

Programação de Aplicações Web

Aula 1: Introdução ao HTML e Estrutura Web

```
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
      self.logduplicates = True
      self.debug = debug
      self.logger = logging.getLogger(__name__)
      if path:
          self.file = open(os.path.join(path, 'responses.log'), 'w')
          self.file.seek(0)
          self.fingerprints.update(fp)
      else:
          self.fingerprints = set()
      self.job_dir = job_dir
      self.settings = settings
      self.debug = settings.getbool('DEBUG', False)
      return cls(job_dir(settings), settings)

    def request_seen(self, request):
        fp = self.request_fingerprint(request)
        if fp in self.fingerprints:
            return True
        self.fingerprints.add(fp)
        if self.file:
            self.file.write(fp + os.linesep)

    def request_fingerprint(self, request):
        fingerprint(request)
```



Como funciona a Web?

Quando você acessa um site, uma série de etapas acontece nos bastidores:

1. **Requisição (Request):** O navegador (cliente) envia um pedido para o servidor onde o site está hospedado.
2. **Resposta (Response):** O servidor processa o pedido e envia de volta os arquivos do site (HTML, CSS, Imagens).
3. **Renderização:** O navegador lê o código HTML e "desenha" a página na sua tela.



Cliente pede Servidor entrega Navegador desenha

O Navegador e o HTML



O navegador funciona como um intérprete que transforma código em visual.

O Processo de Renderização



- **1. Parsing:** O navegador lê o arquivo HTML linha por linha.
- **2. DOM (Document Object Model):** Cria uma estrutura em memória em forma de árvore.
- **3. Pintura:** Aplica estilos (CSS) e desenha os pixels na tela.

Estrutura Básica



Todo documento HTML precisa desta estrutura padrão.

- **DOCTYPE:** Avisa que é HTML5.
- **html:** A raiz do documento.
- **head:** Configurações (não visível).
- **body:** O que o usuário vê.

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="UTF-8">
    <title>Título da Página</title>
  </head>
  <body>
    <h1>Meu Título</h1>
    <p>Meu parágrafo.</p>
  </body>
</html>
```

Tags Principais

O HTML usa etiquetas para dar significado.

Texto e Títulos

- `<h1>` a `<h6>` : Títulos
- `<p>` : Parágrafos
- `` : **Negrito** (Importante)
- `` : *Itálico* (Ênfase)

Estrutura e Links

- `<div>` : Bloco genérico
- `` : Linha genérica
- `` : Imagens
- `<a>` : Links
- `` / `` : Listas

```
<h1>Título</h1>
<p>Texto com <strong>destaque</strong>. </p>
<a href="#">Link</a>
```

Títulos e Parágrafos



Hierarquia e blocos de texto.

Títulos (Headings)

- <h1> é o mais importante.
- <h6> é o menos importante.
- Use apenas **um** <h1> por página.

```
<h1>Manchete Principal</h1>

<h2>Subtítulo da Seção</h2>
<p>Primeiro parágrafo do texto.</p>

<p>Segundo parágrafo.</p>

<h3>Sub-seção menor</h3>
<p>Mais detalhes aqui.</p>
```

Parágrafos

- <p> define blocos de texto.
- Pula uma linha automaticamente.

Manchete Principal

Subtítulo da Seção

Primeiro parágrafo.

Formatação



Dando estilo semântico ao texto.

Estilos Comuns

- `` : **Negrito** (Importante)
- `` : *Itálico* (Ênfase)
- `<mark>` : **Marcado**
- `<small>` : Texto pequeno

Separadores

- `
` : Quebra de linha
- `<hr>` : Linha horizontal

```
<p>
  Este é um aviso <strong>URGENTE</strong>.
</p>
```

```
<p>
  Por favor, <em>leia com atenção</em>.
</p>
```

```
<hr>
```

```
<p>
  Texto final.
</p>
```

Este é um aviso **URGENTE**.

Por favor, *leia com atenção*.

Links (Âncoras)



A tag `<a>` conecta a web.

Atributos

- **href:** O endereço de destino.
- **target:** Onde abrir (`_blank` para nova aba).
- **title:** Texto de ajuda ao passar o mouse.

```
←—— Link para site externo —→  
<a href="https://google.com" target="_blank">  
    Ir para o Google  
</a>
```

```
←—— Link para página interna —→  
<a href="contato.html">  
    Fale Conosco  
</a>
```

Resultado

[Ir para o Google](https://google.com)

Imagens



Incorporando visual com a tag ``.

Atributos

- **src**: Caminho da imagem (URL ou arquivo).
- **alt**: Texto alternativo (Acessibilidade).
- **width**: Largura.
- **height**: Altura.



A tag `img` não tem fechamento!

