



Quem se prepara, não para.

Minicurso: Criando Aplicações com o ChatGPT em Python

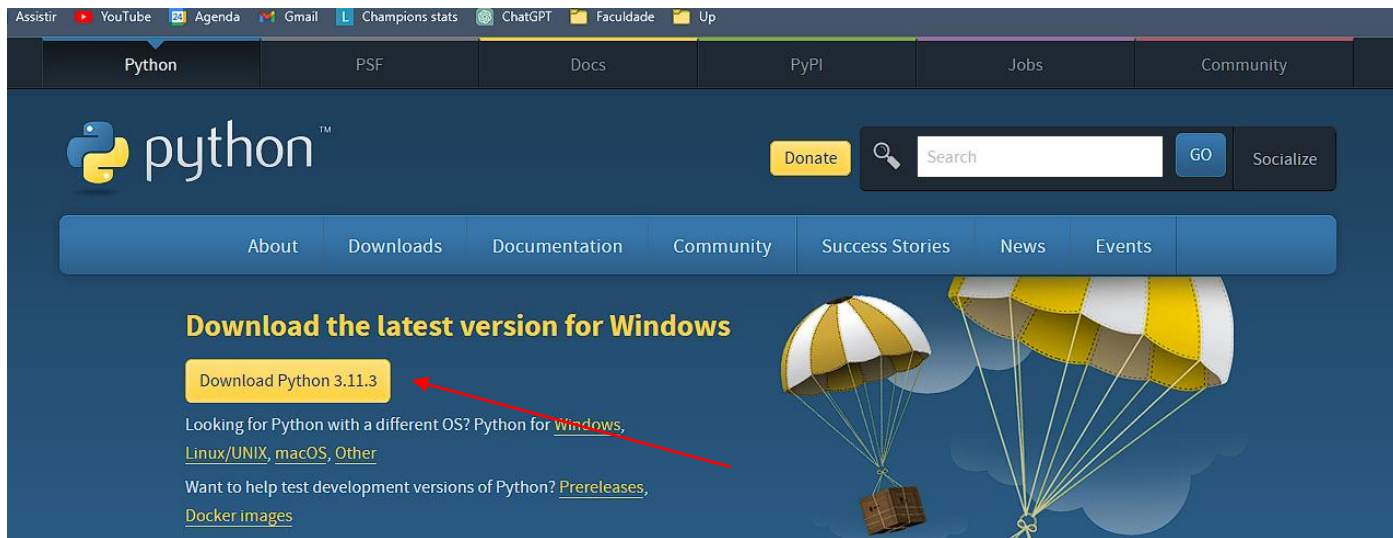
Paulo Ricardo
Marianna Rossi

Sumário

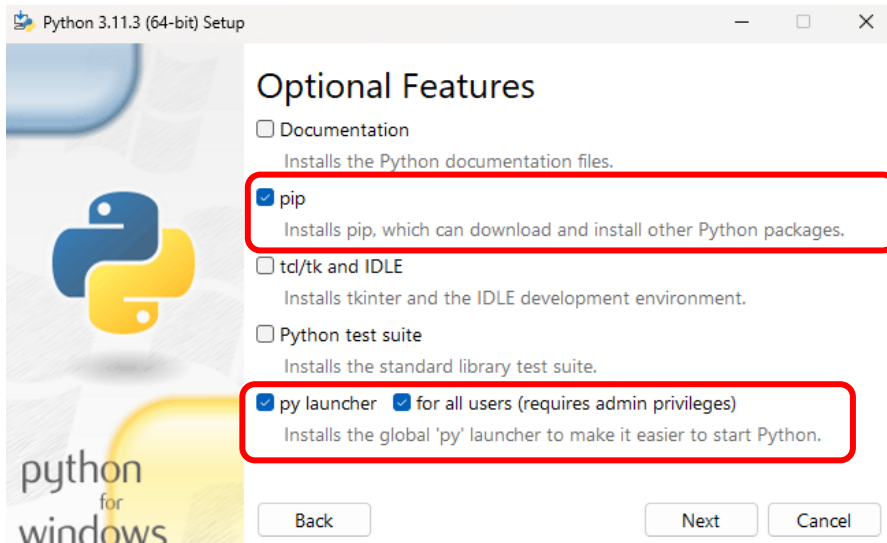
