Python & Jupyter Workshop

https://github.com/ybsuen

Objectives

- Recap Python Lists and Dictionary
- Recap Python Functions
- From Beautiful Soup to Pandas
- Introduction to Pandas
- Storing Data in Google Sheet
- Introduction to Google Refine

Recap Python Lists and Dictionary

Basic Data Types in Python

- Integer
- Float
- String
- Boolean

String

1. The concept of an "index": position of each character in the string
2. The first position is always "0"
3. Notation of the index is expressed as:string_var[beginning position:ending position]
4. The ending position is not included
5. The len(string_var) function will return the length of the string
6. Let's say z = "I am from CUHK."
7. What will the command print(z[1:4]) return?
8. How can you print out the entire string using the index
9. Negative index starts from the end of the string e.g. name[-1:]

Try it out

```
z = "I am from CUHK.”

print(len(z))
print(z[0:14])
print(z[-3:])
print(z[10:14])
```

Useful String Functions

To show the result of your code, make sure "code" display option is chosen instead of the "markdown" option for rendering in Jupyter Notebook and then type:

1. name.upper()
2. name.replace(destination,source)
3. name.find(destination)
4. name.split()
5. name.count(target)

Try it out

```
famous_person = "Prince Charles"
print(famous_person)
famous_person2 = famous_person.replace("Charles","Harry")
print(famous_person, famous_person2)
print(famous_person.split())
print(famous_person.count("e"))
```

Try it out

```
name = "Prince Charles"
print(name)
name = name.replace("Charles","William")
name = name.upper()
print(name)
print(name.find("Charles"))
print(name.find("WILLIAM"))
```

Try it out

```
names = "Prince Charles and Prince Harry"
first_prince_position = names.find("Prince")
print("1st Prince Position:"+str(first_prince_position))
second_prince_position = names.find("Prince",first_prince_position+1)
print("2nd Prince Position:"+str(second_prince_position))
son_of_charles = names[second_prince_position:]
print(son_of_charles)
```

Basic Data Structures in Python

- List and Tuple
- Dictionary and Set

Tuple and List

- Tuples are arrays enclosed with round brackets for storing multiple variables
- Variables with data types can be stored into a Tuple
- A Tuple operates like a string and therefore can be indexed from the beginning (positive) and the end (negative)
- Tuples are immutable
- Inorder to manipulate a tuple, a new one has to be created
- Tuples can be nested e.g. Tuple2 = (1,2,(3,4),5)
- Due to its rigidity, List is more commonly used than Tuple.
- Lists are like Tubles but are mutable and enclosed with square brackets
- List, similar to a Tuple, operates like a string when it comes to access individual element within the data structure

Try it out

```
my_list = [1,2,3,4,5,2.5]
print(sorted(my_list))
print(my_list)
my_list = my_list + [6,7,8]
print(my_list)
my_list.append([9,10,11])
print(my_list)
del(my_list[9])
print(my_list)
```

Try it out

```
score_list = [100,95,85,60,60.5,70]
print(score_list)
print(score_list[3])
score_list.append([50,45,65,55])
print(score_list)
del(score_list[6])
print(score_list[3:7])
print(score_list)
score_list.append([1,2,3,"I am from CUHK.",True])
print(score_list)
second_list = [11,22,33]
second_list.append([44,55,66])
score_list.append(second_list)
print(score_list)
```

Set and Dictionary

- Similar to lists and tuples, sets support different Python types
- Sets use {} (braces) to embed values/elements
- Sets do not allow duplicates
- Lists can be converted into sets with the set function
- Dictionary store data in an array of key-value pairs in braces
- For instance, here is a dictionary instance: dict = {“key1”:1,”key2”: 2,”key3”:3}
- 1st column representing the key and 2nd column representing the value
- <dict_name>.keys() returns all the keys
- <dict_name>.values.() returns all the values

Try it out

```
dict1 = {'Peter':80,"David":90,"Mary":100}
print(dict1)
print(dict1.keys())
print(dict1.values())
print(dict1["David"])
scorelist = []
scorelist.append(dict1)
print(scorelist)
```

Basic Operations in Python

- Variable Assignment
- Mathematical Operations
- Functional Decomposition and Abstraction
- Logical Operations
- Looping Operations

Try it out

```
scorelist = [ ]
test1 = {"Peter":50,"David":60,"Mary":
65,"Harry":80}
test2 = {"Peter":90,"David":90,"Mary":
85,"Harry":70}
scorelist.append(test1)
scorelist.append(test2)
print(scorelist)
```

Looping Operations

```
i = 1
for i in range(1,10):
    print(i)
```

```
hrs_list = [30.0,40.0,50.0,60.0]
rate_list = [65.0,75.0,65.0,75.0]
name_list = ['John',"Mike","Mary","Jane"]
fee_list = []
for number in range(len(name_list)):
        hrs = hrs_list[number]
        rate = rate_list[number]
        fee = hrs * rate
        fee_list.append(fee)
        money_made = name_list[number] + " makes
" + str(fee) + "."
        print(money_made)
money_list = []
money_list.append(name_list)
money_list.append(fee_list)
print(money_list)
```

Try it out

```
hrs_list = [30.0,40.0,50.0,60.0]
rate_list = [65.0,75.0,65.0,75.0]
name_list = ['John',"Mike","Mary","Jane"]
fee_list = []
number = 0
while number < len(name_list):
        hrs = hrs_list[number]
        rate = rate_list[number]
        fee = hrs * rate
        fee_list.append(fee)
        money_made = name_list[number] + " makes "
+ str(fee) + "."
        print(money_made)
        number = number + 1
money_list = []
money_list.append(name_list)
money_list.append(fee_list)
print(money_list)
print(len(name_list))
```

