Introductory Course on Python in Physics

Surajit Sen[†]

[†] Department of Physics
Guru Charan College, Silchar 788004, India
[†] Centre of Advanced Studies & Innovation Lab
18/27 Kali Mohan Road, Tarapur, Silchar 788003, India

December 5, 2019

Overview

Overview

- Introduction (Basic Hello Commands)

Overview

- Introduction (Basic Hello Commands)
- Mathematical (Algebraic, Boolean, Comparison) Operation

Overview

- Introduction (Basic Hello Commands)
- Mathematical (Algebraic, Boolean, Comparison) Operation
- Assignment and Augmented Assignment

Overview

- Introduction (Basic Hello Commands)
- Mathematical (Algebraic, Boolean, Comparison) Operation
- Assignment and Augmented Assignment
- Function, Class, Object, Module etc

Overview

- Introduction (Basic Hello Commands)
- Mathematical (Algebraic, Boolean, Comparison) Operation
- Assignment and Augmented Assignment
- Function, Class, Object, Module etc
- Data Structure or Container

Overview

- Introduction (Basic Hello Commands)
- Mathematical (Algebraic, Boolean, Comparison) Operation
- Assignment and Augmented Assignment
- Function, Class, Object, Module etc
- Data Structure or Container
- Flow control in Programming

Overview

- Introduction (Basic Hello Commands)
- Mathematical (Algebraic, Boolean, Comparison) Operation
- Assignment and Augmented Assignment
- Function, Class, Object, Module etc
- Data Structure or Container
- Flow control in Programming
- Python Programming- Some Simple Projects

Overview

- Introduction (Basic Hello Commands)
- Mathematical (Algebraic, Boolean, Comparison) Operation
- Assignment and Augmented Assignment
- Function, Class, Object, Module etc
- Data Structure or Container
- Flow control in Programming
- Python Programming- Some Simple Projects
- Hands-on Training of some Python packages:

Overview

- Introduction (Basic Hello Commands)
- Mathematical (Algebraic, Boolean, Comparison) Operation
- Assignment and Augmented Assignment
- Function, Class, Object, Module etc
- Data Structure or Container
- Flow control in Programming
- Python Programming- Some Simple Projects
- Hands-on Training of some Python packages:
  - NUMPY - Demonstration with Jupyter Worksheet

Overview

- Introduction (Basic Hello Commands)
- Mathematical (Algebraic, Boolean, Comparison) Operation
- Assignment and Augmented Assignment
- Function, Class, Object, Module etc
- Data Structure or Container
- Flow control in Programming
- Python Programming- Some Simple Projects
- Hands-on Training of some Python packages:
  - NUMPY - Demonstration with Jupyter Worksheet
  - SYMPY - Demonstration with Jupyter Worksheet

Overview

- Introduction (Basic Hello Commands)
- Mathematical (Algebraic, Boolean, Comparison) Operation
- Assignment and Augmented Assignment
- Function, Class, Object, Module etc
- Data Structure or Container
- Flow control in Programming
- Python Programming- Some Simple Projects
- Hands-on Training of some Python packages:
  - NUMPY - Demonstration with Jupyter Worksheet
  - SYMPY - Demonstration with Jupyter Worksheet
  - MATPLOTLIB - Demonstration with Jupyter Worksheet

Overview

- Introduction (Basic Hello Commands)
- Mathematical (Algebraic, Boolean, Comparison) Operation
- Assignment and Augmented Assignment
- Function, Class, Object, Module etc
- Data Structure or Container
- Flow control in Programming
- Python Programming- Some Simple Projects
- Hands-on Training of some Python packages:
  - NUMPY - Demonstration with Jupyter Worksheet
  - SYMPY - Demonstration with Jupyter Worksheet
  - MATPLOTLIB - Demonstration with Jupyter Worksheet
- More Application of Python:

Overview

- Introduction (Basic Hello Commands)
- Mathematical (Algebraic, Boolean, Comparison) Operation
- Assignment and Augmented Assignment
- Function, Class, Object, Module etc
- Data Structure or Container
- Flow control in Programming
- Python Programming- Some Simple Projects
- Hands-on Training of some Python packages:
  - NUMPY - Demonstration with Jupyter Worksheet
  - SYMPY - Demonstration with Jupyter Worksheet
  - MATPLOTLIB - Demonstration with Jupyter Worksheet
- More Application of Python:
  - Numerical Solution of ODE and PDE (To be added)

