

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR & SACHIN BHARDWAJ,
PGT(CS), KV NO.1 TEZPUR

PYTHON MODULE

Group of functions, classes, variables

What is Python Module

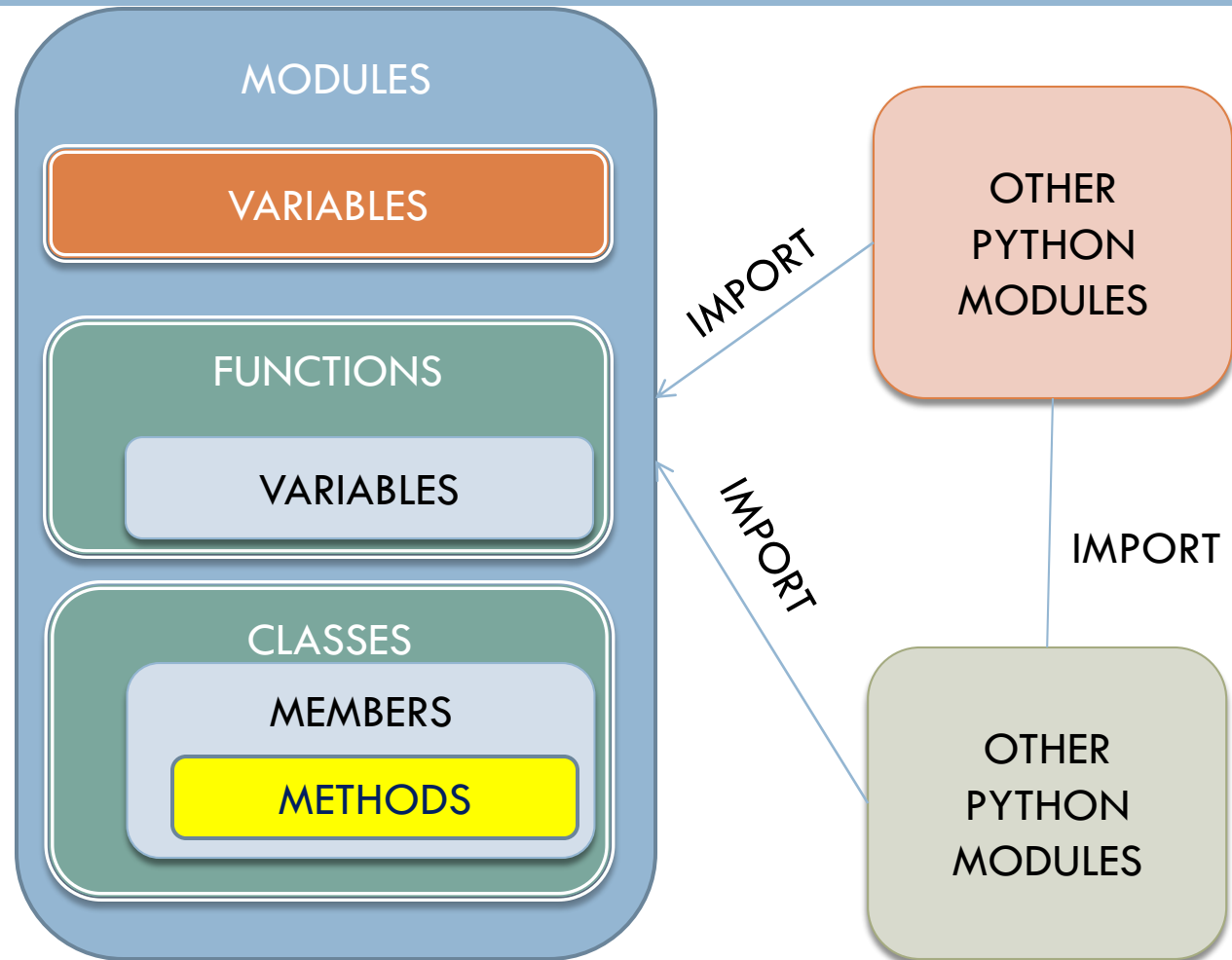
- A Module is a file containing Python definitions (docstrings) , functions, variables, classes and statements.
- Act of partitioning a program into individual components(modules) is called modularity. A module is a separate unit in itself.
 - ▣ It reduces its complexity to some degree
 - ▣ It creates numbers of well-defined, documented boundaries within program.
 - ▣ Its contents can be reused in other program, without having to rewrite or recreate them.

