TechSaksham | SAP | edunet foundation | Microsoft

Chapter: 2

Essentials of Python Programming Language

Disclaimer

The content is curated from online/offline resources and used for educational purpose only.

About the Course

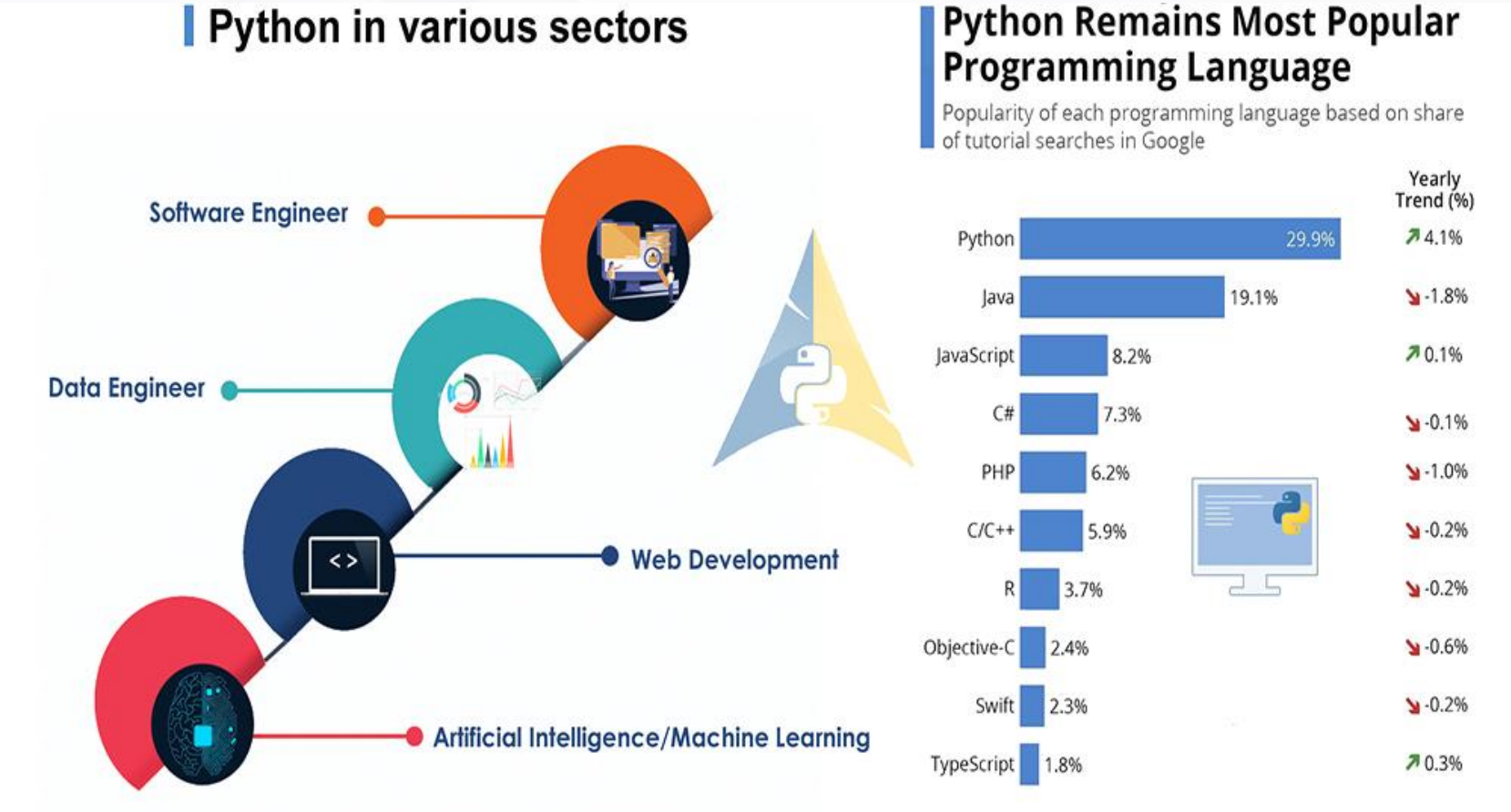
“The "Essentials of Python Programming Language" course introduces Python's history, features, and installation, followed by fundamental syntax covering variables, data types, and operators. It then progresses to teach control structures like conditional statements and loops, along with error handling. Students learn to define, call, and utilize functions, and explore the organization of code through modules. By the course's end, participants gain a solid foundation in Python, enabling them to write basic to moderately complex programs proficiently.”

Learning Objectives

You will learn in this lesson:

- Python Programming
- How to write first program
- Perform basic operations
- Python Data Structures
- Python OOPs Concept

Why do we need to learn Python?



Source : https://jobaxle.com/blog_detail/career-opportunity-in-python/24

Python Introduction

- Python uses an interpreter. Not only can we write complete programs, but we can also work with the interpreter in a statement-by-statement mode enabling us to experiment quite easily.
- Python is especially good for our purposes in that it does not have a lot of “overhead” before getting started.
- It is easy to jump in and experiment with Python in an interactive fashion.

Source : OIP.AQ5rAzGvicbIAudCjnmlyQHaHa (266×266) (bing.com)

Python Interfaces

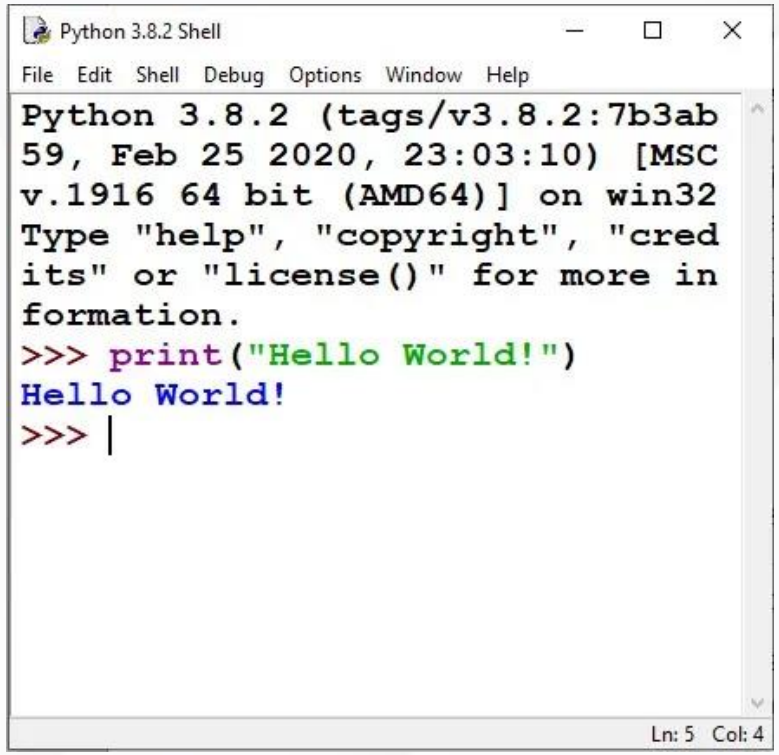
- IDLE – a cross-platform Python development environment
- Anaconda is a Python distribution focused on data driven projects.
- PythonWin – a Windows only interface to Python
- Python Shell – running 'python' from the Command Line opens this interactive shell

Source : OIP.AQ5rAzGvicbIAudCjnmlyQHaHa (266×266) (bing.com)

IDLE - Development Environment

IDLE helps you program in python by:

- Color-coding your program code
- Debugging
- Auto-indent
- Interactive shell



Let's play this

Source : https://www.youtube.com/watch?v=x7X9w_GIm1s

Classroom Activity

Lab 5: Installation of Python and Hello World Program in Python

Solution: GitHub Link

Python Data Structures: List

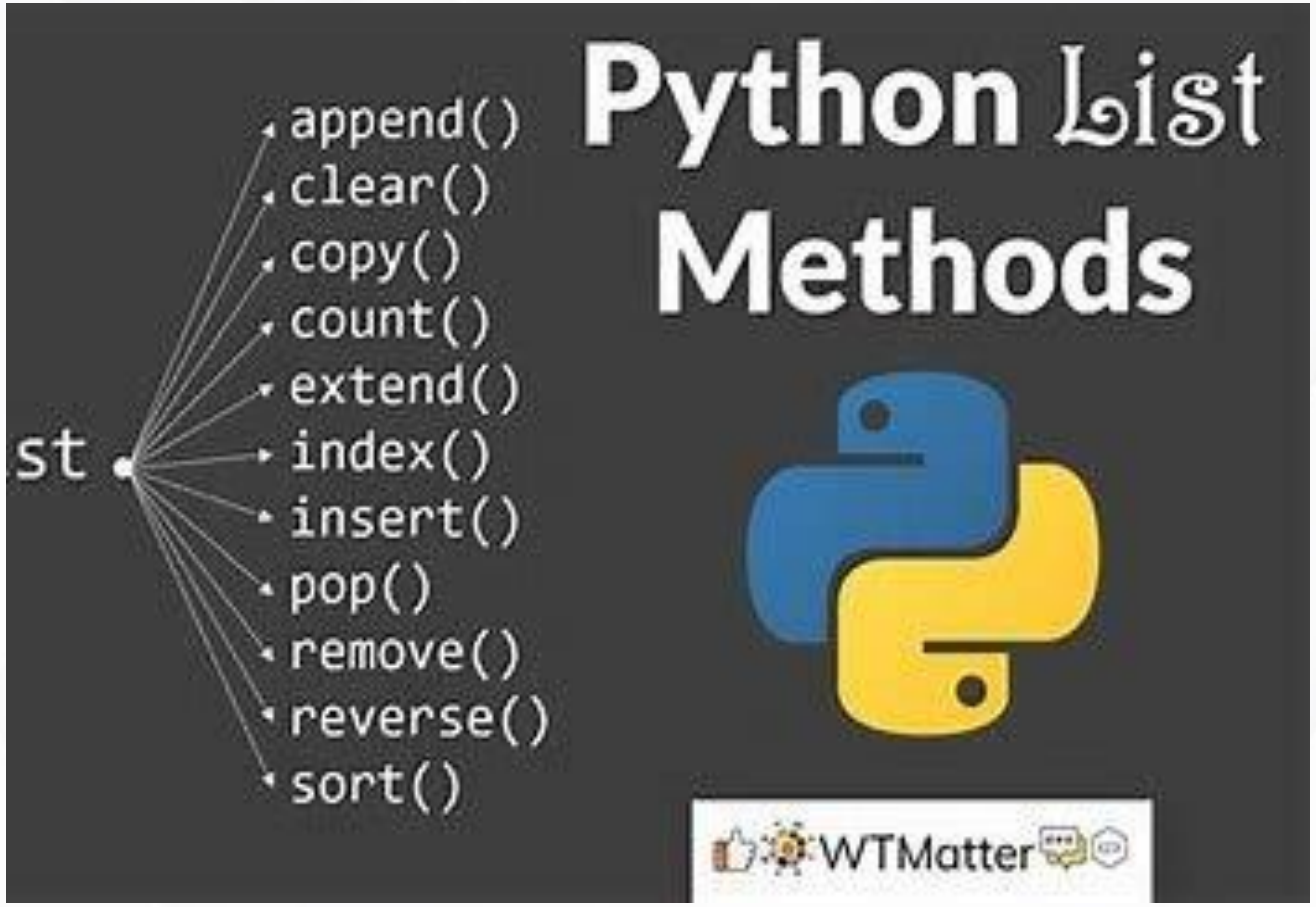
- Think of a list as a stack of cards, on which your information is written
- The information stays in the order you place it in until you modify that order
- Written as var[*index*], index refers to order within set (think card number, starting at 0)

