

# ALGORITMOS E COMPLEXIDADE



ANÁLISE DE ALGORITMO

## WEB SCRAPING II

Aula 06



16/09/2025



A cartoon illustration of a brown capybara with a white patch on its forehead. It has large, expressive eyes and is holding a white coffee cup with both hands. The cup has a small logo on it. The background behind the capybara is a light blue.

Mensagem do Dia

Assunto  
Sério

# Mini-projeto



1 ponto  
(Simulado)

# Mini-projeto - Detalhes

Em Dupla

Entrega - 26/09

Apenas no  
**SAVA**



# Mini-projeto

1. Scrap Notícias;
2. Bot de Login;
3. Documentação



Detalhes do  
projeto Aqui



# Cronograma do Conteúdo

**Aula 01** - Introdução à raspagem de dados.

**Aula 02** - BeautifulSoup - HTML parsing básico.

**Aula 03** - Selenium - automação de navegação web.

**Aula 04** - Playwright - interação rápida com páginas dinâmicas.

**Aula 05** - Scrapy - framework robusto para raspagem.

**Aula 06** - Comparação das ferramentas + boas práticas.

**Aula 07** - Atividade/desafio final.

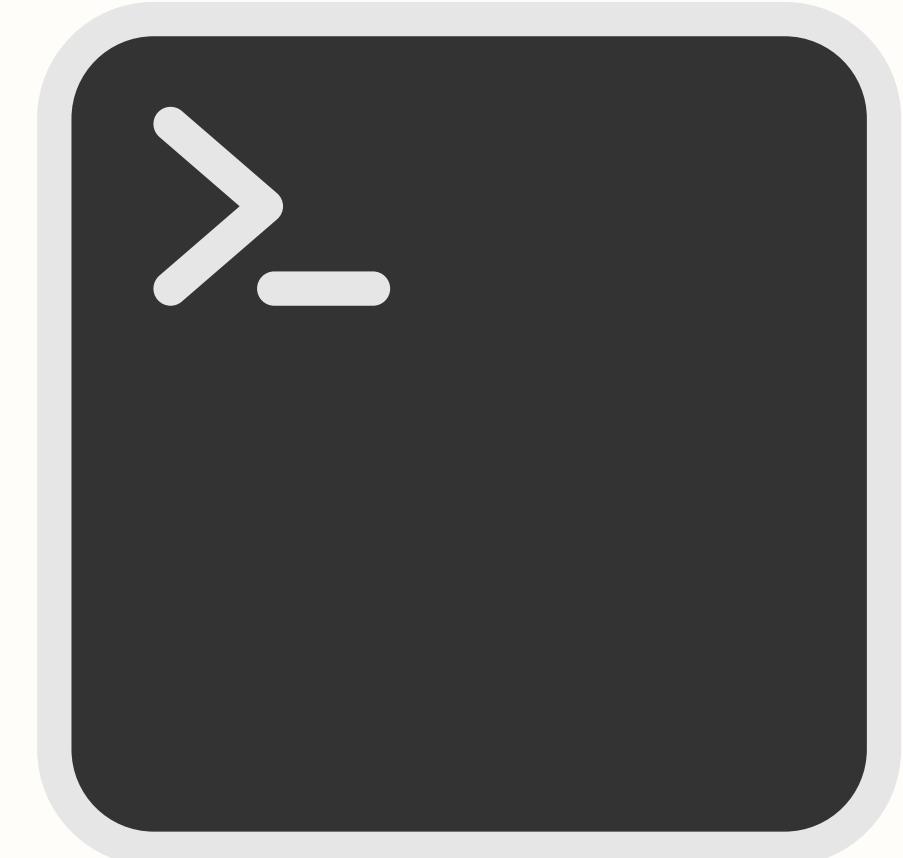
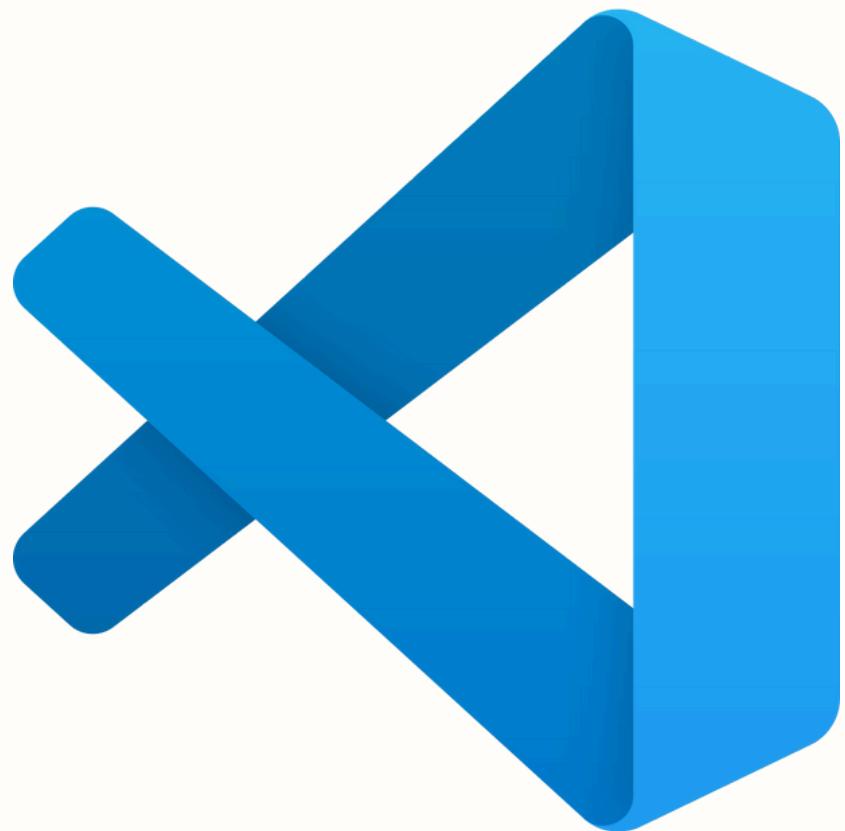
# Objetivo da aula