Try it out

Try it out

```
hrs_list = [30.0,40.0,50.0,60.0]
rate_list = [65.0,75.0,65.0,75.0]
name_list = ['John',"Mike","Mary","Jane"]
hrs = {"John":30.0,"Mike":40,"Mary":50,"Jane":
60}
rates = {"John":65.0,"Mike":75.0,"Mary":
65.0,"Jane":75.0}
for i in name_list:
    fees = hrs[i]* rates[i]
    make_money = " makes " + str(fees) + "."
    print(i,make_money)
```

Conditional Operations

```
my_list = ['red','green','blue','orange','black']
index = 0
for i in my_list:
  if i == 'blue':
    print(index, "Blue is printed.")
    index += 1
  else:
    print(index, my_list[index])
    # index += 1
    index = index + 1
print("This is done.")
```

```
hrs_list = [30.0,40.0,50.0,60.0]
rate_list = [65.0,75.0,65.0,75.0]
name_list = ['John',"Mike","Mary","Jane"]
hrs = {"John":30.0,"Mike":40,"Mary":50,"Jane":60}
rates = {"John":65.0,"Mike":75.0,"Mary":65.0,"Jane":75.0}
less_than_3200 = False
greater_equal_3200 = False
for i in name_list:
    fees = hrs[i]* rates[i]
    make_money = " makes " + str(fees) + "."
    if (fees < 3200.0):
        if (less_than_3200 == False):
            print("Less than 3200")
            less_than_3200 = True
        print(" ",i,make_money)
    else:
        if (greater_equal_3200 == False):
            print("More than or equal to 3200")
            greater_equal_3200 = True
        print(" ",i,make_money)
```

Try it out

Recap Python Functions

Try it out

```
def compute_ta_fees(hrs,rate):
    fee = hrs*rate
    return fee
ta_fees = compute_ta_fees(30,60.40)
print(ta_fees)
```

Try it out

```
def compute_ta_fees(hrs,rate,name):
    fee = hrs*rate
    return name + " has received $" + str(fee) + "."
ta_fees = compute_ta_fees(30,60.40,'Bernard')
print(ta_fees)
```

Try it out

```
def compute_ta_fees(hrs,rate,name):
    fee = hrs*rate
    money_to_mom = fee*.2
    return name + "'s mom has received $" +
str(money_to_mom) + "."
ta_fees = compute_ta_fees(36,67.30,'Mary')
print(ta_fees)
```

```
def compute_ta_fees(hrs,rate,name):
    fee = hrs*rate
    money_to_mom = fee*.2
    return name + "'s mom has received $" + str(money_to_mom) + "."
# print ta_fees
hrs_list = [30.0,40.0,50.0,60.0]
rate_list = [65.0,75.0,65.0,75.0]
name_list = ['John',"Mike","Mary","Jane"]
index = 0
output_list = []
for number in name_list:
        hrs = hrs_list[index]
        rate = rate_list[index]
        name = name_list[index]
        ta_fees = compute_ta_fees(hrs,rate,name)
        print(ta_fees)
        output_list.append(ta_fees)
        index = index + 1
print(output_list)
```

Try it out

Try it out

```
purchase_list = ["ABC 2T HD", "MS wireless mouse", "TS Wireless keyboard"]
inventory = {
    "ABC 2T HD":60,
    "MS wireless mouse":0,
    "TS Wireless keyboard":32,
    "CC 500G USB Drive":25
}

prices = {
    "ABC 2T HD":800,
    "MS wireless mouse":200,
    "TS Wireless keyboard":120,
    "CC 500G USB Drive":450
}

# Write your code below!
def compute_bill(part_list):
    total = 0
    for item in part_list:
        if inventory[item] > 0:
            total = total + prices[item]
            inventory[item] = inventory[item] - 1
            print(item,":",inventory[item])
    return total

print("Total:",compute_bill(purchase_list))
```

From Beautiful Soup to Pandas

Introduction to Pandas

- Pandas allows us to deal with 2 data structures: series and data frame
- In our workshop, we'll only concentrate on data frame as it is more commonly used
- A data frame is consisted of rows and columns. It can be made from Python list and dictionary objects
- Pandas can put a dictionary of list into a data frame
- There are build-in Panas functions for reading (e.g. df=pd.read_csv('<file name>')) and writing to CSV files (e.g df.to_csv('<file name>', sep='\t', encoding='utf-8'))

Try it out

```
import pandas as pd
df1 = pd.DataFrame({
    # Define dataframe as a dictionary object
    'Product ID': [1, 2, 3, 4],
    # add Product Name and Color here
    'Product Name': ['t-shirt','jeans','shirt','skirt'],
    'Color':['blue','green','red','black'],
    'Units Sold':[250,300,180,200]
})
print(df1)
```

With a little bit of background in HTML/CSS/JS and Python, it's now time to scrap!

