



Shri Vile Parle Kelavani Mandal's

**DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING**

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



JUNAID GIRKAR

60004190057

TE COMPS A4

## EXPERIMENT - 5

**AIM: Python program to explore different types of modules**

### OS MODULE:

The OS module in Python provides functions for interacting with the operating system. OS comes under Python's standard utility modules. This module provides a portable way of using operating system-dependent functionality. The `*os*` and `*os.path*` modules include many functions to interact with the file system.

There are some functions in the OS module which are given below:

- `os.name()`
- `os.mkdir()`
- `os.getcwd()`
- `os.chdir()`
- `os.rmdir()`
- `os.error()`
- `os.popen()`
- `os.close()`
- `os.rename()`
- `os.access()`

CODE:

```
import os

def nameOfOS():
    """Name of the operating system module that it imports"""
    return os.name

def getCurrentDirectory():
    """Function to get current directory"""
    return os.getcwd()
```



```
def changeDirectory():
    """Function to change current directory to parent directory"""
    print(getCurrentDirectory())
    os.chdir("..")
    print(getCurrentDirectory())

def makeNewDirectory():
    """Function to make a new directory."""
    return os.mkdir("d:\\newdir")

def removeDirectory():
    return os.rmdir("d:\\newdir")

def listDirectories():
    return os.listdir()

def getCurrentProcessID():
    """This function returns current process ID or PID, as it is popularly known."""
    return os.getpid()

def getTimeOfLastModificationOfPath():
    """This method returns the time of the last modification of the path."""
    return os.path.getmtime()

def getTimeOfLastAccessOfPath():
    """This method returns the time of the last access of the path."""
    return os.path.getatime()

def determineIfAPathExists():
    """This method returns True for existing paths. It returns False for broken symbolic links."""
    return os.path.exists("/Files/Engineering")
```



```
def joinPaths():  
    """Function to join various path components"""  
    path = "\\Files\\Engineering\\Python\\Semester - 5"  
    return os.path.join(path, "python")  
  
if __name__ == "__main__":  
    print(nameOfOS())  
    print(getCurrentDirectory())  
    print(determineIfAPathExists())  
    print(joinPaths())  
    print(getCurrentProcessID())
```

OUTPUT:

```
> python os.py  
nt  
C:\\Users\\junai\\Downloads  
False  
\\Files\\Engineering\\Python\\Semester - 5\\python  
25456
```

## DateTime Module

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.

The DateTime module is categorized into 6 main classes –

**date** – An idealized naive date, assuming the current Gregorian calendar always was, and always will be, in effect. Its attributes are year, month and day.



**time** – An idealized time, independent of any particular day, assuming that every day has exactly 24\*60\*60 seconds. Its attributes are hour, minute, second, microsecond, and tzinfo.

**datetime** – Its a combination of date and time along with the attributes year, month, day, hour, minute, second, microsecond, and tzinfo.

**timedelta** – A duration expressing the difference between two date, time, or datetime instances to microsecond resolution.

**tzinfo** – It provides time zone information objects.

**timezone** – A class that implements the tzinfo abstract base class as a fixed offset from the UTC

### 1) **datetime.date()**

The date class is used to instantiate date objects in Python. When an object of this class is instantiated, it represents a date in the format YYYY-MM-DD. Constructor of this class needs three mandatory arguments year, month and date.

CODE:

```
import datetime
d=datetime.date(2020,11,29)
print(d)
```

OUTPUT:

```
2020-11-29
```

### 2) **datetime.date.today()**

To return the current local date today() function of date class is used. today() function comes with several attributes (year, month and day). These can be printed individually.

CODE:

```
import datetime
td=datetime.date.today()
print(td)
print(td.year)
print(td.month)
print(td.day)
```

OUTPUT:

```
2021-10-29
2021
10
```



### 3) **weekday(), isoweekday()** :

weekday() returns the day of the week as integer where Monday is 0 and Sunday is 6.

isoweekday() returns the day of the week as integer where Monday is 1 and Sunday is 7.

CODE:

```
import datetime
td=datetime.date.today()
print(td.weekday())
print(td.isoweekday())
```

OUTPUT:

```
4
5
```

### 4) **datetime.time()** :

The time class creates the time object which represents local time, independent of any day. It can take minutes, hours, seconds, milliseconds as parameters, it works without parameters as well.

CODE:

```
import datetime
t=datetime.time(10,35,55,200)
print(t)
print(t.hour)
print(t.minute)
print(t.second)
```

OUTPUT:

```
10:35:55.000200
10
35
55
```

### 5) **datetime.datetime()** :

The datetime class contains information on both date and time. Like a date object, datetime assumes the current Gregorian calendar extended in both directions; like a time object, datetime assumes there are exactly 3600\*24 seconds every day.