Example:

Lists (mutable sets of strings)

var = [] # create list

var = [‘one’, 2, ‘three’, ‘banana’]

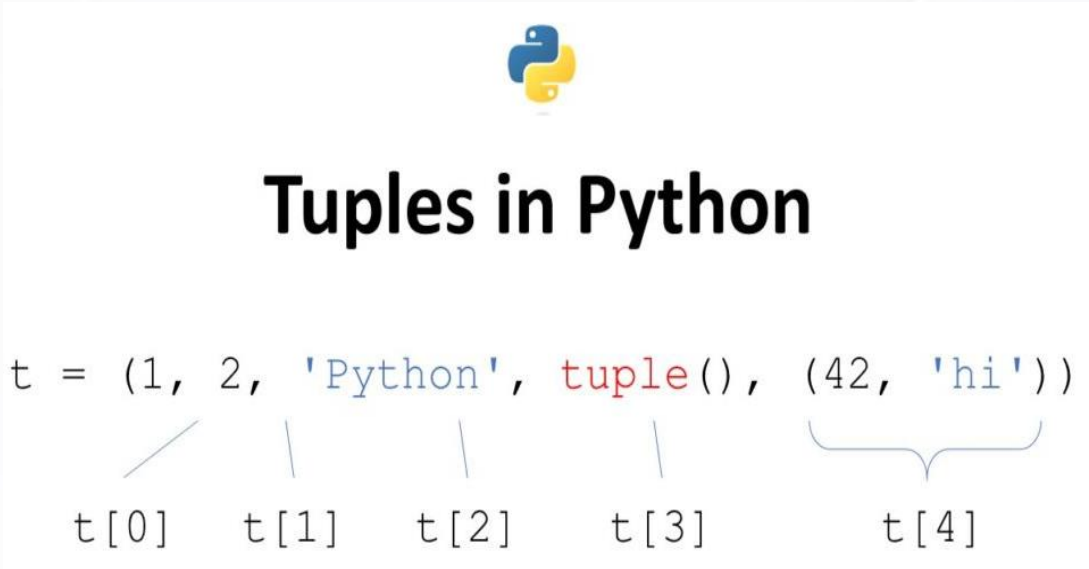


Source : OIP.puQxxCnV_sa5ageMV--A-wHaEj (351×234) (bing.com)

Python Data Structure: Tuples

Like a list, tuples are iterable arrays of objects
Tuples are immutable -
once created, unchangeable

To add or remove items, you must redeclare
Example uses of tuples
County Names
Land Use Codes
Ordered set of functions



Source : https://blog.finxter.com/the-ultimate-guide-to-python-tuples/

Tuple Methods

count()	count element on tuple
index()	find an element on tuple
len()	find length on tuple
min()	find minimum on tuple
max()	find maximum on tuple
sum()	to sum all element on tuple
sort()	to sort all element on tuple
loop tuple	loop all element on tuple

Source : https://miro.medium.com/v2/resize:fit:1400/format:webp/1*KAaTE3P0IEOGELp4IfCNXA.png

Python Data Structure: Dictionary

Dictionaries are sets of key & value pairs

Allows you to identify values by a descriptive name instead of order in a list

Keys are unique:

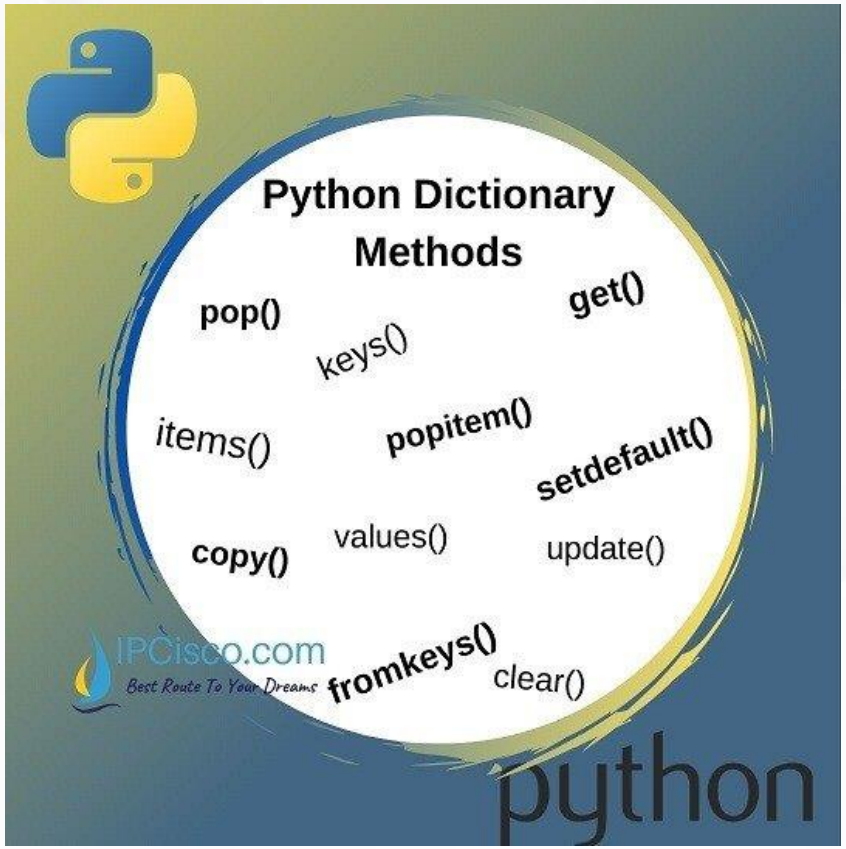
var[‘item’] = “apple”

var[‘item’] = “banana”

print var[‘item’] prints just banana

Dictionaries (associative arrays or ‘hashes’)

```
var = {} # create dictionary
var = {‘lat’: 40.20547, ‘lon’: -74.76322}
var[‘lat’] = 40.2054
```



Source : https://ipcisco.com/wp-content/uploads/2021/04/python-dictionary-methods-ipcisco.com_.jpg

Classroom Activity

Lab 6: Data Structures in Python

Solution: **GitHub Link**

Self Practice

1. Predict proper output for the below code
2. Write a python program to concatenate two strings
3. Write a python program to reverse string

Solution: GitHub Link

Project

Basic Student Record Management System

Overview: This project uses tuples, lists, and dictionaries to manage student records. It will perform actions based on predefined choices and data.

Solution: GitHub Link

Indentation and Blocks

- Python uses whitespace and indents to denote blocks of code
- Lines of code that begin a block end in a colon:
- Lines within the code block are indented at the same level
- To end a code block, remove the indentation
- You'll want blocks of code that run only when certain conditions are met

Conditional Branching

```
if and else
    if variable == condition:
                #do something based on v == c
   else:
                #do something based on v != c
```

elif allows for additional branching

```
if condition:
    ……..……..
elif another condition:
   ……..……..
else: #none of the above
   ……..……..
```

Looping with For

- For allows you to loop over a block of code a set number of times

For is great for manipulating lists:

Example:

```
a = ['cat', 'window', 'defenestrate']
    for x in a:
        print x, len(x)
```

Results:

- cat 3
- window 6
- defenestrate 12

Classroom Activity

Lab 7: Control Statements in Python

Solution: GitHub Link

Self Practice

1. Python Program to Check Prime Number
2. Python Program to Find the Factorial of a Number
3. Python Program to Display the multiplication Table

Solution: GitHub Link

Project

Student Record Management System

Objective: Create a Python program to manage student records using different data structures. This project will help you understand and implement lists, dictionaries, tuples and Control Statements in Python.

