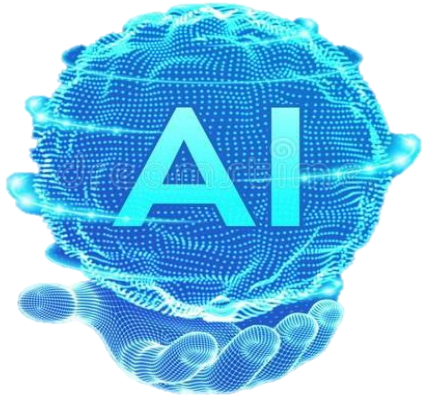
AI is a technique of getting machines to work and behave like humans.



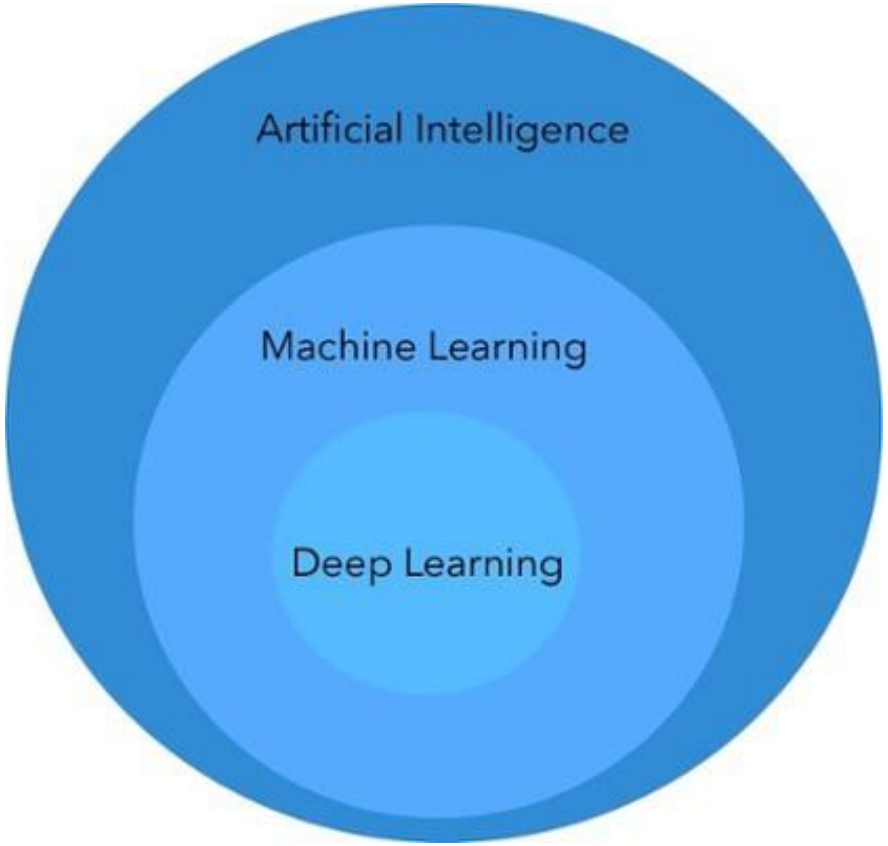
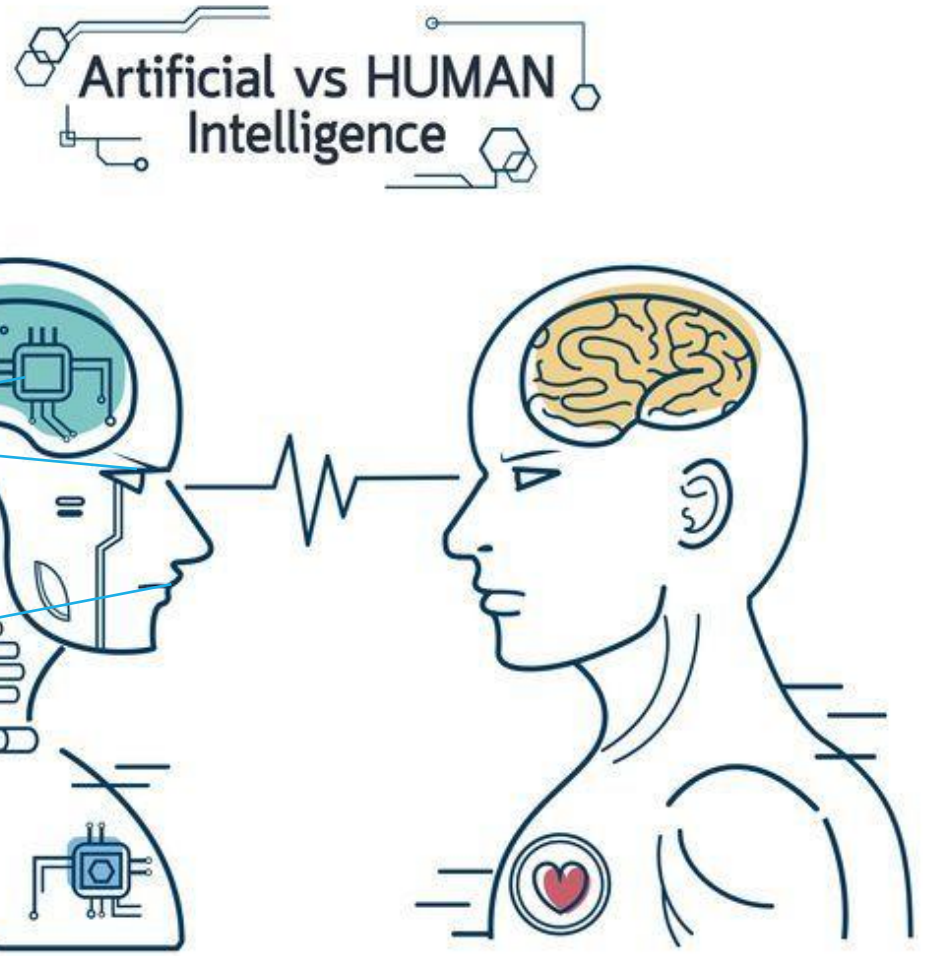


Image Processing

Computer Vision

Neural Network

Natural Language Processing



Machine Learning

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.

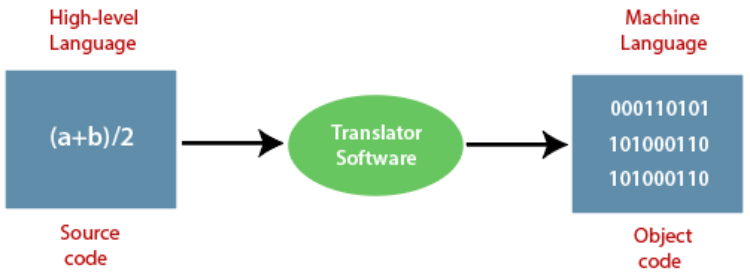





Machine learning and AI are built on mathematical principles like Calculus, Linear Algebra, Probability, Statistics, and Optimization

Static Programming

Compiler



Java

```
int a=9;
int b =4;
int c;
c = a/b  2.25
println(c)  2
```

take input

translator

t.V

t.V & I.v

Result

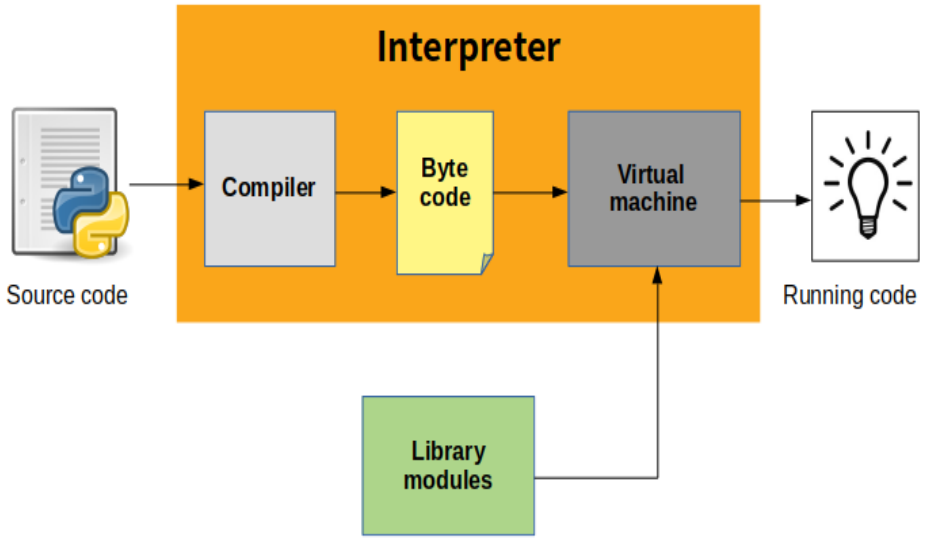
Dynamic Programming

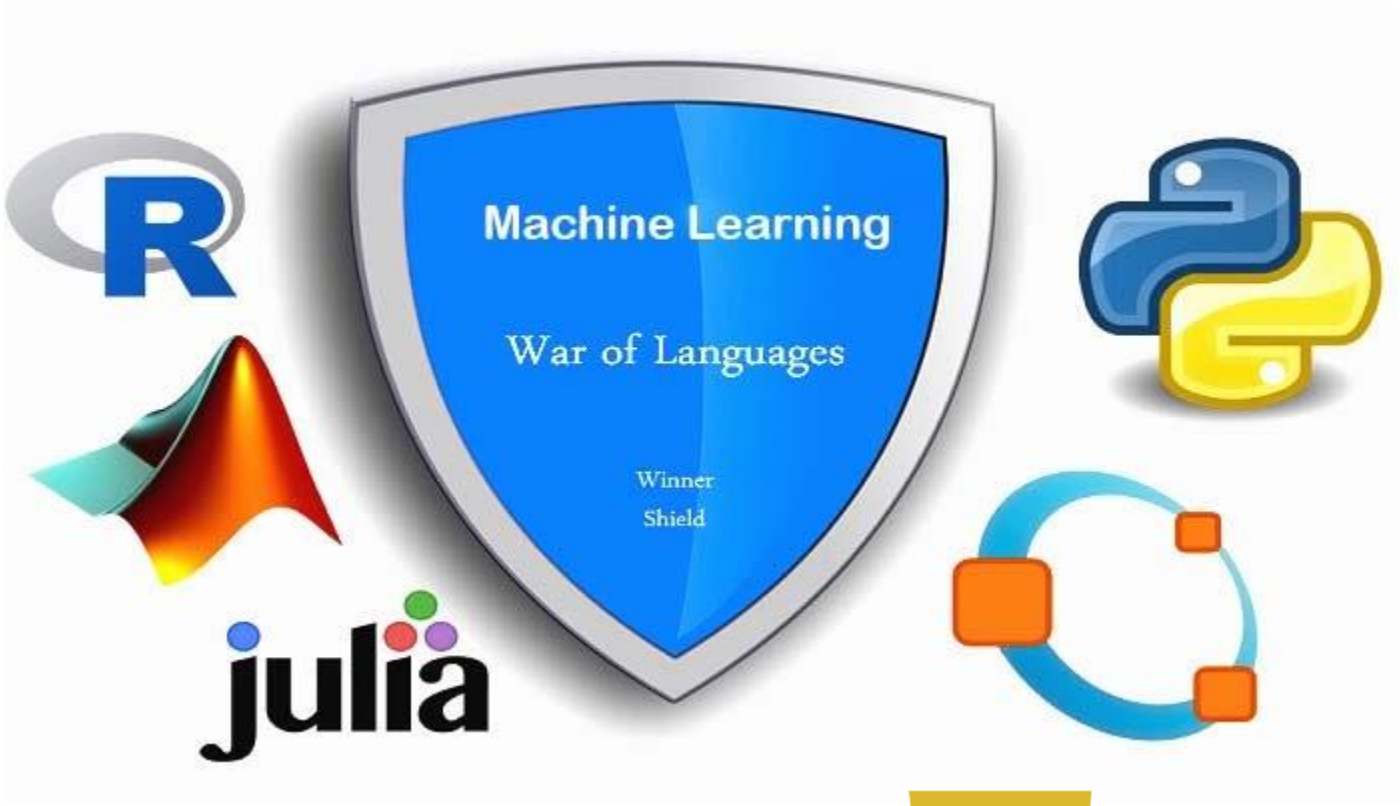
Interpreter

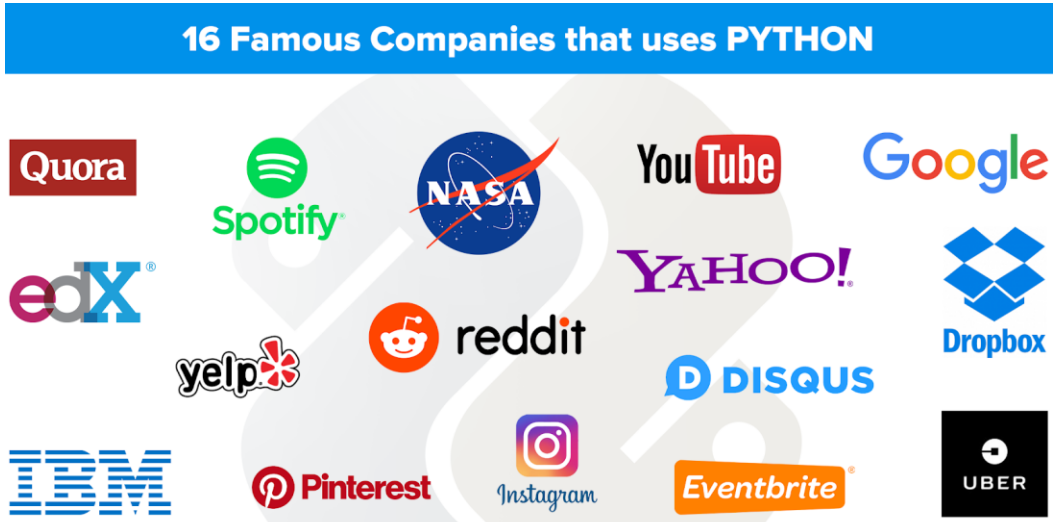
Take input

Translator

Result




Machine Learning
War of Languages
Winner
Shield
julia
sas



GitHub: Python-Tutorials

Python is

High-Level Language

Interpreted Language

Dynamically Typed Language

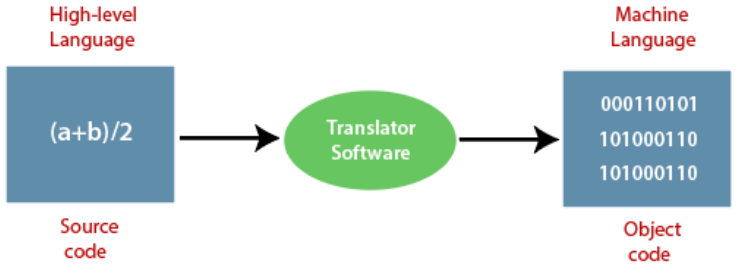
Large Standard Library

Portable language

Extensible feature

Object-Oriented Language

Free and Open Source



1 Domain Knowledge

2 Data collection

3 Data Analysis:
- Null values
- cleaning data
- Process the data
- Data Selection [Features and Target]: Supervised

4 Splitting into training and testing

5 Making the Algorithm: Logistic Regression

6 Find the Accuracy: 77%

7 Accuracy Analysis

8 User Testing [only 1 user at a time]

9 Project Deployment [Django, Flask, Streamlit]

Machine Learning Algorithm

Supervised
(Data is labelled)

Un-Supervised
(Data is unlabelled)

Re-inforement Learning

Weather forecast Detection
Cancer Detection
Churn Prediction
2 type of Target is available

Recommendation System

Automatic Car
AI Games

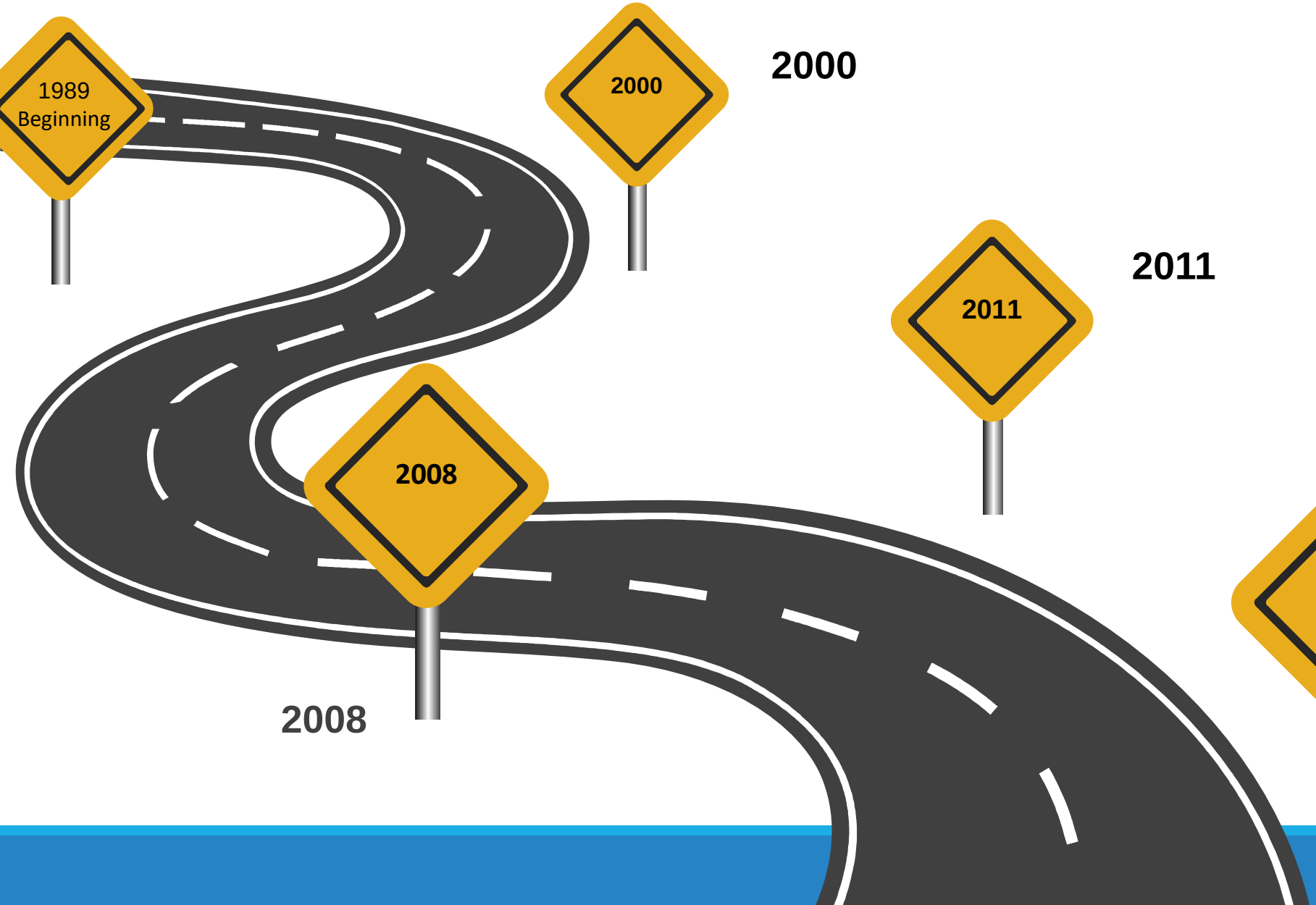
Regression: Target is continuous
Linear Regression

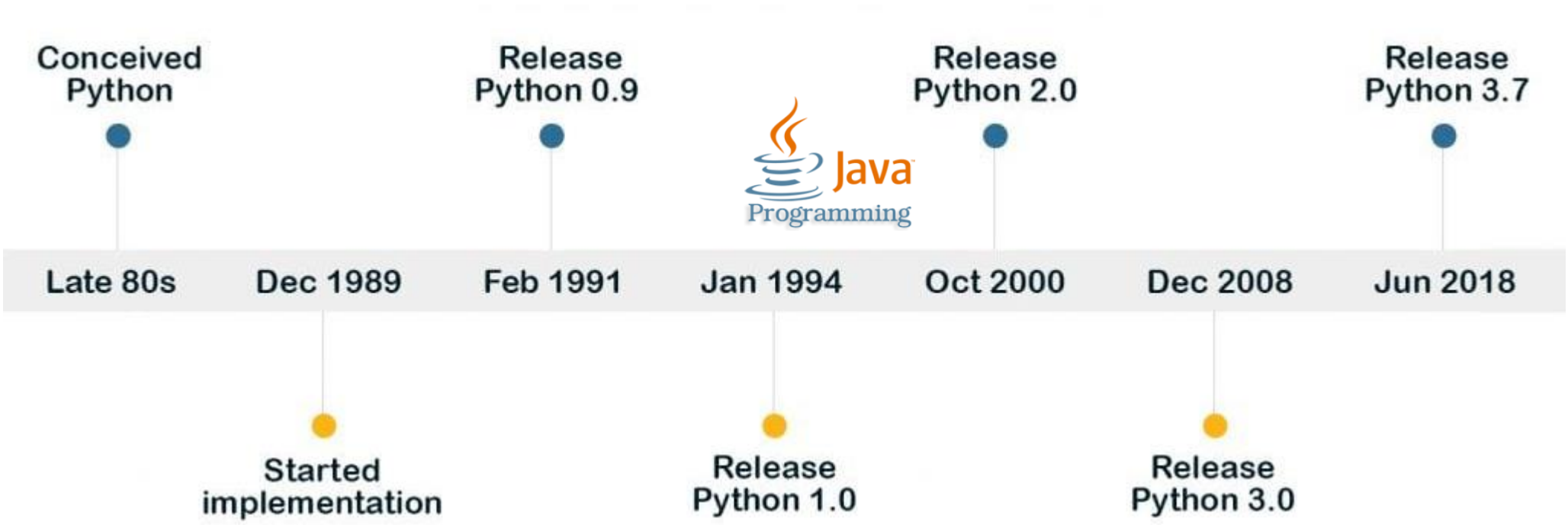
Logistic Regression
K-Nearest Neighbour
Support vector Machine
Decision Tree
Random Forest
Xgboosting

Label: Target, output, y

Classification: Target is discrete
Naive Biased

History of Python




Conceived Python
Late 80s
Started implementation
Dec 1989
Release Python 0.9
Feb 1991
Release Python 1.0
Jan 1994
Java Programming
Release Python 2.0
Oct 2000
Release Python 3.0
Dec 2008
Release Python 3.7
Jun 2018

Basics and Intermediate

Basics of Python
Operators
Data Types
Conditional statement
Loop
Function
Module
Introduction of Analysis: Pandas, Numpy & Maplotlib

Python.org

Anaconda:

Jupyter notebook
Spyder

Pycharm: S/W

VB code: Web Development

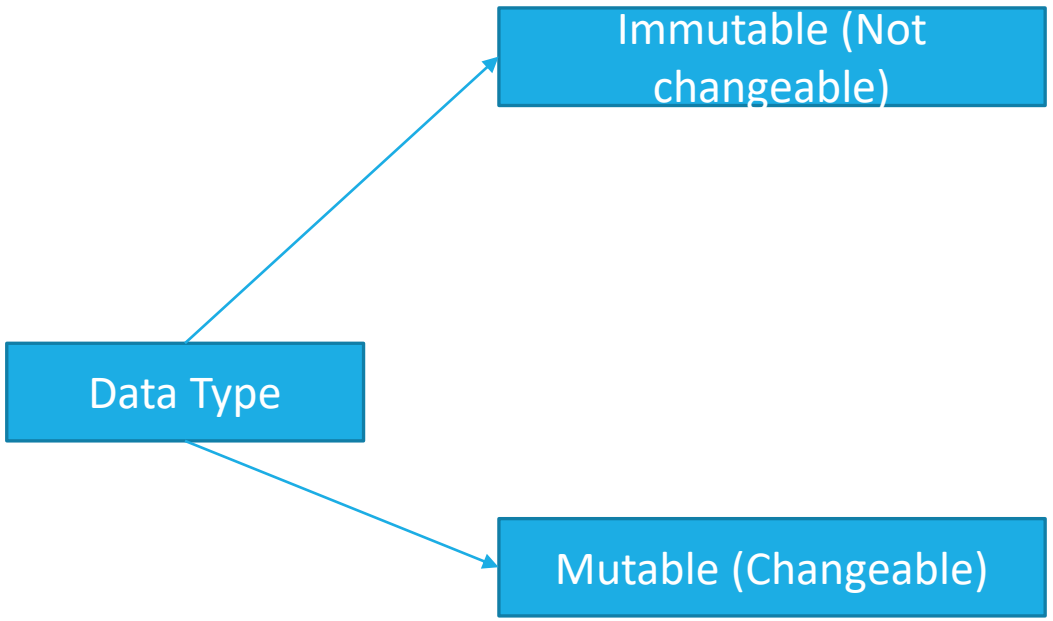
Advance Python

OOPs
Multithreading
File Handling
Exception Handling

Variable: variable is memory allocation which contain the data according to the data type.

a = 7
b = “Hi”

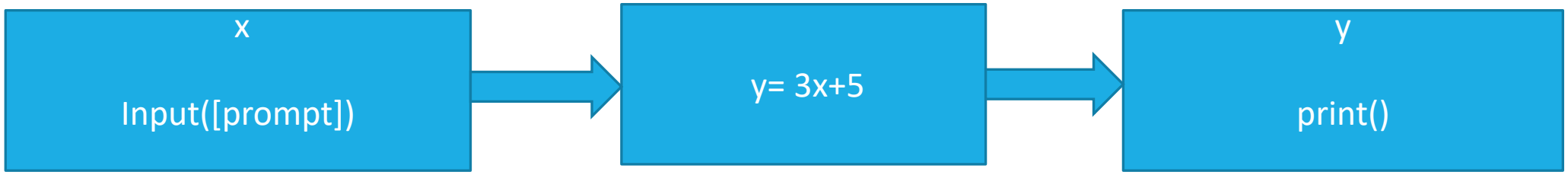
DataType: It’s defined the type of data.



Numeric: int, float, complex, None,
String: “------” ‘--------’
Tuple: (4,5,6,7,6,8,9.8,’Hi’)

List: [4,5,6,7.8,8,’K’]
Dictionary: {key:value}
Set: {4,5,6,7.9}

Input/Output function



Prompt by default taking as string

Keyword: is pre-defined word which used for specific work, can't be use as a identifier.

if, elif, else = conditional statement

Python:

```
>>> import keyword
>>> keyword.kwlist
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is',
'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']
>>> len(keyword.kwlist)
35
```