```

```



Exemplo de Gato

Listas



Agrupando itens.

Não Ordenada (``)

- Itens com marcadores (bolinhas).
- A ordem não importa.

Ordenada (``)

- Itens numerados (1, 2, 3...).
- A ordem importa.

Código:

```
<ul>
  <li>Café</li>
  <li>Leite</li>
</ul>

<ol>
  <li>Acordar</li>
  <li>Codar</li>
</ol>
```

Resultado:

- Café 1. Acordar
- Leite 2. Codar

Div vs Span



A diferença entre Bloco e Linha.

<div> (Bloco)

- Ocupa toda a largura da linha.
- Empurra o próximo elemento.
- Usado para layout.

 (Linha)

- Ocupa só o espaço do conteúdo.
- Fica ao lado.
- Usado para destacar texto.

```
←— Blocos —>
<div style="background: blue;">Bloco 1</div>
<div style="background: red;">Bloco 2</div>

←— Linha —>
<p>
    Texto com <span style="color: yellow;">destaque</span>.
</p>
```

Bloco 1

Bloco 2

Texto com destaque no meio.

Developer Tools (DevTools)

O "Raio-X" do navegador.

O que são?

Ferramentas integradas (Chrome, Firefox, Safari) para **olhar por baixo do capô** de qualquer página.

Por que usar?

- **Debugging:** Encontrar e corrigir erros.
- **Experimentação:** Testar mudanças em tempo real.
- **Aprendizado:** Ver como sites foram feitos.
- **Performance:** Analisar lentidão.



Como abrir:

- F12
- Ctrl + Shift + I (Win/Linux)
- Cmd + Opt + I (Mac)
- Botão Direito > Inspecionar

Funcionalidades Principais

O que você mais vai usar no dia a dia.

1. Elements (Inspetor)

- Mostra o **HTML** e **CSS** do site.
- Permite editar qualquer coisa na tela.
- **Dica:** Use a "setinha" (select) para clicar e inspecionar elementos.

2. Console

- Mostra **erros** (em vermelho).
- Mostra avisos (em amarelo).
- Permite rodar comandos JavaScript.

3. Network (Rede)

- Monitora o download de arquivos.
- Vê se imagens quebraram (404).
- Mede o tempo de carregamento.

4. Device Toolbar (Simulador)

- Botão "Celular/Tablet".
- Simula telas pequenas.
- Essencial para testar **sites responsivos**.

Mão na Massa: Inspecionar



As alterações feitas no DevTools são **temporárias** e **locais**. Elas existem apenas no seu navegador até você recarregar a página.

Tente você mesmo

1. Abra o arquivo de exemplo (`exemplo-html.html`) ou qualquer site.
2. Clique com o **Botão Direito** em cima de um Título.
3. Escolha a opção **Inspecionar**.
4. No painel que abrir (Elements), dê um duplo-clique no texto do título.
5. Escreva outra coisa e aperte **Enter**.

Hacker Mode



Você acabou de "hackear" a página (mas só para você ver!).



Exercício de Sala: Sua Primeira Página

Agora é a sua vez de colocar a mão na massa! Vamos criar uma página HTML simples a partir do zero.

⚠️ Atenção: Este exercício deve ser feito **100% à mão**. O uso de ferramentas de Inteligência Artificial (ChatGPT, Copilot, etc.) **não é permitido** nesse momento. O objetivo é fixar as tags que acabamos de aprender!

Passo a Passo

1. Crie uma pasta nova no seu computador para este projeto.
2. Dentro dessa pasta, crie um arquivo chamado `index.html`.
3. Abra esse arquivo no seu editor de código.
4. Digite a estrutura básica do HTML e adicione o conteúdo usando as tags listadas abaixo.
5. Salve o arquivo (`Ctrl+S` / `Cmd+S`) e abra-o no seu navegador para ver o resultado!



Exercício de Sala: Sua Primeira Página

Tags Obrigatórias

Sua página deve conter, no mínimo, as seguintes tags:

- **Estrutura Básica:** `<html>` , `<head>` , `<title>` e `<body>`
- **Títulos:** Pelo menos um `<h1>` e um `<h2>`
- **Textos:** Pelo menos dois parágrafos (`<p>`)
- **Listas:** Uma lista (`` ou ``) com itens (``)
- **Imagens:** Uma imagem (``)
- **Links:** Um link (`<a>`)