Future
Connect
Media

Python
Part G

Part of Future Connect Media’s IT Course

By Abhishek Sharma

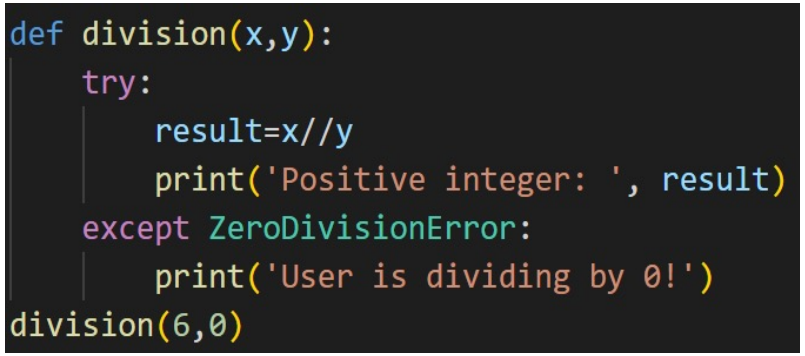
Future
Connect
Media

Topics to be covered:

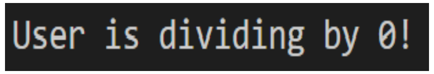
- Error and Exception Handling
- Module packages and libraries
- File management
- Future Consideration

- Try Except:

In Python, there are two different sorts of errors: syntax errors and exceptions. Errors are issues in a programme that cause it to halt during execution. On the other hand, exceptions are raised when internal events take place that alter the program's flow.

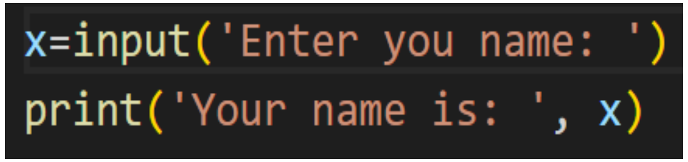


```
def division(x,y):
    try:
        result=x//y
        print('Positive integer: ', result)
    except ZeroDivisionError:
        print('User is dividing by 0!')
division(6,0)
```

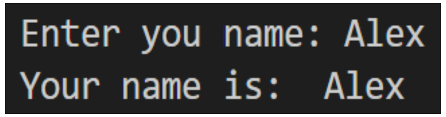


```
User is dividing by 0!
```

- Python user input:



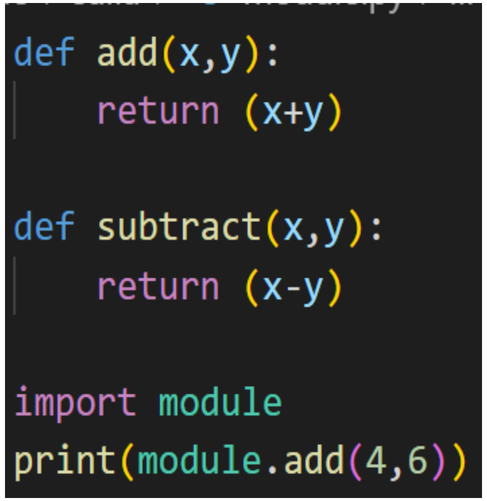
```
x=input('Enter you name: ')
print('Your name is: ', x)
```



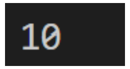
```
Enter you name: Alex
Your name is:  Alex
```

- Python modules:

A file with Python definitions and statements is known as a module. Variables, classes, and functions can all be defined in a module. Runnable code may also be included in a module.

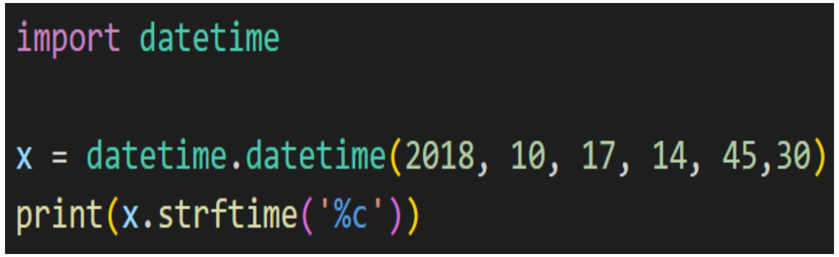


```
def add(x,y):
    return (x+y)

def subtract(x,y):
    return (x-y)

import module
print(module.add(4,6))
```