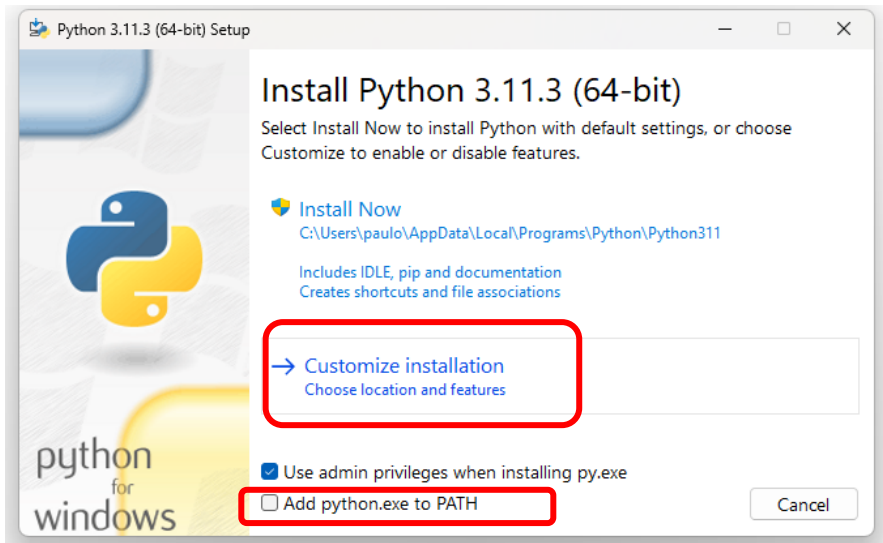
- Apresentação Geral
- Instalando o Python
- Criando a Key no ChatGPT
- Tutorial 1 – *Arquivo .env e acesso a chave*
- Tutorial 2 – *Prompt simples*
- Tutorial 3 – Tradutor
- Tutorial 4 – Comando Shell