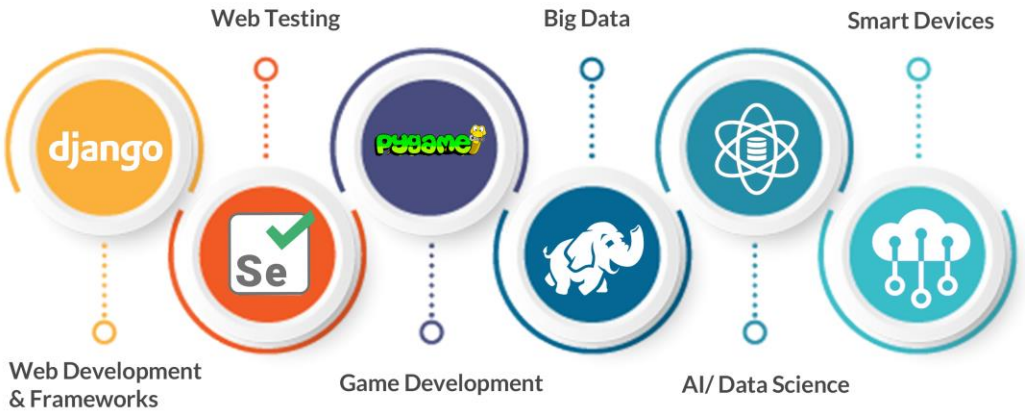
Identifier: variable name, function name, class name

Comment: single line comment: #

Multiline comment: ‘’’----------

---------’’’

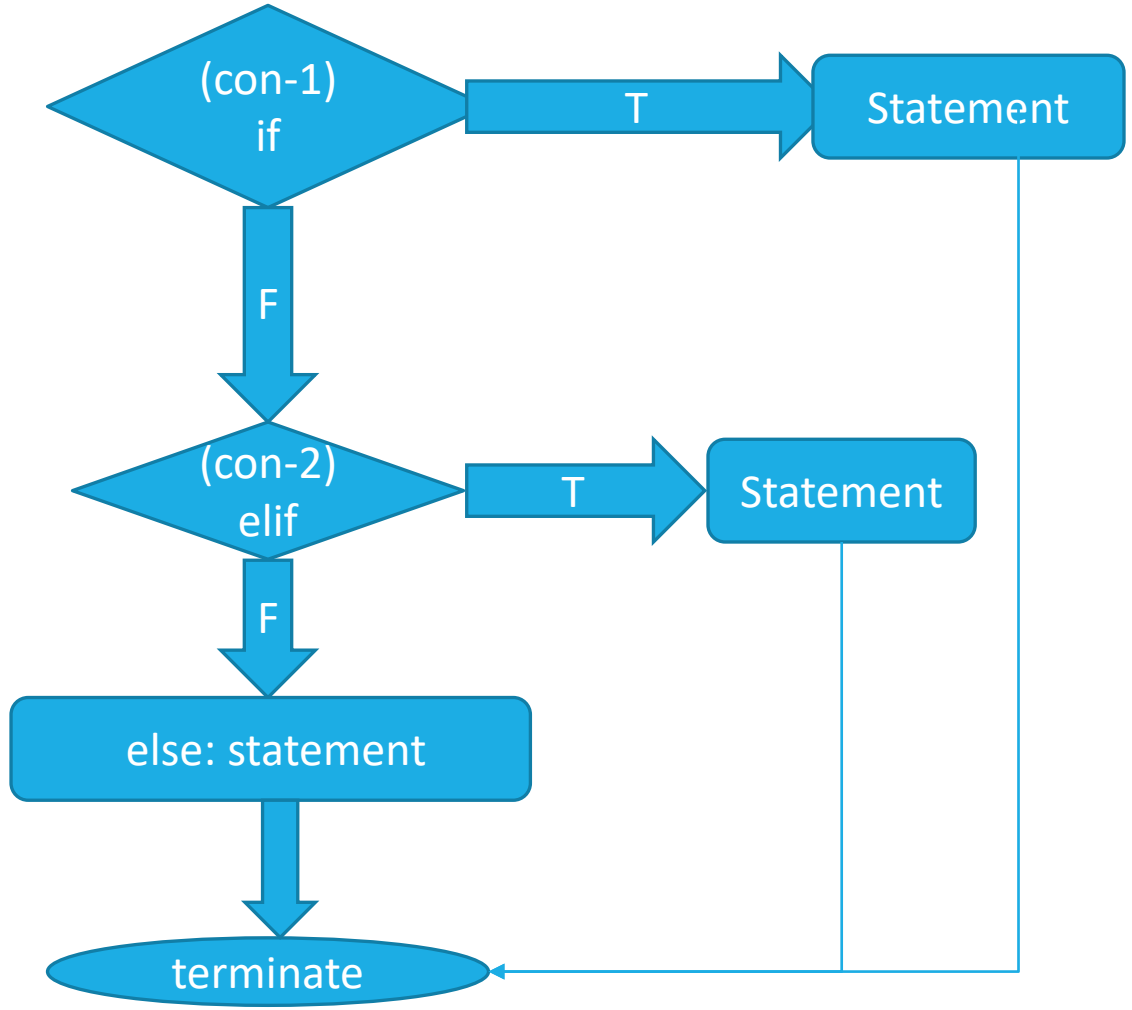

Making Machine leaning project after deploy on websites and connect with cloud
It means, I have to use multiple programming Languages.....
main hoon na


Web Testing
Big Data
Smart Devices
django
Se
pygame
Web Development & Frameworks
Game Development
AI/ Data Science

