

# PYTHON

---

BASIC OF PYTHON PROGRAMMING LANGUAGE

LECTURE – 02

# Contents

---

- Introduction to Python
- Environment Setup
- Syntax of Python
  - Identifiers
  - Reserve Keywords
  - Lines & Indentation
  - Multi-line Statements
  - Comments in Python
  - Variables
- Creating basic programs

# What is Python?

---

“Python is an interpreted, **object-oriented, high-level Programming Language** with dynamic semantics.” It was created by *Guido van Rossum*, and released in 1991 (Current Version of python is: 3).

- What is **object-oriented**?
- What does **high-level** programming language mean?
- Used for: *web development, software development, mathematics, game development, research*

# Why Python?

---

- It's almost **close to HUMAN** language.
- Python is **simple and so easy** to learn.
- Python is **Open Source** which means its available free of cost.
- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python is much in demand and ensures **high salary** in tech related JOB.
- Python has **powerful development libraries** include **AI, ML** etc.

```
>>>
>>> print("Hello World")
Hello World
>>>
>>> 5+6
11
>>>
>>> 2*10
20
>>>
>>> 15/6
2.5
>>>
```

# Python Job Markets

---

Python is very high in demand and all the major international companies nowadays.

Today a Python Programmer with 3-5 years of experience is asking for around **\$150,000** annual package.

- Google
- Intel
- NASA
- PayPal
- Facebook
- IBM
- Amazon
- Netflix
- Pinterest
- Uber and many more.

# Careers in Python

---

- Game developer
- Web designer
- Python developer
- Full-stack developer
- Machine learning engineer
- Data scientist
- Data analyst
- Data engineer
- DevOps engineer
- Software engineer

# Characteristics of Python

---

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to bytecode for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection. It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

# Our Goal

---

- Basic of Python
- Creating simple program
- Introducing 2/3 modules/library (*Pandas, NumPy, MatSciPy*)
- Implementing simple *AI/ML* based Application



# Python Environment Setup

---

- ☐ Download and install latest version of python  
(<https://www.python.org/downloads/windows/>)
- ☐ IDE (Integrated Development Environment)
  - ☐ **PyCharm** (Community version): <https://www.jetbrains.com/pycharm/>

