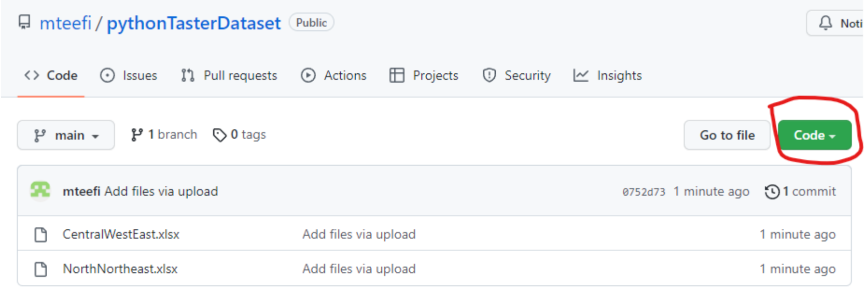
Python Cheatsheet: https://perso.limsi.fr/pointal/_media/python:cours:mementopython3-english.pdf
Recommended Python Course: https://ace.nus.edu.sg/python-for-data-science/
Recommended Udemy Course: https://www.udemy.com/course/100-days-of-code/

Topic 1: What is Python	
● Most widely used programming language currently (27.6% worldwide); followed by Java and JS ● High-level, interpreted, general purpose language ● Simple to use ● Very tolerant ● Widely supported by libraries - ML, CV, etc ● Perceived as “slower”; not as “hardcore”; no street cred ● Lots of peer help and support → stackoverflow	https://www.python.org/doc/essays/blurb/ https://en.wikipedia.org/wiki/Python_(programming_language) https://stackoverflow.com/questions/tagged/python
Topic 2: Repl IDE	
● Python is the programming language, Integrated Development Environment (IDE) are separate applications that facilitates the development of other python apps ○ Better GUIs ○ Better Workflows ○ Neater File Structure ● Many IDEs available → Pycharm, Jupyter, Repl etc ● Repl is a web-based IDE → no need to install anything on computer → just get started ○ SaaS → Actually fires up a VM that runs linux in the background everytime you log in ● Parts of Repl ○ Code Editor ○ Run Button ○ Console → output of results	https://replit.com/ Reddit for IDE recommendation: https://www.reddit.com/r/learnpython/comments/qcs6s7/best_python_ides_for_beginners/ Log in and check all ok shell: pwd uname -a Whoami Install openpyxl: Pip install openpyxl

- Shell → linux command prompt
- Main.py
- Uploading of files
    - [Go to Github Link](https://github.com/mteefi/pythonTasterDataset)
    - Click the “Green Code” Button
    - Choose “Download ZIP
    - Extract the files; note where you place them

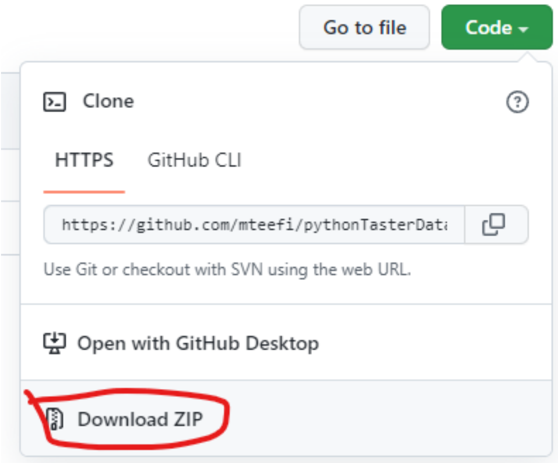
    - Go back to Repl
    - Click the 3 dots of the “Files”Tab
    - Choose “Upload File”
    - Upload the 2 files