Conditional Statement & Loop

Conditional statement

if, elif: condition
else: statement



If is always primary condition

We are using if: when the block coming from True part

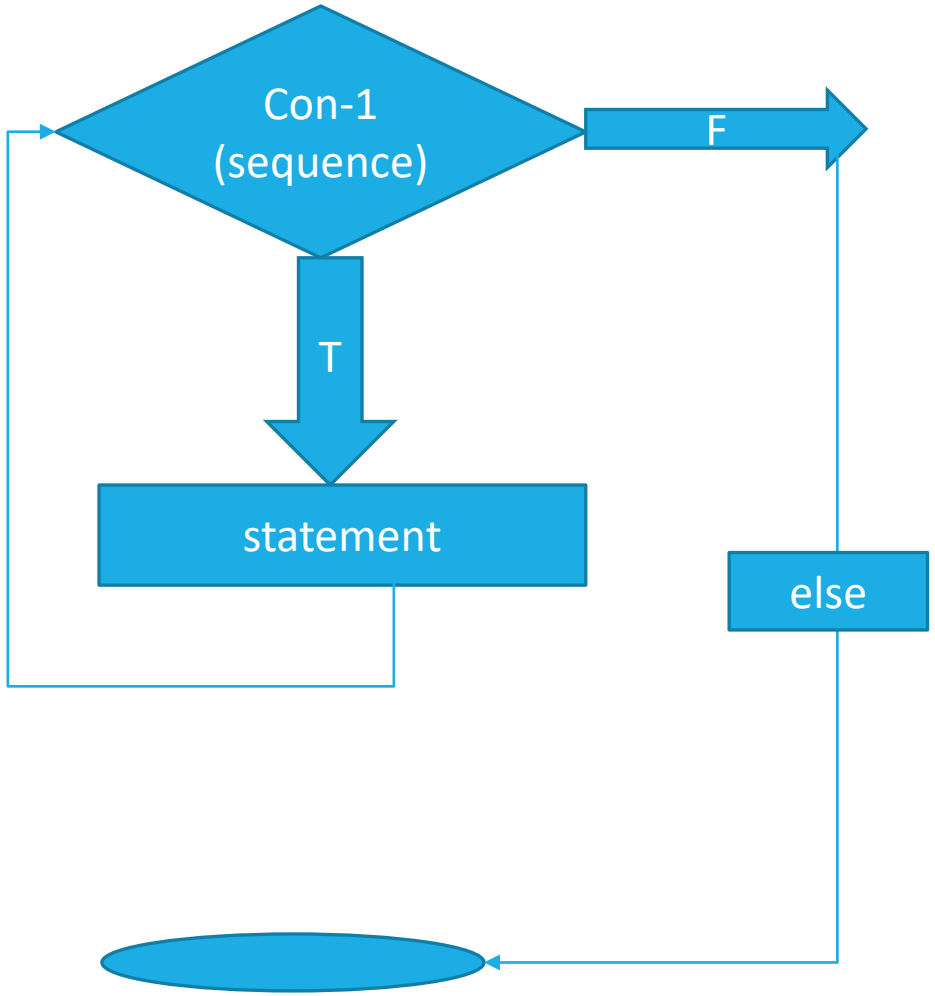
We are using elif: when the block coming from False part

