

Este notebook contém instruções para carregar e salvar dados de fontes externas.

## ▼ Sistema de arquivos local

### ▼ Fazer upload de arquivos a partir de sistema de arquivos local

`files.upload` retorna um dicionário com os arquivos incluídos no upload. O dicionário é indexado com o nome do arquivo e os valores são os dados enviados por upload.

```
from google.colab import files

uploaded = files.upload()

for fn in uploaded.keys():
    print('User uploaded file "{name}" with length {length} bytes'.format(
        name=fn, length=len(uploaded[fn])))
```

### ▼ Fazer download de arquivos para o sistema de arquivos local

`files.download` invocará o download do arquivo pelo navegador para seu computador local.

```
from google.colab import files

with open('example.txt', 'w') as f:
    f.write('some content')

files.download('example.txt')
```

## ▼ Google Drive

É possível acessar os arquivos no Drive de várias maneiras. Por exemplo:

- Montar o Google Drive na máquina virtual do ambiente de execução
- Usar um wrapper ao redor da API, [como o PyDrive](#)
- Usar a [API REST nativa](#)

Veja abaixo exemplos de cada um deles.

### ▼ Montar o Google Drive localmente

O exemplo abaixo mostra como montar o Google Drive no seu ambiente de execução usando um código de autorização, além de como gravar e ler arquivos nele. Depois de executado, você verá o novo arquivo (foo.txt) no <https://drive.google.com/>.

Isto permite somente ler, gravar e mover arquivos. Para modificar de maneira programática as configurações de compartilhamento ou outros metadados, use uma das opções abaixo.

**Observação:** ao usar o botão "Montar Drive" no navegador de arquivos, não é necessário usar nenhum código de autenticação para notebooks que tenham sido editados somente pelo usuário atual.

```
from google.colab import drive
drive.mount('/content/drive')

Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client\_id=947318989803-6bn6qk8qdf4n4g3nfee6491hc0brc4i.apps.googleusercontent.com&redirect\_uri=urn%3Aietf%3Aparameters%3Aoauth%3Aopen-id%3Aconnect%3Aprofile&scope=drive%20readonly%20files%20download%20upload%20offline
Enter your authorization code:
.....
Mounted at /content/drive
```

```
with open('/content/drive/My Drive/foo.txt', 'w') as f:
    f.write('Hello Google Drive!')
!cat /content/drive/My Drive/foo.txt

Hello Google Drive!
```

```
drive.flush_and_unmount()
print('All changes made in this colab session should now be visible in Drive.')

All changes made in this colab session should now be visible in Drive.
```

### ▼ PyDrive

Os exemplos abaixo demonstram a autenticação e o upload/download de arquivos usando o PyDrive. Existem mais exemplos disponíveis na [documentação do PyDrive](#).

```
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
from google.colab import auth
from oauth2client.client import GoogleCredentials
```

Autentique e crie o cliente do PyDrive.

```
auth.authenticate_user()
gauth = GoogleAuth()
gauth.credentials = GoogleCredentials.get_application_default()
drive = GoogleDrive(gauth)
```

Crie e faça upload de um arquivo de texto.

```
uploaded = drive.CreateFile({'title': 'Sample upload.txt'})
uploaded.SetContentString('Sample upload file content')
```

```
uploaded.Upload()
print('Uploaded file with ID {}'.format(uploaded.get('id')))

Uploaded file with ID 14vDAdqp7B5CQnoougmgyl8BexIr2AQx2T
```

Use o ID do arquivo para carregá-lo e imprima seu conteúdo.

```
downloaded = drive.CreateFile({'id': uploaded.get('id')})
print('Downloaded content "{}".format(downloadd.GetContentString()))

Downloaded content "Sample upload file content"
```

## ▼ API REST do Drive

Para usar a API REST do Drive, primeiro é necessário autenticar e construir um cliente da API.

```
from google.colab import auth
auth.authenticate_user()
from googleapiclient.discovery import build
drive_service = build('drive', 'v3')
```

Com esse cliente, é possível usar todas as funções da [Referência da API do Google Drive](#). Veja os exemplos a seguir.