Overview

- Introduction (Basic Hello Commands)
- Mathematical (Algebraic, Boolean, Comparison) Operation
- Assignment and Augmented Assignment
- Function, Class, Object, Module etc
- Data Structure or Container
- Flow control in Programming
- Python Programming- Some Simple Projects
- Hands-on Training of some Python packages:
  - NUMPY - Demonstration with Jupyter Worksheet
  - SYMPY - Demonstration with Jupyter Worksheet
  - MATPLOTLIB - Demonstration with Jupyter Worksheet
- More Application of Python:
  - Numerical Solution of ODE and PDE (To be added)
  - Simulation Based Studies (To be added)

Overview

- Introduction (Basic Hello Commands)
- Mathematical (Algebraic, Boolean, Comparison) Operation
- Assignment and Augmented Assignment
- Function, Class, Object, Module etc
- Data Structure or Container
- Flow control in Programming
- Python Programming- Some Simple Projects
- Hands-on Training of some Python packages:
  - NUMPY - Demonstration with Jupyter Worksheet
  - SYMPY - Demonstration with Jupyter Worksheet
  - MATPLOTLIB - Demonstration with Jupyter Worksheet
- More Application of Python:
  - Numerical Solution of ODE and PDE (To be added)
  - Simulation Based Studies (To be added)
  - IBM-Q - Python in Quantum Information Science (Advanced Course to be added)

Overview

- Introduction (Basic Hello Commands)
- Mathematical (Algebraic, Boolean, Comparison) Operation
- Assignment and Augmented Assignment
- Function, Class, Object, Module etc
- Data Structure or Container
- Flow control in Programming
- Python Programming- Some Simple Projects
- Hands-on Training of some Python packages:
  - NUMPY - Demonstration with Jupyter Worksheet
  - SYMPY - Demonstration with Jupyter Worksheet
  - MATPLOTLIB - Demonstration with Jupyter Worksheet
- More Application of Python:
  - Numerical Solution of ODE and PDE (To be added)
  - Simulation Based Studies (To be added)
  - IBM-Q - Python in Quantum Information Science (Advanced Course to be added)
- Some Resources

Section I: Mathematical Operation

Basic Hello Commands:

Basic Hello Commands:

Algebraic Operation:

Basic Hello Commands:

Algebraic Operation:

- Expression (10±7, 10∗7, 10/7, 10 //7, 10%7, 10∗∗7)

Basic Hello Commands:

Algebraic Operation:

- Expression (10±7, 10∗7, 10/7, 10 //7, 10%7, 10∗∗7)
- Operator (+, -, ∗, /, //, %, ∗∗)

Basic Hello Commands:

Algebraic Operation:

- Expression (10±7, 10∗7, 10/7, 10 //7, 10%7, 10∗∗7)
- Operator (+, -, ∗, /, //, %, ∗∗)
- Operand (10, 7)

Basic Math Operations & Their Prioritization:

Basic Hello Commands:

Algebraic Operation:

- Expression (10±7, 10∗7, 10/7, 10 //7, 10%7, 10∗∗7)
- Operator (+, -, ∗, /, //, %, ∗∗)
- Operand (10, 7)

Basic Math Operations & Their Prioritization:

- Parenthesis ()

Basic Hello Commands:

Algebraic Operation:

- Expression (10±7, 10∗7, 10/7, 10 //7, 10%7, 10∗∗7)
- Operator (+, -, ∗, /, //, %, ∗∗)
- Operand (10, 7)

Basic Math Operations & Their Prioritization:

- Parenthesis ()
- Exponentiation ∗∗

Basic Hello Commands:

Algebraic Operation:

- Expression (10±7, 10∗7, 10/7, 10 //7, 10%7, 10∗∗7)
- Operator (+, -, ∗, /, //, %, ∗∗)
- Operand (10, 7)

Basic Math Operations & Their Prioritization:

- Parenthesis ()
- Exponentiation ∗∗
- Multiplication ∗

Basic Hello Commands:

Algebraic Operation:

- Expression (10±7, 10∗7, 10/7, 10 //7, 10%7, 10∗∗7)
- Operator (+, -, ∗, /, //, %, ∗∗)
- Operand (10, 7)

Basic Math Operations & Their Prioritization:

- Parenthesis ()
- Exponentiation ∗∗
- Multiplication ∗
- Floor Division / & Division //

Basic Hello Commands:

Algebraic Operation:

- Expression (10±7, 10∗7, 10/7, 10 //7, 10%7, 10∗∗7)
- Operator (+, -, ∗, /, //, %, ∗∗)
- Operand (10, 7)

Basic Math Operations & Their Prioritization:

- Parenthesis ()
- Exponentiation ∗∗
- Multiplication ∗
- Floor Division / & Division //
- Modulo %

Basic Hello Commands:

Algebraic Operation:

- Expression (10±7, 10∗7, 10/7, 10 //7, 10%7, 10∗∗7)
- Operator (+, -, ∗, /, //, %, ∗∗)
- Operand (10, 7)

Basic Math Operations & Their Prioritization:

- Parenthesis ()
- Exponentiation ∗∗
- Multiplication ∗
- Floor Division / & Division //
- Modulo %
- Addition, Substraction +, −

Boolean Operation:

Boolean Operation:

- True

Boolean Operation:

- True
- False

Boolean Operation:

- True
- False
- Logical And

Boolean Operation:

- True
- False
- Logical And
- Logical Or

Boolean Operation:

- True
- False
- Logical And
- Logical Or
- Logical Not

Comparison Operation:

Boolean Operation:

- True
- False
- Logical And
- Logical Or
- Logical Not

Comparison Operation:

- Equal (=)

Boolean Operation:

- True
- False
- Logical And
- Logical Or
- Logical Not

Comparison Operation:

- Equal (=)
- Not-equal (≠)

Boolean Operation:

- True
- False
- Logical And
- Logical Or
- Logical Not

Comparison Operation:

- Equal (=)
- Not-equal (≠)
- Greater Than (>)

Boolean Operation:

- True
- False
- Logical And
- Logical Or
- Logical Not

Comparison Operation:

- Equal (=)
- Not-equal ($\neq$)
- Greater Than (>)
- Less Than (<)

Boolean Operation:

- True
- False
- Logical And
- Logical Or
- Logical Not

Comparison Operation:

- Equal $(=)$
- Not-equal $(\neq)$
- Greater Than $(>)$
- Less Than $(<)$
- Greater Than Equal To $(\geq)$

Boolean Operation:

- True
- False
- Logical And
- Logical Or
- Logical Not

Comparison Operation:

- Equal $(=)$
- Not-equal $(\neq)$
- Greater Than $(>)$
- Less Than $(<)$
- Greater Than Equal To $(\geq)$
- Less Than Equal To $(\leq)$

Section III: Assignment

Variable Assignment:

Variable Assignment:

- Integer

Variable Assignment:

- Integer
- Float

Variable Assignment:

- Integer
- Float
- String

Variable Assignment:

- Integer
- Float
- String
- Type

Variable Assignment:

- Integer
- Float
- String
- Type
- Class and Order

Variable Assignment:

- Integer
- Float
- String
- Type
- Class and Order

Augmented Assignment:

Variable Assignment:

- Integer
- Float
- String
- Type
- Class and Order

Augmented Assignment:

- $+=$ Assignment (For example, if $x = 4$ with $x+=1$, then x refers 5)

Variable Assignment:

- Integer
- Float
- String
- Type
- Class and Order

Augmented Assignment:

- $+=$ Assignment (For example, if $x = 4$ with $x+=1$, then x refers 5)
- $-=$ Assignment (For example, if $x = 4$ with $x-=1$, then x refers 3)

Variable Assignment:

- Integer
- Float
- String
- Type
- Class and Order

Augmented Assignment:

- $+=$ Assignment (For example, if $x = 4$ with $x+=1$, then x refers 5)
- $-=$ Assignment (For example, if $x = 4$ with $x-=1$, then x refers 3)
- $*=$ Assignment (For example, if $x = 4$ with $x*=2$, then x refers 8)

Variable Assignment:

- Integer
- Float
- String
- Type
- Class and Order

Augmented Assignment:

- $+=$ Assignment (For example, if $x = 4$ with $x+=1$, then x refers 5)
- $-=$ Assignment (For example, if $x = 4$ with $x-=1$, then x refers 3)
- $*=$ Assignment (For example, if $x = 4$ with $x*=2$, then x refers 8)
- $/=$ Assignment (For example, if $x = 4$ with $x/=2$, then x refers 2)