Structure of Python module

- A python module is simply a normal python file(.py) and contains functions, constants and other elements.
- Python module may contains following objects:

docstring	Triple quoted comments. Useful for documentation purpose
Variables and constants	For storing values
Classes	To create blueprint of any object
Objects	Object is an instance of class. It represent class in real world
Statements	Instruction
Functions	Group of statements

Composition/Structure of python module



Importing Python modules

- To import entire module
 - **import <module name>**
 - **Example:** `import math`

- To import specific function/object from module:
 - **from <module_name> import <function_name>**
 - **Example:** `from math import sqrt`

- **import *** : can be used to import all names from module into current calling module

Accessing function/constant of imported module

- To use function/constant/variable of imported module we have to specify module name and function name separated by dot(.). This format is known as dot notation.
 - `<module_name>.<function_name>`
 - **Example:** `print(math.sqrt(25))`
- However if only particular function is imported using **from** then module name before function name is not required. We will see examples with next slides.

Types of Modules

- There are various in-built module in python, we will discuss few of them
 - ▣ Math module
 - ▣ Random module
 - ▣ Statistical module

Math module

- This module provides various function to perform arithmetic operations.
- Example of functions in math modules are:

sqrt	ceil	floor	pow
fabs	sin	cos	tan

- Example of variables in math modules are:
 - pi
 - e

Math module functions

- **sqrt(x)** : this function returns the square root of number(x).

```
>>> import math
>>> print(math.sqrt(49))
7.0
```

module name is required before function name here

- **pow(x,y)** : this function returns the $(x)^y$

```
>>> from math import pow
>>> print(pow(2,6))
64.0
```

module name is not required before function name here

- **ceil(x)** : this function return the x rounded to next integer.

```
>>> import math
>>> print(math.ceil(45.25))
46
```

Math module functions

- **floor(x)** : this function returns the x rounded to previous integer.

```
>>> import math
>>> print(math.floor(5.9))
5
```

- **fabs(x)** : this function returns absolute value of float x. absolute value means number without any sign

```
>>> import math
>>> a=-8.5
>>> print(math.fabs(a))
8.5
```

- **sin (x)** : it return sine of x (measured in radian)

```
>>> import math
>>> print(math.sin(30))
-0.9880316240928618
```

```
>>> import math
>>> print(math.sin(30*math.pi/180))
0.49999999999999994
```

```
>>> import math
>>> print(round(math.sin(30*math.pi/180),1))
0.5
```

Math module functions

- **cos(x)** : it return cosine of x (measured in radian)

```
>>> import math
>>> print(math.cos(90))
-0.4480736161291701
```

- **tan(x)** : it return tangent of x (measured in radian)

```
>>> import math
>>> print(math.tan(45))
1.6197751905438615
```

- **pi** : return the constant value of pi (22/7)

```
>>> print(math.pi)
3.141592653589793
```

- **e** : return the constant value of constant e

```
>>> print(math.e)
2.718281828459045
```

Using Random Module

- Python has a module namely random that provides random – number generators. Random number means any number generated within the given range.
- To generate random number in Python we have to import random module
- 2 most common method to generate random number in python are :
 - random() function
 - randint(a,b) function

random() function

- It is floating point random number generator between 0.0 to 1.0. here lower limit is inclusive where as upper limit is less than 1.0.
- $0 \leq N < 1$
- Examples:

```
>>> import random
>>> a = random.random()
>>> print(a)
0.0888880146536
>>> |
```

Output is less than 1

random() function

- To generate random number between given range of values using random(), the following format should be used:
 - ▣ $\text{Lower_range} + \text{random()} * (\text{upper_range} - \text{lower_range})$
 - ▣ For example to generate number between 10 to 50:
 - $10 + \text{random()} * (40)$

randint() function

- Another way to generate random number is randint() function, but it generate integer numbers.
- Both the given range values are inclusive i.e. if we generate random number as :
 - ▣ randint(20,70)
 - In above example random number between 20 to 70 will be taken. (including 20 and 70 also)

```
>>> import random
>>> a = random.randint(10,20)
>>> print(a)
18
>>> █
```

