**Tutorial de Web Scraping com Scrapy em Python**

**Neste tutorial, você aprenderá como configurar um ambiente virtual, instalar o Scrapy e criar um spider para web scraping. Certifique-se de usar essas técnicas com responsabilidade e ética, respeitando os termos de serviço dos sites que você planeja raspar.**

**Passo 1: Configurar a Política de Execução de Scripts no PowerShell**

**No PowerShell do Windows, você pode definir a política de execução de scripts como "Unrestricted" usando o seguinte comando:**

Set-ExecutionPolicy Unrestricted -Scope CurrentUser

**Isso permitirá a execução de scripts no PowerShell sem restrições. Lembre-se de que a alteração da política de execução de scripts pode afetar a segurança do seu sistema, portanto, use essa configuração com cuidado e apenas se souber a origem e a segurança dos scripts que está executando.**

**Passo 2: Criar o Ambiente Virtual**

**No terminal integrado, você pode criar um ambiente virtual usando o módulo venv do Python. Substitua <nome-do-seu-ambiente-virtual> pelo nome que deseja dar ao seu ambiente virtual:**

python -m venv <nome-do-seu-ambiente-virtual>

**Por exemplo, se você deseja criar um ambiente chamado "meuambiente", o comando será:**

python -m venv meuambiente

**Passo 3: Ativar o Ambiente Virtual**

**Agora, você precisa ativar o ambiente virtual que acabou de criar. No terminal, digite o seguinte comando:**

.<nome-do-seu-ambiente-virtual>\Scripts\Activate

**Usando o exemplo anterior, você digitará:**

.\meuambiente\Scripts\Activate

**Você saberá que o ambiente está ativado quando o nome do ambiente virtual aparecer antes do prompt de comando no terminal.**

**Passo 4: Instalar Dependências do Projeto**

**Agora que o ambiente virtual está ativado, você pode instalar as dependências do seu projeto usando o pip. Por exemplo:**

pip install scrapy

pip install pandas

pip install pillow

**Passo 5: Criar um Projeto Scrapy**

**Agora que o Scrapy está instalado, você pode criar um novo projeto. Navegue até a pasta onde deseja criar o projeto e execute o seguinte comando:**

scrapy startproject nome\_do\_projeto

**Substitua "nomedoprojeto" pelo nome que deseja dar ao seu projeto.**

**Passo 6: Criar uma Spider**

**Uma spider é um script que define como o Scrapy navegará e coletará dados de um site específico. Vá para a pasta do seu projeto e crie uma spider com o seguinte comando:**

cd nome\_do\_projeto

scrapy genspider nome\_da\_spider exemplosite.com

**Substitua "nome\_da\_spider" pelo nome da sua spider e "exemplosite.com" pelo site que deseja raspar. Isso criará um arquivo da spider na pasta "spiders" do seu projeto.**

**Passo 7: Executar a Spider**

**Para executar a spider, use o seguinte comando:**

scrapy crawl nome\_da\_spider

**Passo 8: Salvar os Dados Coletados**

**Você pode salvar os dados coletados em diferentes formatos, como CSV, JSON ou XML. Por exemplo, para salvar em CSV, use o seguinte comando:**

scrapy crawl nome\_da\_spider -o dados.csv

**Lembre-se de que as etapas podem variar dependendo do site que você está raspando. Sempre verifique a estrutura do site e ajuste suas seleções de XPath conforme necessário.**

**Passo 9: Usar o Scrapy Shell para Depuração**

**O Scrapy Shell é uma ferramenta útil para verificar a integridade do XPath e depurar seu código. Use o seguinte comando para iniciar o Scrapy Shell:**

scrapy shell https://www.seusite.com

**Você pode então interagir com o site e testar seus comandos XPath para garantir que estejam corretos.**

**Com este tutorial, você deve estar pronto para iniciar projetos de web scraping com Scrapy em Python. Lembre-se sempre de respeitar os termos de uso do site que está raspando e garantir que suas ações sejam legais e éticas.**

