

## UNIT 11 — MODULES

---


Hey there! Welcome to Unit 11. Today we are going to learn about one of the most powerful features of Python: MODULES!

Imagine you are building a huge LEGO castle. Instead of making every single brick yourself, you use pre-made sets (like a window set, a door set, etc.).

In Python, a **MODULE** is just a file containing Python code (functions, variables, classes) that you can reuse in other programs.

### Why use Modules?

1. **Reusability:** Write once, use everywhere!
2. **Organization:** Keep your code clean and manageable.
3. **Namespace:** Avoid variable name conflicts.

 **Pro Tip!** A Python module is simply a `.py` file. If you save code in 'myfile.py', you have created a module named 'myfile'!

## 1. Importing Modules

To use a module, we use the `import` keyword. There are a few ways to do it:

### Method A: Basic Import

```
import math
print(math.sqrt(16))
```

Output: 4.0

### Method B: Import with Alias (Short name)

```
import datetime as dt
print(dt.datetime.now())
```

Output: 2026-05-18 17:30:00...

## Method C: Import specific things

```
from random import randint
print(randint(1, 10))
```

-----  
Output: 7 (random)

## The 'math' Module

The 'math' module is a built-in module that provides mathematical functions.

### Common Math Functions:

- `math.sqrt(x)` - Returns square root of x
- `math.pow(x, y)` - Returns x raised to power y
- `math.ceil(x)` - Rounds UP to nearest integer
- `math.floor(x)` - Rounds DOWN to nearest integer
- `math.pi` - Constant value of PI (3.1415...)

```
import math
print(math.ceil(4.2)) # Output: 5
print(math.floor(4.8)) # Output: 4
print(math.pi) # Output: 3.141592653589793
```

-----  
Output:

5


4

3.141592653589793

### ⚠ Common Mistake:

Trying to use math functions without importing.

**Fix:** Always add `import math` at the top of your file!

 **Real-world Example:** Engineers use the math module to calculate trajectories, areas of circles, and complex physics simulations.

## The 'random' Module

Want to build a game? Or pick a lucky winner? The 'random' module is your best friend!

### Key Functions:

- `random.random()` - Random float between 0.0 and 1.0

- `random.randint(a, b)` - Random integer between a and b (inclusive)
- `random.choice(list)` - Picks a random item from a list
- `random.shuffle(list)` - Mixes up the items in a list

```
import random
fruits = ['Apple', 'Banana', 'Cherry']
print(random.choice(fruits))
print(random.randint(1, 100))
```

Output:

```
Banana
42
```

### ? Interview Question:

**Q:** What is the difference between `random()` and `randint()` ?

**A:** `random()` returns a float between 0 and 1, while `randint(a, b)` returns a whole number between the range you specify.

## The 'datetime' Module

Handling dates and times can be tricky, but the `'datetime'` module makes it easy.


### Getting Current Date & Time:

```
import datetime
now = datetime.datetime.now()
print(now)
print(now.year)
print(now.strftime("%A")) # Day name
```

Output:

```
2026-05-18 17:30...
2026
Monday
```

We use `.strftime()` to format dates into readable strings. For example, `'%Y'` is Year, `'%m'` is Month, and `'%d'` is Day.

 **Memory Trick!** `strftime` = String Format Time. It turns a date object into a String!

## Mini Exercise

Try to write a program that imports `'random'` and `'math'` to calculate the square root of a random number between 1 and 100.

```
# Solution
import math
import random
num = random.randint(1, 100)
res = math.sqrt(num)
print(f"Sqrt of {num} is {res}")
```

---

Output: **Sqrt of 64 is 8.0**