E
X
A
M
P
L
E

```
import random
count=3
ans='y'
win=False
print("Guess what number computer generated between 20-30")
print("Total 3 chances are there ")
print("-----")
while ans=='y':
    num1 = random.randint(20,30)
    print("Change Remaining :",count)
    guess = int(input("Enter your answer :"))
    if num1 == guess:
        print("Congratulation! you guessed it right")
        win=True
    else:
        print("Wrong!")
        count-=1
        if count==0:
            print("Oops! You lost all your chances ")
            print("Number was :",num1)
    if win==True or count==0:
        ans=input("Play Again?")
        if ans=='y':
            count=3
            win=False
```



```
Guess what number computer generated between 20-30
```

```
Total 3 chances are there
```

```
-----  
Change Remaining : 3
```

```
Enter your answer :21
```

```
Wrong!
```

```
Change Remaining : 2
```

```
Enter your answer :22
```

```
Wrong!
```

```
Change Remaining : 1
```

```
Enter your answer :23
```

```
Wrong!
```

```
Oops! You lost all your chances
```

```
Number was : 25
```

```
Play Again?y
```

```
Change Remaining : 3
```

```
Enter your answer :28
```

```
Wrong!
```

```
Change Remaining : 2
```

```
Enter your answer :27
```

```
Wrong!
```

```
Change Remaining : 1
```

```
Enter your answer :29
```

```
Congratulation! you guessed it right
```

```
Play Again?n
```

Just a Minute...

- Give the following python code, which is repeated four times. What could be the possible set of output(s) out of four sets (ddd is any combination of digits)

```
import random
```

```
print(15 + random.random()*5)
```

a)	b)	c)	d)
17.ddd	15.ddd	14.ddd	15.ddd
19.ddd	17.ddd	16.ddd	15.ddd
20.ddd	19.ddd	18.ddd	15.ddd
15.ddd	18.ddd	20.ddd	15.ddd

Just a Minute...

- What could be the minimum possible and maximum possible numbers by following code

```
import random
print(random.randint(3,10)-3)
```
- In a school fest, three randomly chosen students out of 100 students (having roll number 1 -100) have to present the bouquet to the guests. Help the school authorities choose three students randomly

Just a Minute...

What possible outputs(s) are expected to be displayed on screen at the time of execution of the program from the following code? Also specify the minimum values that can be assigned to each of the variables BEGIN and LAST.

```
import random

VALUES=[10,20,30,40,50,60,70,80];
BEGIN=random.randint(1,3)
LAST =random.randint(BEGIN,4)

for I in range(BEGIN, LAST+1):
    print VALUES[I], "-",
```

(i) 30 - 40 - 50 -

(ii) 10 - 20 - 30 - 40 -

(iii) 30 - 40 - 50 - 60 -

(iv) 30 - 40 - 50 - 60 - 70 -

Just a Minute...

Look at the following Python code and find the possible output(s) from the options (i) to (iv) following it. Also, write the maximum and the minimum values that can be assigned to the variable PICKER.

Note:

- Assume all the required header files are already being included in the code.
- The function randint() generates an integer between 1 to n

```
import random
```

```
PICKER=1+random.randint(0,2)
```

```
COLOR=["BLUE","PINK","GREEN","RED"]
```

```
for l in range(1,PICKER+1):
```

```
    for j in range(l+1):
```

```
        print(COLOR[j],end="")
```

```
    print()
```

(i)	BLUEPINK	(ii)	PINKGREEN	(iii)	BLUE	(iv)	BLUEPINK
	BLUEPINKGREEN		PINKGREENRED		BLUEPINK		BLUEPINKGREEN
					BLUEPINKGREEN		BLUEPINKGREENRED

What are the possible outcome(s) executed from the following code? Also specify the maximum and minimum values that can be assigned to variable PICK

```
import random
PICK = random.randint(0, 3)
CITY = ["DELHI", "MUMBAI", "CHENNAI", "KOLKATA"]
for i in CITY:
    for j in range(1, PICK):
        print(i, end=" ")
    print()
```