Shri Vile Parle Kelavani Mandal's

**DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING**

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



CODE:

```
import datetime
dt=datetime.datetime(2021,10,29,12,30,45,1000)
print(dt)
dt_today=datetime.datetime.today()
dt_now=datetime.datetime.now()
dt_utcnow=datetime.datetime.utcnow()
print(dt_today)
print(dt_now)
print(dt_utcnow)
```

OUTPUT:

```
2021-10-29 12:30:45.001000
2021-10-29 22:51:49.223397
2021-10-29 22:51:49.223397
2021-10-29 17:21:49.223397
```

#### 6) **pytz module :**

pytz brings the Olson tz database into Python and thus supports almost all time zones. This module serves the date-time conversion functionalities and helps user serving international client's base.

CODE:

```
import pytz
for tz in pytz.all_timezones:
    print(tz)
```

OUTPUT:

```
Africa/Abidjan
Africa/Accra
Africa/Addis_Ababa
Africa/Algiers
.
.
.
US/Mountain
US/Pacific
US/Samoa
UTC
Universal
W-SU
```



Shri Vile Parle Kelavani Mandal's

**DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING**

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



WET  
Zulu

### 7) `datetime.datetime.strptime()` :

Returns a DateTime object corresponding to the date string. You cannot create datetime object from every string. The string needs to be in a certain format

CODE:

```
import datetime
dt_str = 'September 24, 2001'
dt=datetime.datetime.strptime(dt_str,'%B %d, %Y')
print(dt)
```

OUTPUT:

```
2001-09-24 00:00:00
```

### 8) `strftime()`

The `strftime()` method is defined under classes `date`, `datetime` and `time`. The method creates a formatted string from a given date, datetime or time object.

CODE:

```
import datetime
x = datetime.datetime.today()
print(x.strftime("%B"))
print(x.strftime("%A"))
print(x.strftime("%Y"))
print(x.strftime("%X"))
print(x.strftime("%p"))
```

OUTPUT:

```
October
Friday
2021
22:56:51
PM
```

**MATH MODULE:**



Python math module is defined as the most famous mathematical functions, which includes trigonometric functions, representation functions, logarithmic functions, etc. Furthermore, it also defines two mathematical constants, i.e., Pie and Euler number, etc.

Pie ( $\pi$ ): It is a well-known mathematical constant and defined as the ratio of circumference to the diameter of a circle. Its value is 3.141592653589793.

Euler's number ( $e$ ): It is defined as the base of the natural logarithmic, and its value is 2.718281828459045.

There are different math modules which are given below:

Method	Description
<code>math.acos()</code>	Returns the arc cosine of a number
<code>math.acosh()</code>	Returns the inverse hyperbolic cosine of a number
<code>math.asin()</code>	Returns the arc sine of a number
<code>math.asinh()</code>	Returns the inverse hyperbolic sine of a number
<code>math.atan()</code>	Returns the arc tangent of a number in radians
<code>math.atan2()</code>	Returns the arc tangent of $y/x$ in radians
<code>math.atanh()</code>	Returns the inverse hyperbolic tangent of a number
<code>math.ceil()</code>	Rounds a number up to the nearest integer
<code>math.comb()</code>	Returns the number of ways to choose $k$ items from $n$ items without repetition and order
<code>math.copysign()</code>	Returns a float consisting of the value of the first parameter and the sign of the second parameter
<code>math.cos()</code>	Returns the cosine of a number
<code>math.cosh()</code>	Returns the hyperbolic cosine of a





	number
math.degrees()	Converts an angle from radians to degrees
math.dist()	Returns the Euclidean distance between two points (p and q), where p and q are the coordinates of that point
math.erf()	Returns the error function of a number
math.erfc()	Returns the complementary error function of a number
math.exp()	Returns E raised to the power of x
math.expm1()	Returns $E^x - 1$
math.fabs()	Returns the absolute value of a number

**CODE:**

```
import math
```

```
euler = f"Value of Euler e is: {math.e}"
pi = f"Value of Pi is: {math.pi}"
tau = f"Value of Tau is: {math.tau}"
inf = f"Value of Infinity is: {math.inf}"
ceil = f"Ceiling value of 5.5 is {math.ceil(5.5)}."
floor = f"Floor value of 5.5 is {math.floor(5.5)}."
fact = f"Factorial of 4 is {math.factorial(4)}"
gcd = f"GCD of 12 and 16 is {math.gcd(12, 16)}"
fabs = f"Absolute value of -25 is {math.fabs(-25)}"
fmod = f"When {7} divides {17} the remainder is: {math.fmod(17, 7)}"
expInt = f"Exponent of {3} is {math.exp(3)}."
expNegativeInteger = f"Exponent of {-3} is {math.exp(-3)}."
expFloat = f"Exponent of {2.7} is {math.exp(2.7)}."
power = f"{3} to the power of {4} is {math.pow(3, 4)}"
logWithBaseGiven = f"The log value of 2 with base 3 is: {math.log(2, 3)}"
logWithBaseTwo = f"Log value of 16 with base 2: {math.log2(16)}."
logWithBaseTen = f"Log value of 10000 with base 10: {math.log10(10000)}."
radianToDegree = f"Corresponding degree value of {math.pi/6} radians is:
```