Scraping Data with Beautiful Soup and Storing the Result as CSV File Using Startup Beat as Example

```
import requests
import csv
from bs4 import BeautifulSoup

quote_page = requests.get('http://startupbeat.hkej.com/?tag=fintech&paged=1')
soup = BeautifulSoup(quote_page.content,'html.parser')
data = []
for article in soup.find_all('div',class_='archive-text'):
    url = article.a.get('href')
    post_date = article.div.ul.li.text
    data.append((url, post_date))

with open('startup_beat_demo.csv', 'w') as csv_file:
    writer = csv.writer(csv_file)
    header = ['url','post date']
    writer.writerow(header)
    for url, post_date in data:
        writer.writerow([url, post_date])
```

Try it out

Selecting and Displaying the Data Before Cleaning

- Look at the some basic stats for the 'Player' column: df.Player.describe()
- Select a column: df['Player']
- Select the first 10 rows of a column: df['Player'][:10]
- Select multiple columns: df[['Player','College']]
- Select all entries over a particular value: df[df['G'] > 160]
- Select empty entries: df[df['Player'].isnull()]

Storing Data in Google Sheets

1. Create a Google Developer Credential to Use Google Drive API
2. Add Credentials to Project
3. Create Service Account Credential
4. Connect Jupiter to Google Sheet
5. Share a Google Sheet to Your Notebook

1. Create a Google Developer Credential to Use Google Drive API

Go to the Google Developer Console (https://console.developers.google.com/project).

Google APIs

Manage resources

CREATE PROJECT

DELETE

Google APIs

New Project

You have 22 projects remaining in your quota. Request an increase or delete projects.
Learn more

MANAGE QUOTAS

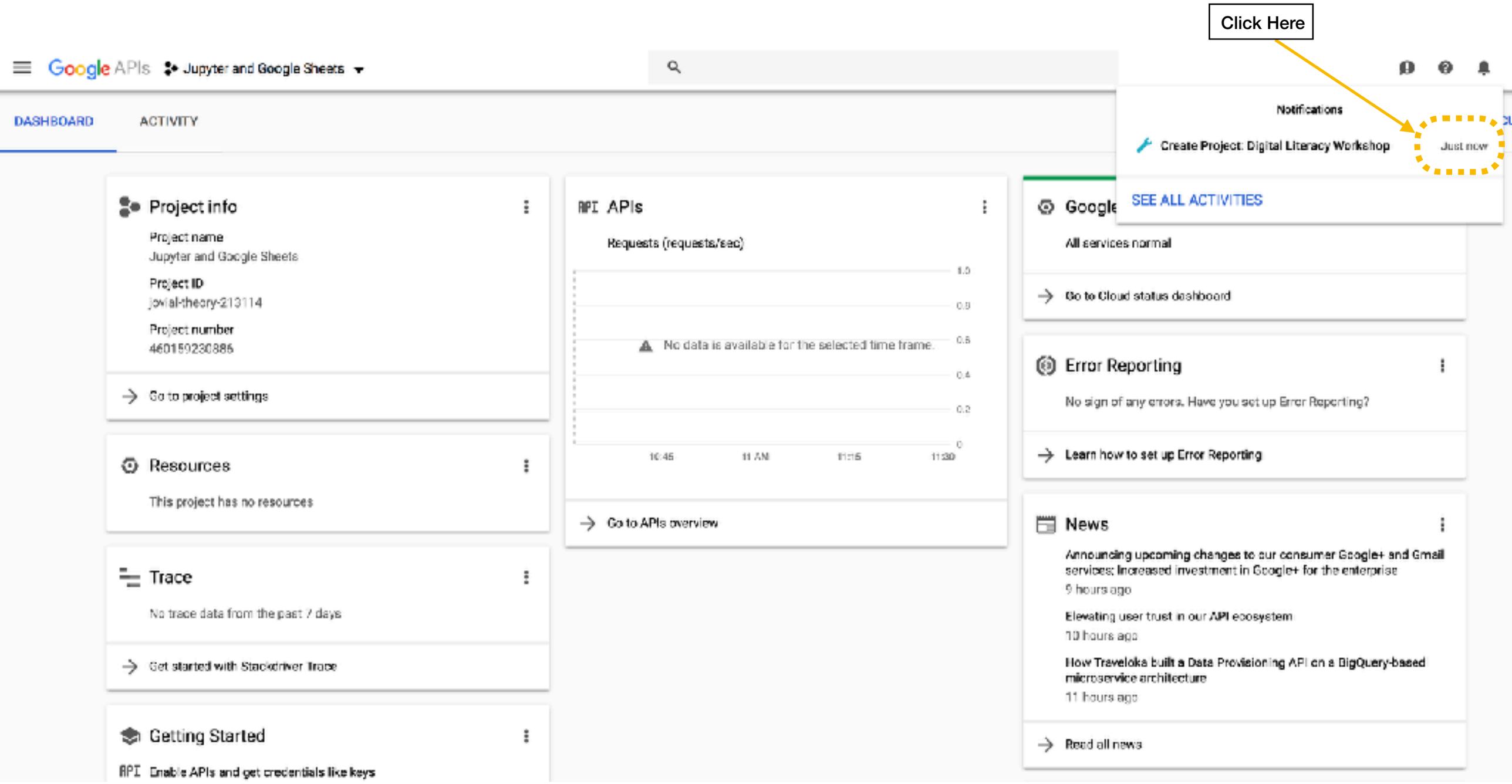
Project Name *
Digital Literacy Workshop

Project ID: digital-literacy-workshop. It cannot be changed later. EDIT

Location *
No organization BROWSE

Parent organization or folder

CREATE CANCEL


Click Here
Google APIs
Jupyter and Google Sheets
DASHBOARD
ACTIVITY
Notifications
Create Project: Digital Literacy Workshop
Just now
SEE ALL ACTIVITIES
Project info
Project name
Jupyter and Google Sheets
Project ID
jovial-theory-213114
Project number
460159230886
Go to project settings
Resources
This project has no resources
Trace
No trace data from the past 7 days
Get started with Stackdriver Trace
Getting Started
Enable APIs and get credentials like keys
APIs
Requests (requests/sec)
No data is available for the selected time frame.
Go to APIs overview
All services normal
Go to Cloud status dashboard
Error Reporting
No sign of any errors. Have you set up Error Reporting?
Learn how to set up Error Reporting
News
Announcing upcoming changes to our consumer Google+ and Gmail services; Increased investment in Google+ for the enterprise
9 hours ago
Elevating user trust in our API ecosystem
10 hours ago
How Traveloka built a Data Provisioning API on a BigQuery-based microservice architecture
11 hours ago
Read all news

