

**DATA SCIENCE**

**PYTHON**

Whitespace Team



# OUR TEAM :



Andhika  
Utomo



Dean  
Setyawan



Heru  
Stiawan



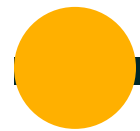
M Haikal  
Febrian p



Raisya  
Amanah N

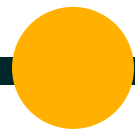
# Porto Overview

## Data Science Track – Python



### ***def***

one of built in feature in python to make a function/formula so we can call it with def name anytime in python and don't need to write the formula everytime we need it



### ***class***

class is one of built in feature in python that consist of two or more def



### ***library python***

library is consist of some module that have been made by someone so we can use it as a simple shortcut for some program that should have required a lot of coding to run

# def function in python

## def overall meaning

def is a keyword that define function in python. so, def use for make a function different with write formula manual. with def feature we can make formula in the beginning and call it with name of the function with its parameter inside parentheses beside function name

# few rules in def



## simple def coding format

```
def function_name(parameter):  
    variabel=your_function  
    return variabel
```



## return rules

it's possible to write def coding without return statement. Functions like this are called void, and they return None. so, no value will be return to def. if we try to call its def will come out none for any parameter we input



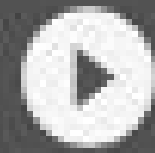
## print vs return in def

print and return make a similar output in python but there is a different between them. if we use print instead of return mean we want to show value after parameter input into function at console(output) and return mean we return its value to the def name so its still inside program as one of variable without using print function

# example def

simple def

**code  
program**



```
def exponential(x,y):  
    quadratic=x**y  
    return quadratic  
  
print(exponential(2,3))
```

**output**



8

# example def

a little complex def

**code  
program**



```
def fun_looping(n):  
    if 0<=n<=100:  
        for i in range(n):  
            if i%2==0:  
                print(f"{i} is an even number")  
            else:  
                print(f"{i} is an odd number")  
    else:  
        while n>100:  
            print(n)  
            n-=100  
n=int(input("n= "))  
fun_looping(n)
```

**output**

```
n= 300  
300  
200
```





# iterable parameter

---

for iterable parameter we need a special symbol so python can read it as iterable parameter and not as single parameter

## dictionary (\*\*)

to read iterable dictionary need a few thing to note. first, add \*\* infront of parameter name and for calling

## other iterable (\*)

to read other iterable except dictionary. we need to add \* infront of parameter nam and for calling



# example iterable parameter

**code  
program**



```
def info(*position,**dictionary):  
    for i in position:  
        print("argument position is ",i)  
    for key, value in dictionary.items():  
        print("argument key word {}:{}".format(key,value))  
  
info("one",key="two",value="3")
```

**output**

```
argument position is one  
argument key word key:two  
argument key word value:3
```

# lambda

## another function in python

### lambda overall meaning

a lambda function is just like any normal python function, except that it has no name when defining it, and it is contained in one line of code

# example lambda function

**code  
program**



```
energy= lambda momentum, massa, potensial: (momentum/(2*massa))+potensial  
energy(20,10,5)
```

**output**

6.0

# Class

In Python, classes are defined using the class keyword. Class is made to represent a certain object, so it will assist in the process of solving complex problems by stating the physical characteristics of a problem. Classes in Python The default has a public Access Specifier.

Inside the class there are class variables. A class variable is a variable that is shared or shared by all instances (children) of the class. class variable defined inside the class, but outside the methods in the class.

Then there is the object. Object is an instance or embodiment of a class. If class is the prototype, and object is the finished product.

# Example :

## Program Code :

```
[ ] from abc import ABC,abstractmethod
    class phona(ABC):
        @abstractmethod
        def total_feet(self):
            pass

    class cat(phona):
        def total_feet(self,total_feet1):
            print("Total feet of cat is : ",total_feet1)

    class chicken(phona):
        def total_feet(self,total_feet2):
            print("Total feet of chicken is : ",total_feet2)

    animal1 = cat()
    animal2 = chicken()

    animal1.total_feet(4)
    animal2.total_feet(2)
```

## Output :

```
Total feet of cat is : 4
Total feet of chicken is : 2
```

# Constructor Method

The constructor method is a special method used Python to initialize the creation of an object of a class. The constructor method is defined as the "init()" function, which always executed when the class is being started. Function "init()" function to assign values to object properties, or other operations that need to be performed when the object is created.

