19I210 Python Programming Laboratory

Dr.B.Sangeetha

Assistant Professor (Sr.Gr)

Objectives

› To inculcate analytical and programming skills

› To introduce syntax and semantics of Python

Outcomes

› To create programs and application in Python

› To demonstrate mastery in application development

Syllabus

- Basic Programs
- Control Statements and Looping
- Functions and Lambda Functions
- String Handling
- Lists, Tuples, Set and Dictionary
- Files and I/O Handling
- Packages & Modules
- Exception Handling
- Class and Inheritance
- Database Connectivity

Python

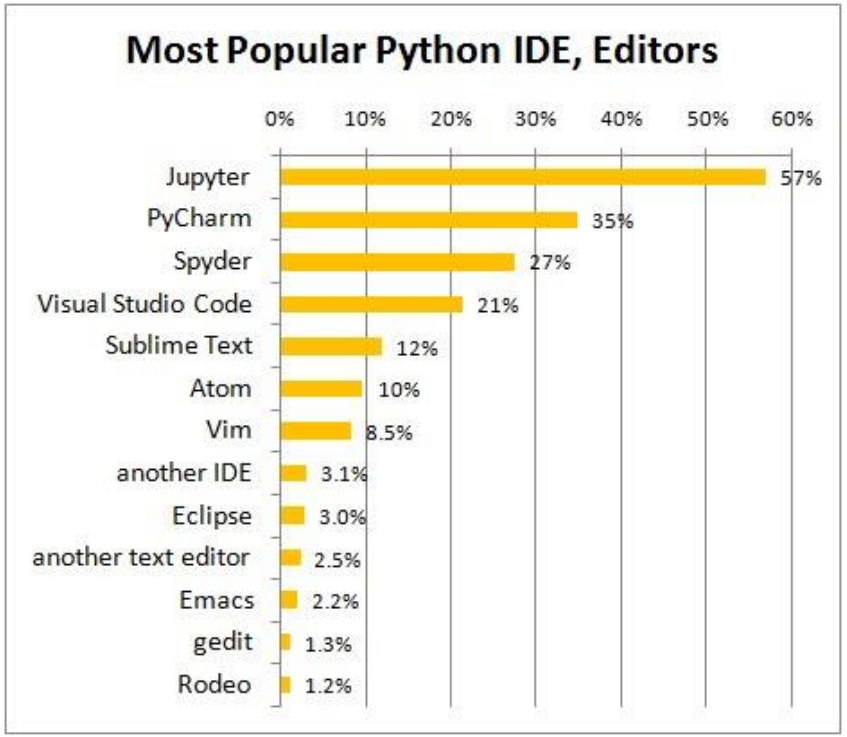
- **High-level programming language.**
- Emphasizes **code readability** with the use of **significant indentation**.
- Creator - **Guido van Rossum**

1991	Python 0.2.0
2000	Python 2.0
2008	Python 3.0 (not backward compatible)

Install Python : https://www.python.org/downloads/ (Compatible to the OS and System Architecture)

Python Editors

› IDLE
(*Integrated Development Learning Environment)*
› Jupyter
› Spyder



› Google Colab (*colab.research.google.com*)

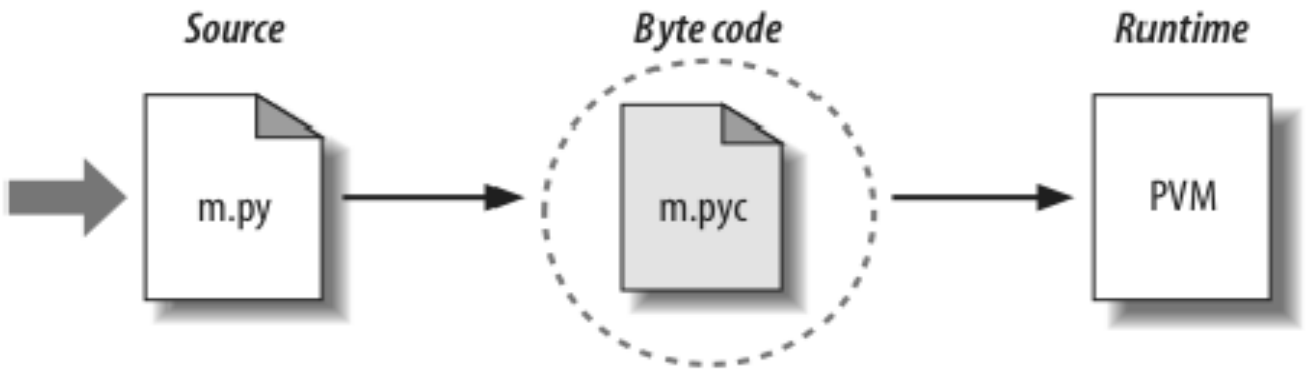
Free web service, offers run-time environment through

Jupyter Notebooks

Compiler vs Interpreter

- Translates source code → machine code
- Compiler – Entire file at a time
- Interpreter – One instruction at a time

Python *No explicit compilation*



Why Python?