Google APIs Digital Literacy Workshop

DASHBOARD ACTIVITY

Project info

Project name
Digital Literacy Workshop

Project ID
digital-literacy-workshop

Project number
199551338147

Go to project settings

Resources

This project has no resources

Trace

No trace data from the past 7 days

Get started with Stackdriver Trace

Getting Started

Enable APIs and get credentials like keys

APIs

Requests (requests/sec)

Go to APIs overview

Google Cloud Platform status

All services normal

Go to Cloud status dashboard

Error Reporting

No sign of any errors. Have you set up Error Reporting?

Learn how to set up Error Reporting

News

Announcing upcoming changes to our consumer Google+ and Gmail services; Increased investment in Google+ for the enterprise
9 hours ago

Elevating user trust in our API ecosystem
10 hours ago

How Travelokа built a Data Provisioning API on a BigQuery-based microservice architecture
11 hours ago

Read all news

Google APIs Digital Literacy Workshop

API APIs & Services

Dashboard

Library

Credentials

Dashboard ENABLE APIS AND SERVICES

No APIs or services are enabled

Browse the Library to find and use hundreds of available APIs and services

Popular APIs and services VIEW ALL (214)

Google Drive API
Google
The Google Drive API allows clients to access resources from Google Drive

Gmail API
Google
Flexible, RESTful access to the user's inbox

Maps SDK for Android
Google
Maps for your native Android app.

Cloud Translation API
Google
The Google Cloud Translation API lets websites and programs integrate with Google Translate...

Geocoding API
Google
Convert between addresses and geographic coordinates.

Click Here

Google APIs Digital Literacy Workshop

API Library

Google Drive API

Google

The Google Drive API allows clients to access resources from Google Drive

ENABLE TRY THIS API

Click Here

Type
APIs & services

Last updated
8/1/18, 12:54 PM

Category
Storage
G Suite

Service name
drive.googleapis.com

Overview

The Google Drive API allows clients to access resources from Google Drive.

About Google

Google's mission is to organize the world's information and make it universally accessible and useful. Through products and platforms like Search, Maps, Gmail, Android, Google Play, Chrome and YouTube, Google plays a meaningful role in the daily lives of billions of people.

Tutorials and documentation

Learn more

Terms of service

By using this product you agree to the terms and conditions of the following license(s): Google APIs Terms of Service

More solutions to explore

2. Add Credentials to Project

Google APIs Digital Literacy Workshop

APIs & Services
Google Drive API

Overview DISABLE API PROVIDE FEEDBACK

- Overview
- Metrics
- Quotas
- Credentials
- Drive UI Integration

To use this API, you may need credentials. Click 'Create credentials' to get started.

CREATE CREDENTIALS

Click Here

Details

Name
Google Drive API

By
Google

Service name
drive.googleapis.com

Overview
The Google Drive API allows clients to access resources from Google Drive.

Activation status
Enabled

Traffic by response code

Request/sec (2 hr average)

View metrics

Tutorials and documentation

Learn more

Try in API Explorer

Google APIs Digital Literacy Workshop

API APIs & Services

Dashboard

Library

Credentials

Credentials

Add credentials to your project

1 Find out what kind of credentials you need

We'll help you set up the correct credentials
If you wish you can skip this step and create an API key, client ID, or service account

Which API are you using?

Different APIs use different auth platforms and some credentials can be restricted to only call certain APIs.

Google Drive API

Other API

Click Here

2 Get your credentials

Cancel

Google APIs Digital Literacy Workshop

API APIs & Services

Dashboard

Library

Credentials

Credentials

Add credentials to your project

1 Find out what kind of credentials you need

We'll help you set up the correct credentials
If you wish you can skip this step and create an API key, client ID, or service account

Which API are you using?

Different APIs use different auth platforms and some credentials can be restricted to only call certain APIs.

Google Drive API

Where will you be calling the API from?

Credentials can be restricted using details of the context from which they're called. Some credentials are unsafe to use in certain contexts.

Web browser (Javascript)

Web server (e.g. node.js, Tomcat)

Android

iOS

Chrome application

PlayStation

Other UI (e.g. Windows, CLI tool)

Other non-UI (e.g. cron job, daemon)

nding on the type of

Click Here

Google APIs Digital Literacy Workshop

API APIs & Services

Credentials

- Dashboard
- Library
- Credentials

Add credentials to your project

1 **Find out what kind of credentials you need**

We'll help you set up the correct credentials
If you wish you can skip this step and create an API key, client ID, or service account

Which API are you using?

Different APIs use different auth platforms and some credentials can be restricted to only call certain APIs.

Google Drive API

Where will you be calling the API from?

Credentials can be restricted using details of the context from which they're called. Some credentials are unsafe to use in certain contexts.

Other UI (e.g. Windows, CLI tool)

What data will you be accessing?

Different credentials are required to authorize access depending on the type of data that you request.

- (●) **User data**
  Access data belonging to a Google user, with their permission
- ( ) **Application data**
  Access data belonging to your own application

Click Here

What credentials do I need?

