



Imersão em GenAI para Desenvolvedores

Projeto 2: Explorando Python e APIs Web

Preâmbulo: Nesse projeto você vai configurar o seu ambiente de desenvolvimento Python, criar funções úteis, interagir com APIs e explorar conceitos web.

Versão: 1.0

Sumário

I	Preâmbulo	2
II	Instruções gerais	4
III	Ex00 - O ambiente molda você	6
IV	Ex01 - Barras de progresso são mais do que parecem	8
V	Ex02 - Data is the new oil	10
VI	Ex03 - Todos os caminhos levam à filosofia	12
VII	Entrega e Avaliação entre pares	15
VII.1	Processo de Entrega	15
VII.2	Avaliação entre pares	15
VII.3	Dicas para uma avaliação bem-sucedida	15

Capítulo I

Preâmbulo

O Zen do Python, criado por Tim Peters, é um conjunto de 19 princípios orientadores que capturam a essência da filosofia de programação em Python.

Estes princípios, acessíveis digitando `import this` em um interpretador Python, oferecem uma visão concisa e poética sobre como escrever código Python de qualidade.

Aforismos notáveis:

"Bonito é melhor que feio"
"Explícito é melhor que implícito"
"Simples é melhor que complexo"
"Complexo é melhor que complicado"
"Legibilidade conta"

Impacto e significado:

1. **Influência no Design:** Molda a própria linguagem Python
2. **Guia Ético e Estético:** Orienta desenvolvedores na escrita de código
3. **Promoção de Qualidade:** Encoraja código funcional, elegante e claro

O Zen do Python incentiva os desenvolvedores a valorizar:

- Simplicidade
- Clareza
- Praticidade
- Eficiência
- Sustentabilidade

- Compreensibilidade a longo prazo

Ao seguir o Zen do Python, os desenvolvedores são incentivados a criar soluções que são não apenas eficientes, mas também sustentáveis e compreensíveis a longo prazo. Isso é particularmente relevante no contexto de projetos de grande escala e de longa duração, onde a manutenibilidade e a clareza do código são cruciais.

— *PEP 20 - The Zen of Python*¹

¹<https://peps.python.org/pep-0020/>

Capítulo II


Instruções gerais

- Esta página é sua única referência oficial. Não confie em informações não verificadas.
- Os exercícios estão organizados em ordem crescente de complexidade. É essencial dominar cada exercício antes de prosseguir para o próximo.
- Preste atenção às permissões de seus arquivos e pastas.
- Siga rigorosamente o procedimento de entrega para todos os exercícios.
- Seus exercícios serão avaliados por colegas da Imersão.
- Para exercícios em Shell, utilize `/bin/zsh`.
- Mantenha em sua pasta apenas os arquivos explicitamente solicitados nos enunciados.
- Em caso de dúvidas, consulte seus colegas à direita ou à esquerda.
- Utilize recursos como Google, manuais online e a Internet como referência.
- Leia os exemplos com atenção. Eles podem conter requisitos não explicitamente mencionados no enunciado.
- Para exercícios em Python:
 - Use a versão do Python especificada no exercício de configuração do ambiente.
 - Utilize os modelos e provedores sugeridos para garantir tempos de resposta adequados e consistência nos testes.
- Esteja atento a erros em todos os exercícios. Eles raramente são tolerados durante a avaliação.
- **Aviso sobre o uso de ferramentas de AI (como ChatGPT):**
 - O uso de ferramentas como o ChatGPT não deve ser encarado como um substituto para seu próprio esforço e entendimento.
 - O aprendizado efetivo ocorre quando você interage ativamente com o conteúdo: pesquisando, refletindo e aplicando o que aprendeu.
 - Nas avaliações, serão feitas perguntas para avaliar sua compreensão real sobre o assunto.

- E durante as avaliações, seus colegas também avaliarão seu nível de conhecimento.

Capítulo III

Ex00 - O ambiente molda você

	Exercício : 00
Configurar um ambiente de desenvolvimento Python usando venv e praticar comandos de gerenciamento de pacotes	
Pasta de entrega : <i>ex00/</i>	
Arquivos para entregar : <i>answers.txt, requirements.txt</i>	

Você vai configurar um ambiente de desenvolvimento sólido. Preparando o terreno para o seu trabalho com futuras bibliotecas e frameworks.

Instruções

1. Verifique a versão do Python instalada:

```
$ python3.10 --version
```

Certifique-se de que está usando Python 3.10.14.