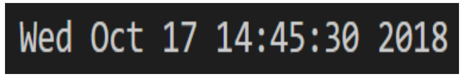


```
10
```

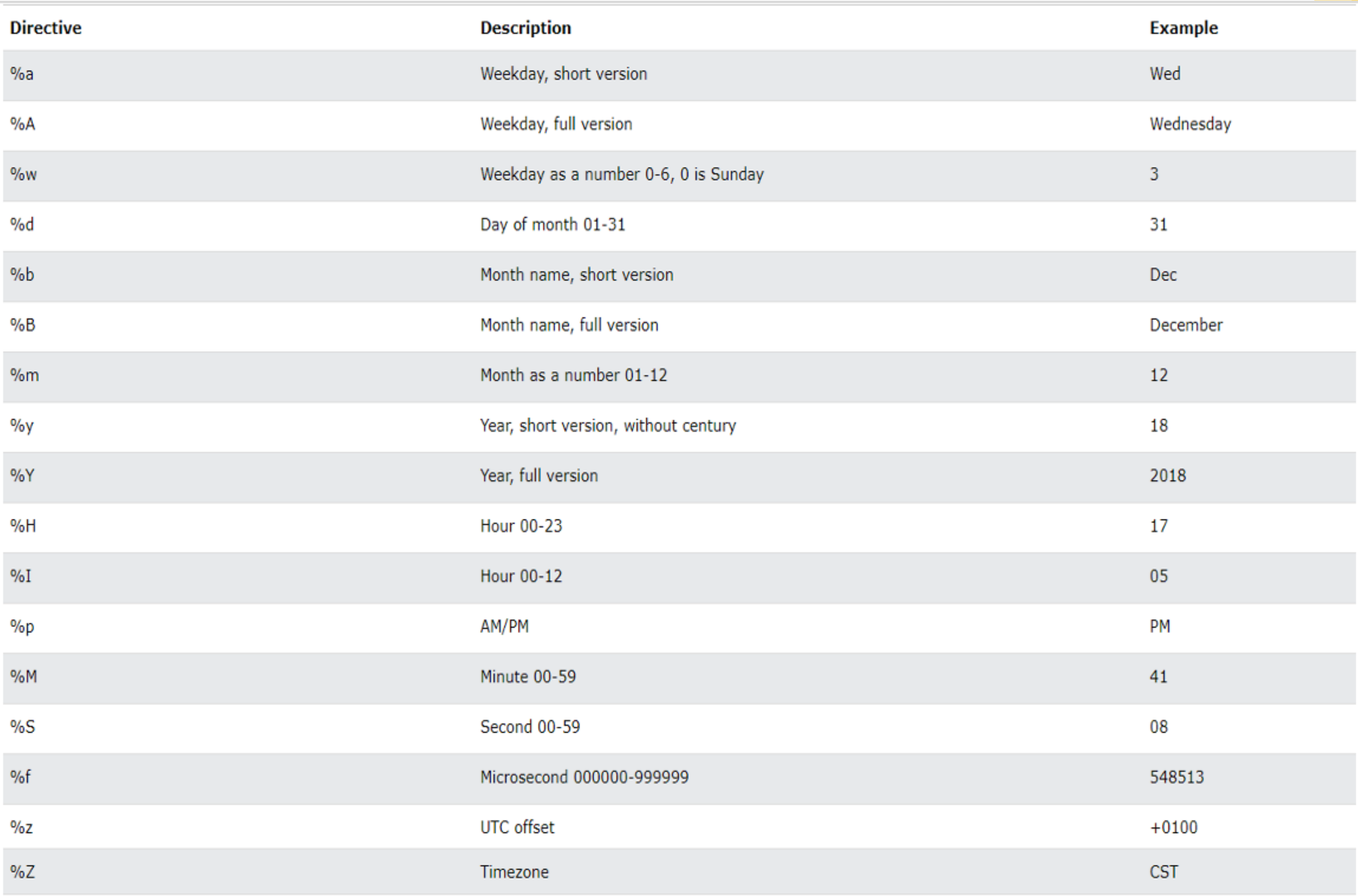
- Datetime module:



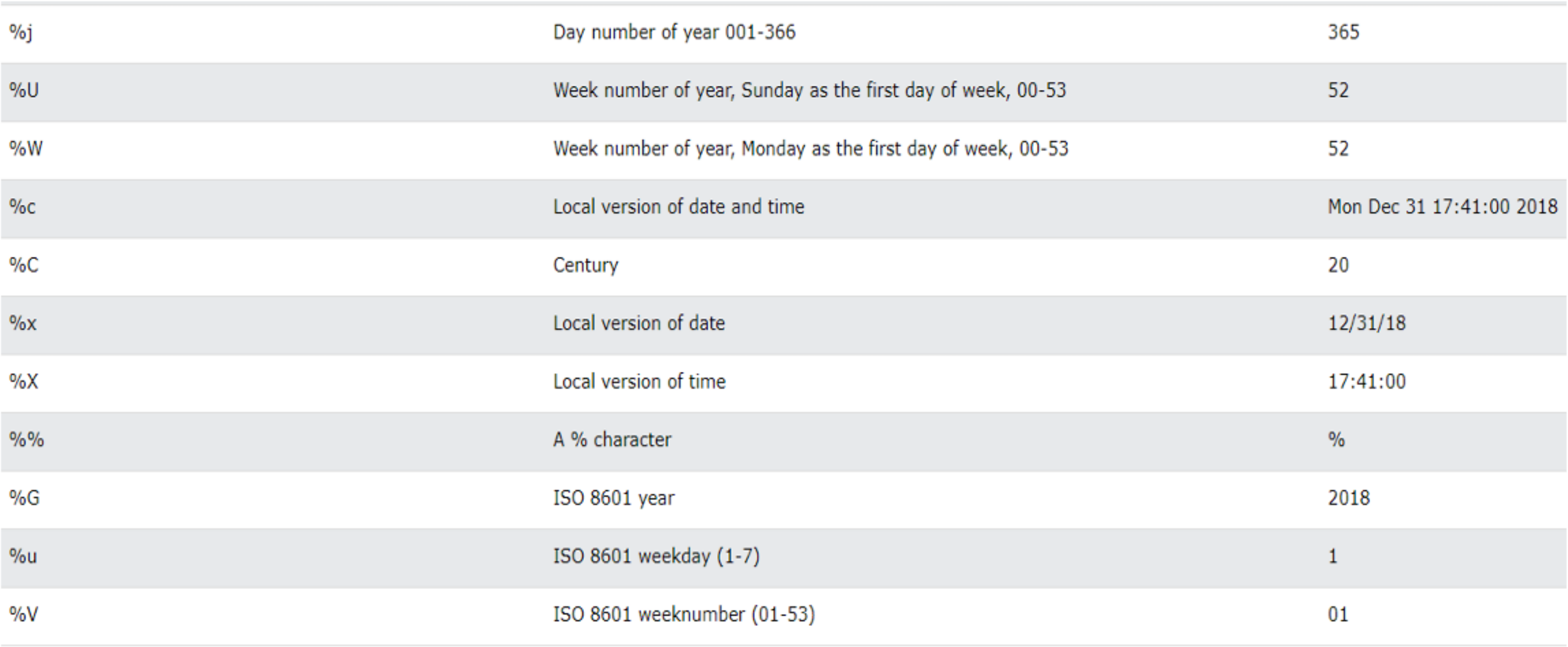
```
import datetime

x = datetime.datetime(2018, 10, 17, 14, 45,30)
print(x.strftime('%c'))
```



```
Wed Oct 17 14:45:30 2018
```