Google APIs Digital Literacy Workshop

API APIs & Services

Credentials

Dashboard

Library

Credentials

Add credentials to your project

Find out what kind of credentials you need
Calling Google Drive API from a UI-based platform

2 Create an OAuth 2.0 client ID

Name

jupyter-google-sheet

Click Here

Create OAuth client ID

3 Set up the OAuth 2.0 consent screen

4 Download credentials

Cancel

Google APIs Digital Literacy Workshop

API APIs & Services

Dashboard

Library

Credentials

Credentials

Add credentials to your project

Find out what kind of credentials you need
Calling Google Drive API from a UI-based platform

Create an OAuth 2.0 client ID
Created OAuth client 'Other client 1'

3 Set up the OAuth 2.0 consent screen

Click Here

Email address
bsysin@gmail.com

Product name shown to users
digital-literacy-worksho

More customization options

Continue

The consent screen will be shown to users whenever you request access to their private data using your client ID. It will be shown for all applications registered in this project.

You must provide an email address and product name for OAuth to work.

Google APIs Digital Literacy Workshop

API APIs & Services

Dashboard

Library

Credentials

Credentials

Add credentials to your project

Find out what kind of credentials you need
Calling Google Drive API from a UI-based platform

Create an OAuth 2.0 client ID
Created OAuth client 'Other client 1'

Set up the OAuth 2.0 consent screen

4 Download credentials

Client ID 199651338147-ph6ij382oqs5oc39acu82ogkag5p83ef.apps.googleusercontent.com

Download this credential information in JSON format. This is always available for you on the credentials page.

Download I'll do this later

Done Cancel

Click Here

3. Create Service Account Credential

Google APIs Digital Literacy Workshop

API APIs & Services

Dashboard

Library

Credentials

Credentials

Credentials OAuth consent screen Domain verification

Create credentials Delete

Create credentials to access your enabled APIs. Refer to the API documentation for details.

OAuth 2.0 client IDs

Name	Creation date	Type	Client ID
jupyter-google-sheet	Oct 9, 2018	Other	199651338147-ph6ij362ops6oc39acu82cgkag5p83ef.apps.googleusercontent.com

Click Here

Google APIs Digital Literacy Workshop

API APIs & Services

Dashboard

Library

Credentials

Credentials

Credentials OAuth consent screen Domain verification

Create credentials Delete

API key
Identifies your project using a simple API key to check quota and access

OAuth client ID
Requests user consent so your app can access the user's data

Service account key
Enables server-to-server, app-level authentication using robot accounts

Help me choose
Asks a few questions to help you decide which type of credential to use

tion for details.

Type	Client ID
Other	199661338147-ph6ij382oqs6oo39ou82cgkag5p83cf.apps.googleusercontent.com

Click Here

Google APIs Digital Literacy Workshop

Create service account key

Service account

Select..

Key type

Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost.

- (●) **JSON**
  Recommended
- ( ) **P12**
  For backward compatibility with code using the P12 format

Create Cancel

Google APIs Digital Literacy Workshop

Create service account key

Service account

New service account

Service account name

Google Sheets

Service account ID

google-sheets @digital-literacy-workshop.iam

Key type

Downloads a file that contains the private key. Store the file secure
be recovered if lost.

JSON
Recommended

P12
For backward compatibility with code using the P12 format

Create Cancel

Role

Select a role

Selected

Project
App Engine
Billing
Cloud IAP
Cloud Security Scanner
Error Reporting
IAM
Logging
Monitoring
Organization Policy
Reserve Partner
Resource Manager
Roles
Service Accounts

Manage roles

Service Account Admin
Service Account Key Admin
Service Account Token Creator
Service Account User

Run operations as the service account

Click Here

Google APIs Digital Literacy Workshop

Create service account key

Service account

New service account

Service account name

Google Sheets

Role

Service Account ...

Service account ID

google-sheets @digital-literacy-workshop.iam

Selected

Service Account User

Run operations as the service account.

Project

App Engine

Billing

Cloud IAP

Cloud Security Scanner

Error Reporting

IAM

Logging

Monitoring

Organization Policy

Reserve Partner

Resource Manager

Roles

Service Accounts

Manage roles

Key type

Downloads a file that contains the private key. Store the file secure
be recovered if lost.

JSON
Recommended

P12
For backward compatibility with code using the P12 format

Create Cancel

Click Here

Google APIs Digital Literacy Workshop

Create service account key

Service account

New service account

Service account name

Google Sheets

Role

Service Account ...

Selected

Service Account User

Project

App Engine

Billing

Cloud IAP

Cloud Security Scanner

Error Reporting

IAM

Logging

Monitoring

Organization Policy

Reserve Partner

Resource Manager

Roles

Service Accounts

Manage roles

Service account ID

google-sheets @digital-literacy-workshop.iam

Key type

Downloads a file that contains the private key. Store the file secure
be recovered if lost