Solution: GitHub Link

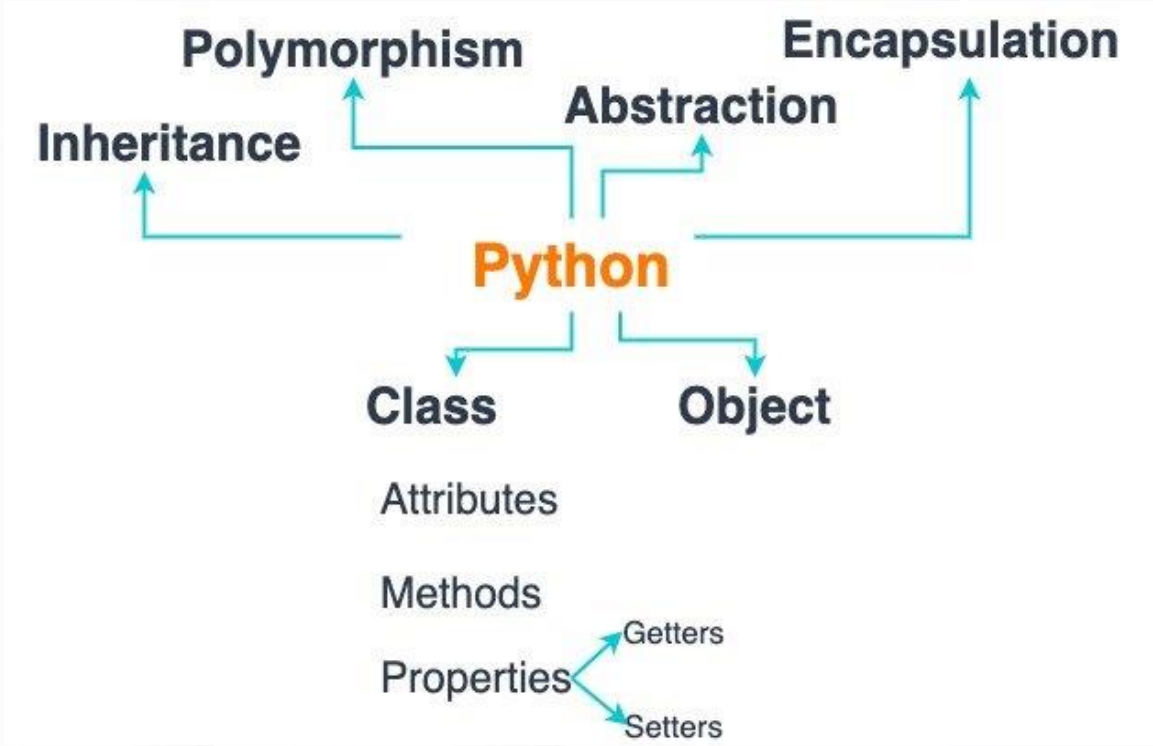
Python OOPs Concepts

- Python is also an object-oriented language since its beginning.
- It allows us to develop applications using an Object-Oriented approach.
- In Python, we can easily create and use classes and objects.

Source : https://editor.analyticsvidhya.com/uploads/50548logo.png

OOPs Principles

- Class
- Object
- Method
- Inheritance
- Polymorphism
- Data Abstraction
- Encapsulation



Source: https://svitla.com/blog/design-object-oriented-code-python

Class

The class can be defined as a collection of objects. It is a logical entity that has some specific attributes and methods. For example: if you have an employee class, then it should contain an attribute and method, i.e. an email id, name, age, salary, etc.

Syntax

```
class ClassName:
    <statement-1>
    .
    .
    <statement-N>
```

Object

- The object is an entity that has state and behavior. It may be any real-world object like the mouse, keyboard, chair, table, pen, etc.
- Everything in Python is an object, and almost everything has attributes and methods. All functions have a built-in attribute __doc__, which returns the docstring defined in the function source code.

Example:

```
class car:
    def __init__(self,modelname, year):
        self.modelname = modelname
        self.year = year
    def display(self):
        print(self.modelname,self.year)

c1 = car("Toyota", 2016)
c1.display()
```

Method

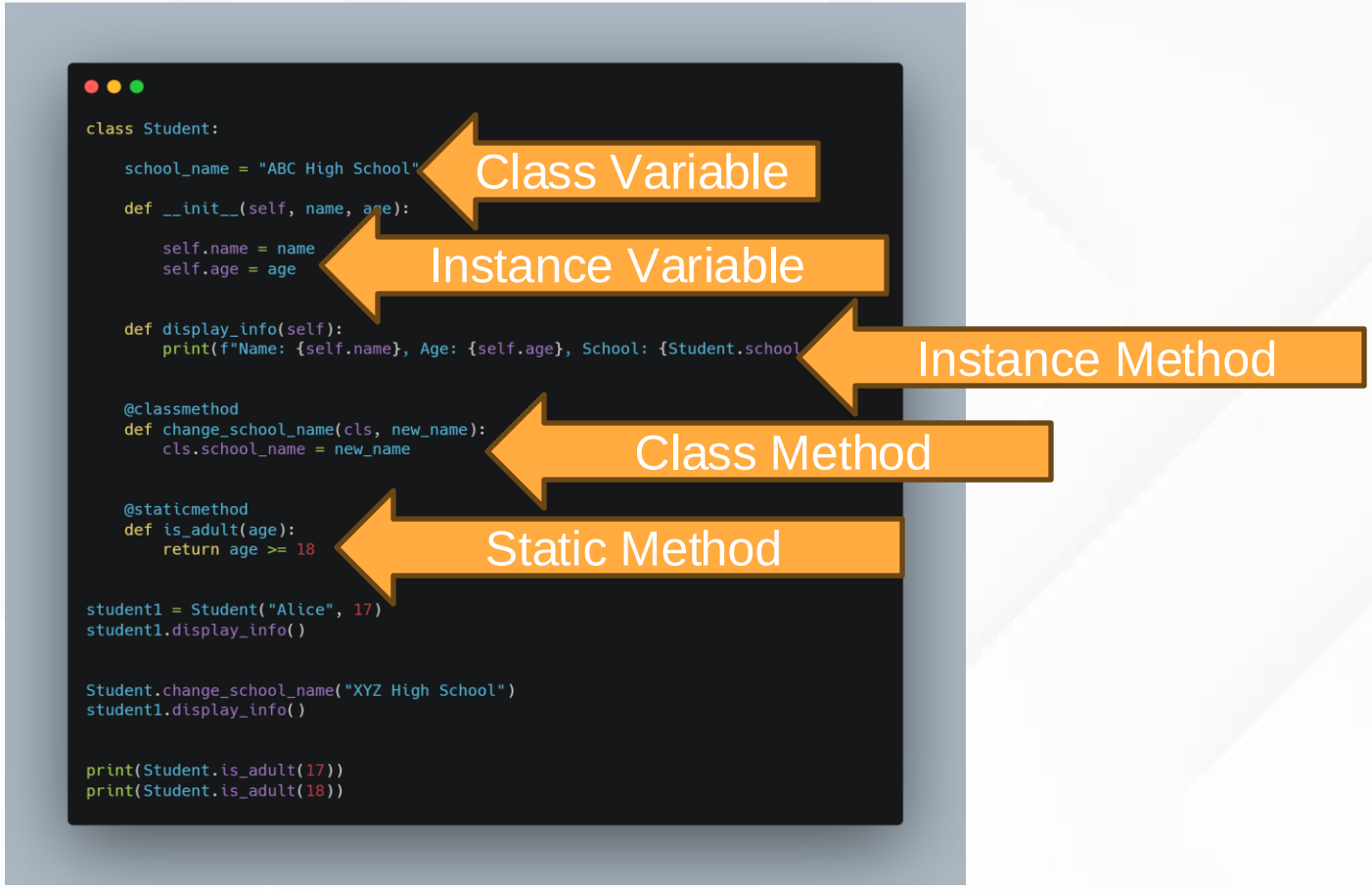
The method is a function that is associated with an object. In Python, a method is not unique to class instances. Any object type can have methods.

Activity Time: List out the Classes, Objects, and Methods

1) Design a system where each object represents a vehicle with attributes for make, model, and year. Implement methods to display the vehicle's details and update its manufacturing year. Create a vehicle object, display its details, update its year, and display the updated details.

2) Build a system to manage a collection of books. Each library object should store a list of book titles. Implement methods to add a book, remove a book, and list all books in the collection. Create a library object, add and remove books, and list the current collection of books.

Activity Time: Identify Instance Variable, Class Variable, Instance, Class and Static methods in given code snippet



