escola britânica de artes criativas & tecnologia

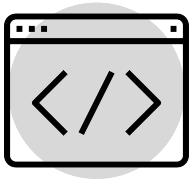
Profissão: Analista de dados

escola britânica de artes criativas & tecnologia

DE OLHO NO CÓDIGO

Coleta de dados II

- **Capte dados da web (crawling)**
- **Extraia dados da web (scrapping)**



Confira boas práticas da linguagem Python por assunto relacionado às aulas.

Capte dados da web (crawling)

O pacote "**requests**" é uma biblioteca Python usada para fazer solicitações HTTP. É uma das bibliotecas mais populares usadas por desenvolvedores Python para enviar solicitações para APIs e páginas da web. Acompanhe as dicas a seguir.

- **Use a versão mais recente**

  O pacote "requests" está em constante evolução, portanto, certifique-se de estar usando a versão mais recente.

- **Trate exceções**

  As solicitações HTTP podem falhar por uma série de motivos, como problemas de rede, erros do servidor e problemas de autenticação. Certifique-se de tratar essas exceções adequadamente em seu código. Por exemplo, use um bloco try-except para lidar com exceções de rede.

Capte dados da web (crawling)

O pacote "**requests**" é uma biblioteca Python usada para fazer solicitações HTTP. É uma das bibliotecas mais populares usadas por desenvolvedores Python para enviar solicitações para APIs e páginas da web. Acompanhe as dicas a seguir.

- **Use as funções de verificação de status**
  O pacote "requests" possui funções para verificar o status da resposta HTTP como "ok" e "status_code". Use essas funções para garantir que a resposta que você recebeu foi bem-sucedida antes de prosseguir com o processamento.

- **Use autenticação segura**
  Se você estiver fazendo solicitações para APIs que exigem autenticação, certifique-se de usar autenticação segura, como OAuth ou tokens de API.

Capte dados da web (crawling)

O pacote "**requests**" é uma biblioteca Python usada para fazer solicitações HTTP. É uma das bibliotecas mais populares usadas por desenvolvedores Python para enviar solicitações para APIs e páginas da web. Acompanhe as dicas a seguir.

escola britânica de artes criativas & tecnologia

- **Use SSL/TLS**
  Use SSL/TLS ao se comunicar com serviços *web* que exigem criptografia de ponta a ponta.

- **Use o método correto**
  Use o método HTTP correto para cada solicitação. Por exemplo, use GET para obter dados e POST para enviar dados para um servidor.

- **Gerencie cookies**
  O pacote "requests" permite o gerenciamento de cookies. Use essa funcionalidade para gerenciar cookies de sessão e autenticação.

- **Use a biblioteca "json“**
  O pacote "requests" pode lidar com solicitações e respostas JSON. Use a biblioteca "json" do Python para serializar e desserializar objetos JSON.

Capte dados da web (crawling)

Selenium é uma ferramenta popular de automação de testes de *software*. Acompanhe as boas práticas que você deve seguir ao usar o Selenium.

- **Identificadores únicos**
  Use identificadores únicos, como IDs ou nomes de classe, para localizar elementos na página da web. Não use identificadores que possam mudar ou não sejam exclusivos.

- **Esperas explícitas**
  Use esperas explícitas para esperar que elementos apareçam na página antes de interagir com eles. Isso ajuda a garantir que o teste funcione corretamente e que o elemento esteja disponível para interação.

Capte dados da web (crawling)

Selenium é uma ferramenta popular de automação de testes de *software*. Acompanhe as boas práticas que você deve seguir ao usar o Selenium.

- **Não dependa de espera implícita**

  A espera **implícita** é uma configuração que faz com que o Selenium aguarde automaticamente um determinado período de tempo antes de executar a próxima ação. Isso pode levar a testes não confiáveis e lentos. Use esperas **explícitas** em vez disso.

- **Crie funções de ajuda**

  Crie funções de ajuda para tarefas repetitivas, como preencher formulários ou clicar em botões. Isso pode economizar tempo e tornar o código mais legível.

Capte dados da web (crawling)

Selenium é uma ferramenta popular de automação de testes de *software*. Acompanhe as boas práticas que você deve seguir ao usar o Selenium.