Section IV: Data Structure

Container or Collections:

Container or Collections:

- List (mylist =[5, 7, ‘x’, True, ‘dog’, ‘sd56’])

Container or Collections:

- List (mylist =[5, 7, ‘x’, True, ‘dog’, ‘sd56’])
- Tuple (mytuple=(5, 7, ‘x’, True, ‘dog’, ‘sd56’))

Container or Collections:

- List (mylist =[5, 7, ‘x’, True, ‘dog’, ‘sd56’])
- Tuple (mytuple=(5, 7, ‘x’, True, ‘dog’, ‘sd56’))
- Set e.g., (myset={5, 7, ‘x’, True, ‘dog’, ‘sd56’})

Container or Collections:

- List (mylist =[5, 7, ‘x’, True, ‘dog’, ‘sd56’])
- Tuple (mytuple=(5, 7, ‘x’, True, ‘dog’, ‘sd56’))
- Set e.g., (myset={5, 7, ‘x’, True, ‘dog’, ‘sd56’})
- Dictionary e.g., (mydict = {’stg’: True, ‘dog’, ‘x’, intg: [5, 7, 10]})

Container or Collections:

- List (mylist =[5, 7, ‘x’, True, ‘dog’, ‘sd56’])
- Tuple (mytuple=(5, 7, ‘x’, True, ‘dog’, ‘sd56’))
- Set e.g., (myset={5, 7, ‘x’, True, ‘dog’, ‘sd56’})
- Dictionary e.g., (mydict = {’stg’: True, ‘dog’, ‘x’, intg: [5, 7, 10]})

Others (Function, Class, Object, Module etc):

Container or Collections:

- List (mylist =[5, 7, ‘x’, True, ‘dog’, ‘sd56’])
- Tuple (mytuple=(5, 7, ‘x’, True, ‘dog’, ‘sd56’))
- Set e.g., (myset={5, 7, ‘x’, True, ‘dog’, ‘sd56’})
- Dictionary e.g., (mydict = {’stg’: True, ‘dog’, ‘x’, intg: [5, 7, 10]})

Others (Function, Class, Object, Module etc):

- Builtin & User Defined Function

Container or Collections:

- List (mylist =[5, 7, ‘x’, True, ‘dog’, ‘sd56’])
- Tuple (mytuple=(5, 7, ‘x’, True, ‘dog’, ‘sd56’))
- Set e.g., (myset={5, 7, ‘x’, True, ‘dog’, ‘sd56’})
- Dictionary e.g., (mydict = {’stg’: True, ‘dog’, ‘x’, intg: [5, 7, 10]})

Others (Function, Class, Object, Module etc):

- Builtin & User Defined Function
- Class

Container or Collections:

- List (mylist =[5, 7, ‘x’, True, ‘dog’, ‘sd56’])
- Tuple (mytuple=(5, 7, ‘x’, True, ‘dog’, ‘sd56’))
- Set e.g., (myset={5, 7, ‘x’, True, ‘dog’, ‘sd56’})
- Dictionary e.g., (mydict = {’stg’: True, ‘dog’, ‘x’, intg: [5, 7, 10]})

Others (Function, Class, Object, Module etc):

- Builtin & User Defined Function
- Class
- Object

Container or Collections:

- List (mylist =[5, 7, ‘x’, True, ‘dog’, ‘sd56’])
- Tuple (mytuple=(5, 7, ‘x’, True, ‘dog’, ‘sd56’))
- Set e.g., (myset={5, 7, ‘x’, True, ‘dog’, ‘sd56’})
- Dictionary e.g., (mydict = {’stg’: True, ‘dog’, ‘x’, intg: [5, 7, 10]})

Others (Function, Class, Object, Module etc):

- Builtin & User Defined Function
- Class
- Object
- Import Module

Section V: Flow Control in Programming

Loop Statement:

Loop Statement:

- For

Loop Statement:

- For
- Nested For

Loop Statement:

- For
- Nested For
- While

Loop Statement:

- For
- Nested For
- While
- Nested While

Loop Statement:

- For
- Nested For
- While
- Nested While
- Continue

Loop Statement:

- For
- Nested For
- While
- Nested While
- Continue
- Pass

Loop Statement:

- For
- Nested For
- While
- Nested While
- Continue
- Pass
- Break

Loop Statement:

- For
- Nested For
- While
- Nested While
- Continue
- Pass
- Break

Conditional Statement:

Loop Statement:

- For
- Nested For
- While
- Nested While
- Continue
- Pass
- Break

Conditional Statement:

- If

Loop Statement:

- For
- Nested For
- While
- Nested While
- Continue
- Pass
- Break

Conditional Statement:

- If
- Else

Loop Statement:

- For
- Nested For
- While
- Nested While
- Continue
- Pass
- Break

Conditional Statement:

- If
- Else
- Elseif

Loop Statement:

- For
- Nested For
- While
- Nested While
- Continue
- Pass
- Break

Conditional Statement:

- If
- Else
- Elseif
- Nested If

Section VI: Some Advanced Features

File Handling in Python:

File Handling in Python:

- Create, Open, Append, Read, Write

File Handling in Python:

- Create, Open, Append, Read, Write
- Simple examples

File Handling in Python:

- Create, Open, Append, Read, Write
- Simple examples
- Concept of Lambda function

File Handling in Python:

- Create, Open, Append, Read, Write
- Simple examples
- Concept of Lambda function

Object Oriented Programming (OOP) in Python:

File Handling in Python:

- Create, Open, Append, Read, Write
- Simple examples
- Concept of Lambda function

Object Oriented Programming (OOP) in Python:

- Abstraction, Encapsulation, Polymorphism, Inheritance

File Handling in Python:

- Create, Open, Append, Read, Write
- Simple examples
- Concept of Lambda function

Object Oriented Programming (OOP) in Python:

- Abstraction, Encapsulation, Polymorphism, Inheritance
- Class, Object & their application

File Handling in Python:

- Create, Open, Append, Read, Write
- Simple examples
- Concept of Lambda function

Object Oriented Programming (OOP) in Python:

- Abstraction, Encapsulation, Polymorphism, Inheritance
- Class, Object & their application

Classification of Control Statements:

File Handling in Python:

- Create, Open, Append, Read, Write
- Simple examples
- Concept of Lambda function

Object Oriented Programming (OOP) in Python:

- Abstraction, Encapsulation, Polymorphism, Inheritance
- Class, Object & their application

Classification of Control Statements:

- Sequential Control

File Handling in Python:

- Create, Open, Append, Read, Write
- Simple examples
- Concept of Lambda function

Object Oriented Programming (OOP) in Python:

- Abstraction, Encapsulation, Polymorphism, Inheritance
- Class, Object & their application

Classification of Control Statements:

- Sequential Control
- Selection Control

File Handling in Python:

- Create, Open, Append, Read, Write
- Simple examples
- Concept of Lambda function

Object Oriented Programming (OOP) in Python:

- Abstraction, Encapsulation, Polymorphism, Inheritance
- Class, Object & their application

Classification of Control Statements:

- Sequential Control
- Selection Control
- Iterative Control

Section VII: Hands-on Training

Link to GITHUB Jupyter Worksheet:

Link to GITHUB Jupyter Worksheet:

- **Basic Python Codes**

Link to GITHUB Jupyter Worksheet:

- **Basic Python Codes**
  - Algebraic Operation

Link to GITHUB Jupyter Worksheet:

- **Basic Python Codes**
  - Algebraic Operation
  - Logical Operation

Link to GITHUB Jupyter Worksheet:

- **Basic Python Codes**
  - Algebraic Operation
  - Logical Operation
  - Assignment

Link to GITHUB Jupyter Worksheet:

- **Basic Python Codes**
  - Algebraic Operation
  - Logical Operation
  - Assignment
  - Data Structure (Container)

Link to GITHUB Jupyter Worksheet:

- **Basic Python Codes**
  - Algebraic Operation
  - Logical Operation
  - Assignment
  - Data Structure (Container)
  - Others (Function, Class, Object, Module etc)

Link to GITHUB Jupyter Worksheet:

- **Basic Python Codes**
  - Algebraic Operation
  - Logical Operation
  - Assignment
  - Data Structure (Container)
  - Others (Function, Class, Object, Module etc)
  - If, Else, Elseif -Conditional Statement

Link to GITHUB Jupyter Worksheet:

- **Basic Python Codes**
  - Algebraic Operation
  - Logical Operation
  - Assignment
  - Data Structure (Container)
  - Others (Function, Class, Object, Module etc)
  - If, Else, Elseif -Conditional Statement
  - For Loop

Link to GITHUB Jupyter Worksheet:

- **Basic Python Codes**
  - Algebraic Operation
  - Logical Operation
  - Assignment
  - Data Structure (Container)
  - Others (Function, Class, Object, Module etc)
  - If, Else, Elseif -Conditional Statement
  - For Loop
  - While Loop

Link to GITHUB Jupyter Worksheet:

- **Basic Python Codes**
  - Algebraic Operation
  - Logical Operation
  - Assignment
  - Data Structure (Container)
  - Others (Function, Class, Object, Module etc)
  - If, Else, Elseif -Conditional Statement
  - For Loop
  - While Loop
  - Some useful Advanced Commands

Section VII: Hands-on Training (Contd.)

Link to GITHUB Jupyter Worksheet:

Link to GITHUB Jupyter Worksheet:

- **Some Python Packages:**

Link to GITHUB Jupyter Worksheet:

- **Some Python Packages:**
  - NUMPY - Package for Numerical Computation

Link to GITHUB Jupyter Worksheet:

- **Some Python Packages:**
  - NUMPY - Package for Numerical Computation
  - SYMPY (Algebra) - Package for Symbolic Computation

Link to GITHUB Jupyter Worksheet:

- **Some Python Packages:**
  - NUMPY - Package for Numerical Computation
  - SYMPY (Algebra) - Package for Symbolic Computation
  - SYMPY (Matrix) - Package for Symbolic Computation

Link to GITHUB Jupyter Worksheet:

- **Some Python Packages:**
  - NUMPY - Package for Numerical Computation
  - SYMPY (Algebra) - Package for Symbolic Computation
  - SYMPY (Matrix) - Package for Symbolic Computation
  - MATPLOTLIB - Package for Graphical Solution

Link to GITHUB Jupyter Worksheet:

- **Some Python Packages:**
  - NUMPY - Package for Numerical Computation
  - SYMPY (Algebra) - Package for Symbolic Computation
  - SYMPY (Matrix) - Package for Symbolic Computation
  - MATPLOTLIB - Package for Graphical Solution
  - Numerical Analysis using Python (To be added)

Link to GITHUB Jupyter Worksheet:

- **Some Python Packages:**
  - NUMPY - Package for Numerical Computation
  - SYMPY (Algebra) - Package for Symbolic Computation
  - SYMPY (Matrix) - Package for Symbolic Computation
  - MATPLOTLIB - Package for Graphical Solution
  - Numerical Analysis using Python (To be added)
  - Numerical Simulation using Python (To be added)

Link to GITHUB Jupyter Worksheet:

- **Some Python Packages:**
  - NUMPY - Package for Numerical Computation
  - SYMPY (Algebra) - Package for Symbolic Computation
  - SYMPY (Matrix) - Package for Symbolic Computation
  - MATPLOTLIB - Package for Graphical Solution
  - Numerical Analysis using Python (To be added)
  - Numerical Simulation using Python (To be added)
  - IBM-Q - Application of Python Programming in Quantum Information Science (To be added)

References

References:

References:

- 7 Reasons To Learn Python In 2019: Medium Magazine

References:

- 7 Reasons To Learn Python In 2019: Medium Magazine
- Practical Programming - Introduction to Computer Science using Python, P Gries, J Campbell, J Montojo (2017) North Caroline

References:

- 7 Reasons To Learn Python In 2019: Medium Magazine
- Practical Programming - Introduction to Computer Science using Python, P Gries, J Campbell, J Montojo (2017) North Caroline
- A web based tutorial on Python

References:

- 7 Reasons To Learn Python In 2019: Medium Magazine
- Practical Programming - Introduction to Computer Science using Python, P Gries, J Campbell, J Montojo (2017) North Caroline
- A web based tutorial on Python
- See, for example, Yet Another Online Course on Python

References:

- 7 Reasons To Learn Python In 2019: Medium Magazine
- Practical Programming - Introduction to Computer Science using Python, P Gries, J Campbell, J Montojo (2017) North Caroline
- A web based tutorial on Python
- See, for example, Yet Another Online Course on Python
- Search 'Google' for free content on Python Programming 'Full Online Course on Python'

Take Home Message

Take Home Message

There are lots of beautiful 'Pythons' around us!

So stay curious and use them in your Project!

Thanks for patient listening