In the above source code "self" is the reference parameter to instance of the current class, and is used to access variables that included in the class. The parameter does not have to be named "self", user can name any user, but the parameter must be the first parameter of each function in the class. Then the way to call a class is slightly different from when we call the function. In order to summon a class, we must first declare an object as in the example above "p1". After successfully creating an object, we can display the contents of the class with the command "print(ObjectName.MethodeName)"

# Example :

## Program Code :

```
▶ class pople:  
    count = 0  
    def __init__(self, name, old):  
        self.name = name  
        self.old = old  
    count += 1  
p1=pople("ryan",20)  
print(p1.name)  
print(p1.old)  
print(p1.count)
```

## Output :

```
ryan  
20  
1
```



The previously defined rectangular class has two attributes (length and width) and three methods: `init()`, `area()`, and `around()`.

The `init()` method inside the class serves to initialize or fill in the required initial values. In the `init` method there are three parameters, namely `self`, `p`, and `l`. The first parameter, `self`, must be included in OBJECT ORIENTED PROGRAMMING MODULE 84 for each method defined in the class and need not be specified at the time of call. `self` in Python is the same as it is in Java and C++.

To define or access the attributes of a class, we also need to use the `self` keyword. For example, in the code above we define two attributes in the Rectangle class i.e. the width is taken from the parameter `l`. So the code is written as follows:



# Example :

## Program Code :

## Output :

## Program Code :

## Output :

```
class rectangular :  
    def __init__(self,p,l):  
        self.tlong = p  
        self.wide = l  
    def large(self):  
        return self.tlong *self.wide  
    def aroung(self):  
        return 2 *(self.tlong +self.wide)  
obj = rectangular(8,6)  
print(obj.large())  
print(obj.aroung())
```

48  
28

```
class sunmation():  
    def add(self,*args):  
        result = 0  
        for i in args:  
            if type (i==int):  
                result+=i  
        return result  
obj_1 = sunmation()  
print(obj_1.add(3,1,3))  
print(obj_1.add(3,1))
```

7  
4

# OS Path

The `os.path` module is a very extensively used module that is handy when processing files from different places in the system.

It is used for different purposes such as for merging, normalizing and retrieving path names in python . All of these functions accept either only bytes or only string objects as their parameters. Its results are specific to the OS on which it is being run.

# txt write or append

using os library we can input data inside txt file we provide or make new txt file if txt file not available. after this slide there will be an example of this code program and result in creation new txt file because there no txt file available

