também
- O navegador converte as regras CSS em um mapa de estilos que ele pode entender e trabalhar.
- O navegador passa por cada conjunto de regras no CSS, criando uma árvore de nós com relacionamentos
  - pai, filho e irmão com base nos seletores CSS.
- A construção do CSSOM é bastante rápido

Interpretando o JavaScript

Durante a construção do CSSOM



JavaScript é interpretado, compilado, parseado e executado



Construindo a árvore de acessibilidade

- O navegador também cria uma árvore de acessibilidade que os dispositivos assistivos usam para analisar e interpretar o conteúdo.
- O modelo de objeto de acessibilidade (AOM) é como uma versão semântica do DOM.
- Quando o DOM é atualizado, o Browser atualiza o AOM
- AOM não é modificável pelas próprias tecnologias assistivas.
- O conteúdo só estará acessível aos leitores de tela quando o AOM estiver pronto

Renderizar

- Estilo, layout, pintura e, em alguns casos, composição.
- CSSOM e DOM criadas na etapa de análise são combinadas em uma árvore de renderização
  - Calcular o layout de cada elemento visível, que é então apresentado na tela.
- O conteúdo pode ser promovido para suas próprias camadas e composto, melhorando o desempenho pintando partes da tela na GPU em vez da CPU, liberando o thread principal.

3. Criação da Árvore de Renderização

- Combinando o DOM e o CSSOM
- Aplicação das regras do CSSOM no DOM.
  - Julga as regras em cascata

4. Layout

- Apresentar na tela o layout da árvore de renderização, calculando a geometria de cada um dos nós
  - Cada largura, altura e localização de nós

5. Pintura

- Cada um dos nós

Atividade

desenhe a árvore DOM para o seguinte código

```
<html>
<head>
  <title>Fundamentos de HTML</title>
  <meta name="keywords" content=”Browsers, Renderização, HTML, Fundamentos da Web">
  <meta name="description" content=”Esta aula descreve o carregamento de páginas Web">
  <meta charset=”utf-8">
  <link rel=”stylesheet" href="../estilo.css">
</head>
<body>
  <h1>Fundamentos da Web</h1>
  <p>Navegadores seguem um processo para a renderização de páginas. Conhecer esse procedimento nos ajuda
a desenvolver sites mais eficientes.</p>
  <ol>
    <li>Construindo o DOM</li>
    <li>Construindo o CSSOM</li>
    <li>Construindo a Árvore de Renderização</li>
    <li>Layout</li>
    <li>Pintura</li>
  </ol>
</body>
</html>
```

Sintaxe HTML

https://w3schools.com/html/html_elements.asp

Elementos HTML

- Definido por uma tag de abertura, algum conteúdo e tag de fechamento

```
<tagname>Conteúdo aqui...</tagname>
```

- Exemplos:

```
<h1>My First Heading</h1>
<p>My first paragraph.</p>
```

Elementos aninhados

```
<!DOCTYPE html>
<html>
<body>

  <h1>My First Heading</h1>
  <p>My first paragraph.</p>

</body>
</html>
```

Elementos HTML vazios

- Elementos HTML sem conteúdo
- Exemplo

```
<p>This is a <br> paragraph with a line break.</p>
```

Quebra de linha

HTML não é case sensitive

MAS... W3C recomenda minúsculas em HTML

Primeiras tags

- <h1> é usado para cabeçalhos de maior nível
  - Exemplo: título de um livro
- <h6> cabeçalhos de menor nível
  - Exemplo: subsubsubseção...

Tag	Descrição
<html>	Define o nó raiz de um documento HTML
<body>	Define o corpo do documento
<h1> a <h6>	Define cabeçalhos de conteúdo em HTML

Atributos HTML

- Todos elementos HTML podem ter **atributos**
- Atributos oferecem **informações adicionais** sobre elementos
- Atributos são sempre especificados na **tag de abertura**
- Atributos normalmente vêm em pares **nome=“valor”**

lang é um atributo que especifica o idioma da página.