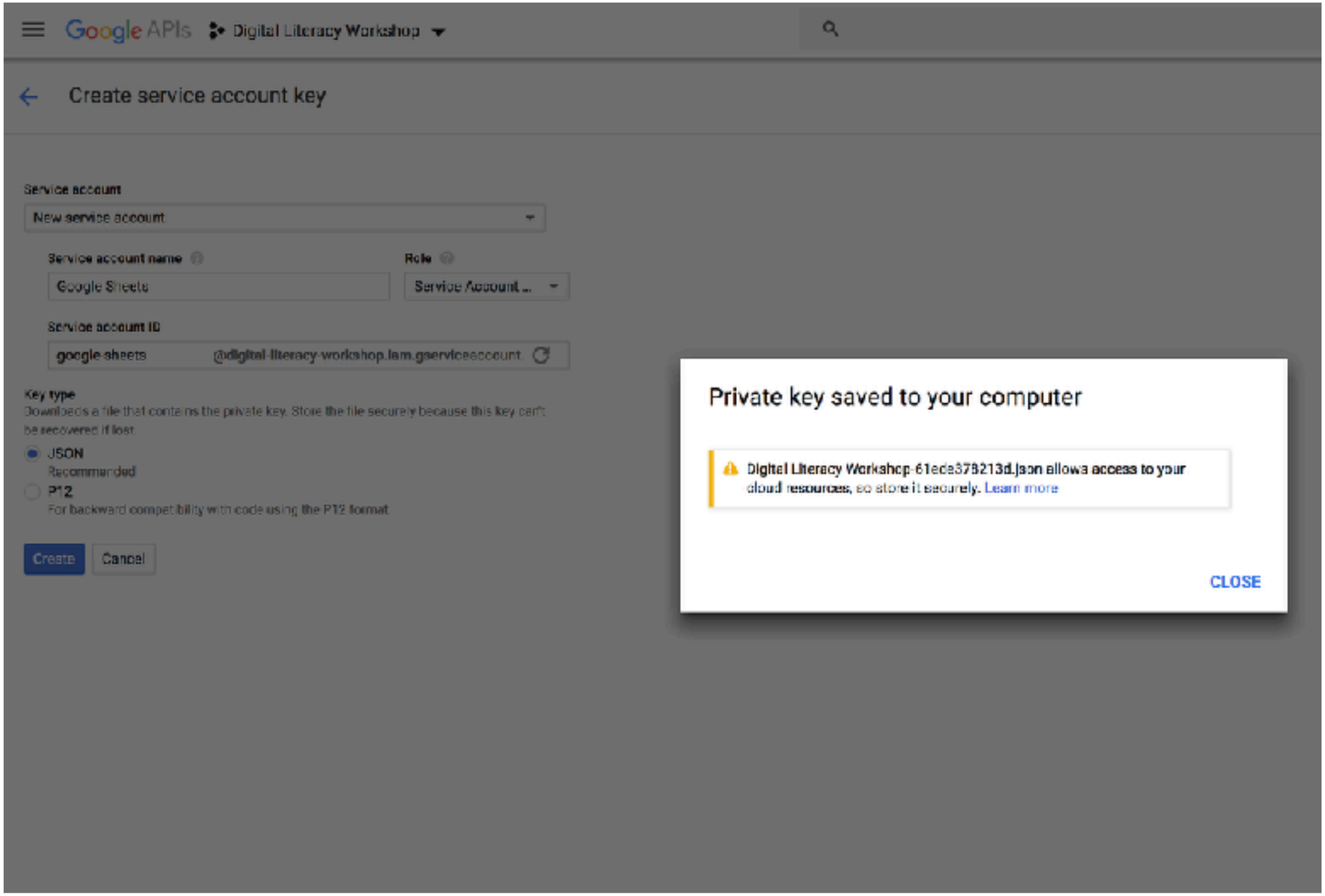
JSON

Recommended

P12

For backward compatibility with code using the P12 format

Create Cancel


Google APIs
Digital Literacy Workshop
Create service account key
Service account
New service account
Service account name
Google Sheets
Role
Service Account ...
Service account ID
google-sheets
@digital-literacy-workshop.iam.gserviceaccount.
Key type
Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost.
JSON
Recommended
P12
For backward compatibility with code using the P12 format
Create
Cancel
Private key saved to your computer
Digital Literacy Workshop-61ede378213d.json allows access to your cloud resources, so store it securely. Learn more
CLOSE

Google APIs Digital Literacy Workshop

API APIs & Services

- Dashboard
- Library
- Credentials

Credentials

Credentials | OAuth consent screen | Domain verification

Create credentials | Delete

Create credentials to access your enabled APIs. Refer to the API documentation for details.