2. Crie um ambiente virtual:

```
$ python3.10 -m venv myenv
```

3. Ative o ambiente virtual:

```
$ source myenv/bin/activate # Em Unix ou MacOS  
$ myenv\Scripts\activate # No Windows
```

4. Verifique se o ambiente virtual está ativado:

```
$ which python  
$ python --version  
$ python -c "print('Hello world!')"
```

5. Responda às seguintes perguntas, salvando as respostas no arquivo *answers.txt*:

- (a) Qual é o comando para listar os pacotes instalados e suas versões?
- (b) Como mostrar os metadados do pacote numpy?
- (c) Qual comando remove o pacote numpy?
- (d) Como criar um arquivo requirements.txt com os pacotes instalados e suas versões?

6. Execute os comandos necessários para responder às perguntas acima.

7. Crie um arquivo requirements.txt com os pacotes instalados e suas versões.




Use o comando `'pip'` para gerenciar pacotes no ambiente virtual.



Pesquise sobre os comandos `'pip list'`, `'pip show'`, `'pip uninstall'`, `'pip install'`, e `'pip freeze'`.

Capítulo IV

Ex01 - Barras de progresso são mais do que parecem

	Exercício : 01
Criar uma função <code>ft_progress(lst)</code> que exibe uma barra de progresso para um loop for	
Pasta de entrega : <code>ex01/</code>	
Arquivos para entregar : <code>loading.py</code>	

Com seu ambiente agora ajustado, você vai criar uma barra de progresso personalizada em Python. Você vai reforçar sua fluência em Python.

Instruções

1. Implemente a função `ft_progress(lst)` que aceita um iterável como argumento.
2. A função deve atuar como um gerador, usando o operador `'yield'`.
3. A barra de progresso deve exibir:
 - ETA (Tempo estimado para conclusão)
 - Porcentagem concluída
 - Barra visual do progresso
 - Número atual/total de iterações
 - Tempo decorrido

Como o seu programa será chamado:

```
a_list = range(1000)
ret = 0
for elem in ft_progress(a_list):
    ret += (elem + 3) % 5
    sleep(0.01)
print()
print(ret)
```

Saída esperada:

```
ETA: 8.67s [ 23%][====>    ] 233/1000 | elapsed time 2.33s
...
2000
```



O operador 'yield' é usado para criar geradores. Ele "pausa" a função e retorna um valor, permitindo que a função seja retomada do mesmo ponto na próxima iteração.



Para calcular o ETA, considere o tempo decorrido e a proporção de itens processados.




Use '\r' para sobrescrever a linha atual no terminal, criando a ilusão de uma barra de progresso em movimento.



A função `time.time()` pode ser útil para medir o tempo decorrido.

Capítulo V

Ex02 - Data is the new oil

	Exercício : 02
Criar um programa que faça requisição à API da Wikipédia, processe os dados recebidos e salve o resultado em um arquivo	
Pasta de entrega : <i>ex02/</i>	
Arquivos para entregar : <code>request_wikipedia.py</code> , <code>requirements.txt</code>	

Para construir uma base de dados de qualidade, você vai criar um script para interagir com a API da Wikipédia, extraíndo e organizando informações que formarão um conjunto de dados valioso.

Instruções

1. Crie um programa chamado `request_wikipedia.py` que:
 - Aceita uma string como parâmetro de linha de comando.
 - Faz uma busca na API da Wikipédia (em português ou inglês).
 - Processa o resultado removendo formatação JSON ou Wiki Markup.
 - Salva o resultado em um arquivo com o nome "`nome_da_busca.wiki`".
2. Requisitos do programa:
 - Deve retornar um resultado mesmo para buscas com erro ortográfico (como o site original).
 - O nome do arquivo não deve conter espaços.
 - Em caso de erro (parâmetro ausente, requisição inválida, etc.), nenhum arquivo deve ser criado e uma mensagem de erro relevante deve ser exibida no console.
3. Crie um arquivo `requirements.txt` com as dependências necessárias.

Exemplo de uso:

```
$ python request_wikipedia.py "barra de chocolate"
$ cat barra_de_chocolate.wiki
Uma barra de chocolate...
...
```



Use a biblioteca 'requests' para fazer a requisição HTTP à API da Wikipedia.



A biblioteca 'json' será útil para processar a resposta da API.



Use 'dewiki' para remover a formatação Wiki do texto recebido.




Considere criar funções separadas para cada etapa do processo (requisição, processamento, escrita no arquivo).