- **Use diferentes navegadores**

  Teste seu aplicativo em diferentes navegadores para garantir que ele funcione em todos eles. O Selenium suporta vários navegadores, incluindo Chrome, Firefox, Safari e Internet Explorer.

- **Gerencie exceções**

  O Selenium pode gerar exceções durante a execução dos testes, como ElementNotVisibleException ou NoSuchElementException. Certifique-se de capturar e lidar com essas exceções adequadamente para tornar seus testes mais robustos.

Capte dados da web (crawling)

Selenium é uma ferramenta popular de automação de testes de *software*. Acompanhe as boas práticas que você deve seguir ao usar o Selenium.

escola britânica de artes criativas & tecnologia

- **Gerencie janelas e abas**

  Quando o Selenium abre uma nova janela ou guia, ele precisa mudar para essa janela para interagir com ela. Certifique-se de gerenciar adequadamente as janelas e guias para que seus testes funcionem corretamente.

- **Use a arquitetura Page Object**

  A arquitetura Page Object é um padrão comum para estruturar o código do Selenium. Ele separa a lógica de teste da lógica da página da web, o que pode tornar o código mais legível e manutenível.

Capte dados da web (crawling)

Scrapy é uma biblioteca Python usada para extrair dados da *web*. Acompanhe as boas práticas que você deve seguir ao usar o Scrapy.

- **Use políticas de rastreamento**

  Use políticas de rastreamento para evitar rastrear páginas em excesso ou para impedir que rastreie páginas sensíveis. O Scrapy possui recursos embutidos para definir limites de profundidade de rastreamento e ignorar certos URLs.

- **Use a arquitetura de projeto Scrapy**

  Use a arquitetura de projeto Scrapy, que separa o código de rastreamento, o pipeline e o armazenamento em arquivos separados. Isso torna o código mais legível e manutenível.

Capte dados da web (crawling)

Scrapy é uma biblioteca Python usada para extrair dados da *web*. Acompanhe as boas práticas que você deve seguir ao usar o Scrapy.

- **Use pipelines para armazenar dados**

  Use pipelines para armazenar dados extraídos em um banco de dados ou arquivo. Isso torna mais fácil para você analisar os dados posteriormente.

- **Use parâmetros de comando**

  Use parâmetros de comando para definir URLs de entrada, profundidade de rastreamento e outros parâmetros importantes. Isso torna mais fácil personalizar o rastreamento e reutilizar o código.

Capte dados da web (crawling)

Scrapy é uma biblioteca Python usada para extrair dados da *web*. Acompanhe as boas práticas que você deve seguir ao usar o Scrapy.

escola britânica de artes criativas & tecnologia

- **Use proxies**

  Use proxies para evitar ser detectado como um robô e para evitar bloqueios de IP. O Scrapy suporta proxies HTTP e SOCKS.

- **Gerencie exceções**

  O Scrapy pode gerar exceções durante a extração de dados, como ConnectionError ou TimeoutError. Certifique-se de capturar e lidar com essas exceções adequadamente para tornar o código mais robusto.

Capte dados da web (crawling)

Scrapy é uma biblioteca Python usada para extrair dados da *web*. Acompanhe as boas práticas que você deve seguir ao usar o Scrapy.

- **Use cabeçalhos falsos**

  Essa ação evita evitar ser detectado como um robô. O Scrapy pode definir cabeçalhos falsos automaticamente para você.

- **Use middlewares**

  Use middlewares para modificar as solicitações e respostas do Scrapy. Por exemplo, você pode adicionar um middleware para redirecionar todas as solicitações HTTP para HTTPS.

Capte dados da web (crawling)

Robots.txt é um arquivo usado para informar aos robôs de mecanismos de busca quais páginas ou seções de um *site* devem ou não ser rastreadas. Aqui estão algumas boas práticas que você deve seguir ao usar o arquivo robots.txt.

- **Coloque o arquivo robots.txt na raiz do seu site**

  O arquivo robots.txt deve ser colocado na raiz do seu *site* (por exemplo, https://www.seusite.com/robots.txt). Isso torna mais fácil para os mecanismos de busca encontrá-lo.

- **Use as diretrizes User-agente**

  As diretrizes User-agent são usadas para especificar quais robôs de mecanismos de busca o arquivo robots.txt se aplica. Use-as para definir restrições diferentes para diferentes robôs.

