

## PYTHON PROGRAMMING COURSE - 2 MONTHS

### ✓ Module 1: PYTHON BASICS

- ✓ Variables and data types
- ✓ Type casting
- ✓ Constants in Python
- ✓ Input and output (input(), print())
- ✓ Comments and docstrings
- ✓ Reserved keywords

### MODULE 2: OPERATORS IN PYTHON

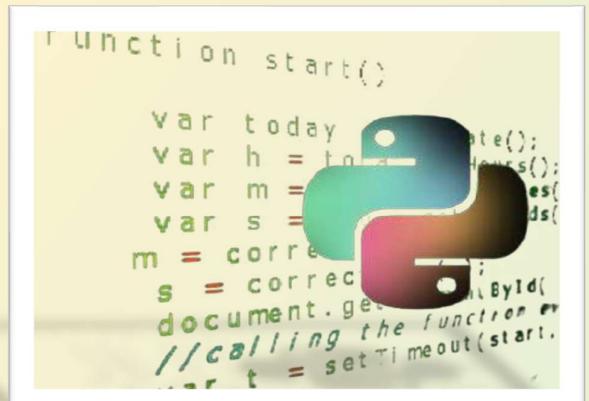
- ✓ Arithmetic operators
- ✓ Relational (comparison) operators
- ✓ Logical operators
- ✓ Assignment operators
- ✓ Identity operators (is, is not)
- ✓ Membership operators (in, not in)
- ✓ Operator precedence

### MODULE 3: CONTROL FLOW STATEMENTS

- ✓ Decision-making:
- ✓ if, if-else, if-elif-else
- ✓ Nested conditions
- ✓ Looping:
- ✓ for loop
- ✓ while loop
- ✓ Loop control:
- ✓ break, continue, pass
- ✓ else with loops

### MODULE 4: DATA STRUCTURES IN PYTHON

- ✓ Lists:
- ✓ Creation, indexing, slicing
- ✓ List methods
- ✓ Tuples:
- ✓ Immutable sequences
- ✓ Tuple packing and unpacking
- ✓ Sets:
- ✓ Creating sets
- ✓ Set operations (union, intersection, difference)
- ✓ Dictionaries:
- ✓ Key-value pairs
- ✓ Dictionary methods
- ✓ String operations and formatting



## MODULE 5: FUNCTIONS IN PYTHON

- ✓ Defining and calling functions
  - ✓ Function parameters and return values
  - ✓ Default, keyword, and variable-length arguments (\*args, \*\*kwargs)
  - ✓ Lambda functions
  - ✓ Scope and lifetime of variables
  - ✓ Recursion

## MODULE 6: FILE HANDLING

- ✓ Reading and writing text files
  - ✓ Reading and writing binary files
  - ✓ File methods (open, read, write, close)
  - ✓ File modes
  - ✓ Exception handling in file operations

## MODULE 7: EXCEPTION HANDLING

- ✓ Errors vs exceptions
  - ✓ Try-except block
  - ✓ Else and finally clauses
  - ✓ Raising exceptions
  - ✓ Custom exceptions

MODULE 8: OBJECT-ORIENTED PROGRAMMING IN PYTHON

- ✓ Classes and objects
  - ✓ Instance and class variables
  - ✓ Instance and class methods
  - ✓ Constructors ( init )
  - ✓ Inheritance (single, multiple, multilevel)
  - ✓ Method overriding
  - ✓ Polymorphism
  - ✓ Encapsulation and abstraction
  - ✓ Magic methods ( str , len , etc.)

## MODULE 9: ADVANCED PYTHON CONCEPTS

- ✓ Iterators and generators
  - ✓ Decorators
  - ✓ Comprehensions (list, set, dictionary)
  - ✓ zip(), map(), filter(), reduce()
  - ✓ Regular expressions (re module)
  - ✓ Date and time operations
  - ✓ JSON handling

MODULE 10: PROJECT WORK S PRACTICE

- ✓ Mini-project ideas:
    - Personal expense tracker
    - Weather app using API
    - Web scraper for news headlines
    - Simple quiz game
  - ✓ Debugging and best coding practices



```
self.logger = logging.getLogger(__name__)
self.logger.setLevel(logging.INFO)
self.path = os.path.join(os.getcwd(), 'fingerprint')
self.file = open(self.path, 'a')
self.fingerprints = {}
self.fingerprints_lock = threading.Lock()

@classmethod
def from_settings(cls, settings):
    debug = settings.getboolean('verbose', False)
    return cls(debug=debug, settings=settings)

def request_seen(self, request):
    fp = self.request_fingerprint(request)
    if fp in self.fingerprints:
        return True
    self.fingerprints[fp] = request
    self.file.write(fp + os.linesep)
```



Python Turtle - Drawing Rainbow Colours

In this Lesson, we build on the Python Turtle Drawing Skills which we have progressively developed in previous lessons. [Eg. You need to have done Lessons 1-6](#)  
We will be learning to continually change the Rainbow Colors in a Drawing, by using a STORAGE ARRAY of RGB Color Values that go from Red to Purple.