Instalando o Python 3.11

Acessar o site: <https://www.python.org/downloads/>

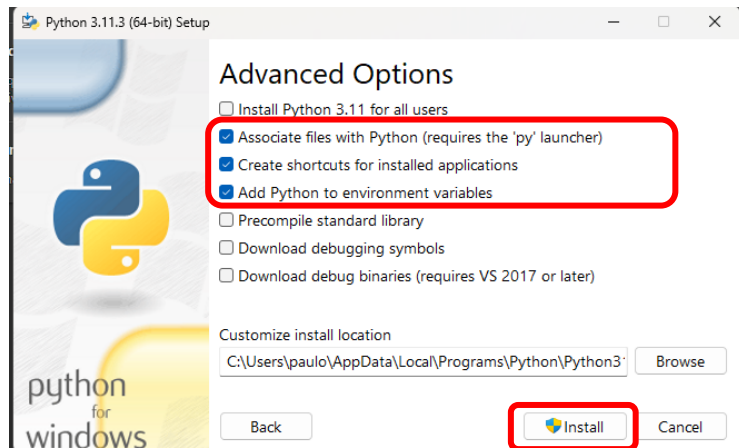


Instalando o Python 3.11



Seguir o passo a passo

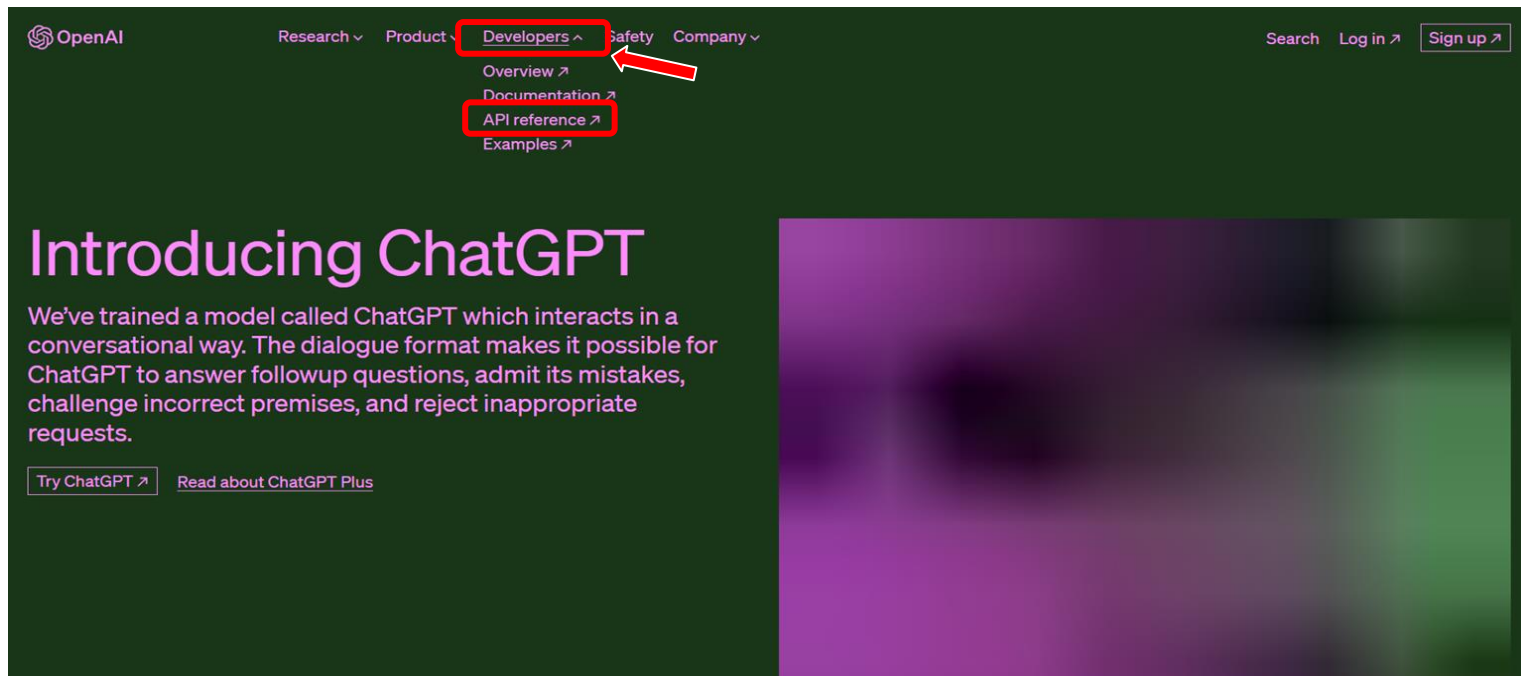
Instalando o Python 3.11



Seguir o passo a passo

Criando a Key - chatGPT

Link: <https://openai.com/blog/chatgpt>

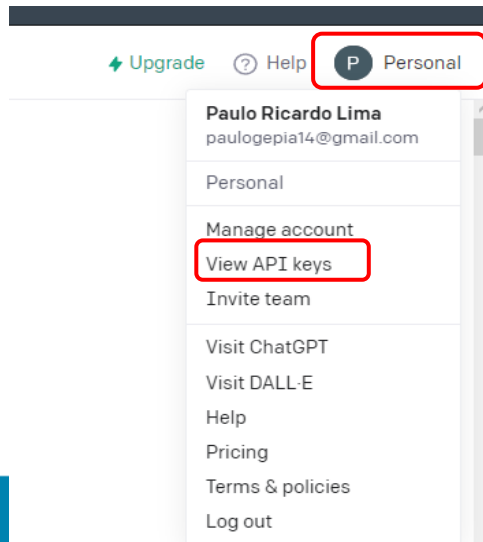


Criando a Key - chatGPT

Passos:

- Criar uma conta ou fazer login (vai precisar de e-mail e telefone)
- Dar um nome para identificar sua KEY
- **CÓPIA SUA KEY E SALVA!!!!!!!**

1)



2)

API keys

Your secret API keys are listed below. Please note that we do not display your secret API keys again after you generate them.

Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may also automatically rotate any API key that we've found has leaked publicly.