## ▼ Criar um novo arquivo do Drive com dados do Python

Primeiro, crie um arquivo local para fazer upload.

```
with open('/tmp/to_upload.txt', 'w') as f:
    f.write('my sample file')

print('/tmp/to_upload.txt contains:')
!cat /tmp/to_upload.txt

/tmp/to_upload.txt contains:
my sample file
```

Faça upload usando o método [files.create](#). Mais detalhes sobre como fazer upload de arquivos estão disponíveis na [documentação do desenvolvedor](#).

```
from googleapiclient.http import MediaFileUpload

file_metadata = {
    'name': 'Sample file',
    'mimeType': 'text/plain'
}
media = MediaFileUpload('/tmp/to_upload.txt',
                        mimetype='text/plain',
                        resumable=True)
created = drive_service.files().create(body=file_metadata,
                                       media_body=media,
                                       fields='id').execute()
print('File ID: {}'.format(created.get('id')))

File ID: 1Cw9CqiyU6zbXFD9ViPZu_3yX-syF4W17
```

Depois de executar a célula acima, você verá um novo arquivo chamado "Sample file" no <https://drive.google.com/>.

## ▼ Fazer download de dados de um arquivo do Drive para Python

Faça do download do arquivo enviado por upload anteriormente.

```
file_id = created.get('id')

import io
from googleapiclient.http import MediaIoBaseDownload

request = drive_service.files().get_media(fileId=file_id)
downloaded = io.BytesIO()
downloader = MediaIoBaseDownload(downloaded, request)
done = False
while done is False:
    # _ is a placeholder for a progress object that we ignore.
    # (Our file is small, so we skip reporting progress.)
    _, done = downloader.next_chunk()

downloaded.seek(0)
print('Downloaded file contents are: {}'.format(downloaded.read()))

Downloaded file contents are: b'my sample file'
```

Para fazer o download de um arquivo diferente, configure o `file_id` acima com o ID desse arquivo, que será similar a este: "1uBtlaggVyWshwcyP6kEI-yW3P8D26sz".

## ▼ Planilhas Google

Os exemplos abaixo usam a biblioteca [gspread](#) de código aberto para interagir com o Planilhas Google.

Importe a biblioteca, autentique e crie a interface com o Planilhas.

```
from google.colab import auth
auth.authenticate_user()

import gspread
from oauth2client.client import GoogleCredentials
```

```
gc = gspread.authorize(GoogleCredentials.get_application_default())
```

Veja abaixo alguns exemplos de `gspread`. Há outros exemplos disponíveis na [página sobre gspread do GitHub](#).

▼ Criar uma nova planilha com dados do Python

```
sh = gc.create('My cool spreadsheet')
```

Depois de executar a célula acima, você verá uma nova planilha chamada "My cool spreadsheet" no <https://drive.google.com/>.  
Abra sua nova planilha e adicione alguns dados aleatórios.

```
worksheet = gc.open('My cool spreadsheet').sheet1

cell_list = worksheet.range('A1:C2')

import random
for cell in cell_list:
    cell.value = random.randint(1, 10)

worksheet.update_cells(cell_list)

{'spreadsheetId': '1dsQeN0YzXuM3871_CuyEbsYzL2ew9TJFzR-E-RQnwxS',
 'updatedCells': 6,
 'updatedColumns': 3,
 'updatedRange': 'Sheet1!A1:C2',
 'updatedRows': 2}
```

▼ Fazer o download de dados de uma planilha em Python como um DataFrame da pandas

Leia novamente os dados aleatórios inseridos anteriormente e converta o resultado para um [Pandas DataFrame](#).

```
worksheet = gc.open('My cool spreadsheet').sheet1

# get_all_values gives a list of rows.
rows = worksheet.get_all_values()
print(rows)

import pandas as pd
pd.DataFrame.from_records(rows)
```

[[['6', '3', '4'], ['7', '2', '1']]]

1 to 2 of 2 entries <span>Filter</span>			
index	0	1	2
	0 6	3	4
	1 7	2	1

Show 25 per page

▼ Google Cloud Storage (GCS)

Para usar o Colaboratory com GCS, será preciso criar um [projeto do Google Cloud](#) ou usar um preexistente.  
Especifique abaixo o ID do projeto:

```
project_id = 'Your_project_ID_here'
```

Os arquivos no GCS são contidos em [buckets](#).  
Os buckets precisam ter nomes exclusivos mundialmente. Por isso, vamos gerar um aqui.

```
import uuid
bucket_name = 'colab-sample-bucket-' + str(uuid.uuid1())
```

Para acessar o GCS, é preciso autenticar.

```
from google.colab import auth
auth.authenticate_user()
```

O GCS pode ser acessado pelo utilitário `gsutil` da linha de comando ou através da API Python nativa.