# Example make new txt file:

## Program Code :

```
import os

if os.path.isfile('log.txt'):
    writefile = open('log.txt','a')
else:
    writefile = open('log.txt','w')

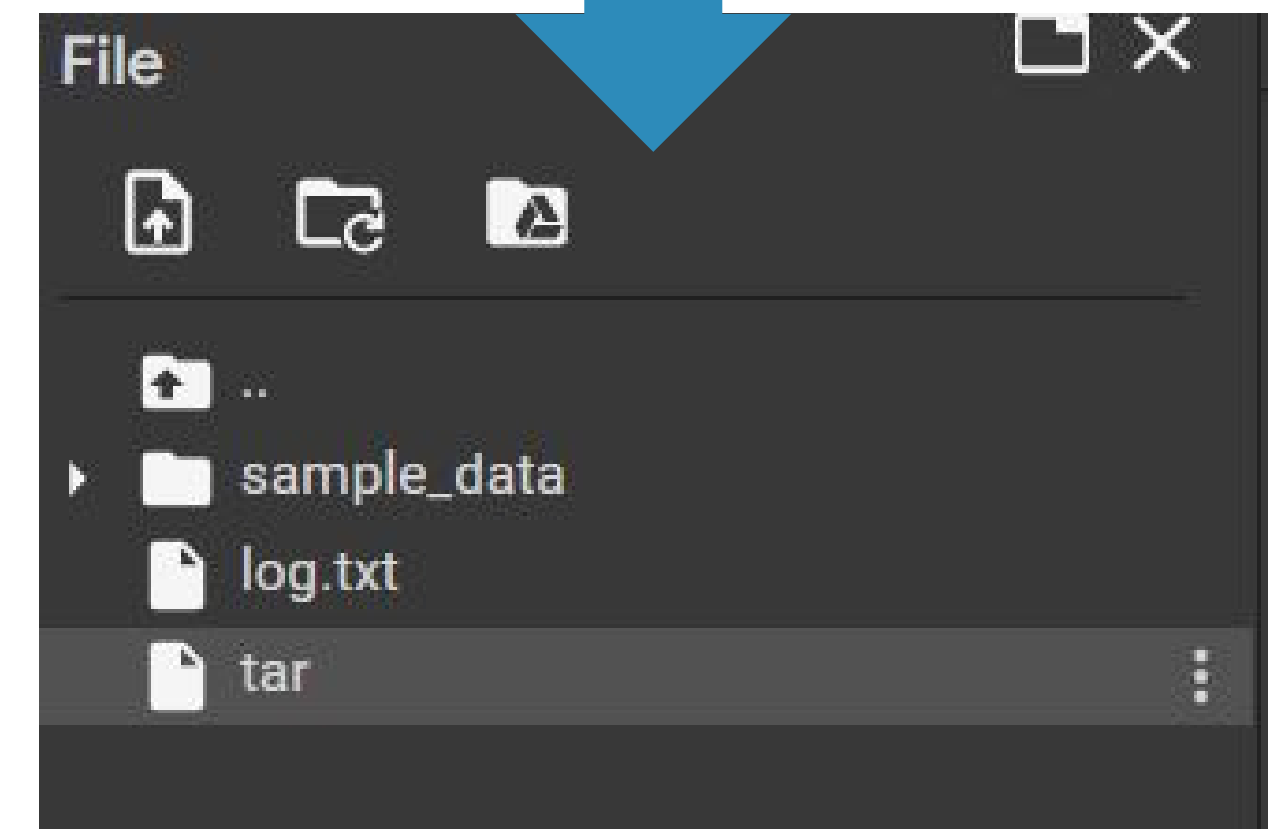
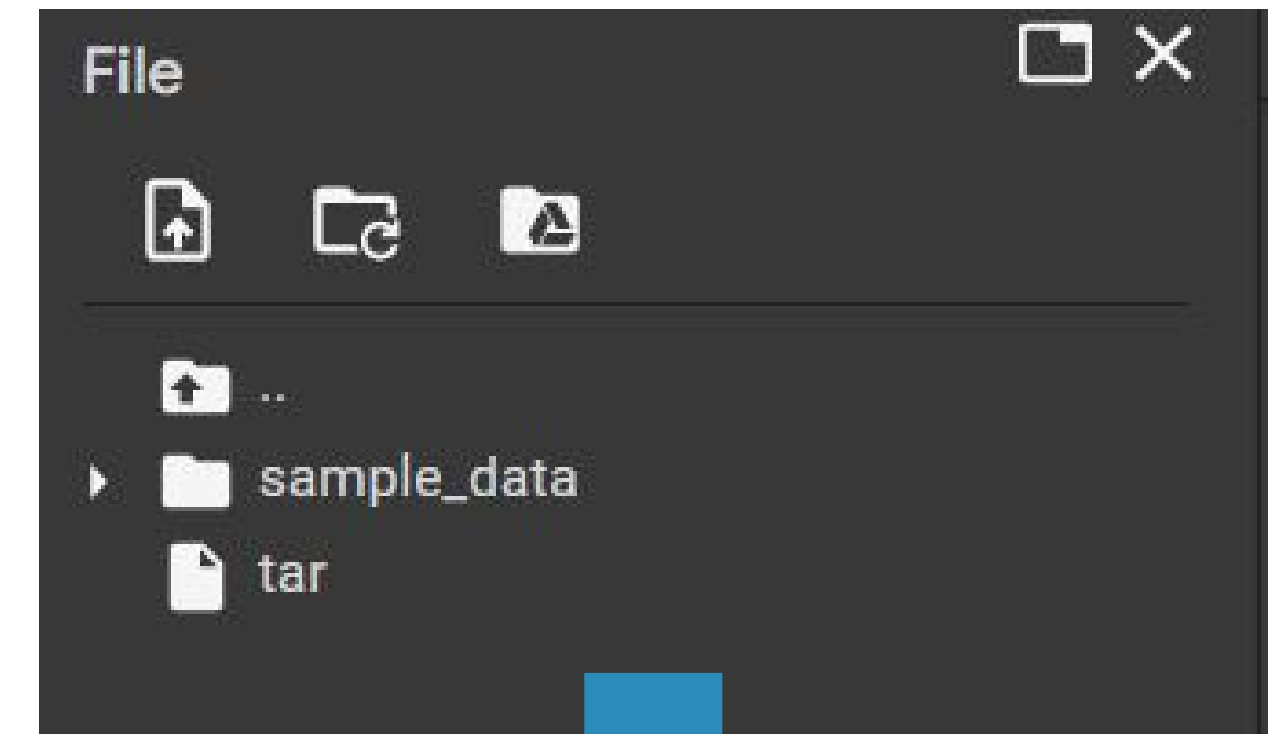
tolog = input("Masukkan angka ")
writefile.write("\n"+tolog)
writefile.close()
```

