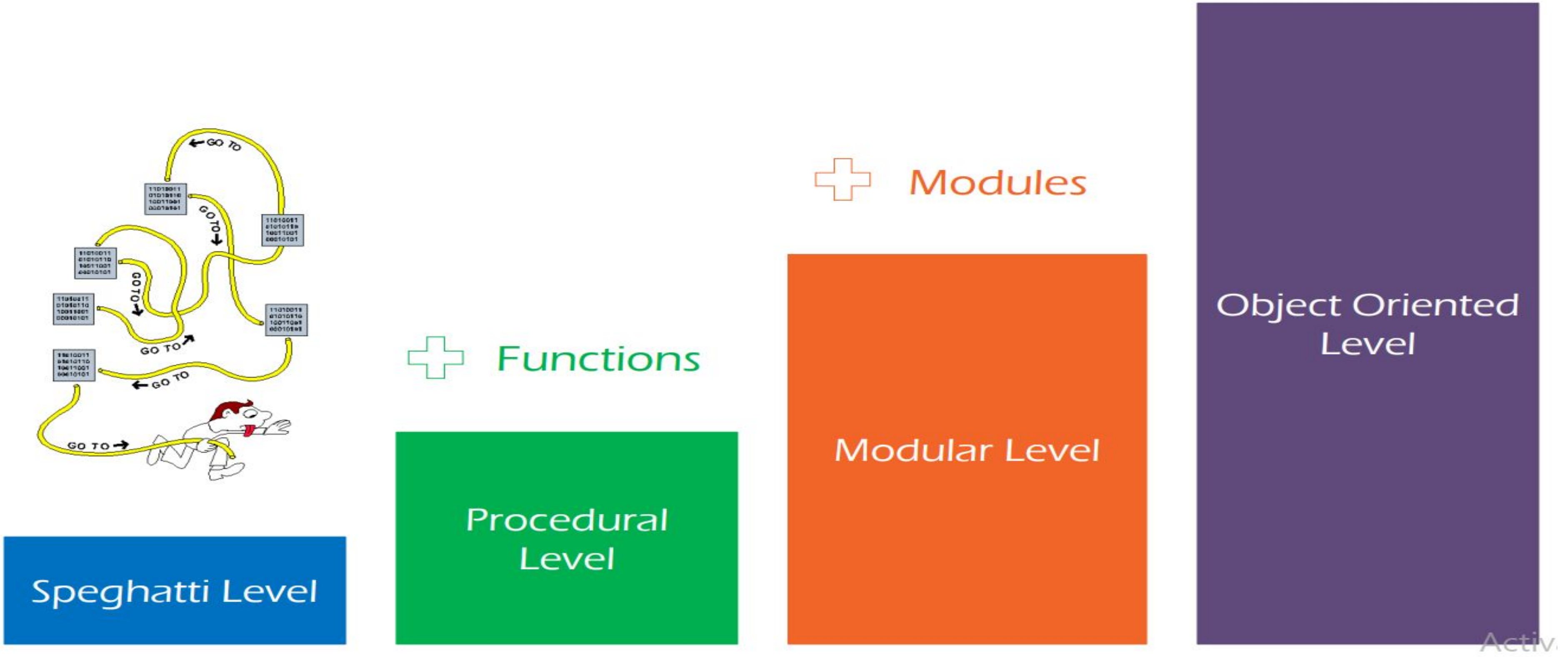
Day 3 Outline

1. Modules and Libraries
2. Exception Handling
3. Read and Write File
4. Standard Libraries
5. External Libraries
6. Day Tips