Capte dados da web (crawling)

Robots.txt é um arquivo usado para informar aos robôs de mecanismos de busca quais páginas ou seções de um *site* devem ou não ser rastreadas. Aqui estão algumas boas práticas que você deve seguir ao usar o arquivo robots.txt.

- **Use as diretrizes Disallow**
  As diretrizes Disallow são usadas para especificar quais páginas ou seções do seu site não devem ser rastreadas pelos robôs. Use-as para evitar que as páginas do seu site que contenham informações confidenciais ou irrelevantes para os mecanismos de busca sejam indexadas.

- **Use as diretrizes Allow**
  As diretrizes Allow são usadas para especificar quais páginas ou seções do seu site devem ser rastreadas pelos robôs. Use-as para permitir que as páginas relevantes do seu site sejam indexadas pelos mecanismos de busca.

Capte dados da web (crawling)

Robots.txt é um arquivo usado para informar aos robôs de mecanismos de busca quais páginas ou seções de um *site* devem ou não ser rastreadas. Aqui estão algumas boas práticas que você deve seguir ao usar o arquivo robots.txt.

- **Use o wildcard () com cautela**

  O wildcard () é usado para representar qualquer string de caracteres. Use-o com cautela, pois pode permitir que os robôs acessem partes do seu site que você não deseja que sejam indexadas.

- **Teste o arquivo robots.txt**

  Teste o arquivo robots.txt para garantir que ele esteja configurado corretamente. Use ferramentas como o Google Search Console para testar o arquivo robots.txt e verificar se ele permite que os mecanismos de busca acessem as páginas relevantes do seu site.

Capte dados da web (crawling)

Robots.txt é um arquivo usado para informar aos robôs de mecanismos de busca quais páginas ou seções de um *site* devem ou não ser rastreadas. Aqui estão algumas boas práticas que você deve seguir ao usar o arquivo robots.txt.

- **Mantenha o arquivo robots.txt atualizado**

  Mantenha o arquivo robots.txt atualizado conforme o seu *site* muda. Isso garantirá que os robôs de mecanismos de busca acessem apenas as partes relevantes do seu *site*.

DDOS, ou Distributed Denial of Service, é um tipo de ataque cibernético em que um grande número de dispositivos, normalmente computadores ou dispositivos da Internet das Coisas, são coordenados para enviar uma quantidade massiva de tráfego para um site ou servidor específico. O objetivo é sobrecarregar o site ou servidor, tornando-o inacessível para usuários legítimos. Acesse o artigo O que é DoS e DDoS? e saiba como se proteger dos ataques.

Extraia dados da web (scrapping)

Beautifulsoup4 é uma biblioteca Python usada para analisar documentos HTML. Usando-a, podemos navegar pelos dados HTML para extrair/deletar/substituir elementos HTML específicos. Ela também vem com funções utilitárias como formatação visual e limpeza da árvore de análise. Acompanhe as dicas.

- **Escolher um backend**
  Escolha um backend adequado para a análise HTML, como lxml ou html5lib12. O backend padrão é o parser HTML do Python, que pode não ser capaz de lidar com HTML malformado ou incompleto.

- **Usar o método find ou find_all**
  Usar o método find ou find_all para localizar elementos HTML por *tag*, atributo, classe ou texto. Esses métodos retornam objetos BeautifulSoup que podem ser manipulados facilmente.

Extraia dados da web (scrapping)

Beautifulsoup4 é uma biblioteca Python usada para analisar documentos HTML. Usando-a, podemos navegar pelos dados HTML para extrair/deletar/substituir elementos HTML específicos. Ela também vem com funções utilitárias como formatação visual e limpeza da árvore de análise. Acompanhe as dicas.

- **Usar o método get_text**
  Use o método get_text para extrair o texto de um elemento HTML sem as tags. Esse método também pode remover espaços em branco extras e quebras de linha.

- **Usar o método prettify**
  Use o método prettify para formatar visualmente a saída HTML e facilitar a depuração. Esse método adiciona recuos e novas linhas ao código HTML para torná-lo mais legível.

Para aprofundar seus conhecimentos, acesse Beautiful Soup 4 Python - PythonForBeginners.com
Encontre a documentação no *link* https://www.crummy.com/software/BeautifulSoup/bs4/doc/

escola britânica de artes criativas & tecnologia

Bons estudos!