Masukkan angka

## Output :

Masukkan angka 789

## result :



# Example make new txt file:

## Program Code :

```
import os

if os.path.isfile('log.txt'):
    writefile = open('log.txt','a')
else:
    writefile = open('log.txt','w')

tolog = input("Masukkan angka ")
writefile.write("\n"+tolog)
writefile.close()
```

Masukkan angka

## Output :

Masukkan angka 789

## result:

log.txt X

1

2 789

log.txt X

1

2 789

3 789

# folder creation

using os library, it not only possible to create txt file or add data into txt file but it also possible to add a new folder like example after this slide



# folder creation

**Example :**

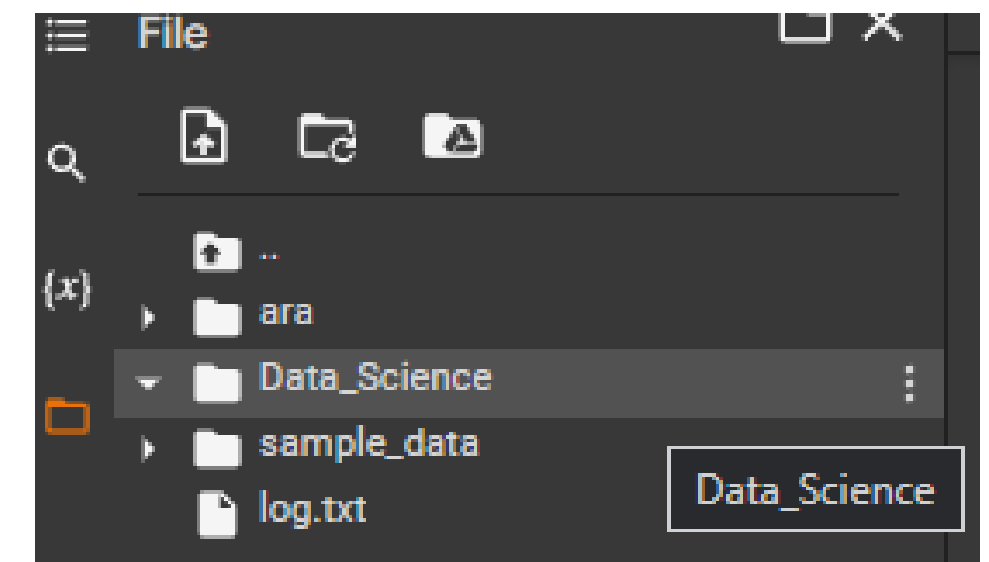
**Program Code :**

```
import os  
  
dirname = input("Masukkan nama folder ")  
os.mkdir(dirname)  
print("directory created")
```

... Masukkan nama folder

**Output :**

```
Masukkan nama folder Data_Sciece  
directory created
```



# delete txt file

using os library it is possible to delete or remove an spesific txt file like example after this slide