▼ `gsutil`

Primeiro, é preciso configurar `gsutil` para usar o projeto especificado anteriormente usando `gcloud`.

```
!gcloud config set project {project_id}

Updated property [core/project].
```

Crie um arquivo local para fazer upload.

```
with open('/tmp/to_upload.txt', 'w') as f:
    f.write('my sample file')

print('/tmp/to_upload.txt contains:')
!cat /tmp/to_upload.txt

/tmp/to_upload.txt contains:
my sample file
```

Faça um bucket para onde você quer fazer o upload ([acesse a documentação aqui](#)).

```
!gsutil mb gs://{bucket_name}
```

```
Creating gs://colab-sample-bucket-44971372-baaf-11e7-ae30-0242ac110002/...
```

Copie o arquivo para o novo bucket ([acesse a documentação aqui](#)).

```
!gsutil cp /tmp/to_upload.txt gs://{bucket_name}/

Copying file:///tmp/to_upload.txt [Content-Type=text/plain]...
/ [1 files][ 14.0 B/ 14.0 B]
Operation completed over 1 objects/14.0 B.
```

Despeje o conteúdo do arquivo recém copiado para conferir se tudo funcionou corretamente ([acesse a documentação aqui](#)).

```
!gsutil cat gs://{bucket_name}/to_upload.txt

my sample file
```

Quando o upload for concluído, os dados aparecerão no navegador do armazenamento do Cloud Console do seu projeto:

[Mostrar código](#)

[https://console.cloud.google.com/storage/browser?project=Your\\_project\\_ID\\_here](https://console.cloud.google.com/storage/browser?project=Your_project_ID_here)

Por último, faça download do arquivo que acaba de ser enviado no exemplo acima. É muito simples: basta inverter a ordem do comando `gsutil cp`.

```
!gsutil cp gs://{bucket_name}/to_upload.txt /tmp/gsutil_download.txt

# Imprima o resultado para conferir se a transferência funcionou.
!cat /tmp/gsutil_download.txt

Copying gs://colab-sample-bucket483f20dc-baaf-11e7-ae30-0242ac110002/to_upload.txt...
/ [1 files][ 14.0 B/ 14.0 B]
Operation completed over 1 objects/14.0 B.
my sample file
```

## ▼ API do Python

Estes snippets se baseiam em [um exemplo maior](#) que mostra outros usos da API.

Primeiro, crie o cliente do serviço.

```
from googleapiclient.discovery import build
gcs_service = build('storage', 'v1')
```

Crie um arquivo local para fazer upload.

```
with open('/tmp/to_upload.txt', 'w') as f:
    f.write('my sample file')

print('/tmp/to_upload.txt contains:')
!cat /tmp/to_upload.txt

/tmp/to_upload.txt contains:
my sample file
```

Crie um bucket no projeto especificado acima.

```
# Use um nome de bucket exclusivo mundialmente do exemplo gsutil acima.
import uuid
bucket_name = 'colab-sample-bucket-' + str(uuid.uuid1())

body = {
    'name': bucket_name,
    # For a full list of locations, see:
    # https://cloud.google.com/storage/docs/bucket-locations
    'location': 'us',
}
gcs_service.buckets().insert(project=project_id, body=body).execute()
print('Done')

Done
```

Faça upload do arquivo para o bucket recém criado.

```
from googleapiclient.http import MediaFileUpload

media = MediaFileUpload('/tmp/to_upload.txt',
                        mimetype='text/plain',
                        resumable=True)

request = gcs_service.objects().insert(bucket=bucket_name,
                                       name='to_upload.txt',
                                       media_body=media)

response = None
while response is None:
    # _ is a placeholder for a progress object that we ignore.
    # (Our file is small, so we skip reporting progress.)
    _, response = request.next_chunk()

print('Upload complete')

Upload complete
```

Quando o upload for concluído, os dados aparecerão no navegador do armazenamento do Cloud Console do seu projeto:

```
.. ..  
https://console.cloud.google.com/storage/browser?project=Your\_project\_ID\_here
```

Faça o download do arquivo que acaba de ser enviado.

```
from apiclient.http import MediaIoBaseDownload  
  
with open('/tmp/downloaded_from_gcs.txt', 'wb') as f:  
    request = gcs_service.objects().get_media(bucket=bucket_name,  
                                              object='to_upload.txt')  
    media = MediaIoBaseDownload(f, request)  
  
    done = False  
    while not done:  
        # _ is a placeholder for a progress object that we ignore.  
        # (Our file is small, so we skip reporting progress.)  
        _, done = media.next_chunk()  
  
print('Download complete')  
  
    Download complete
```

Inspecione o arquivo salvo.

```
!cat /tmp/downloaded_from_gcs.txt  
  
my sample file
```