Github:
https://github.com/mteefi/pythonTasterDataset



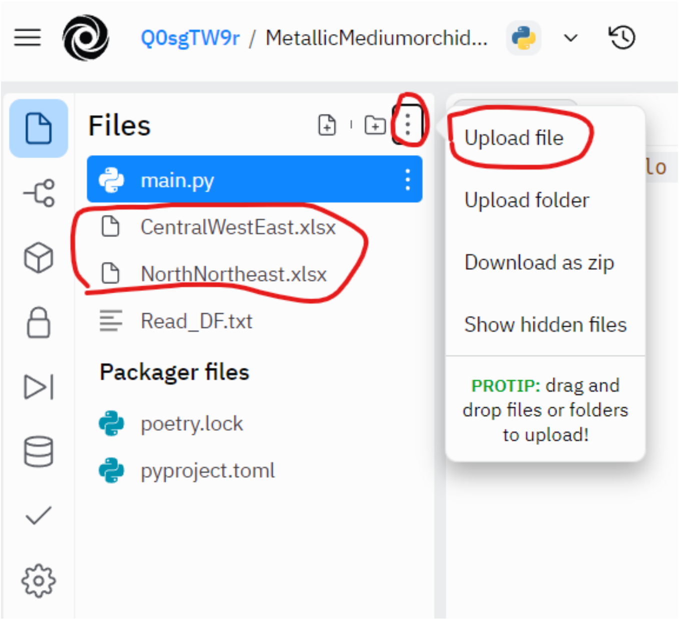
mteefi Add files via upload 0752d73 1 minute ago 1 commit

CentralWestEast.xlsx	Add files via upload	1 minute ago
NorthNortheast.xlsx	Add files via upload	1 minute ago



```
https://github.com/mteefi/pythonTasterDat
```

Use Git or checkout with SVN using the web URL.



Topic 3: Starter Pack Python Program

First Program:

- Hello World

Commenting:

- #
- Ctrl /

Indentation:

- Important in python

Variables (https://stackoverflow.com/questions/11007627/python-variable-declaration)

- There is no such thing as "variable declaration" or "variable initialization" in Python.
- There is simply what we call "assignment", but should probably just call "naming".
- Assignment means "this name on the left-hand side now refers to the result of evaluating the right-hand side, regardless of what it referred to before (if anything)".

Loops

```
print("Hello World!")

n = 1
print(n)
print(type(n))
print("The variable called n holds the value of",n,"and is of type",type(n))

n=1.1
print(n)
print(type(n))
print("The variable called n holds the value of",n,"and is of type",type(n))


n="now text variable"
print(n)
print(type(n))
print("The variable called n holds the value of",n,"and is of type",type(n))
```

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
  print(x)

for x in range(6):
  print(x)

for x in range(2, 6):
  print(x)

for x in range(2, 30, 3):
  print(x)
```

IF-ELSE CONDITIONS:
https://www.w3schools.com/python/python_operators.asp

```
num = -3

if num > 0:
    print("Positive")
elif num == 0:
    print("Zeeeeeeero")
else:
    print("Negative number")
```

```
n=[1,2,3]
print(n)
print(type(n))
print("The variable called n holds the value of",n,"and is of type",type(n))

def describeVariable(variableName):
  print(variableName)
  print(type(variableName))
  print("The variable called n holds the value of",variableName,"and is of type",type(variableName))
  return

n = 1
describeVariable(n)

anotherVar=1.1
describeVariable(anotherVar)

n="now text variable"
describeVariable(n)

n=[1,2,3]
describeVariable(n)
```

Part 4: Reading the Excel File

- What is pandas?
    - https://pandas.pydata.org/docs/user_guide/index.html
    - Popular Python Library for analyzing, reshaping, manipulating tabular data
    - Like “excel”
    - Pandas can clean messy data sets, and make them readable and relevant.
    - most widely used for data science/data analysis and machine learning tasks
    - Just type “import pandas as pd”
- Reading Excel File into Pandas
    - For manipulation using python code
    - Require openpyxl package (that’s why we pip install openpyxl in the

```
import pandas as pd

df_NNE = pd.read_excel (r'NorthNortheast.xlsx')

# dfShape=df_NNE.shape
# print("There are",dfShape[1],"columns or features")
# print("There are",dfShape[0],"rows or data entries")

# print(df_NNE.head())
# print(df_NNE.keys())

def checkShape(dfName):
```

beginning)

EXERCISE: Try reading the other CentralWestEast file and find out the number of columns and rows

```
  dfShape=dfName.shape
  print("There are",dfShape[1],"columns or features")
  print("There are",dfShape[0],"rows or data entries")
  print(dfName.head())
  print(dfName.keys())
  return

checkShape(df_NNE)
```

Topic 5: Merging the 2 Files and Writing Out to Excel File

- CAUTION: Please merge the files only when the columns match up to each other in real world situation
- In this case the 2 files are similar in column (number of columns and column name)
- How to do it?
  - Google stackoverflow lor
    - https://www.google.com/search?q=stackoverflow+how+to+merge+same+columns+df+python&rlz=1C1CHBF_enSG875SG875&sxsrf=ALiCzsZXQeOQ_NyEcq3d7xqi5NlccdiH8w%3A1660987506804&ei=cqgAY53OMJu4z7sPkYeymAo&ved=0ahUKEwidz8bSjNX5AhUb3HMBHZGDDKMQ4dUDCA8&uact=5&oq=stackoverflow+how+to+merge+same+columns+df+python
    - Stackoverflow is your best friend in coding
  - Use pd.concat
    - https://pandas.pydata.org/docs/reference/api/pandas.concat.html

EXERCISE: What is the total number of rows in the combined DF?

```
import pandas as pd

def checkShape(dfName):
  dfShape=dfName.shape
  print("There are",dfShape[1],"columns or features")
  print("There are",dfShape[0],"rows or data entries")
  print(dfName.head())
  print(dfName.keys())
  return

df_NNE = pd.read_excel (r'NorthNortheast.xlsx')
df_CWE = pd.read_excel (r'CentralWestEast.xlsx')

tempHoldingList=[df_NNE, df_CWE]
df_Combined = pd.concat(tempHoldingList)
df_Combined = df_Combined.reset_index(drop=True)

checkShape(df_Combined)

df_Combined.to_excel(r'combinedData.xlsx', index=False)
```

Topic 6: How to remove the clutter in the combined excel file?

- Many ways to skin a cat
  - Either REMOVE the unwanted columns in the data frame OR
  - Create a new data frame and populate it with the columns that you want

- I find that Creating a new data frame and populating it is easier and neater
  - There can be more unwanted columns than the columns you require
  - Columns may not be homogenous across different sources

```
import pandas as pd

def checkShape(dfName):
  dfShape=dfName.shape
  print("There are",dfShape[1],"columns or features")
  print("There are",dfShape[0],"rows or data entries")
  print(dfName.head())
  print(dfName.keys())
  return

df_NNE = pd.read_excel (r'NorthNortheast.xlsx')
df_CWE = pd.read_excel (r'CentralWestEast.xlsx')

tempHoldingList=[df_NNE, df_CWE]
df_Combined = pd.concat(tempHoldingList)
df_Combined = df_Combined.reset_index(drop=True)

df_New = pd.DataFrame(columns=["Reference","Date of
Fault","School Zone","School Name","Fault Type","Fault Criticality"])
df_New["Reference"]=df_Combined["Case Ref"].copy()
df_New["Date of Fault"]=df_Combined["Date Reported"].copy()
df_New["School Zone"]=df_Combined["Zone"].copy()
df_New["School Name"]=df_Combined["Schools"].copy()
df_New["Fault Type"]=df_Combined["Trade Type"].copy()
df_New["Fault Criticality"]=df_Combined["Criticality"].copy()

df_New['Date of Fault']=pd.to_datetime(df_New['Date of
Fault']).dt.date

checkShape(df_New)

df_New.to_excel(r'combinedData.xlsx', index=False)
```

Topic 7: Clean Data Further

- Study the “Fault Criticality” and “Fault Type” of the Excel File, what do you notice?
- How will your observation affect the visualisation effort?

```
import pandas as pd

def checkShape(dfName):
  dfShape=dfName.shape
  print("There are",dfShape[1],"columns or features")
  print("There are",dfShape[0],"rows or data entries")
  print(dfName.head())
  print(dfName.keys())
  return

df_NNE = pd.read_excel (r'NorthNortheast.xlsx')
df_CWE = pd.read_excel (r'CentralWestEast.xlsx')

tempHoldingList=[df_NNE, df_CWE]
```

	df_Combined = pd.concat(tempHoldingList) df_Combined = df_Combined.reset_index(drop=True) df_New = pd.DataFrame(columns=["Reference","Date of Fault","School Zone","School Name","Fault Type","Fault Criticality"]) df_New["Reference"]=df_Combined["Case Ref"].copy() df_New["Date of Fault"]=df_Combined["Date Reported"].copy() df_New["School Zone"]=df_Combined["Zone"].copy() df_New["School Name"]=df_Combined["Schools"].copy() df_New["Fault Type"]=df_Combined["Trade Type"].copy() df_New["Fault Criticality"]=df_Combined["Criticality"].copy() df_New['Date of Fault']=pd.to_datetime(df_New['Date of Fault']).dt.date df_New['Fault Criticality']=df_New['Fault Criticality'].str.upper() df_New["Fault Type"]=df_New["Fault Type"].replace({"Aircon":"Airconditioning","Power Tripped":"Power Trip"}) checkShape(df_New) df_New.to_excel(r'combinedData.xlsx', index=False)