# txt delete

## Program Code :

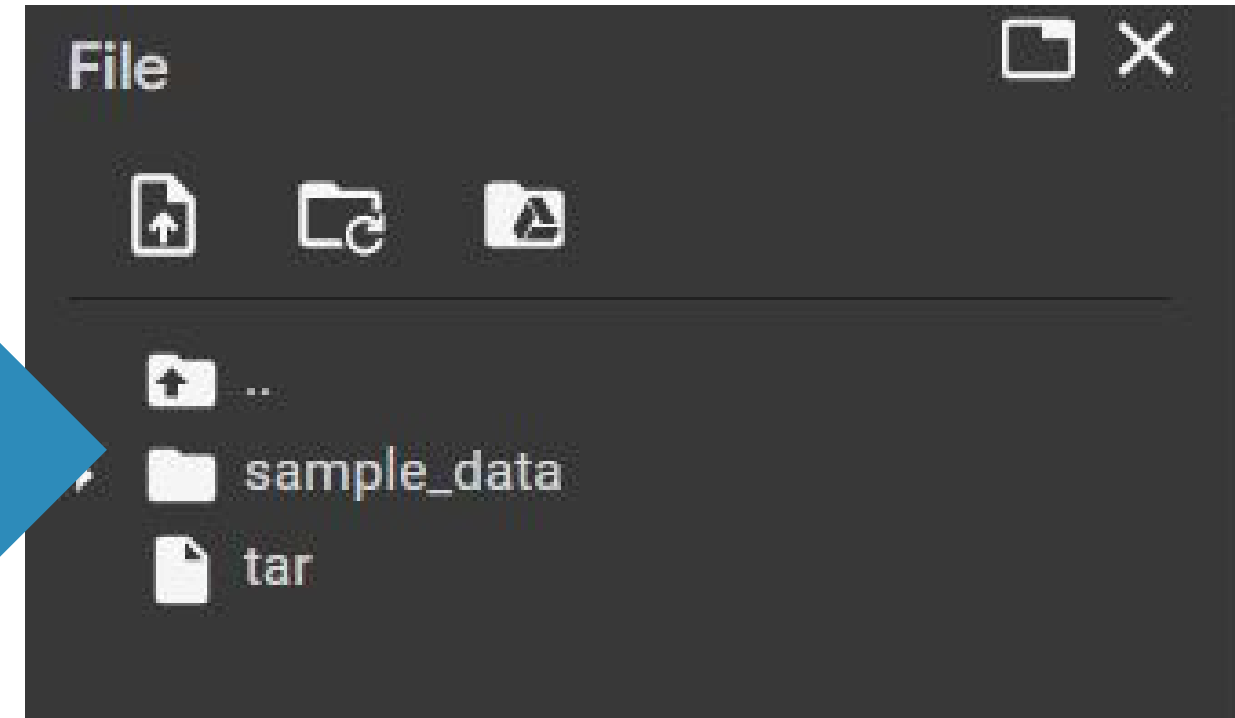
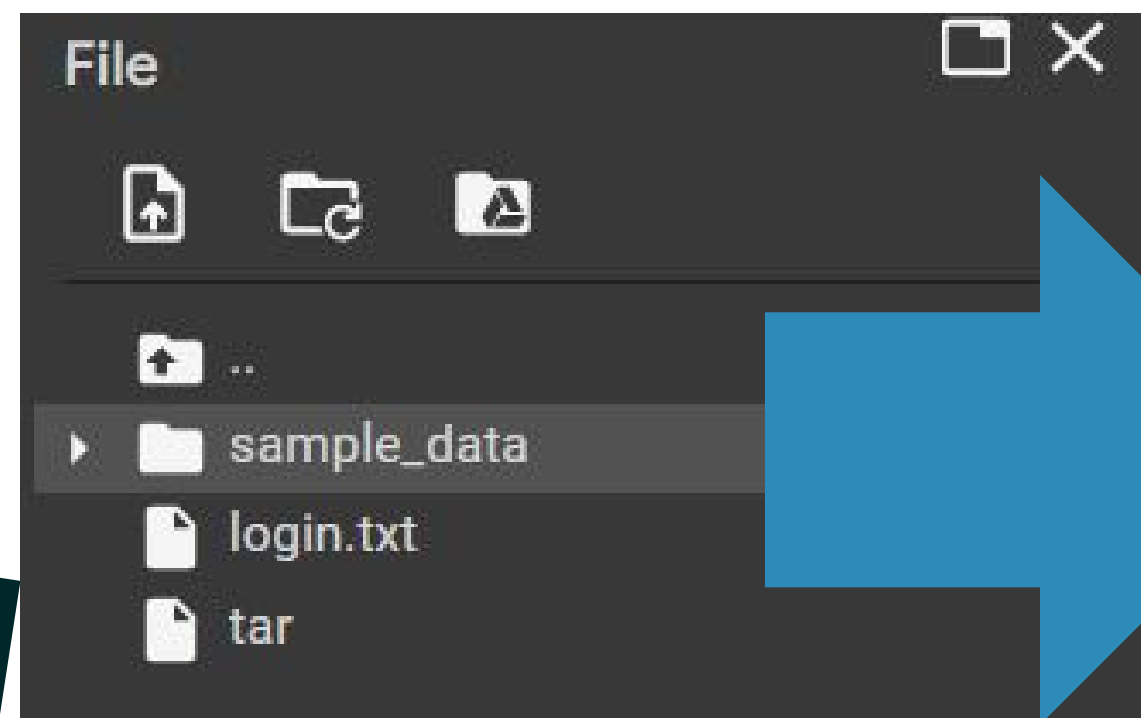
```
import os

[ ] if os.path.isfile('login.txt'):
    os.remove('login.txt')
    print("berhasil")
else:
    print("gagal")
```