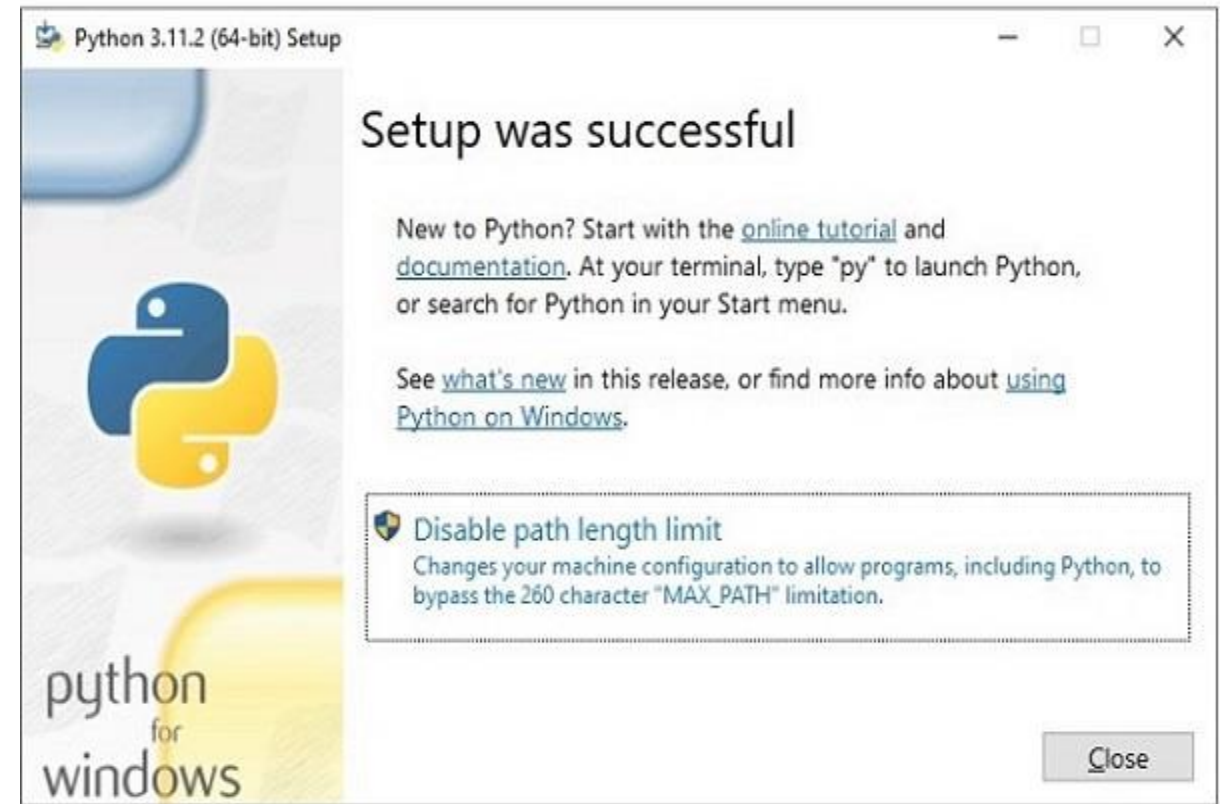
# Python Environment Setup

- Download & Install Python:  
<https://www.python.org/downloads/windows/>
- Double click on the file where it has been downloaded to start the installation.
- Click the Install Now button, it is advised to choose the installation folder with a relatively shorter path, and tick the second check box to update the PATH variable.



# Python Environment Setup

- Download & Install Python:  
<https://www.python.org/downloads/windows/>
- Double click on the file where it has been downloaded to start the installation.
- Click the Install Now button, it is advised to choose the installation folder with a relatively shorter path, and tick the second check box to update the PATH variable.
- Wait a moment it shows Successful message.



# Python Environment Setup

---

Open the Window Command Prompt terminal and run **Python** to check the success of installation.

Open the **Run menu** with **Windows Key + R**, then type "**cmd.**" Press "**Enter**" to open the regular Command Prompt.

or **Ctrl + Shift + Enter** to open as an Administrator.

```
C:\Users\Acer>python
Python 3.11.2 (tags/v3.11.2:878ead1, Feb 7 2023, 16:38:35)
64 bit (AMD64) on win32
Type "help", "copyright", "credits" or "license" for more i
>>>
```

# Python Environment Setup (path setting)

---

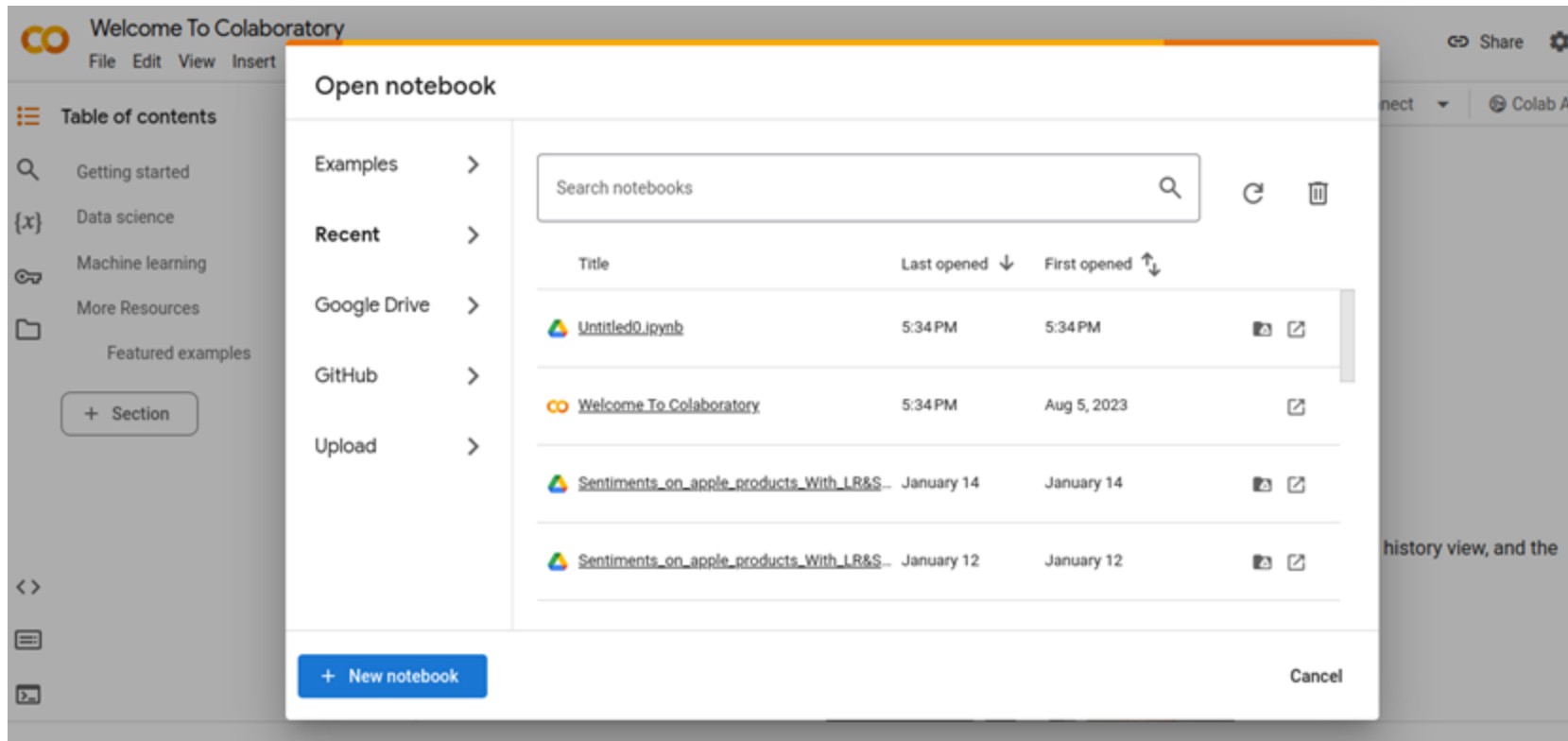
To add the Python directory to the path for a particular session in Windows –