Python Levels



Speghatti Level

Functions

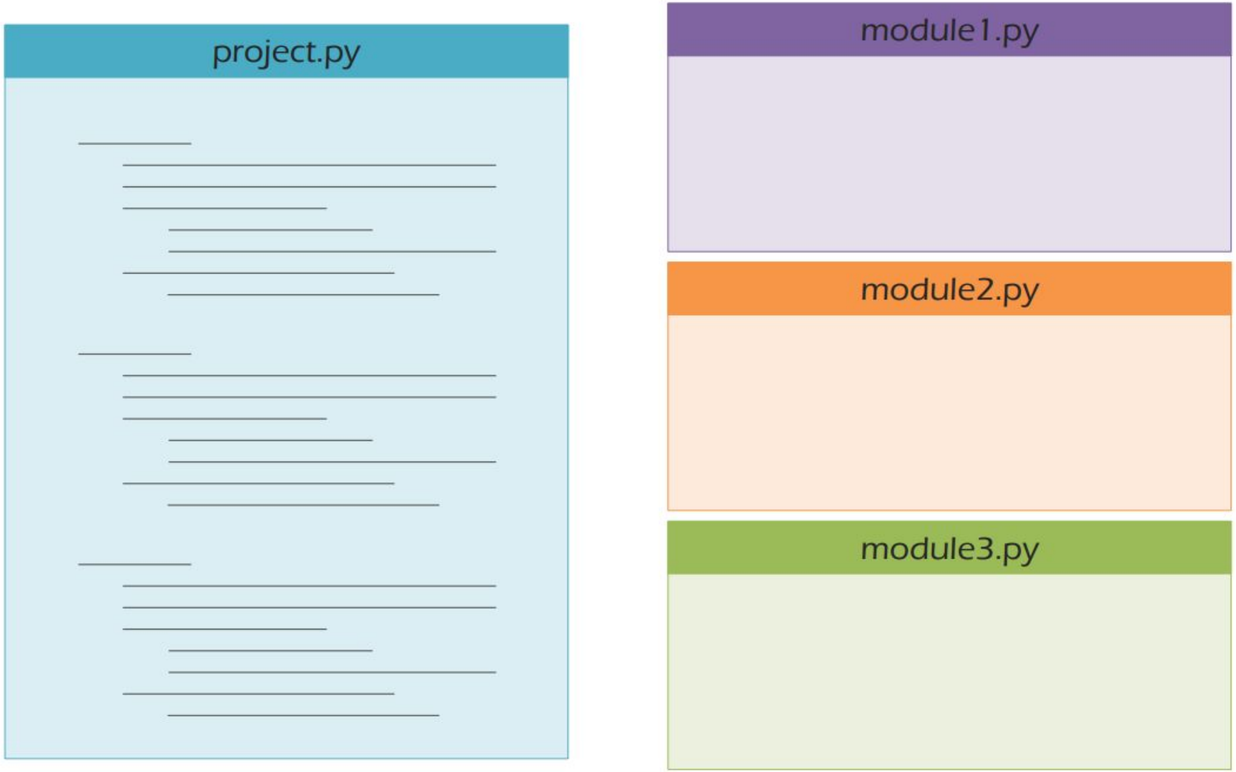
Procedural Level

Modules

Modular Level

Object Oriented Level

Modules



Including Modules in File

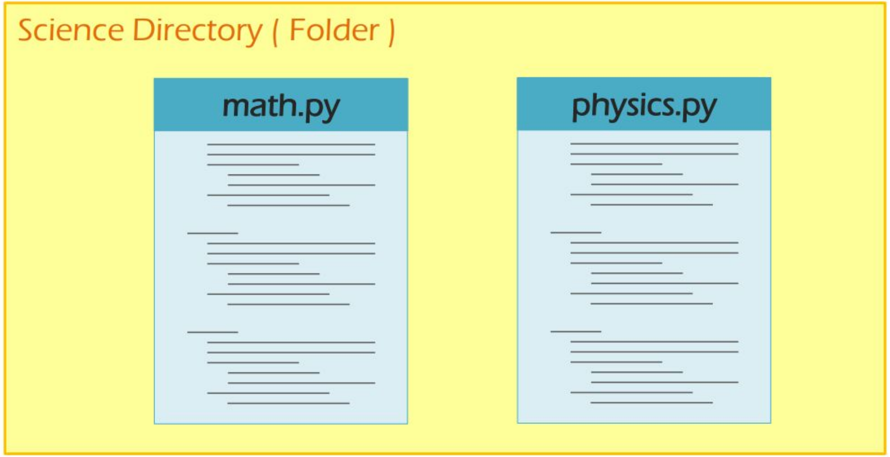
```
from module_name import block_name
```

i.e. from math import tan

A
G

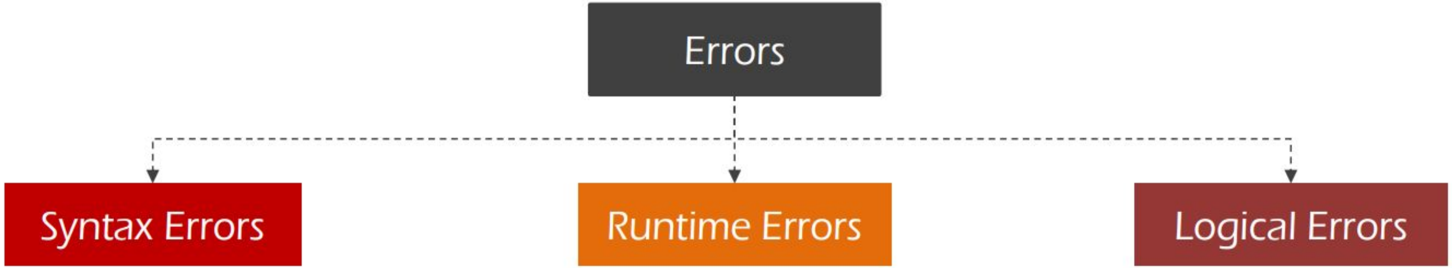
Package Including

```
from pkge_name.module_name import block_name
```

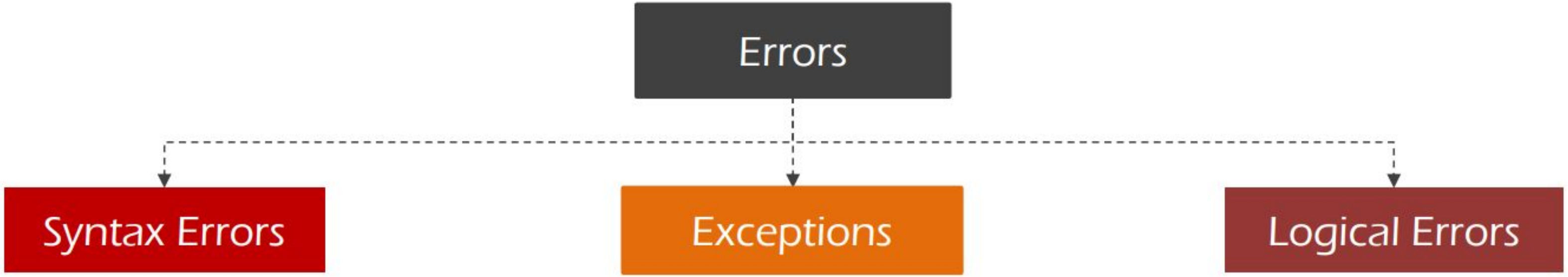


i.e. from science.math import tan

Errors



Errors



Errors

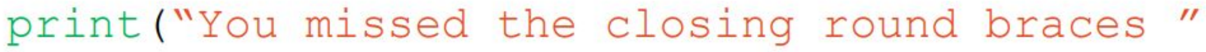
Syntax Errors

Exceptions

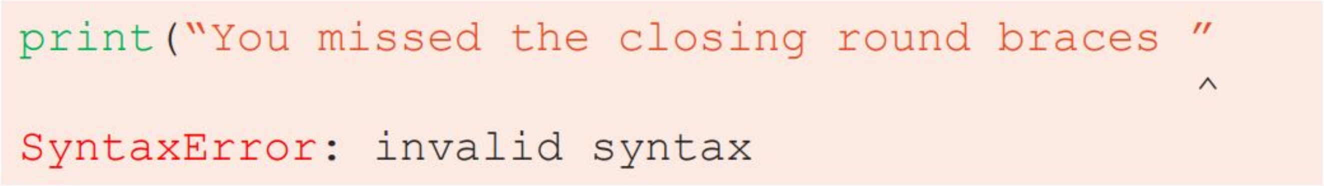
Logical Errors

Syntax Errors

Errors that will show up if you doesn’t follow Python Syntax Rules



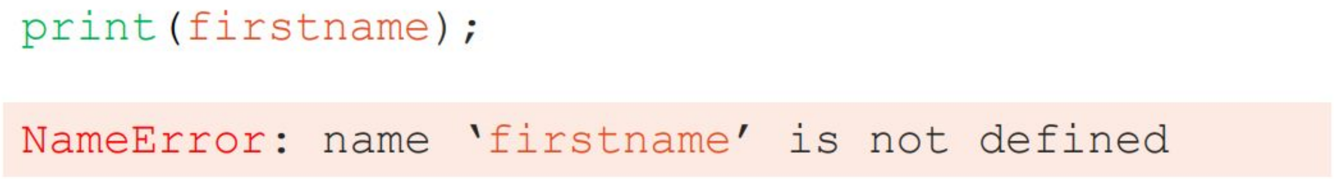
```
print(“You missed the closing round braces ”
```



```
print(“You missed the closing round braces ”
                                           ^
SyntaxError: invalid syntax
```