## output in python:

berhasil

## result :



# Example :

## Program Code :

```
[4] import os

[5] if os.path.isfile('login.txt'):
    os.remove('login.txt')
    print("berhasil")
else:
    print("gagal")

gagal
```

## Output :

the purpose of the program code above is that if there is a file referred to in the program code (login.txt) it will be deleted. if successful it will display the word 'berhasil' otherwise 'gagal'

# math lib

The Python Math Library provides us access to some common math functions and constants in Python, which we can use throughout our code for more complex mathematical computations.

The library is a built-in Python module, therefore you don't have to do any installation to use it. In this article, we will be showing example usage of the Python Math Library's most commonly used functions and constants.

# Example :

**Program Code :**



```
[18] import math  
  
     pi = math.pi  
  
     print(pi)
```

**Output :**

```
3.141592653589793
```

# rounding up and down using math lib

## Example :

**Program Code :**

**Output :**

**Program Code :**

**Output :**

```
✓ [21] upperbound = math.ceil(pi)  
0 d print(upperbound)
```

4

```
✓ [22] lowerbound = math.floor(pi)  
0 d print(lowerbound)
```

3



# date time lib

In Python, date and time are not a data type of their own, but a module named datetime can be imported to work with the date as well as time.

Python Datetime module comes built into Python, so there is no need to install it externally.

Python Datetime module supplies classes to work with date and time.

These classes provide a number of functions to deal with dates, times and time intervals. Date and datetime are an object in Python, so when you manipulate them, you are actually manipulating objects and not string or timestamps.

# Example :

```
✓ [23] import datetime  
0 d      today = datetime.datetime.today()  
          print(today)
```

```
2022-04-10 03:23:41.896083
```

```
✓ [24] todaywithouttime = datetime.date.today()  
0 d      print(todaywithouttime)
```

```
2022-04-10
```

```
[ ] formattingtime = datetime.datetime.strftime(today, "%H: %M: %S")  
    print(formattingtime)
```