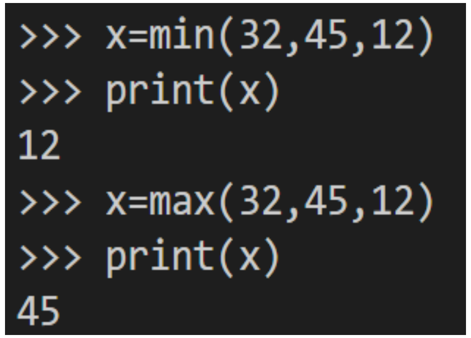


Directive	Description	Example
%a	Weekday, short version	Wed
%A	Weekday, full version	Wednesday
%w	Weekday as a number 0-6, 0 is Sunday	3
%d	Day of month 01-31	31
%b	Month name, short version	Dec
%B	Month name, full version	December
%m	Month as a number 01-12	12
%y	Year, short version, without century	18
%Y	Year, full version	2018
%H	Hour 00-23	17
%I	Hour 00-12	05
%p	AM/PM	PM
%M	Minute 00-59	41
%S	Second 00-59	08
%f	Microsecond 000000-999999	548513
%z	UTC offset	+0100
%Z	Timezone	CST

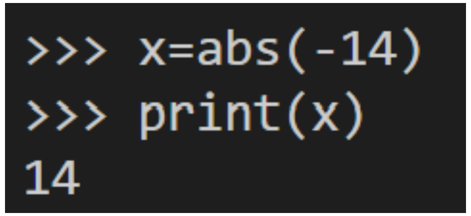


%j	Day number of year 001-366	365
%U	Week number of year, Sunday as the first day of week, 00-53	52
%W	Week number of year, Monday as the first day of week, 00-53	52
%c	Local version of date and time	Mon Dec 31 17:41:00 2018
%C	Century	20
%x	Local version of date	12/31/18
%X	Local version of time	17:41:00
%%	A % character	%
%G	ISO 8601 year	2018
%u	ISO 8601 weekday (1-7)	1
%V	ISO 8601 weeknumber (01-53)	01

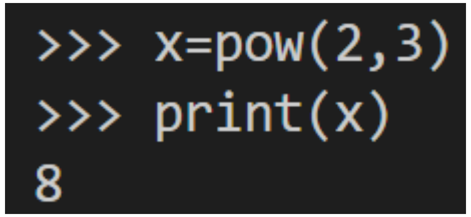
- Built in Math functions and Math module:



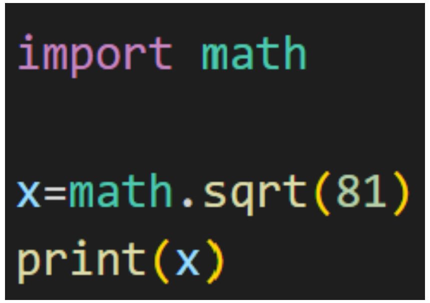
```
>>> x=min(32,45,12)
>>> print(x)
12
>>> x=max(32,45,12)
>>> print(x)
45
```



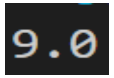
```
>>> x=abs(-14)
>>> print(x)
14
```



```
>>> x=pow(2,3)
>>> print(x)
8
```



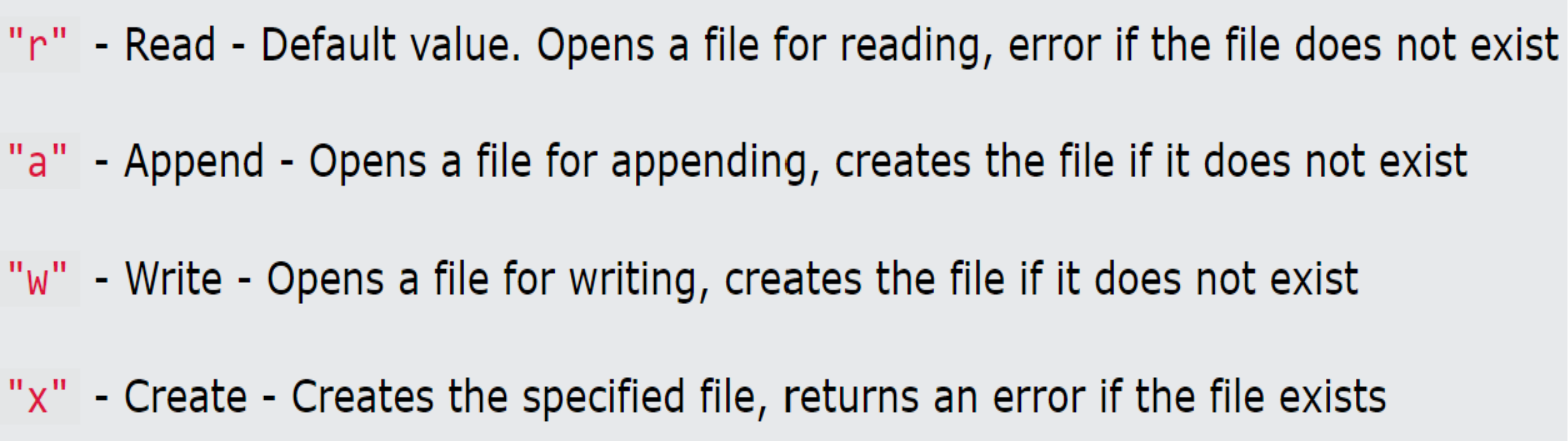
```
import math

x=math.sqrt(81)
print(x)
```



```
9.0
```