NAME	KEY	CREATED	LAST USED
fakenews	sk-...34VI	4 de mai. de 2023	4 de mai. de 2023

[+ Create new secret key](#)

Default organization

If you belong to multiple organizations, this setting controls which organization is used by default when making requests with the API keys above.

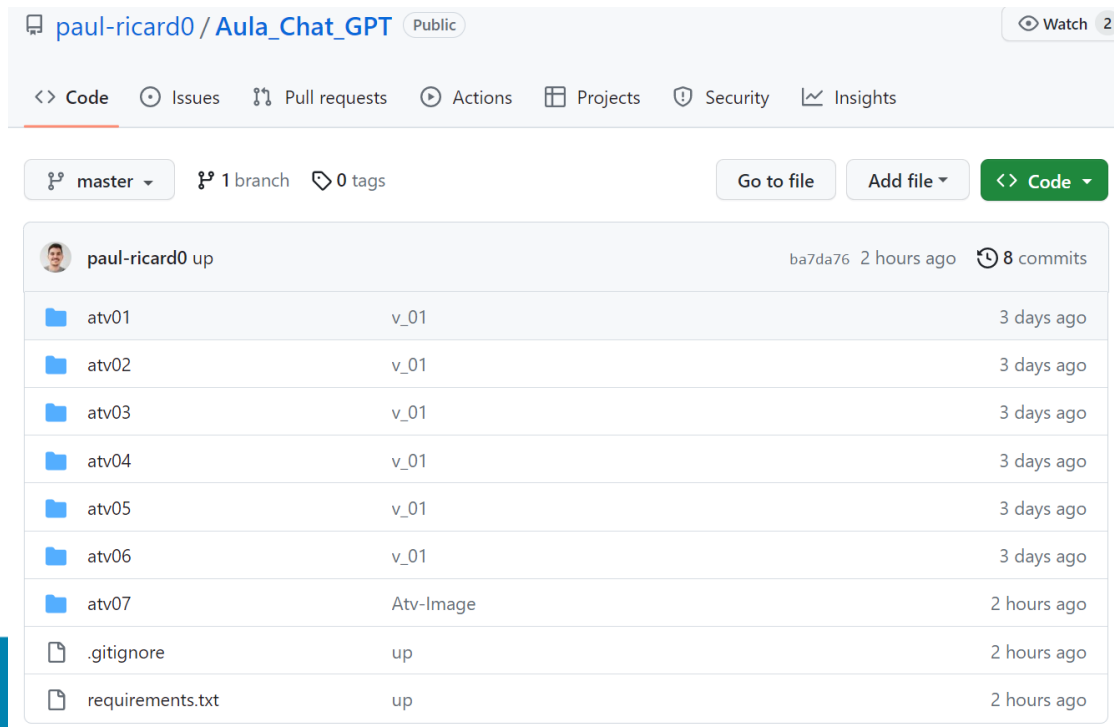
Personal

Note: You can also specify which organization to use for each API request. See [Authentication](#) to learn more.

GitHub dos Tutoriais

Clonar ou fazer o download repositório:

https://github.com/paul-ricard0/Aula_Chate_GPT.git



Repository: paul-ricard0 / Aula_Chate_GPT (Public) Watch 2

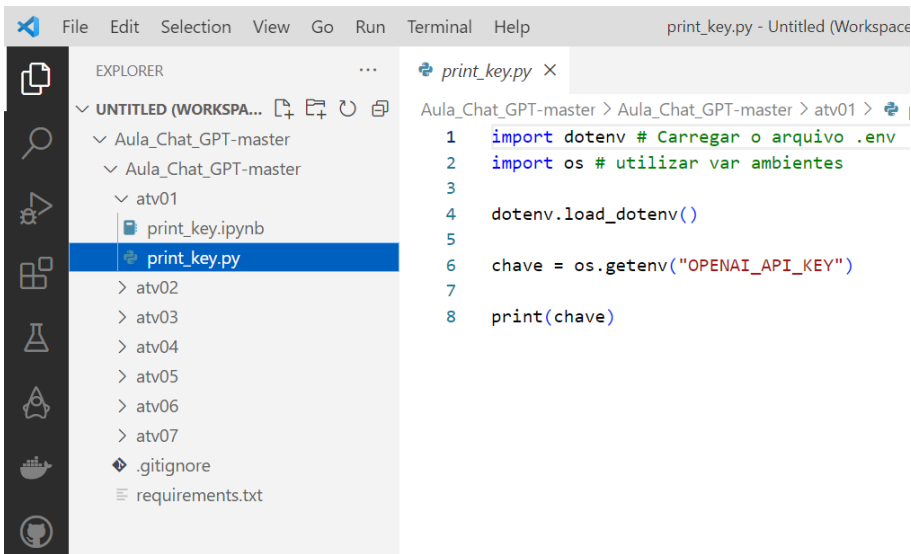
<> Code Issues Pull requests Actions Projects Security Insights