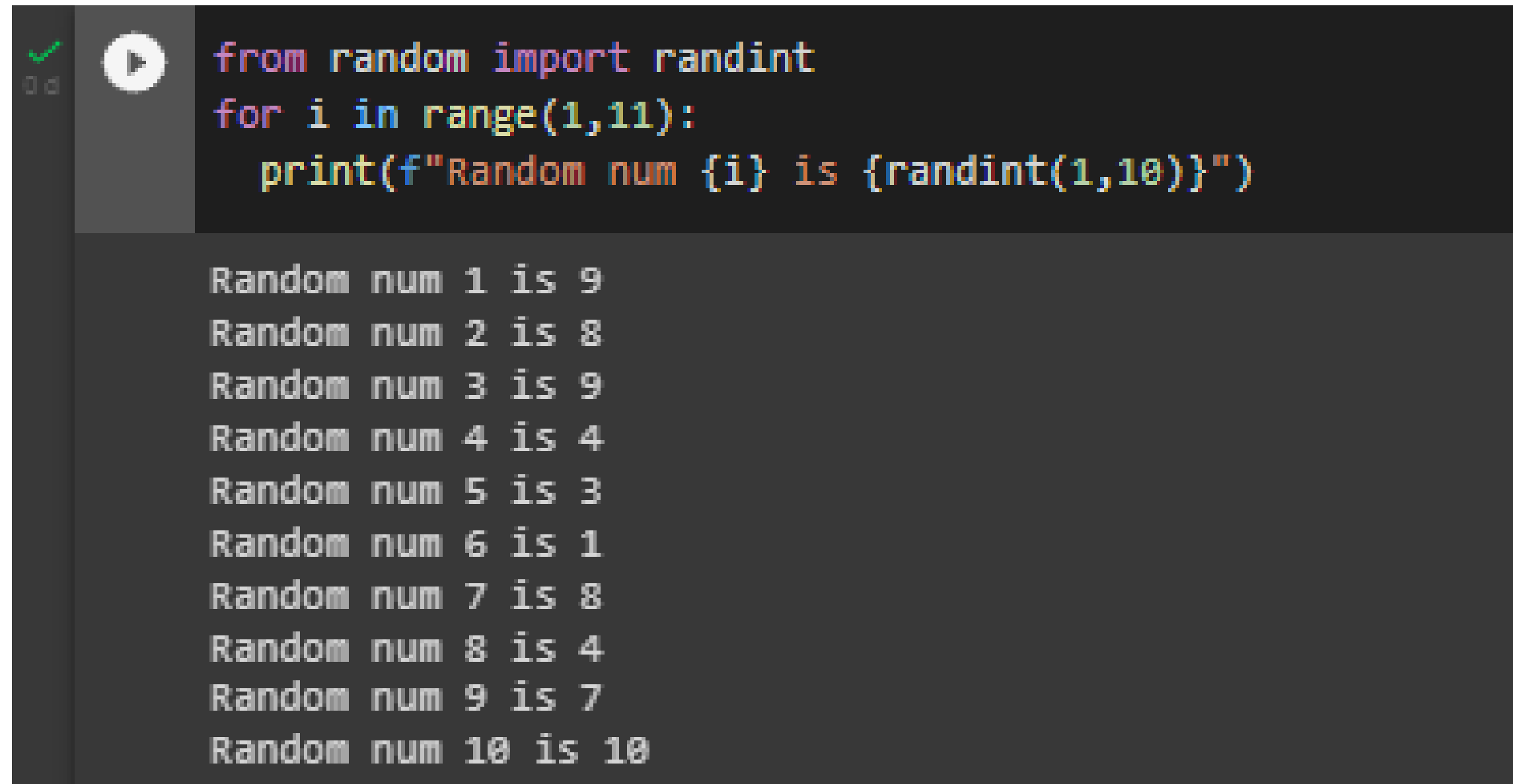
```
✓ [25] formattingtime1 = datetime.datetime.strftime(todaywithouttime, "%d/%m/%y")  
0 d      print(formattingtime1)
```

```
10/04/22
```

# Random lib

Python Random module is an in built module of Python which is used to generate random numbers. These are pseudo-random numbers means these are not truly random. This module can be used to perform random actions such as generating random numbers, print random a value for a list or string, etc.

# Example :

A screenshot of a Python code editor showing a script that generates 10 random numbers. The code is in a dark-themed editor with a play button icon on the left. The output of the script is displayed below the code.

```
from random import randint
for i in range(1,11):
    print(f"Random num {i} is {randint(1,10)}")
```

Random num 1 is 9  
Random num 2 is 8  
Random num 3 is 9  
Random num 4 is 4  
Random num 5 is 3  
Random num 6 is 1  
Random num 7 is 8  
Random num 8 is 4  
Random num 9 is 7  
Random num 10 is 10

the purpose of the code above is that a number with a range of 1-10 will be displayed and it will display a random number that has been set, which is between 1-9.

# sys lib

The sys module in Python provides various functions and variables that are used to manipulate different parts of the Python runtime environment. It allows operating on the interpreter as it provides access to the variables and functions that interact strongly with the interpreter.

# Example :

```
[ ] import sys

[ ] a = float(input("Masukkan angka pertama"))
    b = float(input("Masukkan angka kedua"))

    try:
        print(f"the answer is {a/b}")
    except:
        print(sys.exc_info()[0])
        print("This did not work did you try to devide by zero or something?")
    else:
        print("You successfully used division feature in python")
    finally:
        print("Thank you for playing")

Masukkan angka pertama12
Masukkan angka kedua1
the answer is 12.0
You successfully used division feature in python
Thank you for playing
```

the code will display the result of the division of  $a / b$ . if we input a number divided by 0 then an error will occur and will display an error message according to what we set.



# Contact Us.

## Rimberio Company

Address : 123 Anywhere St., Any City

Phone : +123-456-7890

Social Media : @reallygreatsite

Email Address : hello@reallygreatsite.com

Website : [www.reallygreatsite.com](http://www.reallygreatsite.com)