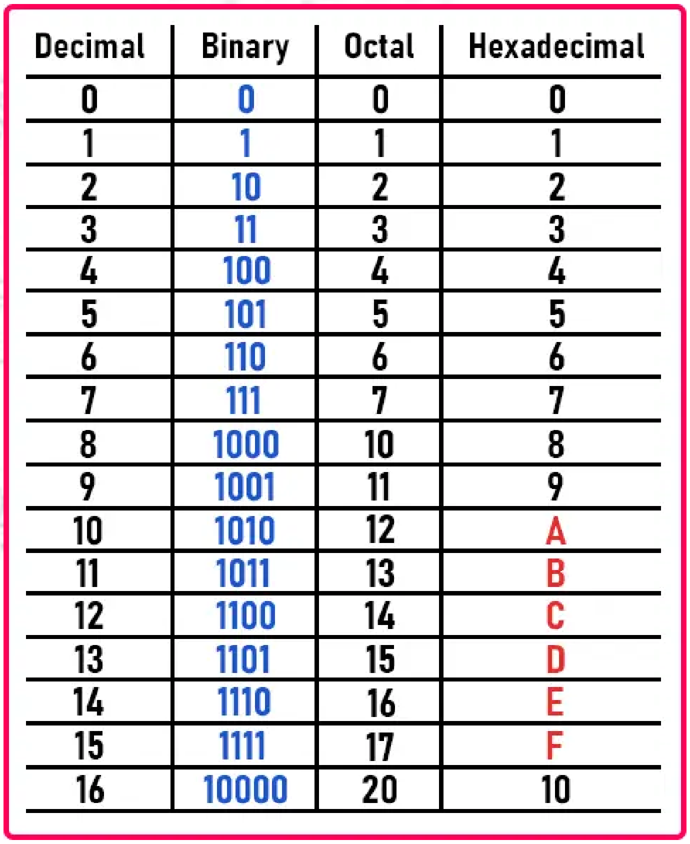
LECTURE SEVEN

Number Data Type in Python

Python supports integers, floating point numbers and complex numbers. They are defined as int, float and complex in Python as described **in lecture 5**.

Numbers we deal with everyday are **decimal (base 10)** number system. But computer programmers (generally embedded programmer) need to work with **binary (base 2)**, **hexadecimal (base 16)** and **octal (base 8)** number systems.



Decimal	Binary	Octal	Hexadecimal
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

In Python, we can represent these numbers by appropriately placing a prefix before that number. Following table lists these prefix.

Number system prefix for Python numbers

System	**Prefix**
Binary	'0b' or '0B'
Octal	'0o' or '0O'
Hexadecimal	'0x' or '0X'

Example: *Python numbers*

```
x=0b10
y=0xFB
z=0o15
print (x)
print (x+ y)
print(z)
```

Output

```
2
253
13
```

Binary, Octal, and Hexadecimal Integers in Python:

- Binary integers are the number represented with base two. Which means in the binary number system, there are only two symbols used to represent numbers: 0 and 1.use the **bin()** function to convert from a decimal value to its corresponding binary value. And similarly, the **int()** with base **2** for the binary number system.

Example (1): Write Python Program for input decimal value and convert it to binary value then return it in decimal value again.

```
x= int(input('Enter a number : '))
y= bin(x)
print("binary is",y)
z= int(y, 2)
print("decimal is", z)
```

- Octal is base eight, which means that eight symbols are used to represent all the quantities. They are 0, 1, 2, 3, 4, 5, 6 and 7. you can use the **oct()** function to convert from a decimal value to its corresponding octal value. Alternatively, you can also use the **int()** function along with the correct base which is **8** for the octal number system.

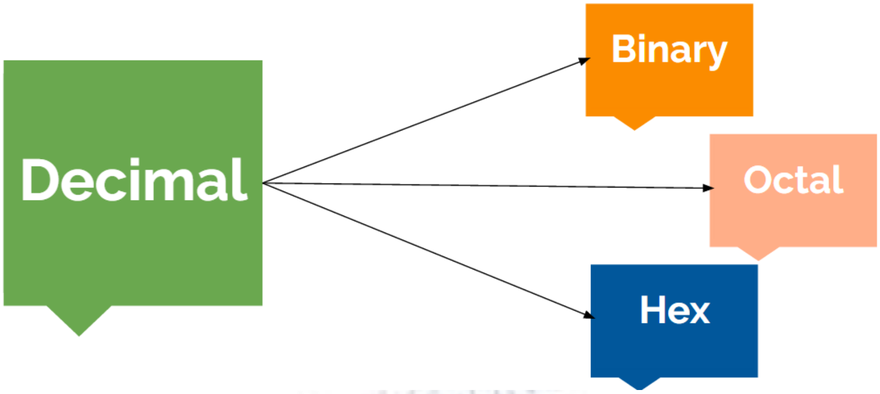
Example (2): Write Python Program for input decimal value and convert it to octal value then return it in decimal value again.

```
x= int(input('Enter a number  : '))
y= oct(x)
print("octal is",y)
z= int(y, 8)
print("decimal is", z)
```

- Hexadecimal is a base 16 number system 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E and F. In Python, you can use the **hex**() function to convert from a decimal value to its corresponding hexadecimal value, or the **int**() function with base **16** for the hexadecimal number system.

Example (3): Write Python Program for input decimal value and convert it to Hexadecimal value then return it in decimal value again.

```
x= int(input('Enter a number  : '))
y= hex(x)
print("Hexadecimal is",y)
z= int(y, 16)
print("decimal is", z)
```

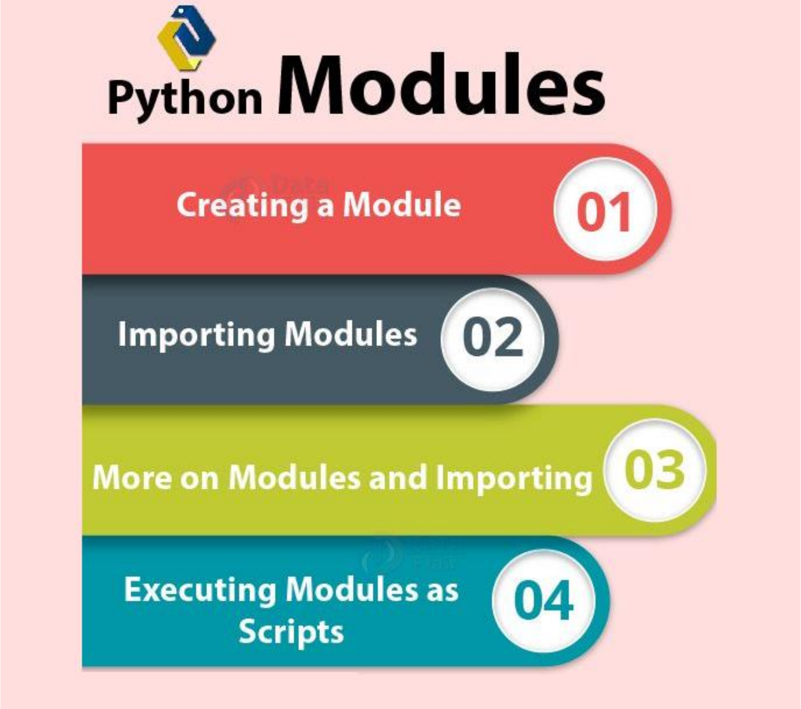


Module in Python

Modules are **Python .py** files that consist of Python code. Any Python file can be referenced as a module. A module can contain **executable statements** as well as **functions**