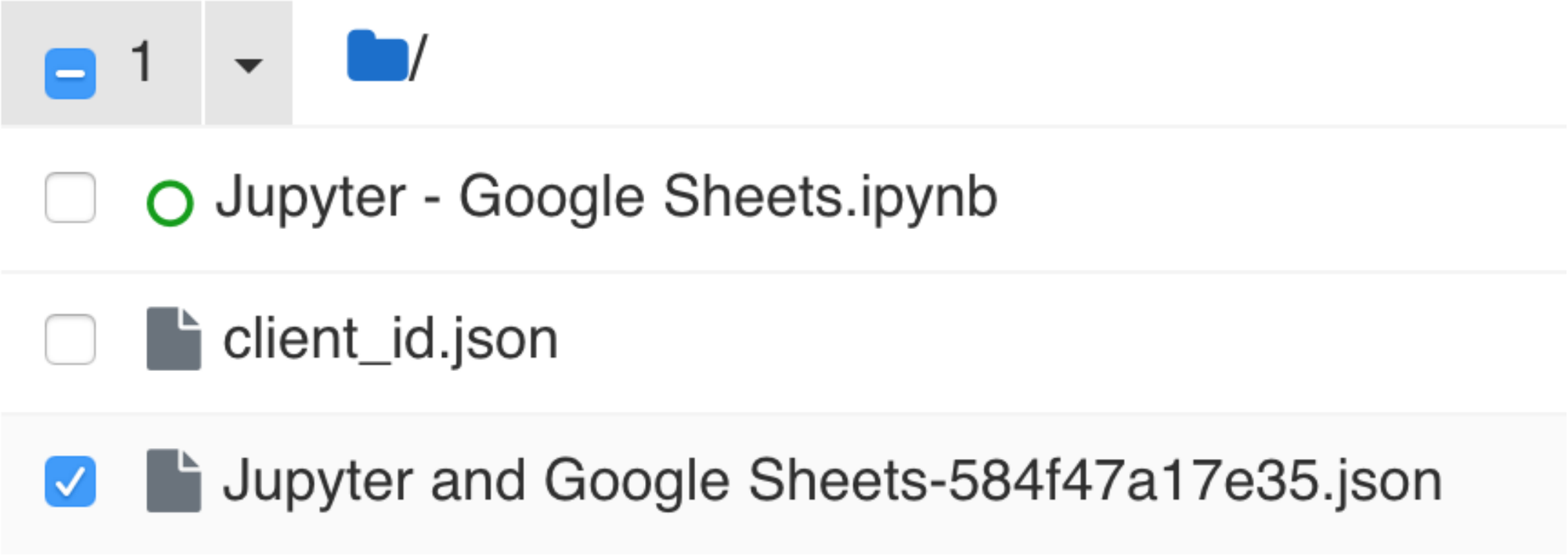
OAuth 2.0 client IDs

Name	Creation date	Type	Client ID
jupyter-google-sheet	Oct 9, 2018	Other	199651338147-ph6ij382oqs5oc39ocu82ogkag5p83cf.apps.googleusercontent.com

Service account keys

ID	Creation date	Service account
61ede378213cf2220274a2c8b919faa29df85869	Oct 9, 2018	Google Sheets

4. Connect Jupiter to Google Sheet


1
/
Jupyter - Google Sheets.ipynb
client_id.json
Jupyter and Google Sheets-584f47a17e35.json

```
6   "client_email": "google-sheets@jupyter-sheets-196102.iam.gserviceaccount.com",
```

5. Share a Google Sheet to Your Notebook

Drive

Search Drive

New

My Drive

Computers

Shared with me

Recent

Starred

Trash

Backups

Storage

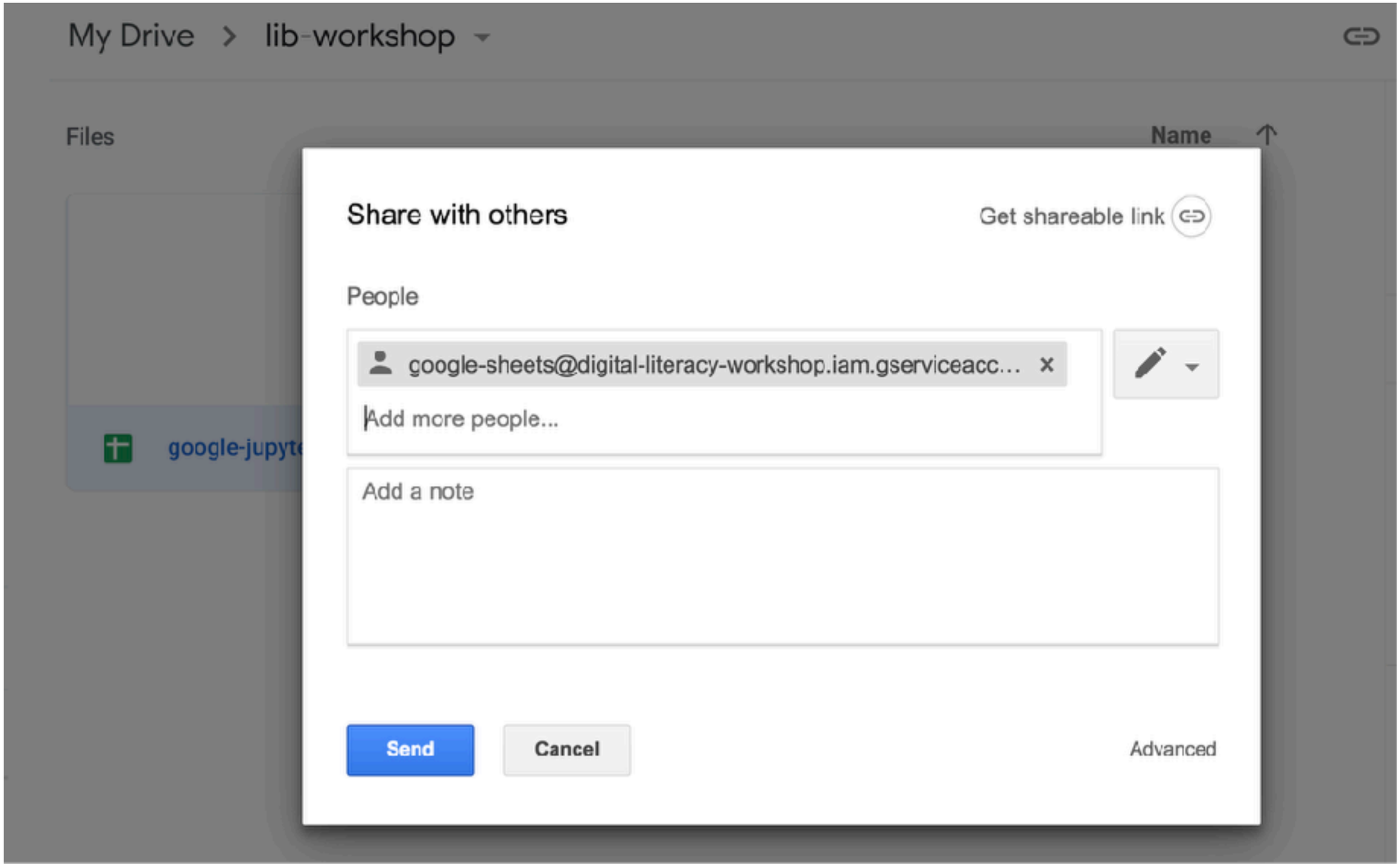
My Drive > lib-workshop

Files

Name

google-jupyte...