At the command prompt – type `path %path%;C:\Python` and press Enter.

Note – C:\Python is the path of the Python directory

# Python Environment (ONLINE)

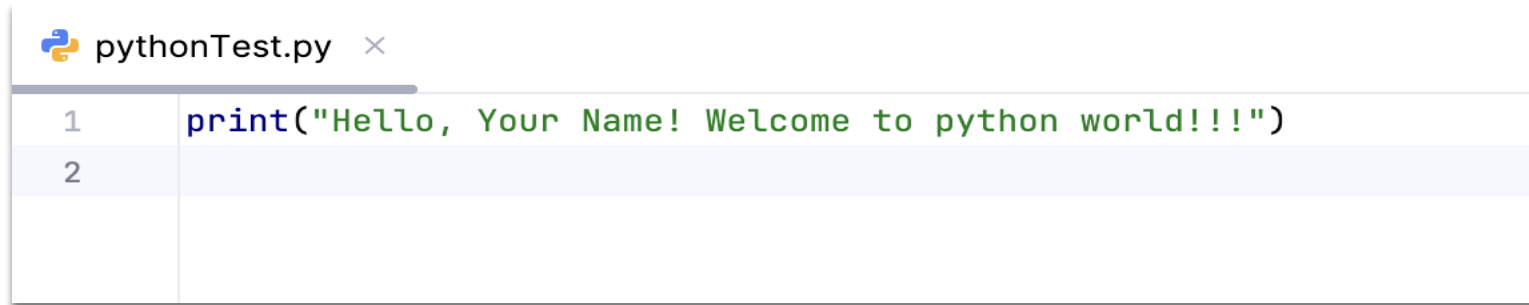
- Google Colab: <https://colab.research.google.com/>



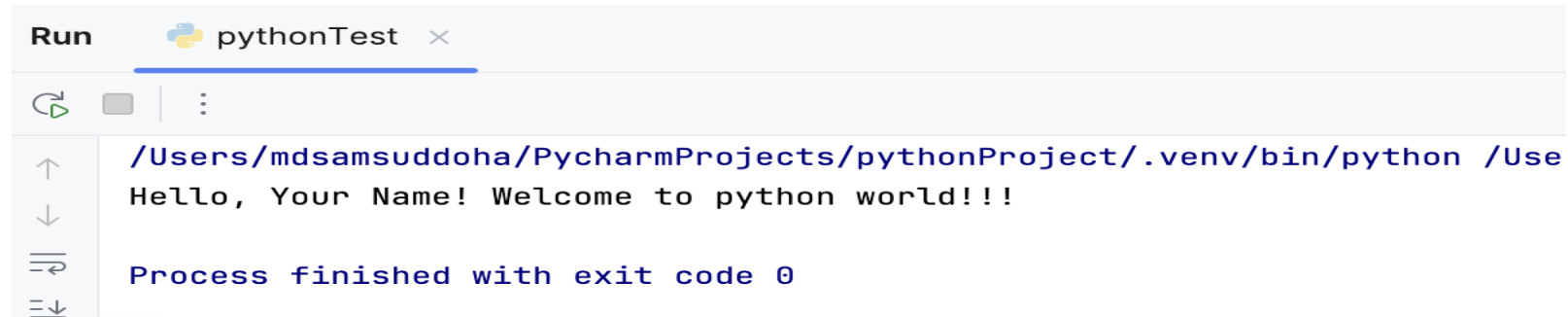
# Start with Python

---

- Open your editor(**PyCharm**) and write your first code.



```
pythonTest.py x
1 print("Hello, Your Name! Welcome to python world!!!")
2
```



```
Run pythonTest x
/Users/mdsamsuddoha/PycharmProjects/pythonProject/.venv/bin/python /Use
Hello, Your Name! Welcome to python world!!!
Process finished with exit code 0
```

# Examples

---

- Write a program that shows your information (name, contact number, address)
- Write a program that displays your name and department in two lines.