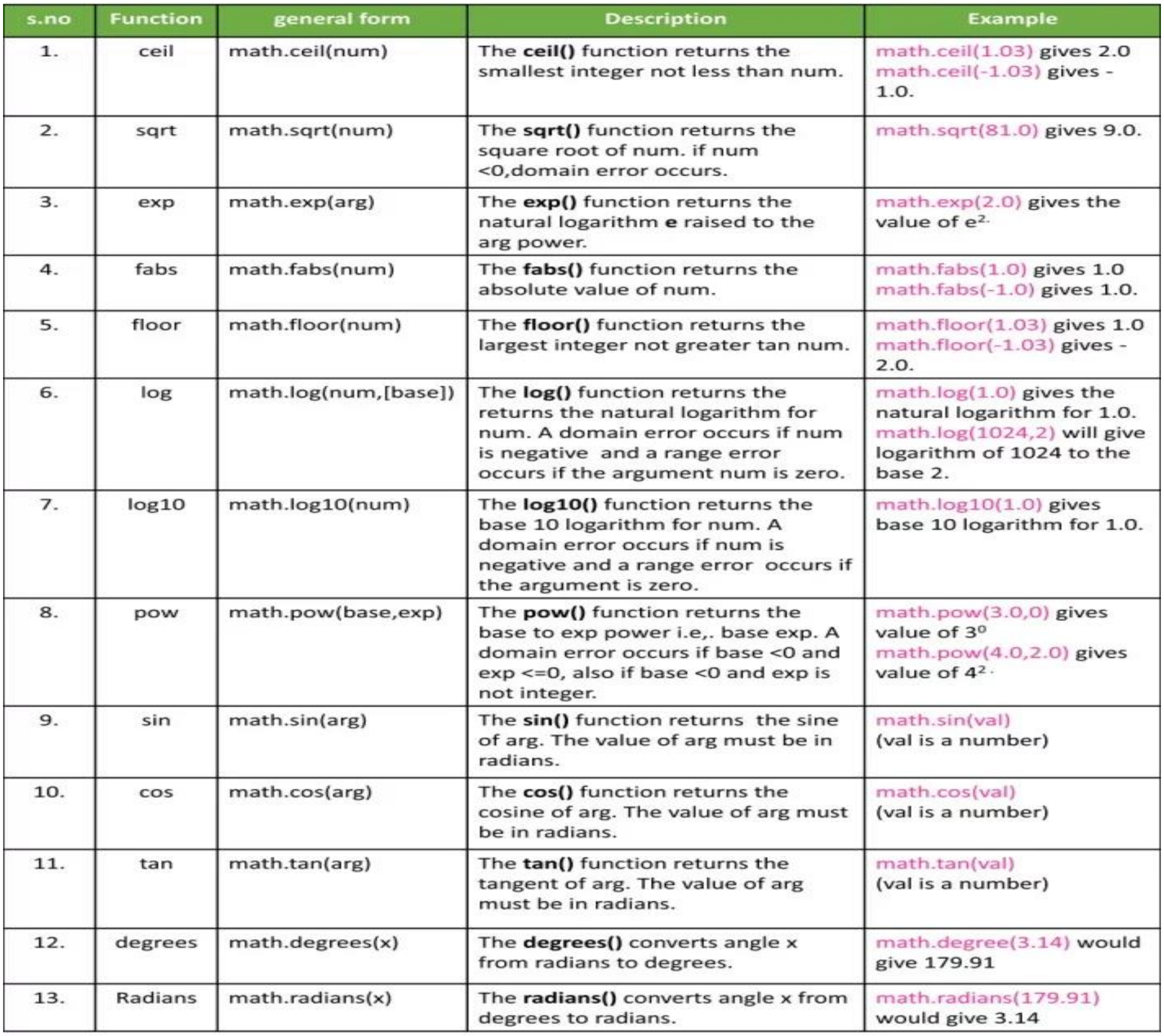
In Python, modules are accessed by using the **import** statement. When you do this, you execute the code of the module, but it is not the only way. Import specific names from a module without importing the module as a whole.

```
import module_name
from module_name import method
from module_name import *
```


Python Modules
Creating a Module 01
Importing Modules 02
More on Modules and Importing 03
Executing Modules as Scripts 04

