

Principais Bibliotecas de Web Scraping com Python

Requests

Uso: Fazer requisições HTTP (GET, POST, etc.)

Ideal para: baixar o HTML de uma página ou acessar uma API.

Instalação: `pip install requests`

Exemplo:

```
import requests
```

```
res = requests.get("https://example.com")
```

```
html = res.text
```

BeautifulSoup (bs4)

Uso: Fazer parsing e navegação em HTML.

Ideal para: pegar textos, links, atributos, tags específicas.

Instalação: `pip install beautifulsoup4`

Exemplo:

```
from bs4 import BeautifulSoup
```

```
soup = BeautifulSoup(html, "html.parser")
```

```
titulos = soup.find_all("h1")
```

lxml

Uso: Parsing de HTML/XML mais rápido que o BeautifulSoup puro.

Ideal para: quando você precisa de performance.

Principais Bibliotecas de Web Scraping com Python

Instalação: `pip install lxml`

Nota: Pode ser usado como motor de parsing no BeautifulSoup.

Selenium

Uso: Automatiza navegadores reais.

Ideal para: lidar com sites JavaScript, botões, login, scroll infinito, etc.

Instalação: `pip install selenium`

Exemplo:

```
from selenium import webdriver
```

```
driver = webdriver.Chrome()
```

```
driver.get("https://example.com")
```

```
html = driver.page_source
```

Playwright

Uso: Automação de navegador moderna (como o Selenium, mas mais rápido e confiável).

Ideal para: sites dinâmicos e complexos.

Instalação: `pip install playwright` && `playwright install`

Pandas

Uso: Manipular dados extraídos (listas, tabelas, etc.).

Ideal para: transformar HTML em DataFrames, salvar em CSV/Excel.

Principais Bibliotecas de Web Scraping com Python

httpx

Uso: alternativa ao requests, suporta requisições assíncronas (mais rápido em alguns casos).

Instalação: `pip install httpx`