File Handling:

The open() function is crucial for working with files in Python. Filename and mode are the two parameters that the open() function accepts. A file can be opened in four different ways/modes.



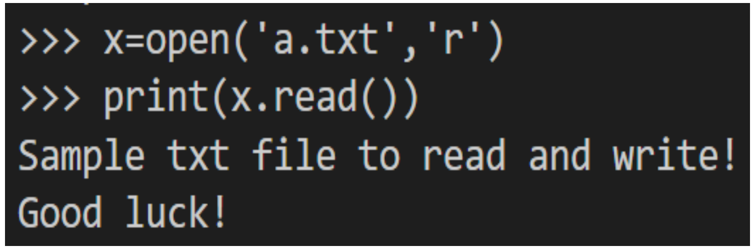
"r" - Read - Default value. Opens a file for reading, error if the file does not exist

"a" - Append - Opens a file for appending, creates the file if it does not exist

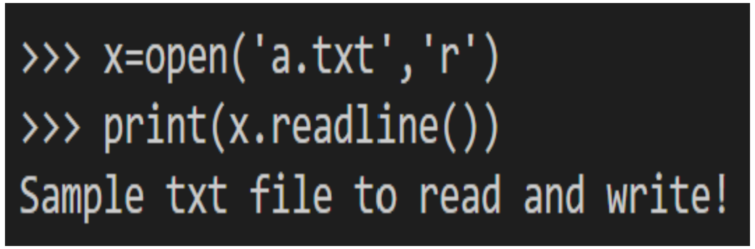
"w" - Write - Opens a file for writing, creates the file if it does not exist

"x" - Create - Creates the specified file, returns an error if the file exists

- readline() is used to read line by line.

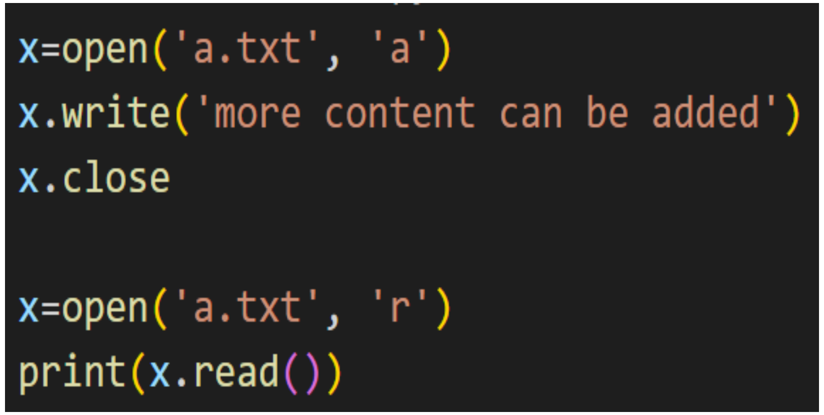


```
>>> x=open('a.txt','r')
>>> print(x.read())
Sample txt file to read and write!
Good luck!
```

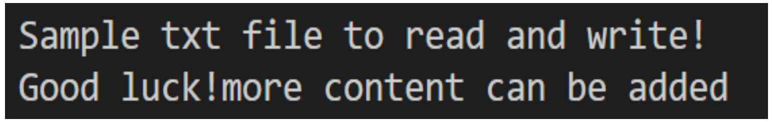


```
>>> x=open('a.txt','r')
>>> print(x.readline())
Sample txt file to read and write!
```

- adding more content to the txt file.