Exception

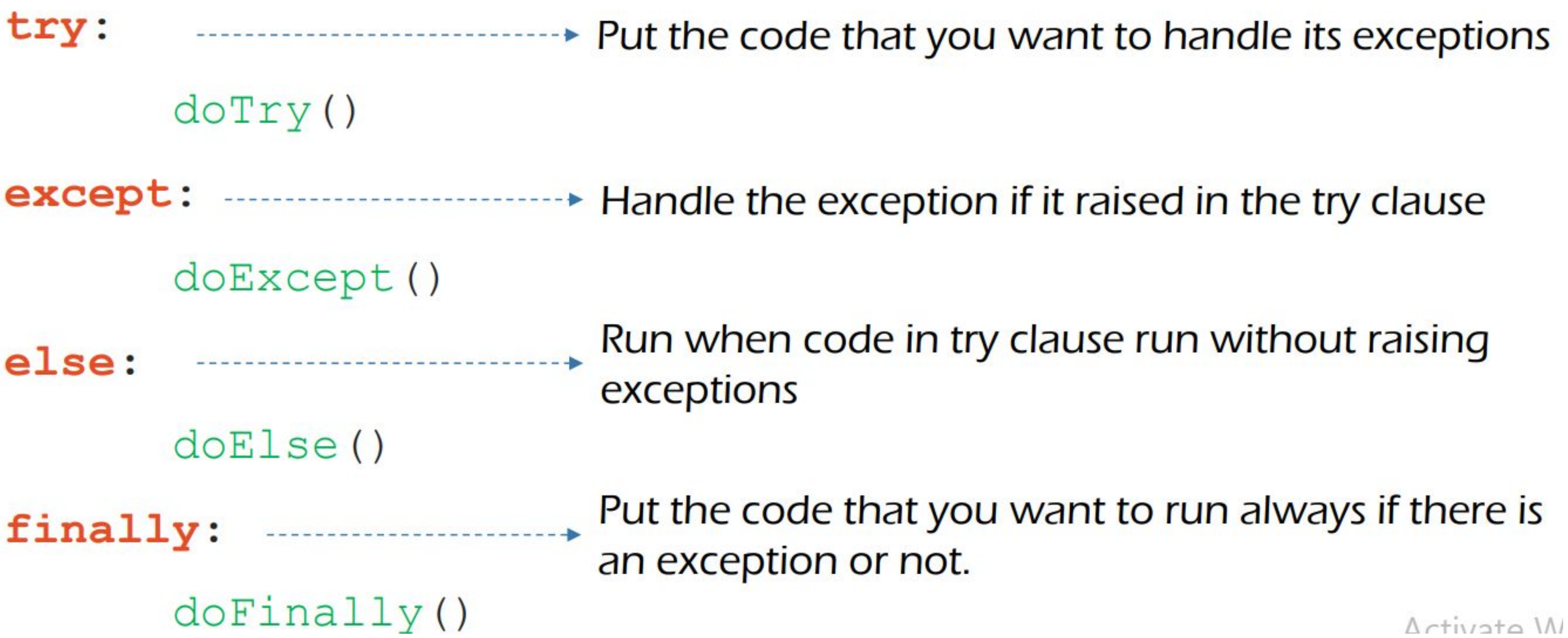
Errors detected during execution are called Exceptions



```
print(firstname);
```

```
NameError: name ‘firstname’ is not defined
```

Exception Handling



```
try:
        doTry()
except:
        doExcept()
else:
        doElse()
finally:
        doFinally()
```

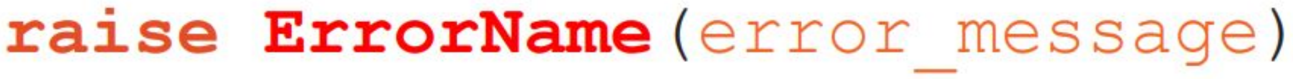
try: → Put the code that you want to handle its exceptions

except: → Handle the exception if it raised in the try clause

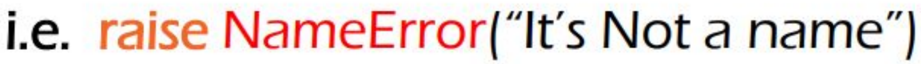
else: → Run when code in try clause run without raising exceptions

finally: → Put the code that you want to run always if there is an exception or not.

Raise Exception



```
raise ErrorName(error_message)
```



i.e. `raise NameError(“It’s Not a name”)`

Exception

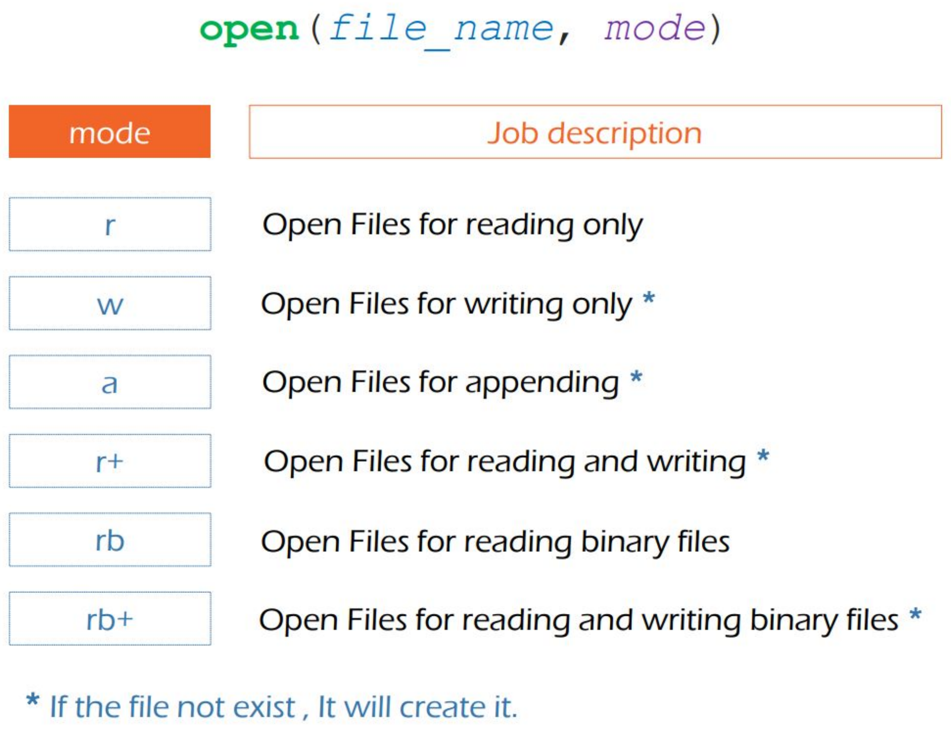
Most Used Exceptions:

1. AssertionError
2. TypeError
3. IndexError
4. NameError
5. ValueError
6. ZeroDivisionError

Problem Time

Program to print the reciprocal of even numbers only , handle zero as Zero Exception and Throw the Exception and Odd as Value Error.