Shri Vile Parle Kelavani Mandal's

**DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING**

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



```
{math.degrees(math.pi/6)}"
```

```
degreeToRadian = f"Corresponding radian value of 30 radians is: {math.radians(30)}"
```

```
sine = f"Sine value of {math.pi/6} is {math.sin(math.pi/6)}"
```

```
cosine = f"Cosine value of {math.pi/6} is {math.cos(math.pi/6)}"
```

```
tangent = f"Tangent value of {math.pi/6} is {math.tan(math.pi/6)}"
```

```
print(f"{euler}\n{pi}\n{tau}\n{inf}\n{ceil}\n{floor}\n{fact}\n{gcd}\n{fabs}\n{fmod}\n{explnt}\n{expNegativeInteger}\n{expFloat}\n{power}\n{logWithBaseGiven}\n{logWithBaseTwo}\n{logWithBaseTen}\n{radianToDegree}\n{degreeToRadian}\n{sine}\n{cosine}\n{tangent}")
```

## OUTPUT:

```
> py math.py
Value of Euler e is: 2.718281828459045
Value of Pi is: 3.141592653589793
Value of Tau is: 6.283185307179586
Value of Infinity is: inf
Ceiling value of 5.5 is 6.
Floor value of 5.5 is 5.
Factorial of 4 is 24
GCD of 12 and 16 is 4
Absolute value of -25 is 25.0
When 7 divides 17 the remainder is: 3.0
Exponent of 3 is 20.085536923187668.
Exponent of -3 is 0.049787068367863944.
Exponent of 2.7 is 14.879731724872837.
3 to the power of 4 is 81.0
The log value of 2 with base 3 is: 0.6309297535714574
Log value of 16 with base 2: 4.0.
Log value of 10000 with base 10: 4.0.
Corresponding degree value of 0.5235987755982988 radians is: 29.999999999999996
Corresponding radian value of 30 radians is: 0.5235987755982988
Sine value of 0.5235987755982988 is 0.49999999999999994
Cosine value of 0.5235987755982988 is 0.8660254037844387
Tangent value of 0.5235987755982988 is 0.5773502691896257
```

## RANDOM MODULE:



Python has a built-in module that you can use to make random numbers.

The random module has a set of methods:

Method	Description
seed()	Initialize the random number generator
getstate()	Returns the current internal state of the random number generator
setstate()	Restores the internal state of the random number generator
getrandbits()	Returns a number representing the random bits
randrange()	Returns a random number between the given range
randint()	Returns a random number between the given range
choice()	Returns a random element from the given sequence
choices()	Returns a list with a random selection from the given sequence
shuffle()	Takes a sequence and returns the sequence in a random order
sample()	Returns a given sample of a sequence
random()	Returns a random float number between 0 and 1
uniform()	Returns a random float number between two given parameters
triangular()	Returns a random float number between two given parameters, you can also set a mode parameter to specify the midpoint between the two other parameters
betavariate()	Returns a random float number between 0 and 1 based on the Beta distribution (used in statistics)



expovariate()	Returns a random float number based on the Exponential distribution (used in statistics)
gammavariate()	Returns a random float number based on the Gamma distribution (used in statistics)
gauss()	Returns a random float number based on the Gaussian distribution (used in probability theories)
lognormvariate()	Returns a random float number based on a log-normal distribution (used in probability theories)
normalvariate()	Returns a random float number based on the normal distribution (used in probability theories)

**CODE:**

```
import random

lst = [1, 2, 3, "Hello", "World", 19.12, -32]
randomValueFromList = f"Random value from list: {random.choice(lst)}"
random.seed(5)
print(random.random())
print(random.random())
randomNumbersInRangeOfIntegers = f"Random numbers in range of integers 5 and 15 is {random.randint(5, 15)}"
print(randomNumbersInRangeOfIntegers)
randomNumberInRangeOfNumbers = f"Random number from 50-100 is: {random.randrange(50,100)}"
print(randomNumberInRangeOfNumbers)
originalList = [1, 2, 3, 4, 5]
original = f"Original list: {originalList}"
print(original)
shuffled = f"After shuffling: {random.shuffle(originalList)}"
print(shuffled)
randomFloatInRange = f"Random float in the range 13.26 and 19.12 is {random.uniform(13.26, 19.12)}"
print(randomFloatInRange)
```



Shri Vile Parle Kelavani Mandal's

**DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING**

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



## OUTPUT:

```
> py random_Module.py
0.6229016948897019
0.7417869892607294
Random numbers in range of integers 5 and 15 is 15
Random number from 50-100 is: 83
Original list: [1, 2, 3, 4, 5]
After shuffling: None
Random float in the range 13.26 and 19.12 is 18.53927688165865
```

**Conclusion :** Python modules are one of the major reasons for its popularity as they help in writing programs in lesser lines of code. We learnt about the Math module, Random module, OS module and DateTime module which has many functionalities. We then implemented it in a python program.

---