Ajuda motores de busca e navegadores

- Exemplo:

```
<!DOCTYPE html>
<html lang=“pt-br”>
<body>

  <h1>My First Heading</h1>
  <p>My first paragraph.</p>

</body>
</html>
```

Estrutura de um documento HTML

https://htmlquick.com/tutorials/document-structure.html

Estrutura básica

- Consiste nas seguintes seções:
  - The DTD (!DOCTYPE declaration).
  - The main container (html element).
  - The head section (head element).
  - The body section (body element).

Doctype declaration

- Documentos HTML precisam iniciar com a declaração de seu tipo
- Indica para o navegador qual tipo de documento está sendo manipulado
- A declaração é inserida com a marcação especial **!doctype**
- Em HTML5, a declaração é feita da seguinte forma:

```
<!doctype html>
```

Doctype

HTML5

```
<!DOCTYPE html>
```

HTML 4

```
<!DOCTYPE html PUBLIC
  "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">
```

Transitional XHTML 1.0

```
<!DOCTYPE html PUBLIC
  "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/
   xhtml1-transitional.dtd">
```

Strict XHTML 1.0

```
<!DOCTYPE html PUBLIC
  "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/
   xhtml1-strict.dtd">
```

XML Declaration

```
<?xml version="1.0" ?>
```

Contêiner principal: elemento html

- <html> deve envolver TODAS as demais marcações

```
<!doctype html>
<html lang="pt-br">
  todos os elementos html aqui!
</html>
```

head de um documento

- Contêiner para metadados do documento
- 5 categorias de metadados:
  1. **Título do documento**: descreve brevemente o tema do documento. É OBRIGATÓRIO.
     - É declarada pelo elemento **<title>**
  2. **Declarações de estilo**: definição de grupos de estilos usados para configurar a apresentação
     - São declaradas por elementos **<style>**
  3. **Scripts client-side:** insere programas para dar funcionalidades e interatividade
     - São declaradas por elementos **<script>**
  4. **Instruções meta:** definem atributos e valores descritores
     - Declaradas por elementos **<meta>**
  5. **Informações relacionais:** indicam recursos relacionados ao documento
     - Declaradas por elementos **<link>**

```
<head>
  <title>Eppur si muove</title>
  <meta name="keywords" content="Galileo Galilei, heliocentrism, geocentrism">
  <meta name="description" content="This document approaches briefly the works of Galileo Galilei about the Heliocentrism...">
  <meta name="Author" content="Mark Rottenberg">
  <style>
     table {
        width: 100%;
        border-color: black;
     }
  </style>
  <script>
     result = 0;
     let increment  = amount => { result += amount; }
  </script>
  <link rel="index" href="../index.html">
  <link rel="alternate" media="print" href="printer-version.html">
</head>
```

body de um documento

- Parte renderizada do documento, *viewport*

```
<!doctype html>
<html lang=“en-US”>
  <head>
    <title>This is the story of me, learning HTML on my own</title>
  </head>
  <body>
    <p>Today I woke up decided to learn <abbr title="Hypertext Markup Language">HTML</abbr> on my own.
    </p>
  </body>
</html>
```

Elementos

<meta>, <link>, <style>

https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/The_head_metadata_in_HTML

Metadados em HTML

- Dados que descrevem dados...
- Auxiliam na definição dos conteúdos da página
- Melhoram a indexação da página por buscadores >> melhor ranqueamento
- Exemplos de metadadas em HTML:
  - author, description, charset, keywords, *open graph datas*