1)
DELHIDELHI
MUMBAIMUMBAI
CHENNAICHENNAI
KOLKATAKOLKATA

2)
DELHI
DELHIMUMBAI
DELHIMUMBAICHENNAI

3)
DELHI
MUMBAI
CHENNAI
KOKLATA

4)
DELHI
DELHIMUMBAI
KOLKATAKOLKATAKOLKATA

randrange() function

- This function is also used to generate random number within given range.
- Syntax
 - ▣ `randrange(start,stop,step)`

```
import random
n1 = random.randrange(5,15)
n2 = random.randrange(5,15)
n3 = random.randrange(5,15)
n4 = random.randrange(5,15)
print(n1,n2,n3,n4)
```

11 8 5 12

It will generate random number between 5 to 14

random output between 5 to 14, may vary

randrange() function

```
import random
for i in range(20):
    n1 = random.randrange(1, 30, 2)
    print(n1, end='\t')
```

It will generate random number between 1 to 29 with stepping of 2 i.e. it will generate number with gap of 2 i.e. 1,3,5,7 and so on

```
25    11    15    9    3    7    19    13    17    7
27    11    27    5    21   7    17    9    25    7
```


Mathematics Game for Kids

```
import random
operators = ['+', '*', '-']
error = 0
score = 0
print("##### WELCOME TO SIMPLE CALCULATION GAME #####")
print("Rule : +4 for correct answer, -2 for wrong answer ")

for i in range(5):
    print("***50")
    n1 = random.randrange(1,100)
    n2 = random.randrange(1,100)
    i = random.randrange(0,3)
    op = operators[i]
    result = 0
    if op=='+' :
        result = n1 + n2
    elif op=='-' :
        if n1<n2:
            n1,n2=n2,n1
        result = n1 - n2
    elif op=='*' :
        result = n1 * n2
    print(n1,op,n2,'=')
    ask = int(input())
    if ask == result:
        score+=4
    else:
        score-=2

print("***50")
print("## YOU SCORED : ",score, " ##")
```

Mathematics Game for Kids

```
import random
operators = ['+', '*', '-', '-']
error = 0
score = 0
print("##### WELCOME TO SIMPLE CALCULATION GAME #####")
print("Rule : +4 for correct answer, -2 for wrong answer ")

for i in range(5):
    print("*"*50)
    n1 = random.randrange(1,100)
    n2 = random.randrange(1,100)
    i = random.randrange(0,3)
    op = operators[i]
    result = 0
    if op=='+':
        result = n1 + n2
    elif op=='-':
        if n1<n2:
            n1,n2=n2,n1
        result = n1 - n2
    elif op=='*':
        result = n1 * n2
    print(n1,op,n2,'=')
    ask = int(input())
    if ask == result:
        score+=4
    else:
        score-=2

print("*"*50)
print("### YOU SCORED : ",score, " ##")
```

```
##### WELCOME TO SIMPLE CALCULATION GAME #####
Rule : +4 for correct answer, -2 for wrong answer
*****
93 * 50 =
11
*****
29 + 29 =
58
*****
80 + 22 =
102
*****
61 - 25 =
36
*****
43 - 43 =
0
*****
## YOU SCORED : 14 ##
```

Statistical Module

- This module provides functions for calculating mathematical statistics of numeric (Real-valued) data.
- We will deal with 3 basic function under this module
 - Mean
 - Median
 - mode

Mean

- The mean is the average of all numbers and is sometimes called the arithmetic mean.

```
>>> import statistics
>>> mynum = [10,20,30,40,50,60,70,80,90,100]
>>> mymean = statistics.mean(mynum)
>>> print(mymean)
55
```

55, is the average of all numbers in the list

Median

- The median is the middle number in a group of numbers.

```
>>> import statistics
>>> mynum = [10,20,30,40,50,60,70,80,90]
>>> mymedian = statistics.median(mynum)
>>> print(mymedian)
50
```

With odd number of elements it will simply return the middle position value

```
>>> import statistics
>>> mynum = [10,20,30,40,50,60,70,80,90,100]
>>> mymedian = statistics.median(mynum)
>>> print(mymedian)
55.0
```

With even number of elements, it will return the average of value at mid + mid-1 i.e. $(50+60)/2 = 55.0$

Mode

- The mode is the number that occurs most often within a set of numbers i.e. most common data in list.

```
>>> import statistics
>>> mynum = [10,20,10,40,20,10,70,80,90]
>>> mymode = statistics.mode(mynum)
>>> print(mymode)
10
```

Here, 10 occurs most in the list.