File Input and Output

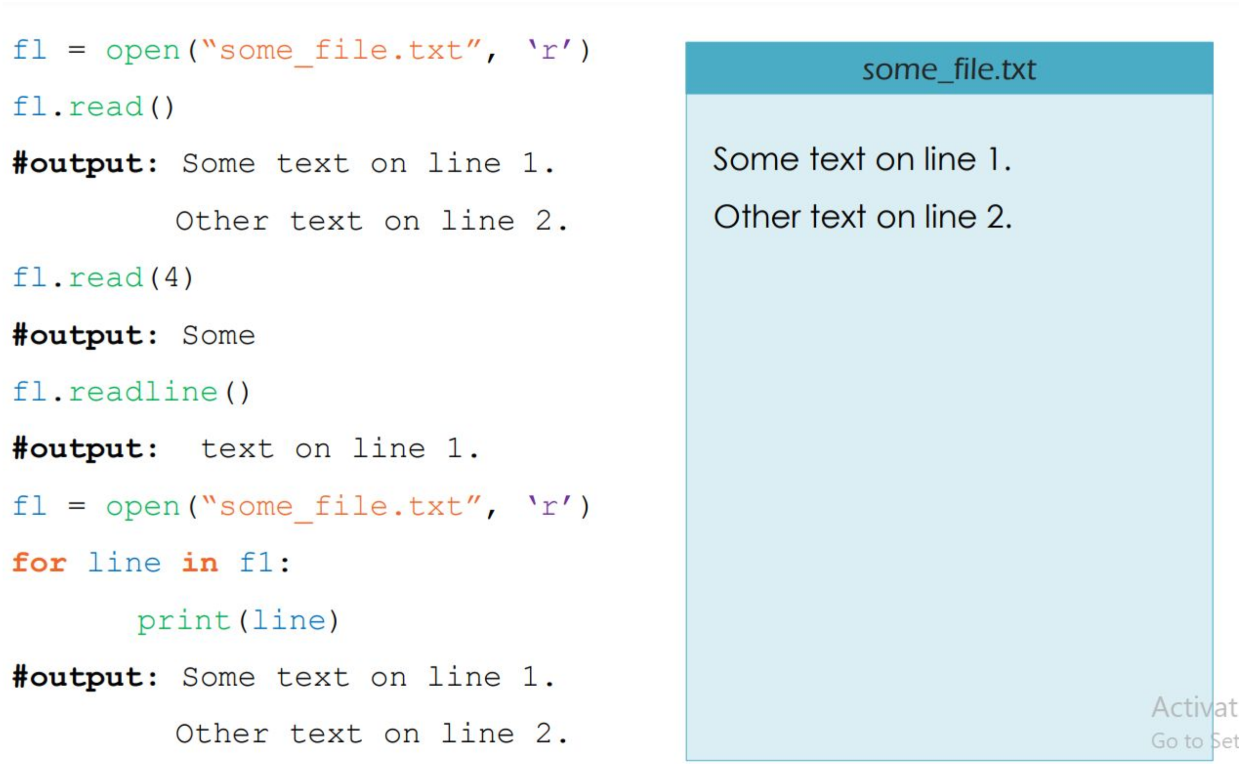


```
open(file_name, mode)
```

mode	Job description
r	Open Files for reading only
w	Open Files for writing only *
a	Open Files for appending *
r+	Open Files for reading and writing *
rb	Open Files for reading binary files
rb+	Open Files for reading and writing binary files *

* If the file not exist , It will create it.

Read File



```
fl = open("some_file.txt", 'r')
fl.read()
#output: Some text on line 1.
         Other text on line 2.
fl.read(4)
#output: Some
fl.readline()
#output:  text on line 1.
fl = open("some_file.txt", 'r')
for line in fl:
      print(line)
#output: Some text on line 1.
         Other text on line 2.
```

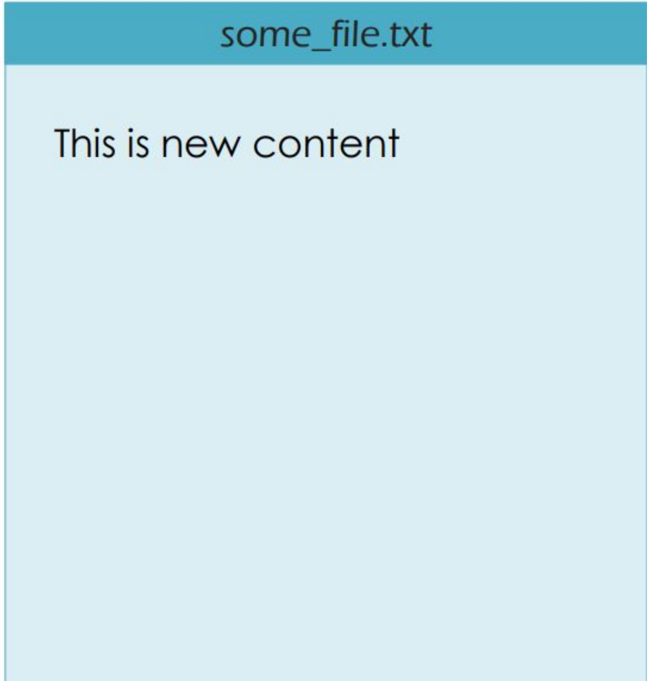
some_file.txt

Some text on line 1.

Other text on line 2.

Write File

```
fl = open(“some_file.txt”, ‘w’)
fl.write(“This is new content”)
```

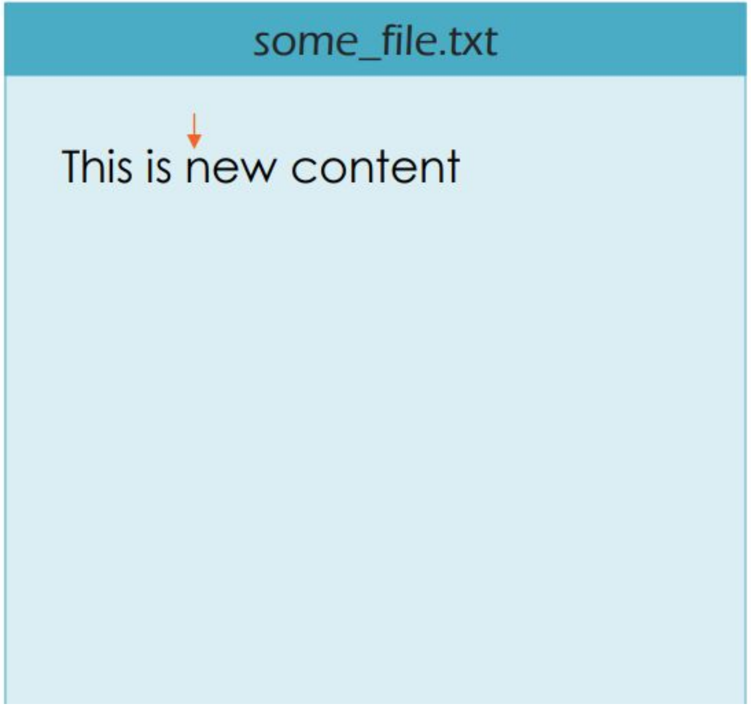


some_file.txt

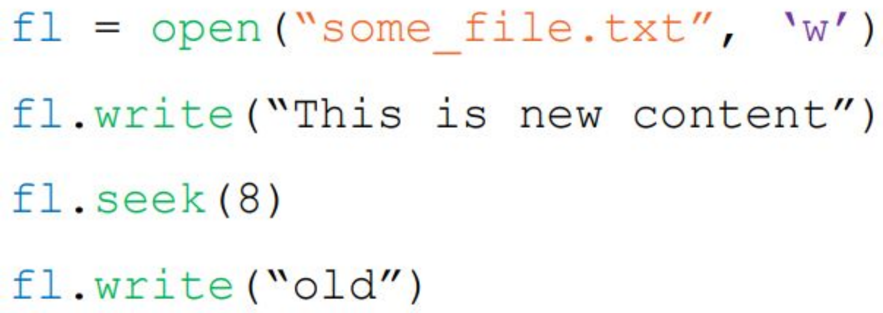
This is new content

Write File

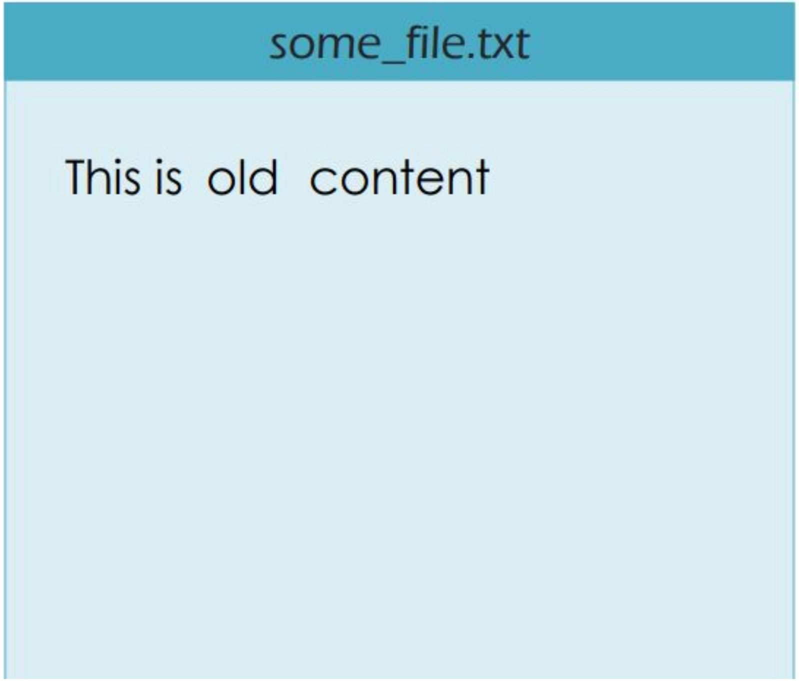
```
fl = open(“some_file.txt”, ‘w’)

fl.write(“This is new content”)

fl.seek(8)
```



Write File



```
fl = open(“some_file.txt”, ‘w’)
fl.write(“This is new content”)
fl.seek(8)
fl.write(“old”)
```

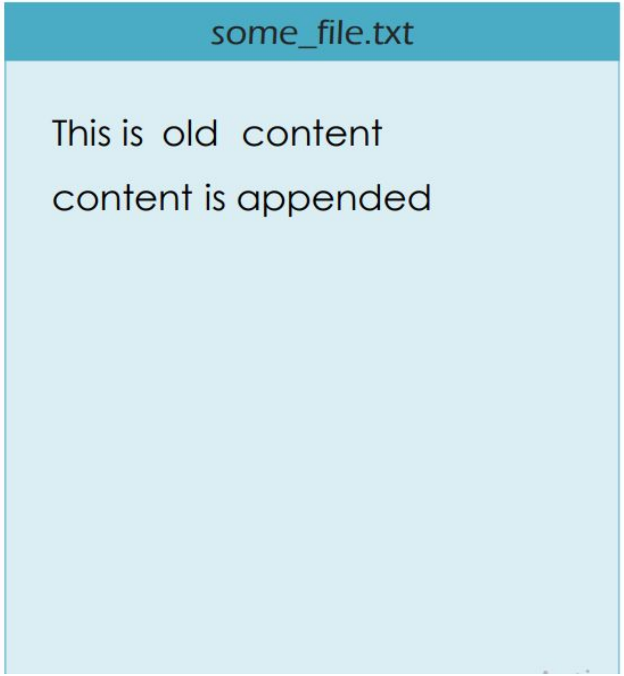


some_file.txt

This is old content

Write File

```
fl = open(“some_file.txt”, ‘w’)

fl.write(“This is new content”)

fl.seek(8)

fl.write(“old”)

fl.close()

fl = open(“some_file.txt”, ‘a’)

fl.write(“\n content is appended”)
```



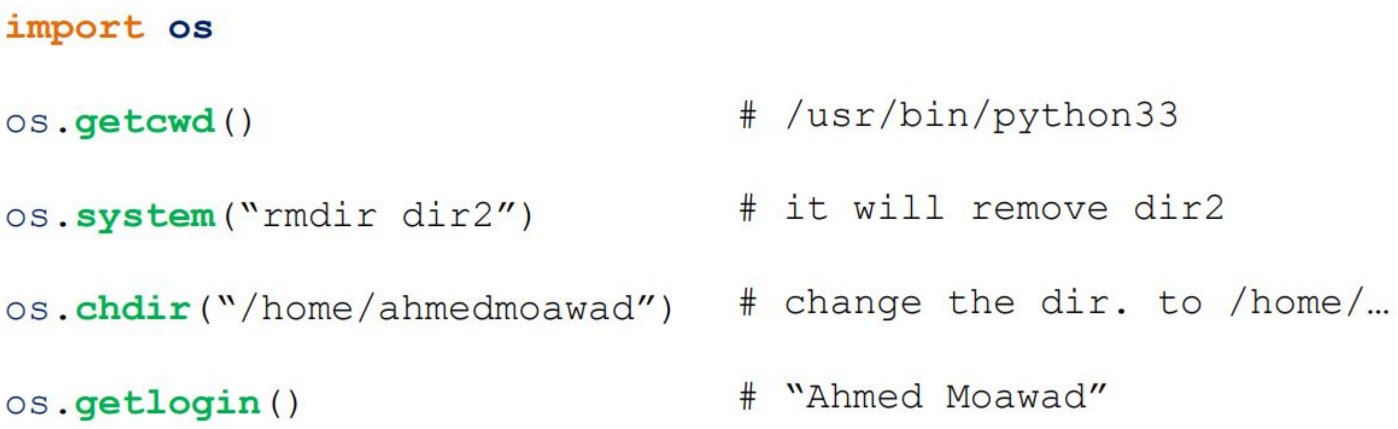
some_file.txt

This is old content

content is appended

Standard Library - OS

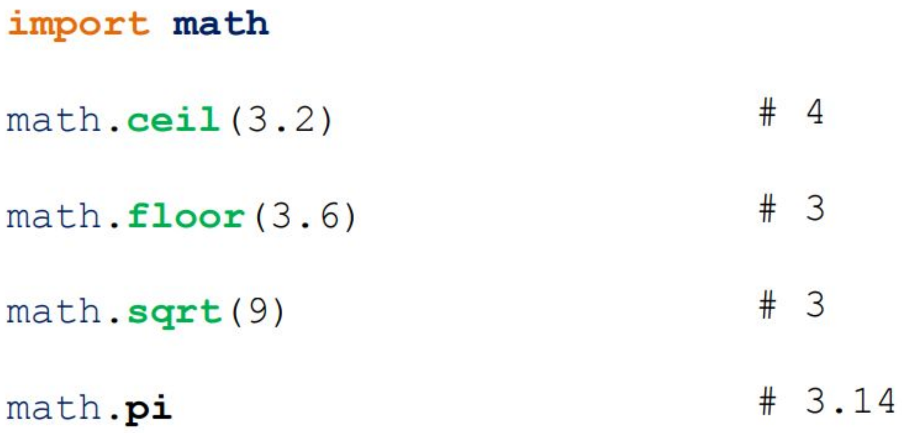
os module provides functions for interacting with the operating system



```
import os

os.getcwd()                       # /usr/bin/python33

os.system(“rmdir dir2”)           # it will remove dir2

os.chdir(“/home/ahmedmoawad”)     # change the dir. to /home/…

os.getlogin()                     # “Ahmed Moawad”
```

Standard Library- math

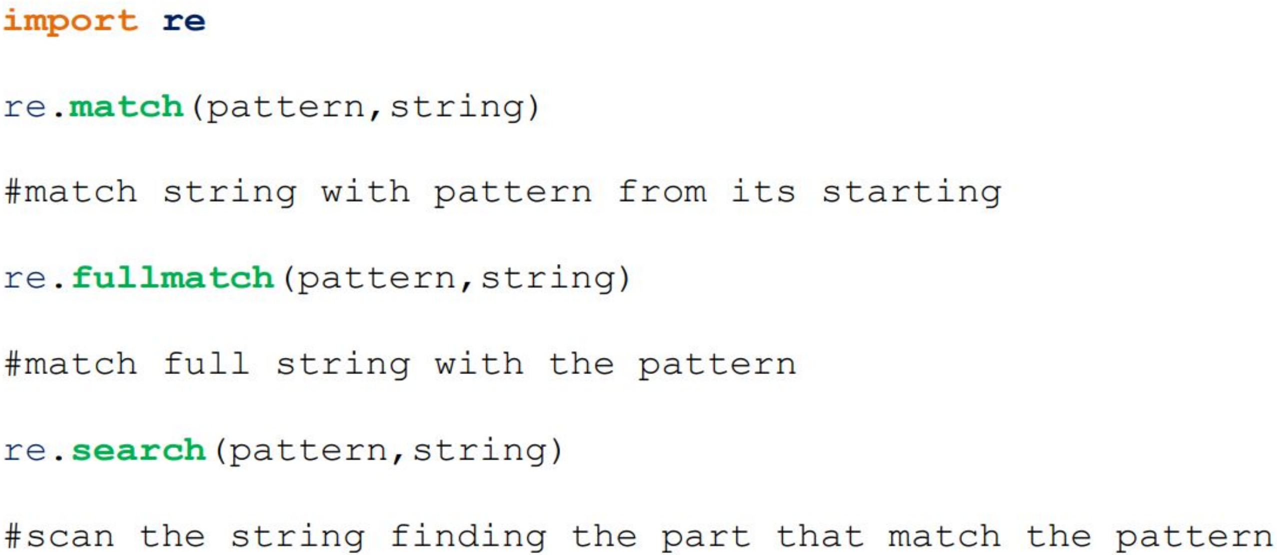
math module provides access to the mathematical functions by the C standard



```
import math

math.ceil(3.2)                  # 4

math.floor(3.6)                 # 3

math.sqrt(9)                    # 3

math.pi                         # 3.14
```