Math Module (Python Mathematic)

The math module is a standard module in Python and is always available. To use mathematical functions under this module, you have to import the module using import math. The math module supplies mathematical functions on floating-point numbers, while the cmath module supplies equivalent functions on complex numbers. For example, math.sqrt(-1) raises an exception, but cmath.sqrt(-1) returns 1j.



s.no	Function	general form	Description	Example
1.	ceil	math.ceil(num)	The **ceil()** function returns the smallest integer not less than num.	math.ceil(1.03) gives 2.0 math.ceil(-1.03) gives -1.0.
2.	sqrt	math.sqrt(num)	The **sqrt()** function returns the square root of num. if num <0,domain error occurs.	math.sqrt(81.0) gives 9.0.
3.	exp	math.exp(arg)	The **exp()** function returns the natural logarithm **e** raised to the arg power.	math.exp(2.0) gives the value of e^{2}.
4.	fabs	math.fabs(num)	The **fabs()** function returns the absolute value of num.	math.fabs(1.0) gives 1.0 math.fabs(-1.0) gives 1.0.
5.	floor	math.floor(num)	The **floor()** function returns the largest integer not greater tan num.	math.floor(1.03) gives 1.0 math.floor(-1.03) gives -2.0.
6.	log	math.log(num,[base])	The **log()** function returns the returns the natural logarithm for num. A domain error occurs if num is negative and a range error occurs if the argument num is zero.	math.log(1.0) gives the natural logarithm for 1.0. math.log(1024,2) will give logarithm of 1024 to the base 2.
7.	log10	math.log10(num)	The **log10()** function returns the base 10 logarithm for num. A domain error occurs if num is negative and a range error occurs if the argument is zero.	math.log10(1.0) gives base 10 logarithm for 1.0.
8.	pow	math.pow(base,exp)	The **pow()** function returns the base to exp power i.e,. base exp. A domain error occurs if base <0 and exp <=0, also if base <0 and exp is not integer.	math.pow(3.0,0) gives value of 3^{0} math.pow(4.0,2.0) gives value of 4^{2}.
9.	sin	math.sin(arg)	The **sin()** function returns the sine of arg. The value of arg must be in radians.	math.sin(val) (val is a number)
10.	cos	math.cos(arg)	The **cos()** function returns the cosine of arg. The value of arg must be in radians.	math.cos(val) (val is a number)
11.	tan	math.tan(arg)	The **tan()** function returns the tangent of arg. The value of arg must be in radians.	math.tan(val) (val is a number)
12.	degrees	math.degrees(x)	The **degrees()** converts angle x from radians to degrees.	math.degree(3.14) would give 179.91
13.	Radians	math.radians(x)	The **radians()** converts angle x from degrees to radians.	math.radians(179.91) would give 3.14

Example (4): Write Python program that will calculate the roots of a quadratic equation: $ax^2 + \mathrm{bx} + c = 0$

Hint: a,b,c as input

d =$\sqrt{b^2 - 4ac}$, **the roots are:** x1 = (–b + d)/2a and x2 = (–b – d)/2a

```
import math
a = int(input("a="))
b = int(input("b="))
c = int(input("c="))
d = math.sqrt((b**2) - (4*a*c))
solution1 = (-b + d)/(2*a)
solution2 = (-b - d)/(2*a)
print('solution of 0=',solution1)
print('solution of 1=,',solution2)
```

Example (5): Write Python program to read number x and calculate the sin(x), cos(x) and tan(x)

```
import math
x = int(input("x="))
x=math.radians(x)
print(math.sin(x))
print(math.cos (x))
print(math.tan (x))
```

WORK SHEET (3)

Homework Write a Python program to read two Hexadecimal number and compute addition between them.

Homework Write a Python program to find the output of z

$$z=\exp(x^2) + \log2(x)$$

Homework Write a Python program to find the output of m

$$m = \sqrt{b^2 - a^2}$$