- Easy to write programs
- Wide range of applications
  - Machine Learning, Web Development, Gaming
- Wider community support

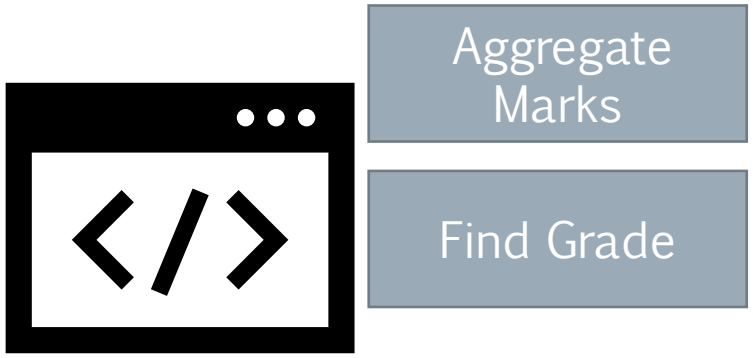
Use cases

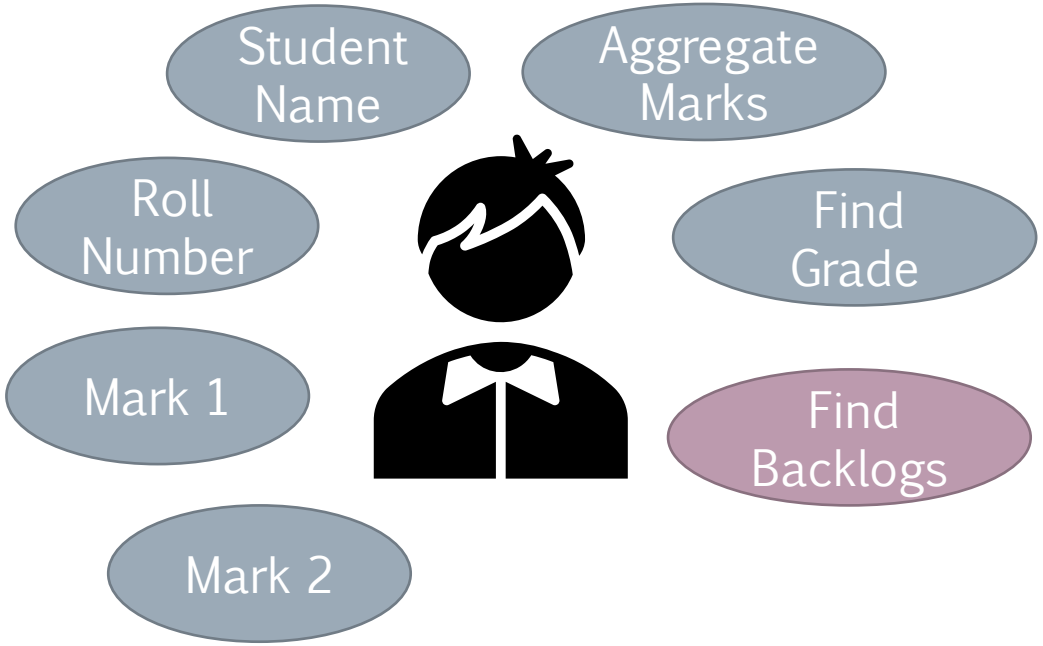
- Netflix
- Uber
- Pinterest
- Quora
- Instagram