google-jupyter-tet


My Drive
lib-workshop
Files
Name
Share with others
Get shareable link
People
google-sheets@digital-literacy-workshop.iam.gserviceacc...
Add more people...
Add a note
Send
Cancel
Advanced
google-jupyte

In [ ]:

```
import pandas as pd
import gspread
from oauth2client.service_account import ServiceAccountCredentials

scope = ['https://spreadsheets.google.com/feeds']
credentials = ServiceAccountCredentials.from_json_keyfile_name('Jupyter and Google Sheets-cda1603fb5ad.json', scope)
gc = gspread.authorize(credentials)

spreadsheet_key = '1VR2nC8KSbt9Hi5mR9MWW6NDeJFHkaUfWtTve7QFovx0'
book = gc.open_by_key(spreadsheet_key)

worksheet = book.worksheet(“nba")
table = worksheet.get_all_values()
```

In [ ]:

```
import pandas as pd
import gspread
from oauth2client.service_account import ServiceAccountCredentials

scope = ['https://spreadsheets.google.com/feeds']
credentials = ServiceAccountCredentials.from_json_keyfile_name('Jupyter and Google Sheets-cda1603fb5ad.json', scope)
gc = gspread.authorize(credentials)

spreadsheet_key = '1VR2nC8KSbt9Hi5mR9MWW6NDeJFHkaUfWtTve7QFovx0'
book = gc.open_by_key(spreadsheet_key)

worksheet = book.worksheet(“nba")
table = worksheet.get_all_values()
```

In [8]:

```python
import pandas as pd
import gspread
from oauth2client.service_account import ServiceAccountCredentials

scope = ['https://spreadsheets.google.com/feeds']
credentials = ServiceAccountCredentials.from_json_keyfile_name('Jupyter and Google Sheets-cda1603fb5ad.json', scope)
gc = gspread.authorize(credentials)


spreadsheet_key = '1VR2nC8KSbt9Hi5mR9MWW6NDeJFHkaUfWtTve7QFovx0'
book = gc.open_by_key(spreadsheet_key)

worksheet = book.worksheet("nba")
table = worksheet.get_all_values()

df = pd.DataFrame(table[1:], columns=table[0])
##Only keep columns we need
df = df[['Rk', 'Pk', 'Tm', 'Player', 'College', 'Yrs', 'G', 'MP']]
df = df.apply(pd.to_numeric, errors='ignore')
df.head()
```

Out[8]:

	Rk	Pk	Tm	Player	College	Yrs	G	MP	MP
0	1	1.0	CLE	Andrew Wiggins	University of Kansas	4.0	327.0	11841.0	36.2
1	2	2.0	MIL	Jabari Parker	Duke University	4.0	183.0	5617.0	30.7
2	3	3.0	PHI	Joel Embiid	University of Kansas	2.0	94.0	2698.0	28.7
3	4	4.0	ORL	Aaron Gordon	University of Arizona	4.0	263.0	6867.0	26.1
4	5	5.0	UTA	Dante Exum		3.0	162.0	3280.0	20.2

Introduction to Google Refine

Not Secure openrefine.org

Apps | Cyberport Bootca... | Blockchain | NLP and Chatbot | FinTech | E-Commerce | Classes | Development Tools | Data Science | Machine Learning | Music | Drupal | Cloud Providers | Other Bookmarks

OpenRefine

A free, open source, powerful tool for working with messy data

- Home
- Community
- Documentation
- Download
- Contact Us
- Blog

Enhanced with Java profiler

JPROFILER

Welcome!

OpenRefine (formerly Google Refine) is a powerful tool for working with messy data: cleaning it; transforming it from one format into another; and extending it with web services and external data.

OpenRefine is available in English, Chinese, Spanish, French, Russian, Portuguese (Brazil), German, Japanese, Italian, Hungarian, Hebrew, Filipino, Cebuano, Tagalog

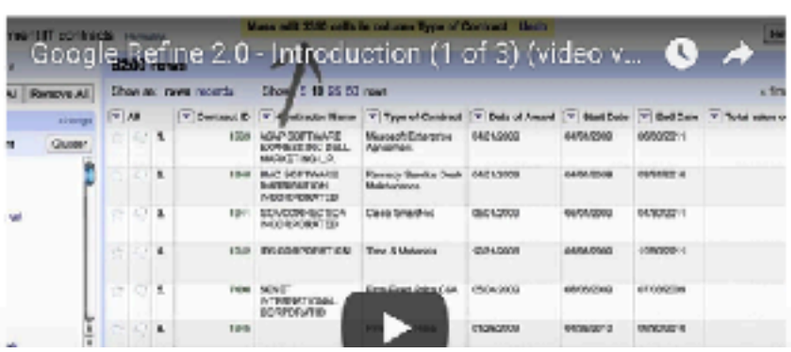
OpenRefine is supported by:

Google News Initiative

Introduction to OpenRefine

1. Explore Data

OpenRefine can help you explore large data sets with ease. You can find out more about this functionality by watching the video below and going through these articles



http://d3-media.blogspot.hk/2013/11/how-to-refine-your-data.html

Thank You!