# Python Syntax (Identifiers)

---

A python identifier is a name to identify a variable, function, class, module or other objects.

## Naming Conventions for Python Identifiers:

- Identifiers cannot be a **keyword**.
- Identifiers are **case-sensitive**. (*AGE, age, Age are three different identifiers*)
- It can contain alpha-numeric characters and underscores (A-z, 0-9, and `_`). However, it must begin with a **letter** or **underscore** (`_`). The first letter of an identifier cannot be a digit.
- It's a convention to start an identifier with a letter rather `_`.
- White Spaces are not allowed.
- We cannot use special symbols like `!`, `@`, `#`, `$`, and so on.

# Python Syntax (Reserved Keywords)

---

<i>and</i>	<i>continue</i>	<i>finally</i>	<i>is</i>	<i>raise</i>
<i>as</i>	<i>def</i>	<i>for</i>	<i>lambda</i>	<i>return</i>
<i>assert</i>	<i>del</i>	<i>from</i>	<i>nonlocal</i>	<i>True</i>
<i>async</i>	<i>elif</i>	<i>global</i>	<i>None</i>	<i>try</i>
<i>await</i>	<i>else</i>	<i>if</i>	<i>not</i>	<i>while</i>
<i>break</i>	<i>except</i>	<i>import</i>	<i>or</i>	<i>with</i>
<i>class</i>	<i>False</i>	<i>in</i>	<i>pass</i>	<i>yield</i>

For Details About Keywords: <https://www.digitalocean.com/community/tutorials/python-keywords-identifiers>

# Python Syntax (Lines & Indentation)

---

Python programming provides **no braces** to indicate **blocks of code** for class and function definitions or flow control.

Blocks of code are denoted by **line indentation**, which is rigidly enforced.

The **number of spaces** in the indentation is variable, but all statements within the block must be indented the **same amount**. For example –

```
if True:
    print ("True")
else:
    print ("False")
```

# Python Syntax (Multi-Line Statements)

---

- Use of the *line continuation character* (\)

```
total = item_one + \  
        item_two + \  
        item_three
```

# Python Syntax (Multi-Line Statements)

Cnt.

---

- Statements contained within the [], {}, or () brackets do not need to use the ***line continuation character***.

```
days = [ 'Monday', 'Tuesday', 'Wednesday',  
         'Thursday', 'Friday' ]
```

# Python Syntax (Comments in Python)

---

A comment is a programmer-readable explanation or annotation in the Python source code that ignored by Python interpreter.

**Single Line Comment:** use #

```
# First comment  
print ("Hello, World!") # Second comment
```

# Python Syntax (Comments in Python)

---

A comment is a programmer-readable explanation or annotation in the Python source code that ignored by Python interpreter.

**Multi Line Comment: use '''text'''**

```
'''  
This is a multiline  
comment.  
'''
```

# Variable

---

- Variables are **containers** for storing data values.
- Python has **no command** for declaring a variable.
- A variable is created the moment you first assign a value to it.
- Variables do not need to be **declared with any particular type**, and can even **change type after** they have been set.

```
x = 5  
y = "John"  
print(x)  
print(y)
```



# Variable Names

---

- A variable can have a ***short name*** (like x and y) or a more ***descriptive name*** (age, carname, total\_volume).
- Rules for Python variables:
  - A variable name must start with a **letter or the underscore character**
  - A variable name cannot start with a number
  - A variable name can only contain **alpha-numeric characters and underscores** (A-z, 0-9, and \_ )
  - Variable names are **case-sensitive** (age, Age and AGE are three different variables)
  - A variable name cannot be any of the Python keywords.

○

# Variable Examples

---

- **Legal Variable:** *myvar, my\_var, \_my\_var, myVar, MYVAR, myvar2*
- **Illegal variable names:** *2myvar, my-var, my var*
- **Multi words variable**
  - Camel case: *myVariableName*
  - Pascal case: *MyVariableName*
  - Snake Case: *my\_variable\_name*

# Homeworks

---

- Write a python program that shows your full address in multiple lines.
- Write a python program that shows your name, age and phone number using variable.

---

# THANK YOU