

Webscraping de metadados de um video de 5 minutos
<https://www.youtube.com/watch?v=JAMloNO0glk>

```
import json
from yt_dlp import YoutubeDL

def extrair_metadados_5min(url_video):
    ydl_opts = {
        "quiet": True,
        "skip_download": True, # Apenas extrai os metadados, sem baixar o arquivo de mídia
    }

    try:
        print(f"Analisando os streams de dados do YouTube para: {url_video}...\n")

        with YoutubeDL(ydl_opts) as ydl:
            # Faz o download das informações estruturadas
            info_dict = ydl.extract_info(url_video, download=False)

            # Estrutura os dados principais para exibição limpa
            metadados_filtrados = {
                "Título": info_dict.get("title"),
                "Canal": info_dict.get("uploader"),
                "Duração": f"{info_dict.get('duration')} segundos (aprox. {info_dict.get('duration') // 60} minutos)",
                "Visualizações": info_dict.get("view_count"),
                "Curtidas": info_dict.get("like_count"),
                "Data de Lançamento": info_dict.get("upload_date"),
                "Resolução Máxima": f"{info_dict.get('width')}x{info_dict.get('height')}",
                "FPS": info_dict.get("fps"),
                "Total de Formatos de Transmissão Otimizados": len(
                    info_dict.get("formats", [])
                ),
            }

        print("=" * 60)
        print("📺 RESULTADOS DA EXTRAÇÃO:")
        print("=" * 60)
        for chave, valor in metadados_filtrados.items():
            print(f"💎 {chave}: {valor}")
        print("=" * 60)

        # Salva o arquivo JSON na memória local do Colab
```

```

nome_arquivo = f"metadados_{info_dict.get('id')}.json"
with open(nome_arquivo, "w", encoding="utf-8") as f:
    json.dump(info_dict, f, ensure_ascii=False, indent=4)

print(f"\n📁 Arquivo completo gerado com sucesso: {nome_arquivo}")
print(
    "💡 Dica: Abra a pasta na barra lateral esquerda do Colab para baixá-lo."
)

```

```

except Exception as e:
    print(f"❌ Erro ao ler o stream de dados: {e}")

```

```

# Execução do teste de 5 minutos
url_exemplo = "https://www.youtube.com/watch?v=JAMloNO0gIk"
extrair_metadados_5min(url_exemplo)

```

=====

Webscraping de metadados do seguinte vídeo de 10 minutos

```
url_exemplo = "https://www.youtube.com/watch?v=x-YqdUJ4bxE"
```

```

# 1. Garante que a biblioteca está atualizada no ambiente do Colab
!pip install -q --upgrade yt-dlp

import json
from yt_dlp import YoutubeDL

def extrair_metadados_receita(url_video):
    # Configurações com User-Agent para simular um navegador comum
    ydl_opts = {
        "quiet": True,
        "skip_download": True, # Apenas lê o manifesto de dados, sem
        # baixar o vídeo
        "http_headers": {
            "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
            AppleWebKit/537.36"
        },
    }

```

```

try:
    print(f"Analisando os streams do vídeo de receita:
{url_video}...\n")

    with YoutubeDL(ydl_opts) as ydl:
        info_dict = ydl.extract_info(url_video, download=False)

        # Organizando os dados de forma limpa para exibição
        metadados_filtrados = {
            "Título da Receita": info_dict.get("title"),
            "Canal": info_dict.get("uploader"),
            "Duração (segundos)": info_dict.get("duration"),
            "Visualizações": info_dict.get("view_count"),
            "Curtidas": info_dict.get("like_count"),
            "Data de Publicação": info_dict.get("upload_date"),
            "Resolução Máxima":
f"{info_dict.get('width')}x{info_dict.get('height')}",
            "FPS": info_dict.get("fps"),
        }

        print("=" * 60)
        print("📊 METADADOS DA RECEITA EXTRAÍDOS:")
        print("=" * 60)
        for chave, valor in metadados_filtrados.items():
            print(f"👁️ {chave}: {valor}")
        print("=" * 60)

        # Salva o arquivo JSON na barra lateral do Colab
        nome_arquivo = f"metadados_receita_{info_dict.get('id')}.json"
        with open(nome_arquivo, "w", encoding="utf-8") as f:
            json.dump(info_dict, f, ensure_ascii=False, indent=4)

        print(f"\n💾 Arquivo completo salvo com sucesso como:
{nome_arquivo}")
        print(
            "💡 Dica: Clique no ícone de 'Pasta' na barra lateral
esquerda para ver o JSON!"
        )

except Exception as e:

```

```

        print(
            f"❌ Erro: Ocorreu uma falha na raspagem de dados.\nDetalhes
técnicos: {e}"
        )

# Executando o teste com o vídeo de receita de 10 minutos
url_receita = "https://www.youtube.com/watch?v=x-YqdUJ4bxE"
extrair_metadados_receita(url_receita)

```

Extraíndo a fala em texto do mesmo vídeo

Método de Extrair a Legenda Existente (Rápido e Leve)

Se o vídeo da receita já tiver legendas (sejam criadas pelo dono do canal ou geradas automaticamente pelo algoritmo do YouTube), podemos usar a biblioteca

`youtube_transcript_api`.

Este método é **instantâneo** porque ele não precisa "ouvir" o áudio; ele apenas faz o download do texto que já está pronto nos servidores do YouTube.

```

# 1. Garante a instalação da biblioteca no ambiente do Colab
!pip install -q --upgrade youtube-transcript-api

from youtube_transcript_api import YouTubeTranscriptApi

# ID fixo do vídeo de receitas do Adilson
ID_DO_VIDEO = "x-YqdUJ4bxE"

def extrair_transcricao_correta_pypi():
    print(f"🤖 [Processo Iniciado] Utilizando ID fixo: {ID_DO_VIDEO}")
    print(
        "⌚ Conectando diretamente ao stream de texto dos servidores do
YouTube...\n"
    )

    try:
        # Instancia a API e puxa a lista de legendas

```

```

lista_de_legendas = YouTubeTranscriptApi().list(ID_DO_VIDEO)

# Seleciona a faixa em português
legenda = lista_de_legendas.find_transcript(["pt"])

# SOLUÇÃO DO BUG: Baixa os dados e converte para dicionários
brutos usando .to_raw_data()
# Isso corrige o erro "'FetchedTranscriptSnippet' object is not
subscriptable"
blocos_texto = legenda.fetch().to_raw_data()

texto_completo = ""

print("=" * 65)
print("📺 TEXTO EXTRAÍDO DA FALA EM TEMPO REAL (LINHA POR
LINHA):")
print("=" * 65)

for bloco in blocos_texto:
    # Agora que convertemos para dicionário bruto, o uso de
colchetes funciona perfeitamente!
    tempo = bloco["start"]
    minutos = int(tempo // 60)
    segundos = int(tempo % 60)
    fala = bloco["text"]

    # Imprime cada frase sincronizada com o tempo do vídeo no seu
terminal
    print(f"🕒 [{minutos:02d}:{segundos:02d}] ➡ {fala}")
    texto_completo += f" {fala}"

print("=" * 65)
print("\n📖 TEXTO INTEGRAL DA RECEITA CAPTURADO:")
print("=" * 65)
print(texto_completo.strip())
print("=" * 65)

# Salva o arquivo final (.txt) na barra lateral do seu Colab
with open("transcricao_sucesso.txt", "w", encoding="utf-8") as f:
    f.write(texto_completo.strip())

```

```
        print(
            "\n📄 Sucesso absoluto! O arquivo 'transcricao_sucesso.txt'
foi gerado na barra lateral."
        )

    except Exception as e:
        print(
            f"❌ Falha técnica ao obter o stream de texto do
YouTube.\nDetalhes do erro: {e}"
        )

# Executa a função imediatamente
extrair_transcricao_correta_pypi()
```