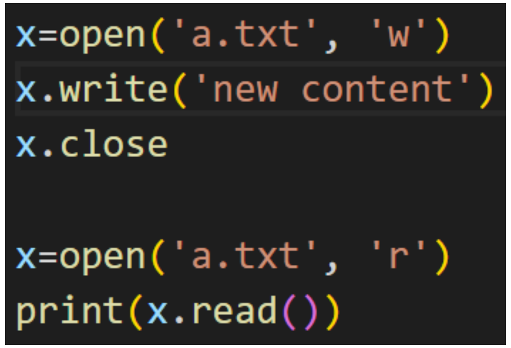


```
x=open('a.txt', 'a')
x.write('more content can be added')
x.close

x=open('a.txt', 'r')
print(x.read())
```

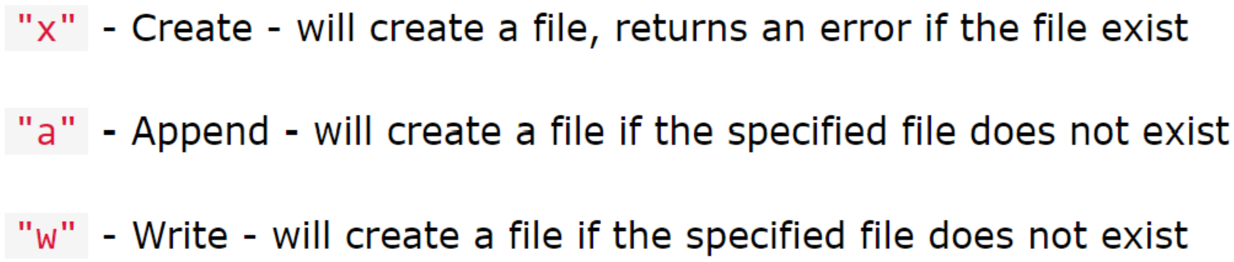


```
Sample txt file to read and write!
Good luck!more content can be added
```

- Overwrites the selected txt file.



```
x=open('a.txt', 'w')
x.write('new content')
x.close

x=open('a.txt', 'r')
print(x.read())
```

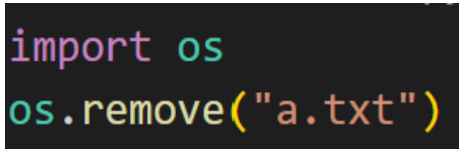


"x" - Create - will create a file, returns an error if the file exist

"a" - Append - will create a file if the specified file does not exist

"w" - Write - will create a file if the specified file does not exist

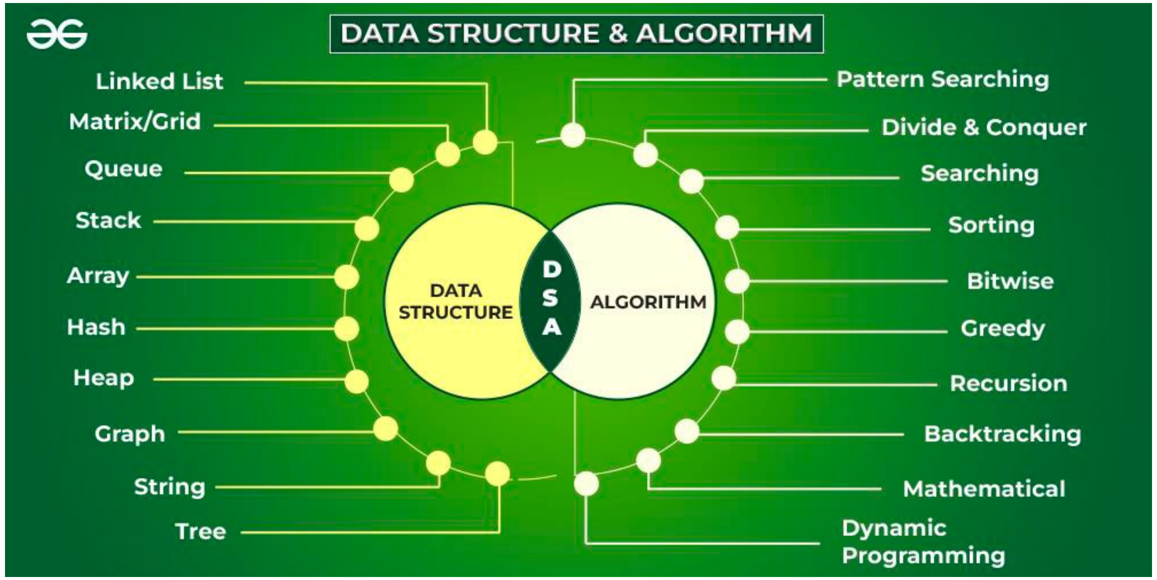
- To delete a txt file.



```
import os
os.remove("a.txt")
```