- Entender diferenças entre scraping estático (`requests + BeautifulSoup`) e scraping dinâmico (`Selenium`).
- Aprender a configurar ambiente (`Python, drivers, webdriver-manager`).
- Extrair dados de páginas reais (**prática 1**).
- Criar um bot que faz login e interage com um site de teste (**prática 2**).
- Salvar dados em `CSV/JSON`, tratar erros, usar waits e técnicas para tornar automação mais robusta.

# Pré-requisitos

- Python 3.9+ instalado.
- **Conhecimentos básicos de Python (requests, pandas são úteis).**
- Editor (VS Code, PyCharm) e terminal.
- Acesso à internet.

# Preparando o Ambiente



Google  
**colab**

# Preparando o Ambiente

No terminal | Prompt de Comando (CMD)

`python -m venv venv`

`source venv/bin/activate`

`# Linux / macOS`

`venv\Scripts\activate`

`# Windows`

`pip install --upgrade pip`

# Ambiente e Dependências

Após prepara instale as dependencias:

```
pip install requests beautifulsoup4 lxml selenium  
webdriver-manager pandas
```

# Ambiente e Dependências

**selenium** – controle do browser.

**webdriver-manager** – evita ter que baixar/georreferenciar manualmente o driver.

**beautifulsoup4, lxml** – parsing.

**requests** – scraping estático quando possível.

**pandas** – salvar/visualizar dados (opcional).

# Conceitos importantes

- **Requests + BeautifulSoup:** ideal para páginas estáticas (conteúdo entregue no HTML inicial).  
Mais rápido e leve.
- **Selenium:** controla um navegador real (Chrome, Firefox), necessário quando a página carrega conteúdo via JavaScript/AJAX, ou quando é necessária interação (login, clique, rolagem).

# Conceitos importantes

- **Headless mode:** navegador sem UI – útil para servidores. Nem sempre evita detecção.
- **Waits:**
  - **Implicit wait** – aviso global para encontrar elementos.
  - **Explicit wait (WebDriverWait)** – esperar por condição específica (elemento visível, clicável).
- **Seletores:** **id, class, name, xpath, css selector.** Prefira id/css quando possível.

# Conceitos importantes

- **Rate limiting & politeness:** delays, limitar frequência de requests, seguir robots.txt.
- **Detecção e bloqueios:** Cloudflare, WAF, CAPTCHAs – técnicas complexas; sempre respeitar TOS.

# Exemplos e Códigos Aqui



# O que é Selenium?

**Selenium** é um projeto de código aberto (open-source) que consiste em um conjunto de ferramentas e bibliotecas que permitem a automatização de navegadores web.

## De forma simples

o Selenium é um "robô" que você programa para controlar um navegador (como Chrome, Firefox, Edge, etc.) exatamente como um ser humano faria, mas de forma automatizada, rápida e repetitiva.



# Para que serve?

- Automação de Testes (Principal uso); !
- Web Scraping / Extração de Dados; !
- Automação de Tarefas Repetitivas. !

# Entendendo o que “caçar elementos” na web



Abram um site;  
Pressione F12;



Prestem atenção ao  
Professor

# Atividade de Sala





# Bot de Login

- Criar um bot usando Selenium que:
  - Faça login em um site de testes (ex: <https://the-internet.herokuapp.com/login>).
  - Navegue até a área autenticada.
  - Capture e salve uma mensagem de sucesso ou erro em um arquivo de log.
  - Tire uma screenshot da tela após o login.

# Próxima Aula

1. ~~BeautifulSoup~~

2. ~~Selenium~~

3. Playwright

4. Scrapy

5. Requests

