Python:- created by Guido van Rossum

It’s a general purpose interpreted, interactive, object oriented and high level programming language. It has few keywords, simple structure and a clearly defined syntax which allows the student to pick up the language quickly

* High level programming language
* General purpose
* Interpreted
* Interactive
* Object oriented
* Frequently uses English keywords instead of punctuation
* Fewer syntactical constructions
* Beginner language

Features:-

* Easy to learn:- (few keywords, simple structure, clear defined syntax)
* Easy to read
* Easy to maintain
* A broad standard library
* Interactive mode
* Portable
* Extendable
* Databases
* GUI Programming
* Scalable
* Can be used as scripting language or can be compiled to byte code
* Provides high level dynamic data types and supports dynamic type checking
* Automatic garbage collection
* Easily integrated with C, C++, COM, CORBA, ActiveX, Java.

Python Identifiers:-

* To identify a variable
* Starts with A to Z or a to z or (\_) followed by digits and alphabets
* Does not allow punctuation characters (@, $, %)
* Case sensitive
* Class start with an uppercase letter and other start with lowercase letter.
* Starting with \_ means its private
* Starting with \_\_ indicates strongly private
* Starts and ends with \_\_ indicates identifier is language defined special name

Lines & Identation:-

* Does not provide braces for block indication
* Line indentation, rigidly enforced
* Allows the use of line continuation character (\) for continuation
* Statements within [], {}, () do not need to use (\)

Quotation:-

* Accepts (‘), (“ “), (‘’’ ‘’’ or “”” “””) quotes
* Word = ‘word’
* Sentence = “sentence”
* Paragraph = “”” This is paragraph”””

Multi Line Assignment:- semicolon (;) allows multiple statements on the single line. It allows multiple assignment.

Header lines begins with keyword & terminate with colon ( : )

Standard data types:

1) Numbers

2) String

3) List

4) Tuple

5) Dictionary

It supports 4 numerical types:-

* Int
* Long (long, octal, hexadecimal)
* Float
* complex

# Day 2:-

Python Data Types:-

1. Python Numbers:-

It supports 4 numerical types:-

* Int (50)
* Long (long, octal, hexadecimal) (518245262L)
* Float (125.0)
* Complex (3.14j)

2. Strings:-

* Continuous set of characters represented in quotation marks
* Subset can be taken using slice operator ([], [:])

e.g. welcomeStr = “Welcome to world of Python”

3. List:-

* Most versatile pyhon compound data type
* Similar to arrays in ruby
* All elements can be of different data type
* Can be accesses using slice operator
* Enclosed with square brackets

list = [‘abcd’, 786, 2.23, ‘john’, 70.2]

print list[1:3] # print elements starting fron 2nd to 3rd

Ans: 786, 2.23

4. Tuples:-

* Similar to list
* Enclosed within parentheses ( () ) unlike list ( [])
* Cannot be updated
* Read only lists

tuple = (‘abcd’, 786, 2.23, ‘john’, 70.2)

list = [‘abcd’, 786, 2.23, ‘john’, 70.2]

tuple[2] = 1000 # invalid with tuple, updating is not allowed within tuple

list[2] = 1000 # it will work fine, modification is allowed in list

5. Dictionary:-

* Hash table type
* Enclosed by curly braces
* Represented by keys and value pairs

dict = {‘name’: ‘Anjali’, ‘code’: 1309, ‘dept’: ‘search in progress’}

dict.keys() & dict.values()

dictionary keys is ['dept', 'code', 'name'] dictionary values is ['search in progress', 1309, 'Anjali']

Data Type Conversion:-

* Performs conversions between the built in types.
* Use the type name as functions

e.g. int(x[,base]), float(x), str(x), eval(str), tuple(s), long(x[,base]), complex(real[,imag]), repr(x), list(s), set(s), dict(d), frozenset(s), chr(x), unichr(x), ord(x), hex(x), oct(x)