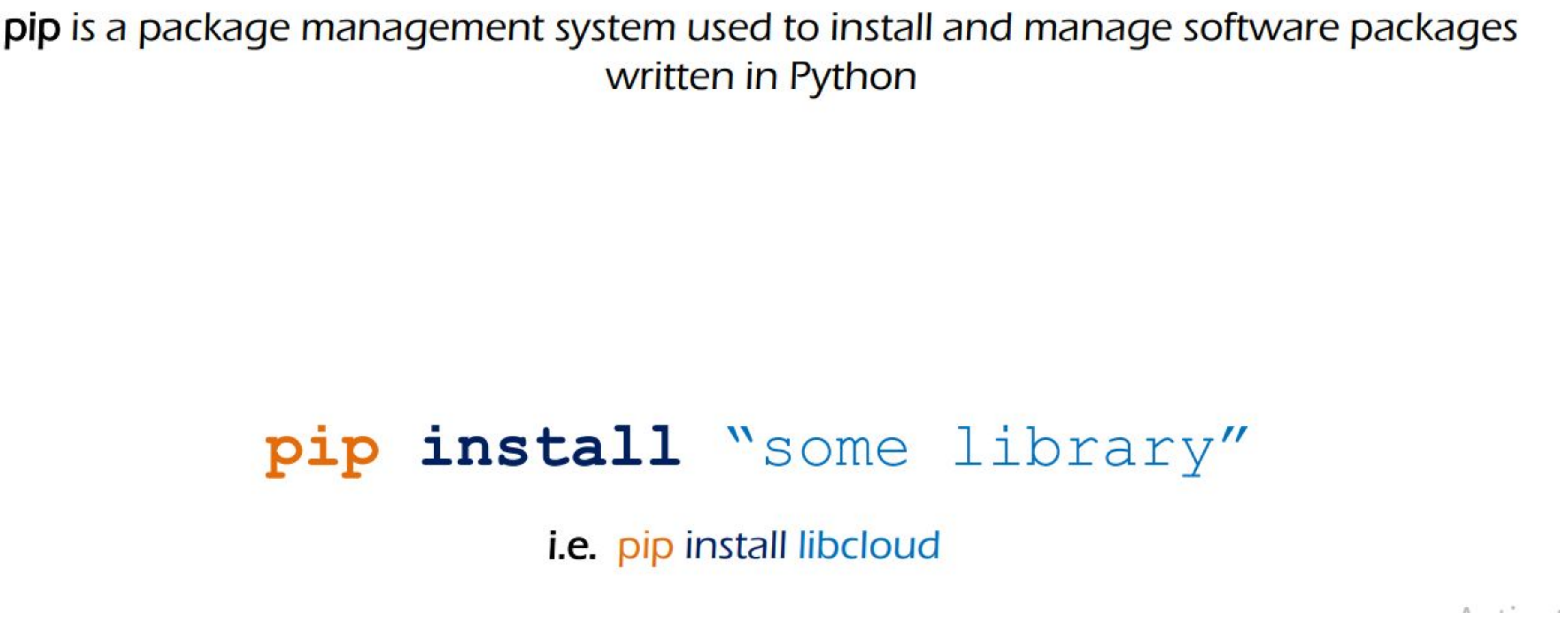
Standard Library - re

re provides regular expression matching operations



```
import re

re.match(pattern,string)

#match string with pattern from its starting

re.fullmatch(pattern,string)

#match full string with the pattern

re.search(pattern,string)

#scan the string finding the part that match the pattern
```

External Library - Pip



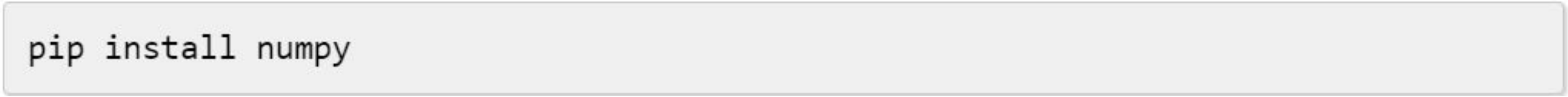
pip is a package management system used to install and manage software packages written in Python

```
pip install "some library"
```

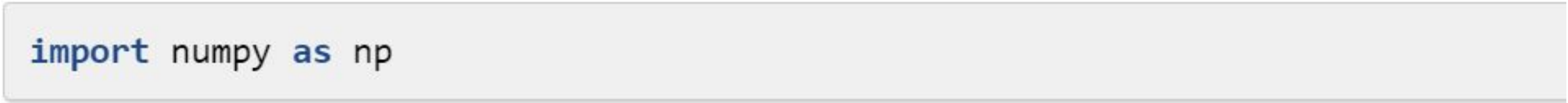
i.e. pip install libcloud

External Library - Numpy

- NumPy is a Python library used for working with arrays.
- It also has functions for working in domain of linear algebra, fourier transform, and matrices.



```
pip install numpy
```

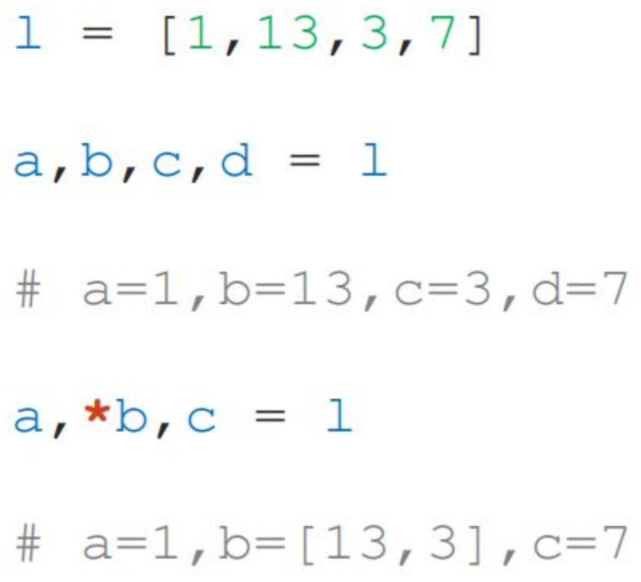


```
import numpy as np
```

External Libraries

- **Matplotlib** -> used in **visualization**
- **Pandas** -> used in opening files and creation of **dataframe**
- **Django** -> Framework built on python for **Web development**
- **Flask** -> Framework built on top of python for **Web development**
- **CherryPy** -> Framework built on python for **Web development**
- **Arcade** -> is a python **game development** library to design 2D video games (openGL)
- **PyGame** -> is a game development library designed for developing video games in Python.
- **Scikit-learn** is one of the most popular ML libraries for classical **ML** algorithms
- **Pytorch , TensorFlow ,,,,,,**