master 1 branch 0 tags Go to file Add file <> Code

ba7da76 2 hours ago 8 commits

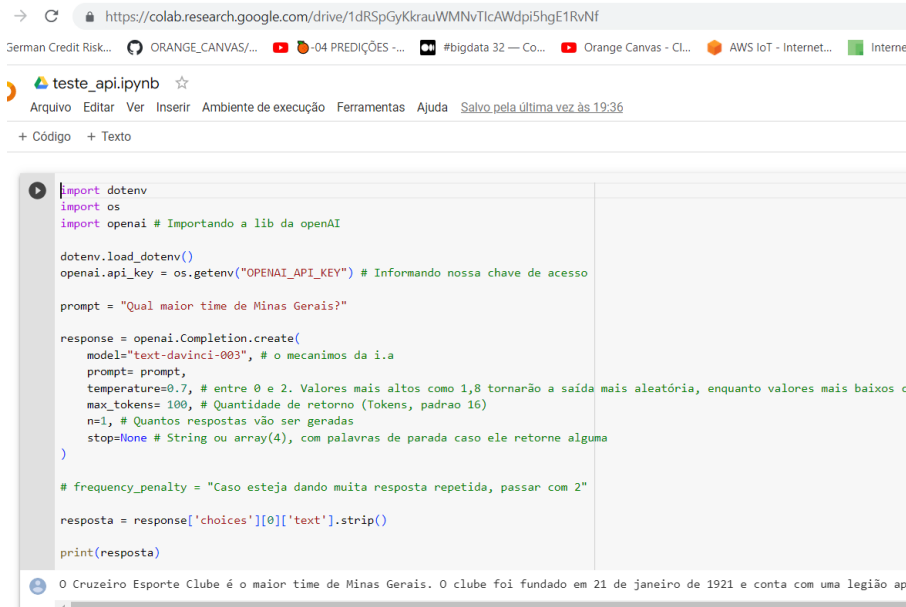
atv01	v_01	3 days ago
atv02	v_01	3 days ago
atv03	v_01	3 days ago
atv04	v_01	3 days ago
atv05	v_01	3 days ago
atv06	v_01	3 days ago
atv07	Atv-Image	2 hours ago
.gitignore	up	2 hours ago
requirements.txt	up	2 hours ago

Abrindo os Arquivos



VS CODE

File-> Adicionar Folder To Workspace
<https://code.visualstudio.com/download>



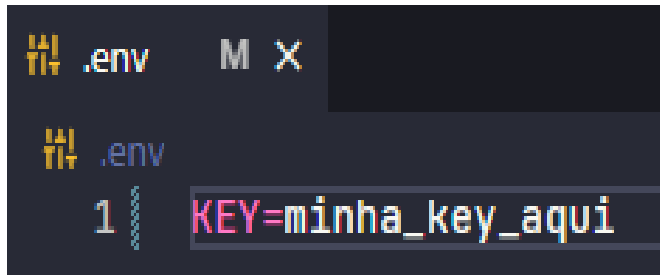
Google Colaboratory

Subir a pasta para o Google DRIVE
<https://colab.research.google.com/>

Tutorial 1 – atv01

Organização do projeto:

- Arquivo **.env** – será utilizado para gravar a *key*
- Arquivo **.gitignore** usado para adicionar a exceção do arquivo **.env** quando subir para o GitHub.