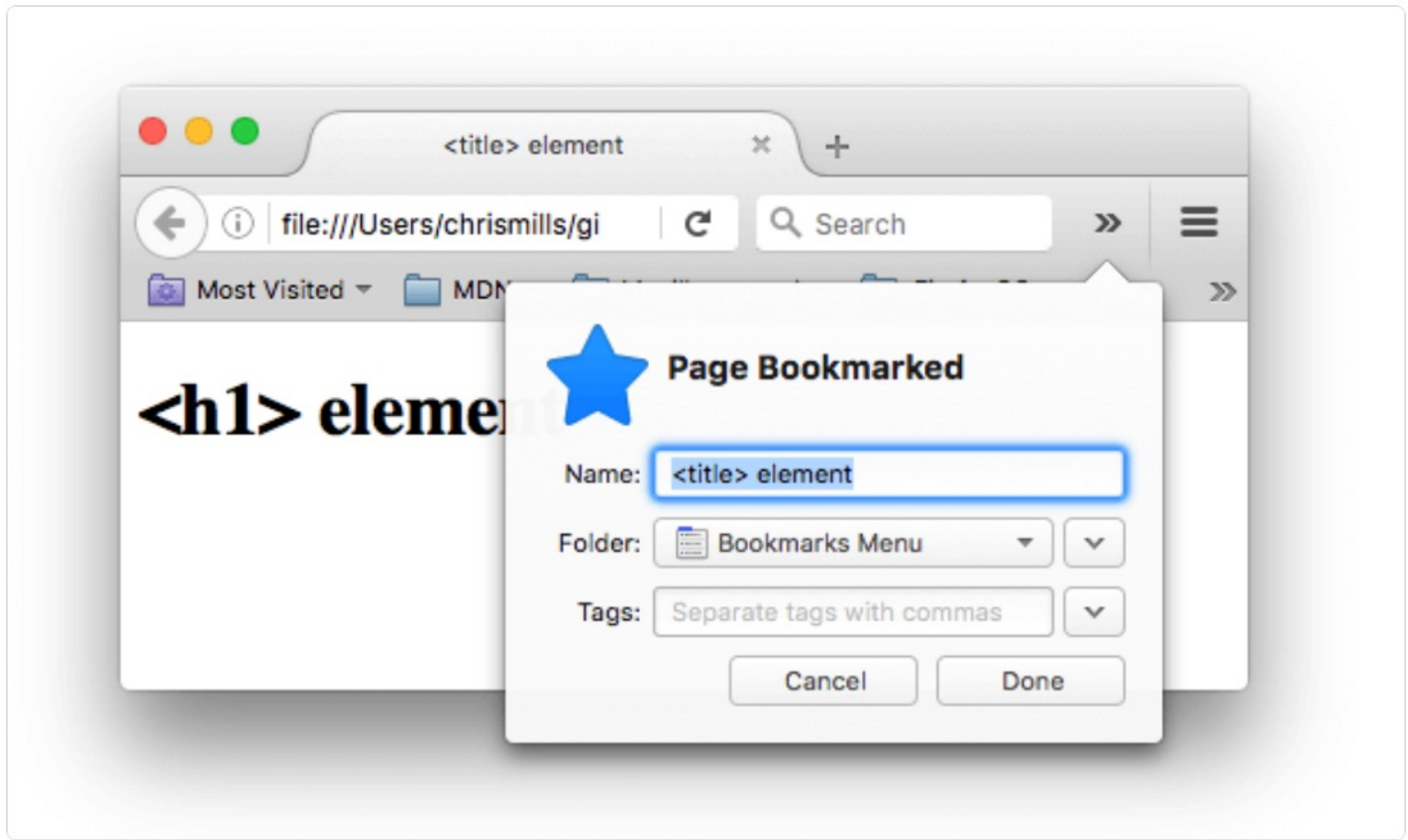
Metadados em HTML

- São declarados no **`head`** do documento e não ficam aparentes na viewport
- O `<head>` contém informações como
  - `<title>` – título da página; importantíssimo para o ranqueamento; aparece na aba do site
  - `<link>` – referencia arquivos externos, como folhas de estilo
  - `<meta>` – para metadescritores

Adicionando um título

- Não confunda title com h1
  - <h1> aparece ao longo da página, na *viewport*
    - Marca os títulos de conteúdos
      - Título de histórias, artigos, ou qualquer outro conteúdo apropriado
  - <title> é um metadados que representa o título geral da página HTML como um todo, não de conteúdo.
    - Só pode haver um.

```
<!doctype html>
<html>
  <head>
    <title>título da página aqui</title>
  </head>
  <body>
    <h1>Título de conteúdo</h1>
    <p>Um conteúdo qualquer</p>
    <h1>Outro título de conteúdo</h1>
    <p>ATENÇÃO: use h1 com critério...</p>
  </body>
</html>
```