Tip7 - Sequence Unpacking



```
l = [1,13,3,7]

a,b,c,d = l

# a=1,b=13,c=3,d=7

a,*b,c = l

# a=1,b=[13,3],c=7
```

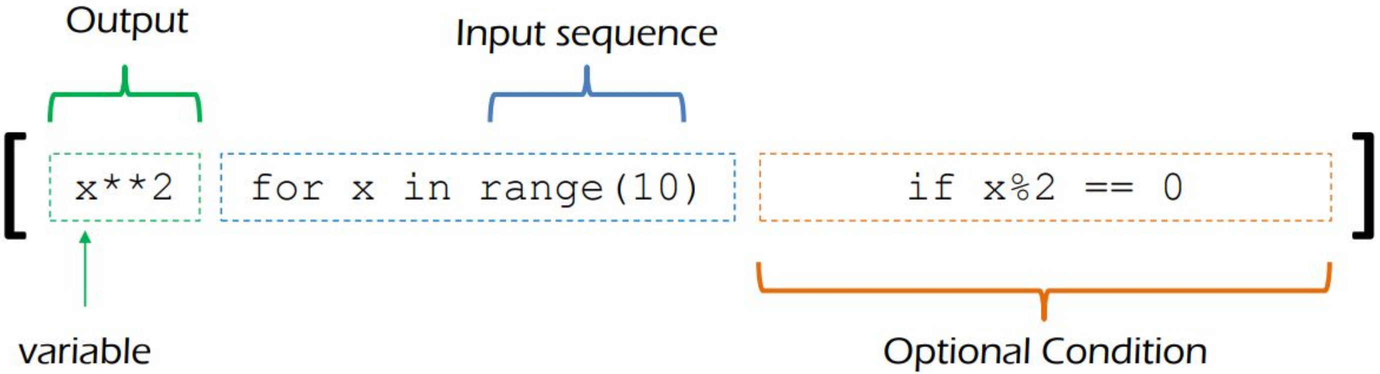
Tip8 - with

with statement is used for handling the entry (set-up) and exit (tear-down) tasks for its input

```
with open(“file.txt”, ‘r’) as fp:
    fp.read()
```

Tip9 - list Instruction

It is an easy method to construct a list



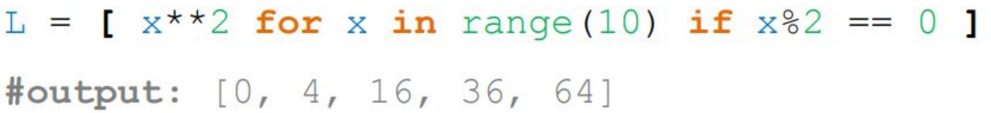
Output

Input sequence

```
[ x**2  for x in range(10)  if x%2 == 0 ]
```

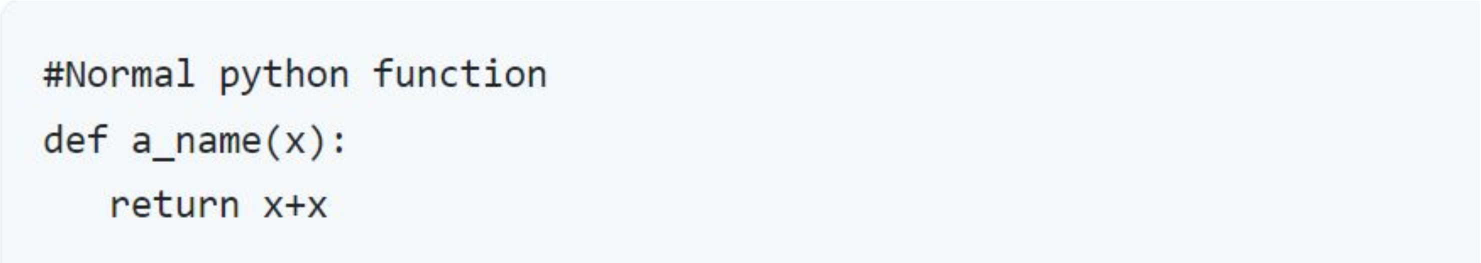
variable

Optional Condition

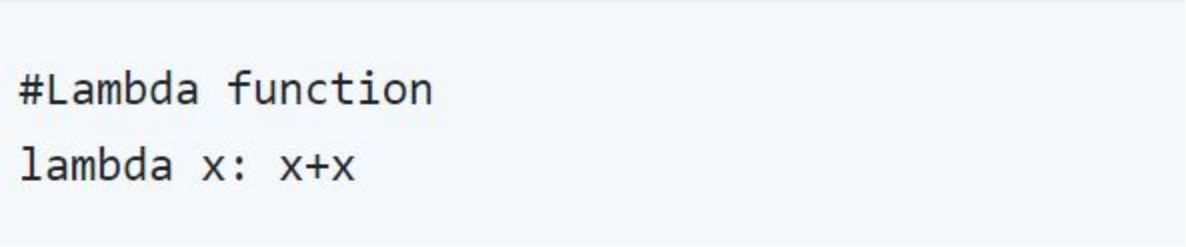


```
L = [ x**2 for x in range(10) if x%2 == 0 ]

#output: [0, 4, 16, 36, 64]
```

Tip10 - lambda Function



```
#Normal python function
def a_name(x):
   return x+x
```



```
#Lambda function
lambda x: x+x
```

Problems

- Write a function in Python that accepts two string parameters. The first parameter will be a string of characters, and the second parameter will be the same string of characters, but they'll be in a different order and have one extra character. The function should return that extra character.
    - For example, if the first parameter is "eueiieo" and the second is "iieoedue," then the function should return "d."
- Import the above function in another program (file) and call it .

Problems

- Write a program that accepts the lengths of three sides of a triangle as inputs. The program output should indicate whether or not the triangle is a right triangle (Recall from the Pythagorean Theorem that in a right triangle, the square of one side equals the sum of the squares of the other two sides).

- Write a function that accepts a number as a parameter. The function should return a number that's the difference between the largest and smallest numbers that the digits can form in the number.
    - For example, if the parameter is "213", the function should return "198", which is the result of 123 subtracted from 321.

Problems

- Create a function in Python that accepts one parameter: a string that’s a sentence. This function should return True if any word in that sentence contains duplicate letters and False if not.   Hint check Data structure = set()