A screenshot of a code editor with a dark theme. The top part shows a file explorer with a folder icon and the file name `.env`, with status indicators 'M' and 'X'. Below, the file is open, showing a single line of code: `KEY=minha_key_aqui`. The line is numbered '1' on the left margin.



A screenshot of a code editor with a dark theme. The top part shows a file explorer with a folder icon and the file name `.gitignore`. Below, the file is open, showing a single line of code: `.env`. The line is numbered '1' on the left margin.

Tutorial 1 – atv01

Instalando as dependências:

Executar o comando: *pip install requirements.txt*

Abrir o arquivo `print_key.py`

```
print_key.py 7 m
import dotenv # Carregar o arquivo .env
import os # utilizar var ambientes

dotenv.load_dotenv()

chave = os.getenv("KEY")

print(chave)
```

Este exemplo carrega a chave do arquivo .env e imprime para conferência.

Tutorial 2 – atv02

```
1  import dotenv
2  import os
3  import openai # Importando a lib da openAI
4
5  dotenv.load_dotenv()
6  openai.api_key = os.getenv("OPENAI_API_KEY") # Informando nossa chave de acesso
7
8  prompt = "Qual maior time de Minas Gerais?"
9
10 response = openai.Completion.create(
11     model="text-davinci-003", # o mecanimos da i.a
12     prompt=prompt,
13     temperature=0.7, # entre 0 e 2. Valores mais altos como 1,8 tornarão a saída mais aleatória, enquanto v
14     max_tokens= 100, # Quantidade de retorno (Tokens, padrao 16)
15     n=1, # Quantos respostas vão ser geradas
16     stop=None # String ou array(4), com palavras de parada caso ele retorne alguma
17 )
18
19 # frequency_penalty = "Caso esteja dando muita resposta repetida, passar com 2"
20
21 resposta = response['choices'][0]['text'].strip()
22
23 print(resposta)
```

*Exemplo simples de prompt
usando o modelo text-davinci-003*

Tutorial 3 – atv03

```
1 import os, dotenv
2 import openai
3
4 dotenv.load_dotenv()
5 openai.api_key = os.getenv("OPENAI_API_KEY")
6
7 def read_file(file_path):
8     with open(file_path, 'r', encoding='utf-8') as f:
9         return f.read()
10
11 def write_file(text, file_path):
12     with open(file_path, 'w', encoding='utf-8') as f:
13         f.write(text)
14
15 def translate(text: str, source_lang: str, target_lang: str) -> str:
16     prompt = f"""
17     Traduza esse texto em {source_lang} para {target_lang}: {text}
18     """
19     response = openai.Completion.create(
20         model="text-curie-001", # Melhor para traduzir texto
21         prompt=prompt,
22         temperature=0.7,
23         max_tokens= 1900,
24         n=1,
25         stop=None
26     )
27     return response['choices'][0]['text'].strip()
28
29 if __name__ == '__main__':
30     idioma_original = 'Inglês'
```

Exemplo tradutor, carrega um arquivo e realiza a tradução de inglês para português

Tutorial 4 – atv04

```
1 import os, dotenv
2 import openai
3 import subprocess
4
5 dotenv.load_dotenv()
6 openai.api_key = os.getenv("OPENAI_API_KEY")
7
8 def gerar_comando(texto: str) -> str:
9     prompt = f"""
10     Me retorne a penas o comando cmd que faça o seguinte: {texto}
11     """
12     response = openai.Completion.create(
13         model="text-davinci-003",
14         prompt=prompt,
15         temperature=0.7,
16         max_tokens=1000,
17         n=1,
18         stop=None
19     )
20     return response['choices'][0]['text'].strip()
21
22 def __executar_comando_shell(comando: str) -> None:
23     try:
24         resultado = subprocess.run(comando, shell=True, check=True)
25         print(resultado)
26     except subprocess.CalledProcessError as e:
27         print(e)
28
29 def validar_comando(comando: str) -> str:
30     print(f'Comando gerado: {comando}')
```

Exemplo shell, traduz comandos de uma linguagem natural para um comando Shell

Sumário

- Tutorial 5 – Manipulando Arquivos (tradução)
- Tutorial 6 – Análise de Sentimentos do Twitter
- Tutorial 7 - Trabalhando com Imagens