Classroom Activity

Lab 8: OOPS concepts in Python

Solution: GitHub Link

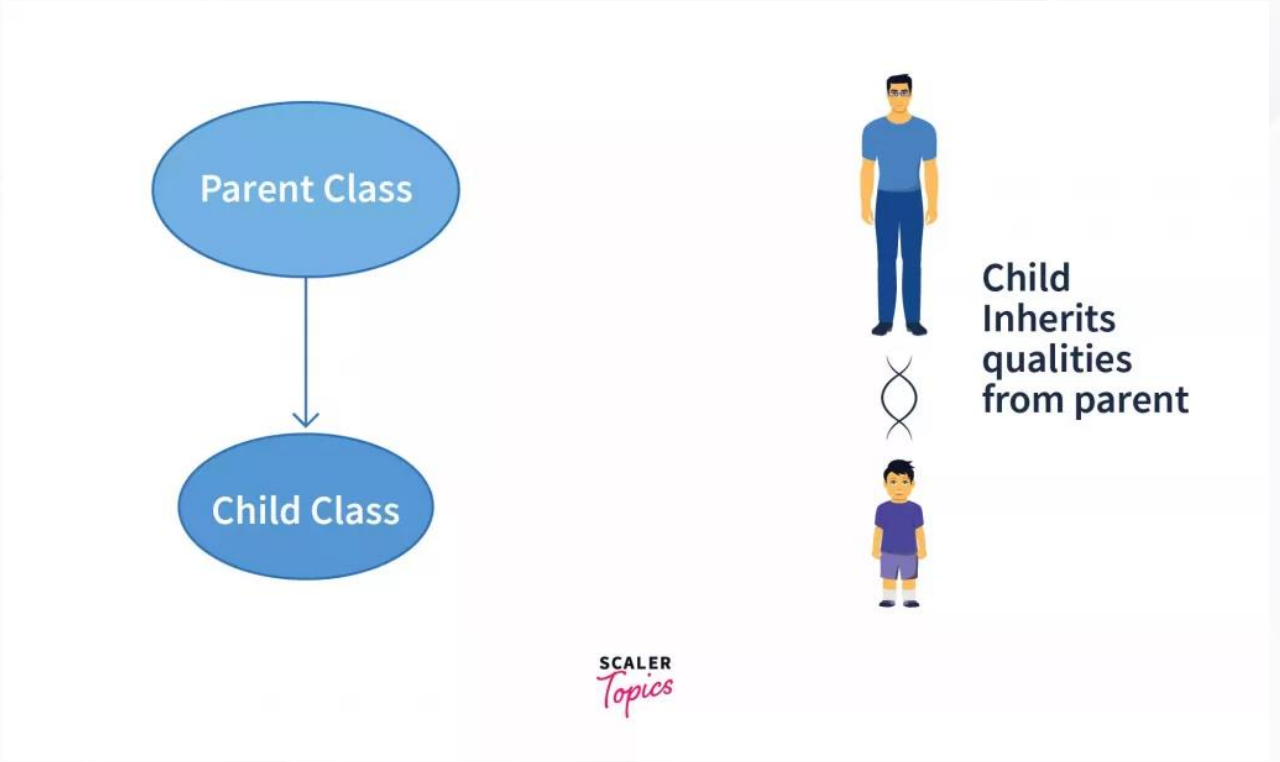
Self Practice

1. Fill in the Line class methods to accept coordinates as a pair of tuples and return the slope and distance of the line.
2. Create a class and attributes name and age and display a greeting message on the display
3. Write a python program describing inheritance for example Animal is a class and you can inherit various properties like all animals eat,bark etc

Solution: GitHub Link

Python Inheritance

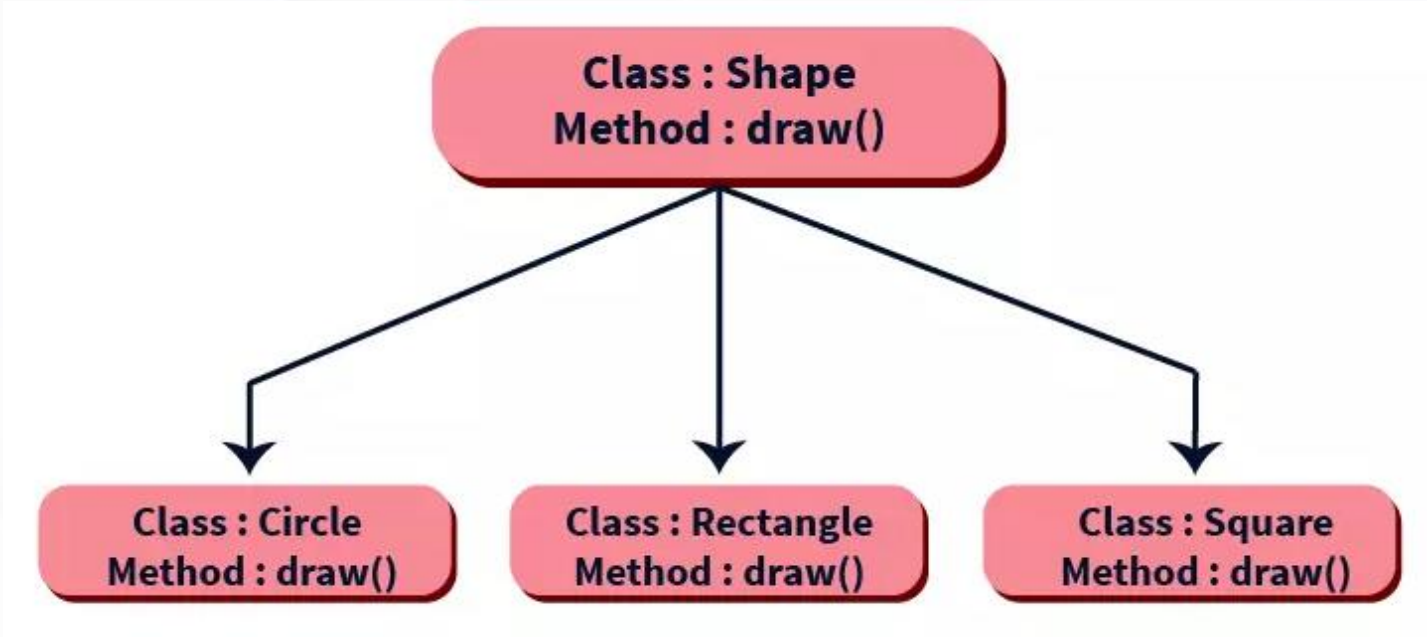
- Inheritance allows us to define a class that inherits all the methods and properties from another class.
- **Parent class** is the class being inherited from, also called base class.
- **Child class** is the class that inherits from another class, also called derived class.



