

## Logs module

logs module typically refers to a module or library that facilitates logging functionality in a program. Logging is the process of recording events, messages, or information about the execution of a program for monitoring, debugging, and auditing purposes.

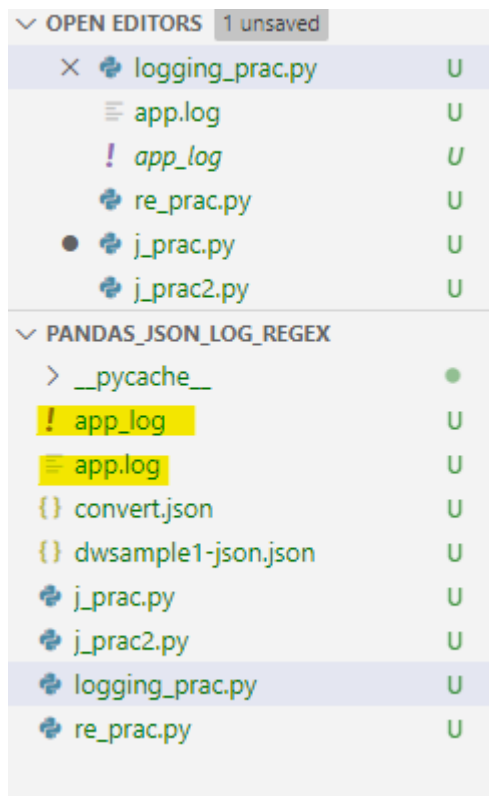
The built-in logging module in Python provides a comprehensive logging framework. It allows developers to configure loggers, handlers, formatters, and filters to control the behavior of logging in their applications.

When I try to configure the basic setting of logging it gives me this error:

```
import logging
File "c:\Users\Syed Taha Alam\Desktop\pandas_json_log_regex\logging.py", line 5, in <module>
    logging.basicConfig(
    ~~~~~~
AttributeError: partially initialized module 'logging' has no attribute 'basicConfig' (most likely due to a circular import)
5 C:\Users\Syed Taha Alam\Desktop\pandas_json_log_regex>
```

After searching about circular import, I learned that it conflicted with my working file name, "logging.py, " so I renamed it.

```
logging_prac.py
1  import logging
2
3
4  # Set the logging configuration
5  logging.basicConfig(
6      filename='app_log',                # Specify the log file
7      level=logging.ERROR,               # Set the logging level
8      format='%(asctime)s - %(levelname)s - %(message)s', # Specify the log message format
9      handlers=[
10         logging.FileHandler("app.log"), # keep log in file
11         logging.StreamHandler() #to print logs on console
12     ]
13
14
15  # Log some messages
16  logging.debug('This is a debug message')
17  logging.info('This is an informational message')
18  logging.warning('This is a warning message')
19  logging.error('This is an error message')
20  logging.critical('This is a critical message')
```



## Regex

Regex, short for regular expression, is a sequence of characters that forms a search pattern. It is a powerful tool used for pattern matching and manipulating strings. Regex allows you to search, extract, validate, and replace text based on specific patterns or rules.

```
re_prac.py > ...
1  import re
2
3  pattern = r'apple'
4  text = 'An apple a day keeps doctor away.'
5
6  # Match
7  match = re.match(pattern, text)
8  if match:
9      print('Match found:', match.group())
10 else: print('No match found')
11
12 # Search
13 search = re.search(pattern, text)
14 if search:
15     print('Search found:', search.group())
16
17 # Find all occurrences
18 matches = re.findall(pattern, text)
19 print('All occurrences:', matches)
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   SQL CONSOLE

```
e" "c:/Users/Syed Taha Alam/Desktop/pandas_json_log_regex/re_prac.py"
No match found
Search found: apple
All occurrences: ['apple', 'apple']
Replaced text: I have an orange and a pineorange.
PS C:\Users\Syed Taha Alam\Desktop\pandas_json_log_regex> & "C:/Users/
e" "c:/Users/Syed Taha Alam/Desktop/pandas_json_log_regex/re_prac.py"
No match found
Search found: apple
All occurrences: ['apple']
Replaced text: An orange a day keeps doctor away.
PS C:\Users\Syed Taha Alam\Desktop\pandas_json_log_regex> []
```

When I see the result I wonder why it's saying 'no match found' and I search and learned that the `re.match` matches at the beginning of the string.

Here is a [link](#)

## JSON

JSON (JavaScript Object Notation) is a lightweight data-interchange format that is widely used for representing structured data. It provides a simple and standardized way to exchange data between different systems and programming languages.

I searched a lot of articles and videos for this topic because it was really confusing for me.

Some of highlighted videos are: [video1](#), [video2](#), [video3](#) and lot other

[Link](#) to download json file for practice

j\_prac.py > ...

```
1
2 import json
3
4 file= '{"fruit": "Apple","size": "Large","color": "Red", "health" : "True"}'
5
6 data = json.loads(file) #convert string(which looks like json) into python dict
7
8 print(data)
9 print(type(data))
10
11 data2 = json.dumps(data) # convert python dict to json string
12
13 print(data2)
14 print(type(data2))
15
16
```

j\_prac2.py > ...

```
1
2 import json
3
4 with open('dwsample1-json.json') as file:
5     data = json.load(file) # convert json to python object
6
7
8 print(data)
9 print(type(data))
10
11 # Convert this data into json file
12
13 with open('convert.json', 'w') as file2:
14     json.dump(data, file2)
15
16
```

## Pandas

Pandas is a powerful open-source data manipulation and analysis library for Python. It provides data structures and functions to efficiently handle structured data, such as tabular data, time series, and more. Pandas is built on top of the NumPy library and is widely used in data science, machine learning, and data analysis workflows.

## Importing CSV File

(variable) df: DataFrame

▶

df = pd.read\_csv("diabetes\_prediction\_dataset.csv")  
df.head()

[2]

Pythor

...

	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
0	Female	80.0	0	1	never	25.19	6.6	140	0
1	Female	54.0	0	0	No Info	27.32	6.6	80	0
2	Male	28.0	0	0	never	27.32	5.7	158	0
3	Female	36.0	0	0	current	23.45	5.0	155	0
4	Male	76.0	1	1	current	20.14	4.8	155	0

## Shape of Dataset