Object Oriented Programming

Concept of class and objects /

Concept of functions and logics





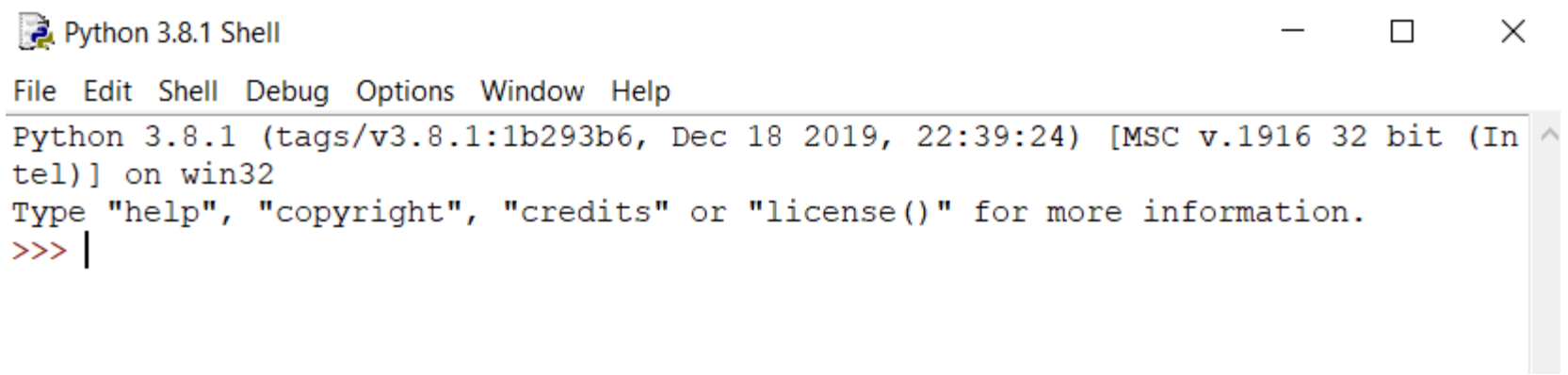
Object Oriented Programming Principles

- Encapsulation
  - Wrap data and code
  - Expose only what is required
- Data Hiding
  - Hide complex details
- Inheritance
  - Reusability of the common data and methods
- Polymorphism
  - Objects exhibit different behavior based on the current context

Basics of Python

- Source file created with .py extension
- Instructions
    - Expressions (Operators , values)

>>> (prompt)



Basics of Python

```
>>> 2
2
>>> 2 + 3
5
```

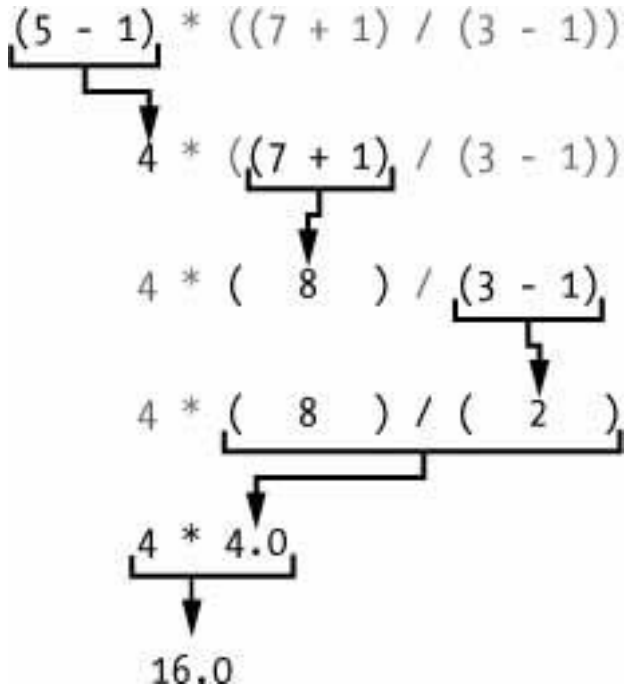
Data

Data type	Examples
Integers	`-2, -1, 0, 1, 2, 3, 4, 5`
Floating-point numbers	`-1.25, -1.0, -0.5, 0.0, 0.5, 1.0, 1.25`
Strings	`'a', 'aa', 'aaa', 'Hello!', '11 cats'`

Operators

Precedence & Associativity

Operator	Operation	Example	Evaluates to . . .
**	Exponent	2 ** 3	8
%	Modulus/remainder	22 % 8	6
//	Integer division/floored quotient	22 // 8	2
/	Division	22 / 8	2.75
*	Multiplication	3 * 5	15
-	Subtraction	5 - 2	3
+	Addition	2 + 2	4

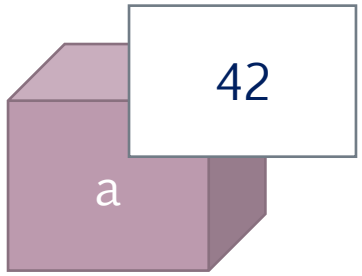


Variables

Storing values

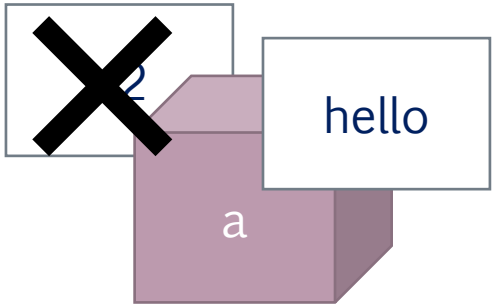
Associate name with a quantity

```
>>> a = 42
```



Can store any type of value

```
>>> a = 'hello'
```



print() and input() functions

Display values

Accepts one argument

Expression (or) value to be displayed

>>> print('hello')

```
>>> print('hello')
hello
```

Accept input , receive and store it as string

>>> a = input()

Waits for user input , Convert it into string

str(), int(), float() functions

str()	convert int (or) float to a string
int()	convert numeric string (whole number) (or) float to integer
float()	convert numeric string (decimal number) (or) integer to float

References

https://automatetheboringstuff.com/