Title é usado no histórico do Browser

Referências

- Como os navegadores interpretam páginas:
  - https://developer.mozilla.org/en-US/docs/Web/Performance/How_browsers_work#parsing
- Elementos básicos de HTML
  - https://w3schools.com/html/html_elements.asp
- Estrutura de um documento HTML
  - https://htmlquick.com/tutorials/document-structure.html
- Metadescritores
  - https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/The_head_metadata_in_HTML

Tags Textuais

Elementos HTML de conteúdo

- Learn the purpose of the following elements:
- 3.2
- 3.3
- 3.4 <time>
- 3.5 <code>
- 3.6
- 3.7
- 3.8 <abbr>
- sub, sup, del, mark, q, quotation,

Elemento

- Tag semântica para dar ênfase em partes de textos.
  - Por padrão, o texto marcado com aparece em *itálico*. Esta aparência pode ser alterada via CSS.

```
<p>Lorem <em>ipsum</em> dolor et amec pur amer avec na muer.</p>
```

Elemento

- Tag semântica para destacar partes de textos, reforçando sua relevância frente o restante do texto.
  - Por padrão, o texto marcado com aparece em **negrito**. Esta aparência pode ser alterada via CSS.

```
<p>Lorem <strong>ipsum</strong> dolor et amec pur amer avec na muer.</p>
```

Elemento <mark>

- Tag semântica para destacar partes de textos, algo semelhante a um marcador de textos.
  - Por padrão, o texto marcado com <mark> aparece com fundo amarelo. Esta aparência pode ser alterada via CSS.

```
<p>Lorem <mark>ipsum</mark> dolor et amec pur amer avec na muer.</p>
```

Elemento <q> <blockquote> e <code>

- <q> Tag semântica utilizada para sinalizar que o conteúdo se refere a uma citação curta.
- <blockquote> Tag semântica utilizada para sinalizar que o conteúdo se refere a uma citação longa.
- <code> Tag semântica utilizada para sinalizar que o conteúdo se refere a um trecho de código

Elemento

- Tag genérica para aplicarmos regras de CSS em partes de textos.

```
<p>Lorem <span>ipsum</span> dolor et amec pur amer avec na muer.</p>
```

- Com este recurso, é possível definir regras de CSS para alterar a visualização do trecho.

Elemento <sup> e <sub>

- <sup> Tag utilizada para deixar um trecho de texto em sobrescrito.

```
<p>X<sup>2</sup> = Y<sup>2</sup> + Z<sup>2</sup>
```

Resultado: $X^2 = Y^2 + Z^2$

- <sub> Tag utilizada para deixar um trecho de texto em subescrito.

```
<p>X<sub>2</sub> = Y<sub>2</sub> + Z<sub>2</sub>
```

Resultado: $X_2 = Y_2 + Z_2$

Elemento <address> e <time>

- <address> Tag semântica utilizada para sinalizar que o conteúdo se refere a um endereço.

```
<p>Sede: <address>Rua João Brasil, 25</address>.</p>
```

- <time> Tag semântica utilizada para sinalizar que o conteúdo se refere a uma data.

```
<p>Data de Fundação: <time>20/01/1984</time>.</p>
```

Elementos <b>, <i>, <u>

- Tag legadas para marcar partes de textos
  - <b> era usado para destacar trechos de texto como **negrito**
  - <i> era usado para destacar trechos de texto como *itálico*
  - <u> era usado para destacar trechos de texto como <u>sublinhado</u>
- Conforme discutimos em aula, a linguagem HTML deve ser usada para estruturar documentos web, e não para definir aspectos de visualização...

Referências

- Marcações textuais
  - https://w3schools.com/html/html_formatting.asp
  - https://developer.mozilla.org/en-US/docs/Web/HTML/Element#text_content]
- Marcações semânticas em linha
  - https://developer.mozilla.org/en-US/docs/Web/HTML/Element#inline_text_semantics