Source: https://www.scaler.com/topics/python/inheritance-in-python/

Python Polymorphism

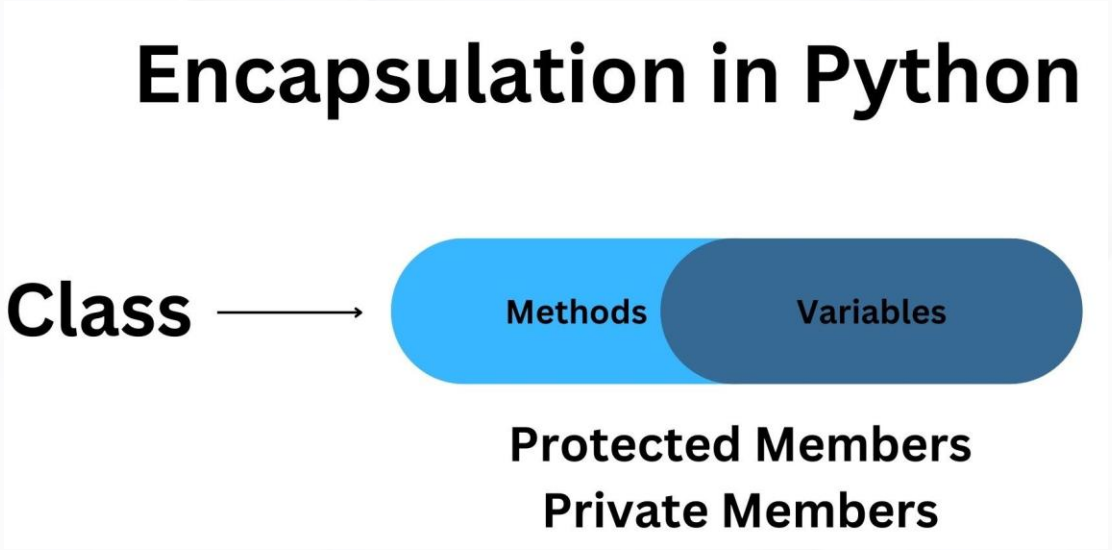
The word "polymorphism" means "many forms", and in programming it refers to methods/functions/operators with the same name that can be executed on many objects or classes.



Source: https://www.scaler.com/topics/python/polymorphism-in-python/

Encapsulation in Python

Encapsulation is a mechanism of wrapping the data (variables) and code acting on the data (methods) together as a single unit. In encapsulation, the variables of a class will be hidden from other classes, and can be accessed only through the methods of their current class.



Source: https://www.boardinfinity.com/blog/understanding-encapsulation-in-python/

Data Abstraction in Python

Data abstraction and encapsulation both are often used as synonyms. Both are nearly synonyms because data abstraction is achieved through encapsulation.

Abstraction is used to hide internal details and show only functionalities. Abstracting something means to give names to things so that the name captures the core of what a function or a whole program does.

Classroom Activity

Lab 9: Advanced OOPS concepts in Python

Solution: GitHub Link

Summary

- Python is a high-level, versatile programming language known for its readability and simplicity.
- Python supports both procedural and object-oriented programming paradigms and has a rich ecosystem of libraries.
- Python's indentation-based syntax enforces code readability.
- Python is widely used for web development, data analysis, artificial intelligence, and automation due to its extensive standard library and active community support.

Quiz

1. What is Python primarily known for?

a) Speed optimization
b) Complex syntax
c) Readable and clean syntax
d) Low-level memory management

Answer: c
Readable and clean syntax

Quiz

2. Which of the following is used to define a block of code in Python?

a) Parentheses ()

b) Brackets []

c) Indentation

d) Semicolons ;

Answer: c
Indentation

Quiz

3. What does the if statement do in Python?

a) Iterates through a sequence
b) Defines a loop
c) Executes a block of code conditionally
d) Prints a value

Answer: c
Executes a block of code conditionally

Quiz

4. Which Python data type is ordered, mutable, and allows duplicate elements?

a) Set
b) List
c) Tuple
d) Dictionary

Answer: b
List

References

- https://docs.python.org/3/library/
- https://www.tutorialspoint.com/numpy
- https://towardsdatascience.com/
- https://pynative.com/
- https://medium.com/javarevisited/top-5-python-frameworks-for-web-development-e034ebe85574
- https://www.projectpro.io/article/machine-learning-frameworks/509
- https://www.javatpoint.com/python-oops-concepts
- https://www.shiksha.com/online-courses/articles/abstraction-in-python/
- https://www.boardinfinity.com/blog/understanding-encapsulation-in-python/

Thank you!