**Comandos Adicionais**

Para criar uma pasta no CMD, você pode usar o comando mkdir, por exemplo: arduino

mkdir nova\_pasta

**Para verificar se a pasta foi criada com sucesso, você pode listar o conteúdo do diretório usando o comando dir no Windows:**

dir

**Para navegar entre diretórios, você pode usar os comandos cd para entrar em uma pasta e cd .. para voltar para o diretório anterior.**

Lembre-se de que o terminal CMD e outros usam a linguagem Batch, então os comandos específicos podem variar ligeiramente em diferentes sistemas operacionais ou versões do Windows.

**Para criar um ambiente virtual dentro do seu diretório, você usa:** python -m venv venv

**E para ativar o ambiente virtual, você usa:**

.\venv\Scripts\activate

**E para desativar o ambiente virtual, você usa o comando:**

deactivate

Certifique-se de estar no diretório correto ao usar esses comandos.

**O Scrapy** é um framework popular em Python para web scraping e crawling. Lembre-se de que você deve estar no diretório do projeto Scrapy para executar esses comandos e que pode ser necessário instalar as dependências adequadas e configurar o spider antes de usá-lo. Certifique-se também de seguir as diretrizes éticas e legais ao realizar web scraping. Aqui estão alguns dos comandos mais comuns relacionados ao Scrapy: **Criar um novo projeto Scrapy:**

scrapy startproject nome\_do\_projeto

**Criar um spider:**

scrapy genspider nome\_do\_spider nome\_do\_site

**Executar um spider:**

scrapy crawl nome\_do\_spider

**Executar um spider e salvar os resultados em um arquivo JSON:** scrapy crawl nome\_do\_spider -o arquivo.json

**Executar um spider em segundo plano e registrar as saídas em um arquivo de log:**

scrapy crawl nome\_do\_spider -o arquivo.json -t json --nolog

**Listar todos os spiders disponíveis em um projeto:**

scrapy list

**Verificar a estrutura de um spider:**

scrapy parse url -c nome\_do\_spider

**Executar um spider com um intervalo de tempo entre as requisições:** scrapy crawl nome\_do\_spider -s DOWNLOAD\_DELAY=2

**Executar um spider com um User-Agent personalizado:**

scrapy crawl nome\_do\_spider -s USER\_AGENT="Meu User-Agent"

**Executar um spider com um proxy:**

scrapy crawl nome\_do\_spider -s HTTP\_PROXY=proxy\_url

**Executar um spider e exportar os resultados para um arquivo CSV:** scrapy crawl nome\_do\_spider -o arquivo.csv -t csv

**Executar um spider e seguir links para páginas seguintes (paginação):** scrapy crawl nome\_do\_spider -s DEPTH\_LIMIT=3

**Varrendo o site com “scrapy shell”**

scrapy shell https://www.confederadosinternacional.org/blog

**#Este comando fará você ficar dentro do site e poderá usar comando do scrapy no site em tempo de execução.**

**#para pegar a div corretamente deve se apagar o id de uma div copiando novamente seu xpath full, após isso devemos apagar a numeração final da dive que estará entre []**

**Desta forma pegamos todas as divs**

**#pegando as divs**

divs = response.xpath("/html/body/div/div/div[3]/div/main/div/div/div/div[2]/div/div/div/section/div[2]/div/div/div/div/div/div/div[3]/div[2]/div[1]/div[2]/div/div/div/div/div/div[2]/div/div/div")

for div in divs: print(div)  
  
**#pegando os links em cada div**  
  
 links = divs.xpath('.//a/@href').getall()  
  
**#imprimindo os links**  
  
 for link in links: print(link)

**#Saia do scrapy shell com o comando**

exit

**#Agora vamos pegar o texto das paginas em cada link**

scrapy shell https://www.confederadosinternacional.org/post/projeto-super-computador

/html/body/div/div/div[3]/div/main/div/div/div/div[2]/div/div/div/section/div[2]/div/div[1]/div/div/div/div/div[3]/div[2]/div/div[2]/div/div[1]/div[1]/article/div/div[2]/article/div/div/div/div

**# Supondo que você tenha a div representada pelo XPath**

minha\_div = response.xpath('/html/body/div/div/div[3]/div/main/div/div/div/div[2]/div/div/div/section/div[2]/div/div[1]/div/div/div/div/div[3]/div[2]/div/div[2]/div/div[1]/div[1]/article/div/div[2]/article/div/div/div/div')

**# Extrair texto diretamente da div e de elementos <p> dentro de <span>**

texto = minha\_div.xpath('.//text()').getall()

**#imprimindo o texto**  
  
for text in texto: print(text)

**# Agora, 'texto' conterá todos os textos dentro da div e elementos <p> dentro de <span>**

Vamos falar sobres como pegar o xpath com método container e pegar a classe que contém um determinado nome.  
Lembre-se de sempre estar dentro do repositório correto, se instalou alguma dependência em seu ambiente virtual lembre-se de ativá lo.  
  
Sempre verifique se os xpath estão corretos com a página no spider .  
  
nos estamos no site [www.confederadosinternacional.org](http://www.confederadosinternacional.org)  
o correto seria para aonde estamos coletando [www.confederadosinternacional.org/blog](http://www.confederadosinternacional.org/blog)   
  
**Vamos criar um novo diretório**

mkdir nome\_projeto

**Navegue para o novo projeto**

cd nome\_projeto

**Vamos criar o ambiente virtual dentro do novo projeto**  
python -m venv venv

**Vamos ativar o ambiente virtual**

.\venv\Scripts\activate

**Para desativar o ambiente virtual use console com linguagem Batch o seguinte comando;**  
deactivate

**Vamos instalar o scrapy;**

**Scrapy: A biblioteca principal que permite a você criar e executar spiders para raspar dados de sites.**

pip install scrapy

**Além disso, é comum usar outros pacotes para armazenar os dados coletados, como o pandas:**

pip install pandas

**Para fazer o download de imagens, você pode usar a biblioteca Pillow:**

pip install pillow

**Isso instalará o Scrapy e suas dependências.**

**Agora que o Scrapy está instalado, você pode criar um novo projeto. Para isso, navegue até a pasta onde deseja criar o projeto e execute o seguinte comando:**

**Vamos criar um projeto scrapy,**

**Dentro da pasta projeto, não do ambiente virtual.**

scrapy startproject nome\_do\_projeto

*Substitua "nomedoprojeto" pelo nome que deseja dar ao seu projeto.*

**Uma spider é um script que define como o Scrapy navegará e coletará dados de um site específico. Vá para a pasta do seu projeto e crie uma spider com o seguinte comando:**

**Navegue até o projeto do scrapy:**

cd nomedoprojeto

**crie um spider:**

scrapy genspider nomedaspider exemplosite.com

**Substitua "nomedaspider" pelo nome da sua spider e "exemplosite.com" pelo site que deseja raspar. Isso criará um arquivo da spider na pasta "spiders" do seu projeto.**

**Para Executar o spider usamos a seguinte linha de comando**  
  
scrapy crawl nome\_spider  
  
**Para executar o spider e salvar os dados usa-se  
Para salvar em CSV**

scrapy crawl nome\_spider -o dados.csv

**Para salvar em JSON**

scrapy crawl nome\_spider -o dados.json

**Para salvar em XML**

scrapy crawl nome\_spider -o dados.xml

**Se o código não estiver salvando os dados remova a linha que contém**allowed\_domains =

**Usar o scrapy shell para depurar alguma etapa do algoritmo criado usando scrapy é válido:**

**Use o scrapy shell para verificar a integridade do xpath, se está correto ou não.**