Lembre-se de tratar possíveis exceções durante a requisição e processamento dos dados.

Capítulo VI

Ex03 - Todos os caminhos levam à filosofia

	Exercício : 03
Criar um programa que simule o fenômeno "roads to philosophy" da Wikipedia, rastreando links de artigo em artigo até chegar à página sobre filosofia	
Pasta de entrega : <i>ex03/</i>	
Arquivos para entregar : <code>roads_to_philosophy.py</code> , <code>requirements.txt</code>	

Você vai criar um programa que navega pela Wikipedia de forma autônoma, seguindo os links dos artigos até chegar, inevitavelmente, à página "Filosofia".

Instruções

1. Crie um programa chamado `roads_to_philosophy.py` que:
 - Aceita uma string como parâmetro (palavra ou grupo de palavras para busca na Wikipedia).
 - Faz uma requisição à página da Wikipédia em inglês correspondente à busca.
 - Analisa o HTML da página usando BeautifulSoup para encontrar o próximo link válido.
 - Repete o processo até chegar à página de Filosofia, um beco sem saída, ou um loop infinito.
2. Requisitos do programa:
 - Deve usar a URL padrão da Wikipedia (<https://en.wikipedia.org/wiki/>), não a API.
 - Deve lidar com redirecionamentos de página (não de URL).

- Deve ignorar links em itálico, entre parênteses, ou que não levem a novos artigos.
- Deve exibir cada título de artigo visitado.
- Ao final, deve exibir o número total de artigos visitados ou uma mensagem apropriada.

3. Tratamento de casos especiais:

- Sucesso: Ao chegar à página de Filosofia, exiba:

`<número> roads from <busca inicial> to philosophy!`

- Beco sem saída: Se não houver links válidos na página, exiba:

`It leads to a dead end!`

- Loop infinito: Se um artigo já visitado for encontrado novamente, exiba:

`It leads to an infinite loop!`

- Erro: Para qualquer erro (conexão, servidor, página não encontrada), exiba uma mensagem de erro detalhada, por exemplo:

`Error: <descrição do erro>. Unable to continue.`

4. Crie um arquivo requirements.txt com as dependências necessárias.

Exemplo de uso:

```
$ python roads_to_philosophy.py "42 (number)"
42 (number)
Natural number
Mathematics
...
Philosophy
17 roads from 42 (number) to philosophy!
```



Use 'requests' para fazer as requisições HTTP.



Use BeautifulSoup para analisar o HTML e encontrar os links relevantes.



Considere criar funções separadas para cada etapa do processo (requisição, parsing, busca de links).



Mantenha uma lista de páginas visitadas para detectar loops.



Use expressões regulares ou métodos de string para limpar os títulos dos artigos.



Ética sobre web scraping: Web scraping pode ser uma ferramenta poderosa, mas deve ser usada com responsabilidade. Sempre verifique os termos de serviço do site e respeite as diretrizes do robots.txt. Evite fazer requisições excessivas que possam sobrecarregar os servidores.

Capítulo VII

Entrega e Avaliação entre pares

VII.1 Processo de Entrega

- Submeta seu trabalho no repositório Git gerado na página principal do projeto.
- Certifique-se de que todos os arquivos necessários estejam incluídos e organizados conforme as instruções do projeto.
- Respeite o prazo de entrega estabelecido.

VII.2 Avaliação entre pares

- Seu projeto será avaliado por um dos seus colegas.
- A avaliação focará na qualidade do seu código e na aderência aos requisitos do projeto.
- Critérios de avaliação podem incluir:
 1. Funcionalidade: O código atende a todos os requisitos especificados?
 2. Legibilidade: O código é claro e bem estruturado?
 3. Eficiência: As soluções implementadas são otimizadas e seguem boas práticas?
 4. Organização: Os arquivos e estrutura do projeto estão bem organizados?
- Feedback detalhado é esperado, mas pode variar em extensão e detalhamento.

VII.3 Dicas para uma avaliação bem-sucedida

- Revise seu código antes da submissão final.

- Teste exaustivamente todas as funcionalidades implementadas.
- Se entender necessário para clareza, documente claramente qualquer decisão ou suposição feita.
- Esteja preparado para explicar suas escolhas de implementação.



A avaliação entre pares é uma oportunidade para aprendizado e crescimento pessoal e profissional. Esteja aberto ao feedback recebido e use-o para aprimorar suas habilidades.