Loop

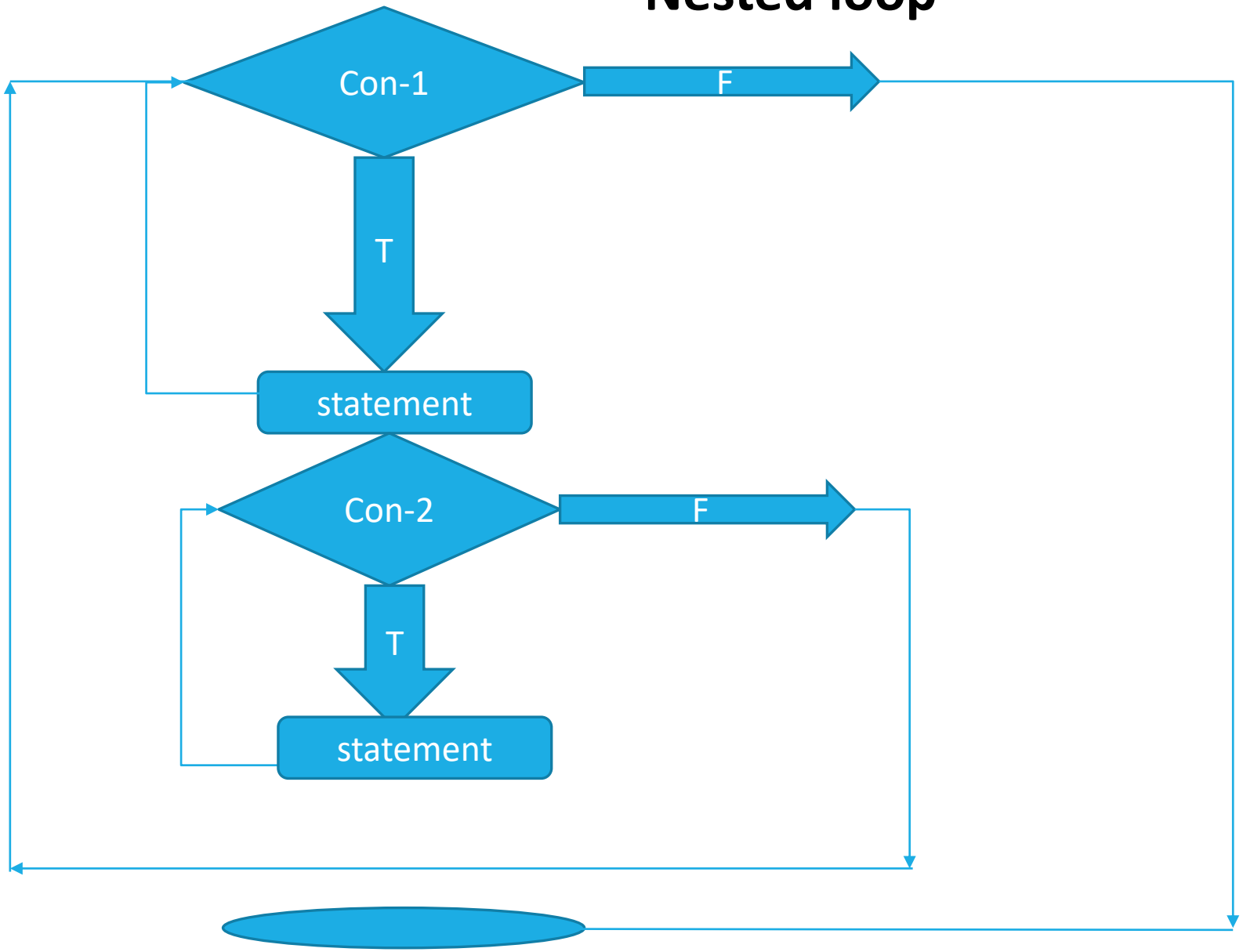
for, while

for var in seq:
#body

while condition:
#body



Nested loop



1

Hello

30

Hi

50

Hello

70

Hey

90

Hello

100

Data Types

1) Numeric data type

Real number

Complex number

Rational number

…….-0.6, -1,0,1.5,2.009877,…….

a+ib

a/b

Natural Number: 1,2,5,6
Whole number: 0,1,2,3……

Integer number:

…….-2,-1,0,1,2,3…….

In Python:

float: decimal number
int: Integer number
Complex: a+jb

2) String

s = “India” or ‘India’

Immutable and **ordered** data type

Positive Index	0	1	2	3	4
	I	n	d	i	a
Negative Index	-5	-4	-3	-2	-1

“””_____
_____”””

‘’’_____
_____’’’

Operation

+: concatenation
* : Repetition
s[3] : Slicing
 s[0:3]

Method	Function
s.upper()	len()
s.lower()	min()
s.capitalize()	max()
s.split()	str()
s.join()	type()
s.count()	
s.index() ; ; Many	

String formatting

%d = digit
%i = integer
 %f = float
%s = string
%c = character

```
name = "Charles"
marks = 97
subject = "Python"

print("Congratulation %s, you got %d percent in %s exam."%(name,marks,subject))
print("Congratulation {}, you got {} percent in {} exam.".format(name,marks,subject))
print("Congratulation Rahul, you got 67 percent in maths exam.")
```

3) Tuple

Immutable and ordered data type

t = (5,6,7,8.7,'k')

Operation

+: concatenation
*
t[3]
t[0:3]

Method	Function
.index()	len()
.count()	min()
	max()
	tuple(seq)
	type()

4) List

Mutable and ordered data type

l = [7,8,9,7,8,9.5,'hii']

Operation

+: concatenation

*

l[3]

l[0:3]

Method	Function
.append(element)	len()
.insert(index, element)	min()
.remove(element)	max()
.pop(index)	list(seq)
.copy()	type()
.sort()	
.reverse()	
.index()	
.count()	
.clear()	

Dictionary

Mutable and un-ordered data type

d = {“name”:”Rahul”,”class”:”12th”}

Method	Function
.keys()	len()
.value()	dict()
.items()	type()
.fromkeys()	
.copy()	
.pop()	
.clear()	
.update()	

Set

Mutable and unordered data types

s = {5,5,3,4,6,7,8.5,4}

Always taking unique values

Method	Function
.union()	len()
.intersection()	min()
.add()	max()
.remove()	set()
.clear()	type()

In python container: No any similar data type

	Numeric	String	Tuple	List	Dictionary	set
	int(), float(), complex()	str()	tuple()	list()	dict()	set()
	5, 5.3, 5+2j	“----” or ‘----’	()	[]	{key:value}	{2}
	Im	Im	Im	M	M	M
	----------	O	O	O	UnO	UnO
add	--------	No	No	append(), insert()	var[“key”] = value	.add()
delete	----------	No	No	.remove(), pop()	.pop()	.remove()

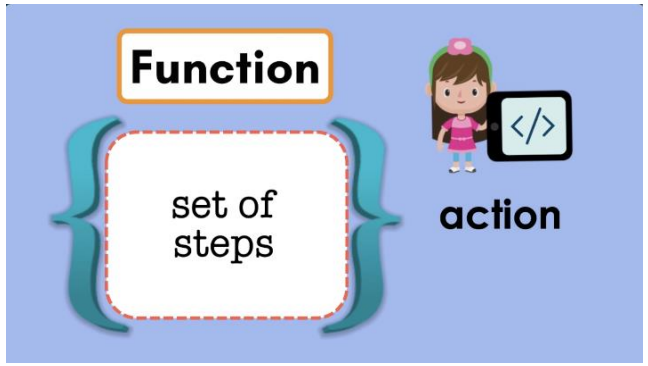
https://github.com/ritesh-maury/Data-Types

Function

A function is a block of organized, reusable code that is used to perform a single, related action.

```
def function_name(par1, par2, par3):
    #Body


function_name(arg1, arg2, arg3)
```



Return statement

Types of function:

1. **Pre-defined function**: min(), max(), int(), print(), input()……. Many
2. **User defined function:** add(), sub()……
3. **Anonymous function:** A function without name which defined with lambda keyword
4. **Recursion function:**

Package & Module

Module

Python files with a .py extension.

A Python module can have a set of functions, classes or variables defined and implemented.

Package

A package is basically a directory with Python files and a file with the name __init__ . py.

Python package can have sub-packages and modules.