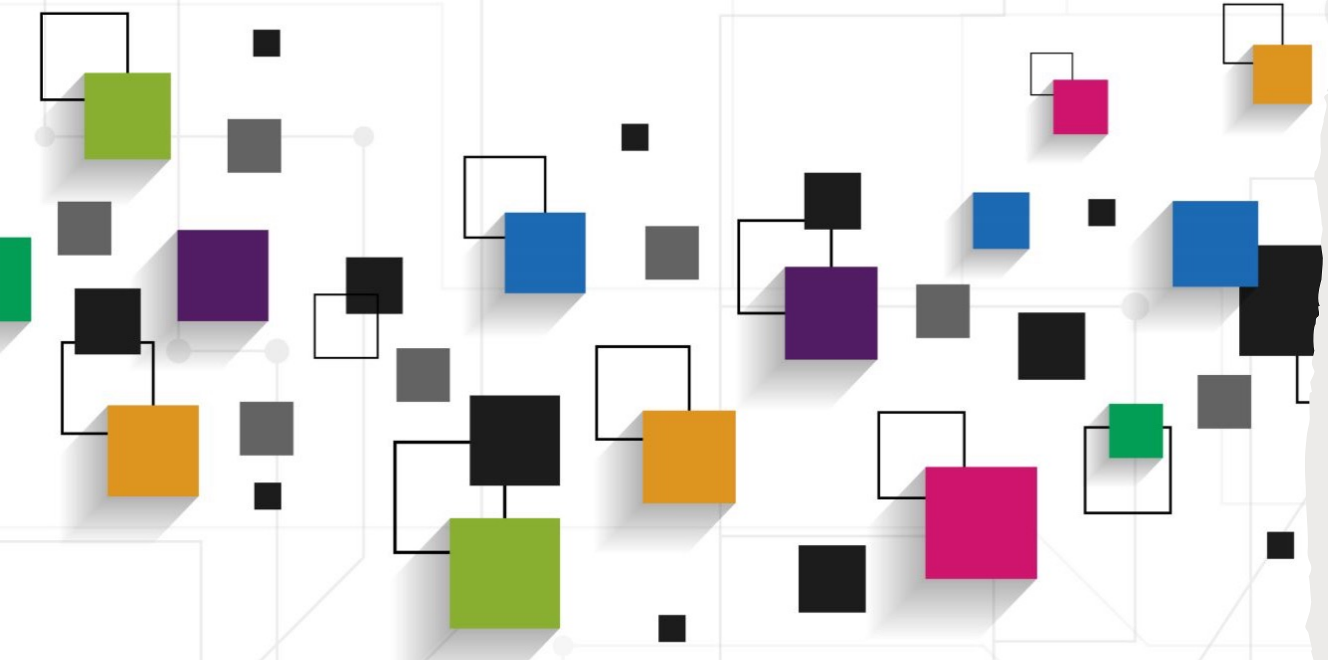
Future Connect Media

Future Consideration



- Data structure algorithms:

Python uses two of the most fundamental ideas in computer science: data structures and algorithms. They are crucial resources for any programmer. In Python, data structures are used to organize and store data in memory while a program processes.



- Web Development

- Graphic User Interface (GUI)

Graphical User Interface (GUI) programming in python supports numerous frameworks or toolkits. From Tkinter, which is typically included with Python, to a number of cross-platform options that you may install as third-party libraries, such PyQT or wxPython.