

Module 12 Python Modules and Packages

Modules:

- ❑ Modules in python are simply Python files with .py extension.
- ❑ The name of the module will be the name of the file.
- ❑ Python module can have set of variables, functions and classes.

Packages:

- ❑ Packages are directories which contain modules and also subpackages.

math module:

- ❑ **math module provides access to the mathematical functions.**

- ❑ To use this module we need to write : `import math`

ceil(n) : Returns the next integer number of the given number.

floor(n): Returns the previous integer number of the given number.

sqrt(n): Returns the square root of the given number.

pow(x, y): Returns base to raise to the exp power.

sin(n): Returns sine of the given radian.

cos(n): Returns cosine of the given radian.

tan(n) :Returns tangent of the given radian.

random module:

- ❑ **random module can be used to generate random integers and floating point numbers.**

- ❑ We can use it for:

- ❑ Generating OTP
- ❑ Pick a random card from a deck of cards.
- ❑ Flip a coin.
- ❑ Generating passwords.

- ❑ To use this module we need to write : `import random`

randrange(stop):

- ❑ It will produce a random integer value in the range of 0 to stop-1.

randrange(start, stop):

- ❑ `start` → base value of the range(inclusive).
- ❑ `stop` → boundary value of the range(exclusive).

randrange(start, stop, stepsize):

- ❑ start → base value of the range(inclusive).
- ❑ stop → boundary value of the range(exclusive).
- ❑ step → value with which number is incremented. Default 0.

random():

- ❑ It will generate a random floating point number in the range of 0.0 and 1.0.

random.randint(a, b)

- ❑ Return a random integer N such that $a \leq N \leq b$. Alias for randrange(a, b+1).

datetime module:

- ❑ **datetime module provides with classes for working with date and time.**
- ❑ To use this module we need to write : `import datetime`

datetime.now(): This method provides current date and time.

- ❑ It also provides the following attributes:
 - ❑ day → between 1 and number of days in the given month of the given year,
 - ❑ month → between 1 and 12 inclusive,
 - ❑ year → current year,
 - ❑ hour → in range(24),
 - ❑ min → in range(60),
 - ❑ sec → in range(60)
 - and
 - ❑ microsecond → in range(1000000)

datetime.now().date(): It returns date with year, month and day

datetime.now().time(): It returns time with hour, min, sec and ms.

Module 12: Python Modules and Packages Test

Q1) What is the result of the following?

```
import math  
print(pi)
```

Options:

- a) It will display 22/7
- b) It will display 3.14
- c) It will display 3
- d) Error

Solution:

Q2) What is the result of the following?

```
from math import sqrt  
print(sqrt(4))  
print(cos(0))
```

Options:

- a) It will display 2.0 and 1.0
- b) It will display 2 and 1
- c) It will display 4 and 0
- d) Error

Solution:

Q3) You are writing an application that uses the sqrt function. The program must reference the function using the name squareRoot. You need to import the function. Which code segment should you use?

Options:

- A. import math.sqrt as squareRoot
- B. import sqrt from math as squareRoot
- C. from math import sqrt as squareRoot
- D. from math.sqrt as squareRoot

Solution:

Q4) Tailspin Toys uses Python Practice Test to control its new toy Happy Clown. The program has errors and you have been hired to help debug the following Happy Clown code. Line numbers are included for reference only.

```
1. import math
2. power = True
3. move = 0
4. while power:
5.     if move == 0:
6.         turnValue = math.pi / move
7.         move += 5
8.     else:
9.         turnValue = 0
10.        move = 0
```

Which error exists in the code?

Options:

- A. Line 04 has a syntax error because it should read (power == True).
- B. Line 07 has a syntax error because + = is an invalid statement.
- C. Line 06 causes a runtime error due to division by zero.
- D. Line 04 causes a runtime error because the expression is incomplete.

Solution:

Q5) What is the result of the following?

```
import random
print(random.randrange(3))
```

Options:

- a) It will display random number is the range of 1 to 3
- b) It will display random number is the range of -2 to 2
- c) It will display random number is the range of 0 to 2
- d) Error

Solution:

Q6) You need to write code that generates a random **float** with a minimum value of 0.0 and a maximum value of 1.0. Which statement should you use?

Options:

- A. `random.randrange(0.0, 1.0)`
- B. `random.randrange()`
- C. `random.random()`
- D. `random.randint(0, 1)`

Solution:

Q7) You are writing code that generates a random integer with a minimum value of 5 and a maximum value of 11. Which two functions should you use?

Options:

- A. `random.randint(5, 12)`
- B. `random.randint(5,11)`
- C. `random.randrange(5, 12, 1)`
- D. `random.randrange(5, 11, 1)`

Solution:

Q8) You work on a team that is developing a game for AdventureWorks. You need to write code that generates a random number that meets the following requirements: 0

- ❖ . The number is a multiple of 5.
- ❖ . The lowest number is 5.
- ❖ . The highest number is 100.

Which code segment will meet the requirements?

Options:

A. <code>from random import randrange; print(randrange(5, 100, 5))</code>	B. <code>from random import randint; print(randint(1, 20) * 5)</code>
C. <code>from random import randint; print(randint(0, 20) * 5)</code>	D. <code>from random import randrange; print(randrange(0, 100, 5))</code>

Solution:

Q9) Consider the following code in the module m1.py:

```
def f1():  
    print('hello')
```

Which two of the following are the correct options of using the above module.

Options:

a) import m1; m1.f1()	b) from m1 import f1(); f1()
c) from m1 import *; f1()	d) from m1 import f1; m1.f1()

Solution: