# Python Programming Language

#### Prepared by: Mohamed Ayman

Algorithm Engineer at Valeo
Deep Learning Researcher and Teaching Assistant
at The American University in Cairo (AUC)
spring 2020







sw.eng.MohamedAyman@gmail.com



<u>linkedin.com/in/cs-MohamedAyman</u>



github.com/cs-MohamedAyman



codeforces.com/profile/Mohamed\_Ayman



#### **Mohamed Ayman**

#### **Experience**



- Valeo
  - Deep Learning Researcher
  - Algorithm Software Engineer



- The American University in Cairo (AUC)
  - Research Assistant
  - Teaching Assistant



- BBI Consultancy
  - Research and Development Engineer (R&D Engineer)
  - Data Engineer



#### **Education**

- MSc in Deep Learning, Cairo University
- BSc in Computer Science, Cairo University

# Python Programming Language Training

# Lecture Agenda

We will discuss in this lecture the following topics

- 1- Python Features
- 2- Python Content
- 3- Practice on Online Judges
- 4- Tutorials and References
- 5- Online Courses
- 6- Python Installation

4



# Lecture Agenda



#### ✓ Section 1: Python Features

Section 2: Python Content

Section 3: Practice on Online Judges

Section 4: Tutorials and References

**Section 5: Online Courses** 

Section 6: Python Installation





1 A simple language which is easier to learn

Python has a very simple and elegant syntax. It's much easier to read and write Python programs compared to other languages like: C++, Java, C#. Python makes programming fun and allows you to focus on the solution rather than syntax. If you are a newbie, it's a great choice to start your journey with Python.

You can freely use and distribute Python, even for commercial use. Not only can you use and distribute softwares written in it, you can even make changes to the Python's source code.

Python has a large community constantly improving it in each iteration.

3 Portability

You can move Python programs from one platform to another, and run it without any changes.

It runs seamlessly on almost all platforms including Windows, Mac OS X and Linux.



A high-level, interpreted language

Unlike C/C++, you don't have to worry about daunting tasks like memory management, garbage collection and so on. Likewise, when you run Python code, it automatically converts your code to the language your computer understands. You don't need to worry about any lower-level operations.

5 Large standard libraries to solve common tasks
Python has a number of standard libraries which m

Python has a number of standard libraries which makes life of a programmer much easier since you don't have to write all the code yourself. For example: Need to connect MySQL database on a Web server? You can use MySQLdb library using import MySQLdb.

Standard libraries in Python are well tested and used by hundreds of people. So you can be sure that it won't break your application.

6 Object-oriented

Everything in Python is an object. Object oriented programming (OOP) helps you solve a complex problem intuitively.

With OOP, you are able to divide these complex problems into smaller sets by creating objects.



Simple & Easy to Learn



```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, world");
    }
}

Python

print("Hello, world")
```



• Portable & Extensible



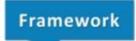


#### • Testing Frameworks

- > Python supports testing with cross-platform & cross-browser
- Built in testing framework which covers debugging time and fastest workflows















• Web Development





• Computer Graphics





















#### • Data Science















#### Big Data

- Python handles BIG DATA!
- > Python supports parallel computing
- You can write MapReduce codes in Python









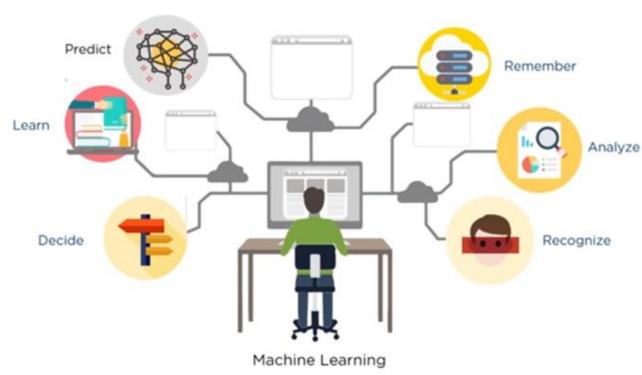






Artificial Intelligence





# Lecture Agenda



✓ Section 1: Python Features

Section 2: Python Content

Section 3: Practice on Online Judges

Section 4: Tutorials and References

Section 5: Online Courses

Section 6: Python Installation



## **Python Content**



Part 1: Python Basics and Functions Part 2: Python Collections and Strings

Lecture 1: Python Overview Lecture 7: Strings

Lecture 2: Variable Types Lecture 8: Lists

Lecture 3: Basic Operations Lecture 9: Tuples

Lecture 4: Conditions Lecture 10: Dictionaries

Lecture 5: Loops Lecture 11: Sets

Lecture 6: Functions Lecture 12: Numbers

## **Python Content**



Part 3: Python Object Oriented

Lecture 13: Object Oriented Overview

Lecture 14: Data Encapsulation

Lecture 15: Operator Overloading

**Function Overloading** 

Lecture 16: Inheritance

**Function Overriding** 

Lecture 17: Polymorphism

**Abstract Class** 

Part 4: Python Files and Standard Libraries

**Lecture 18: Modules** 

**Standard Libraries** 

Lecture 19: File Handling

Lecture 20: Exception Handling

#### Hands-on Projects & Practices



#### Python Basics Projects (4 Projects)

Project 1: Tic-Tac-Toe Game (python basics application)

Project 2: Connect Four Game (python basics application)

Project 3: Sudoku Game (python basics application)

Project 4: 2048 Game (python basics application)

#### Python Object Oriented Projects (4 Projects)

Project 1: Advanced Mathematical Calculator (python object oriented application)

Project 2: Library Management System (python object oriented application)

Project 3: Bank Management System (python object oriented application)

Project 4: Project Management System (python object oriented application)

Python Practices (30+ Practice Problems) on each Lecture

# Lecture Agenda



- ✓ Section 1: Python Features
- ✓ Section 2: Python Content

Section 3: Practice on Online Judges

Section 4: Tutorials and References

**Section 5: Online Courses** 

Section 6: Python Installation



### Practice on Online Judges







codeforces.com

hackerearth.com





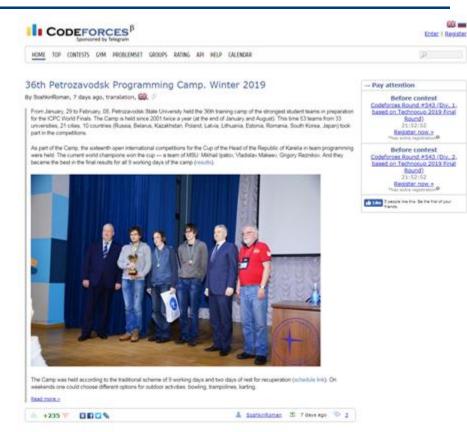
hackerrank.com

atcoder.jp

#### **Codeforces Online Judge**



Codeforces is a website that hosts competitive programming contests. It is maintained by a group of competitive programmers from ITMO University led by Mikhail Mirzayanov.





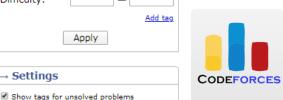
#### **Codeforces Online Judge**





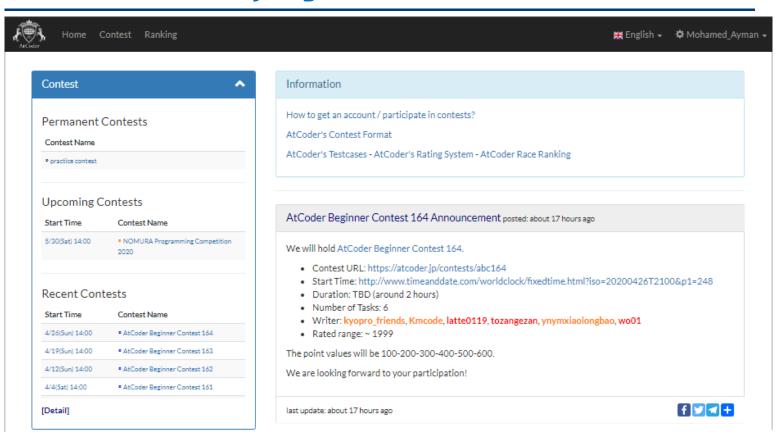
Mohamed Ayman | Logout

0 CONTESTS PROBLEMSET HELP CALENDAR ACMSGURU PROBLEMS SUBMIT STATUS STANDINGS CUSTOM TEST Problems :■ → Pay attention # Name **∮** ♦ Before contest **⋞**\* • Educational Codeforces Round 81 (Rated for Div. 2) 1294F Three Paths on a Tree dfs and similar, dp, trees 2100 x1488 3 days 1294E Obtain a Permutation greedy, implementation, math 2000 x1886 137 people like this. Be the first of your MEX maximizing data structures, math 1600 x4272 Product of Three Numbers greedy, math, number theory 1300 → Filter Problems x8603 Collecting Packages implementation, sortings 1200 Difficulty: x9042 Add tag 1294A Collecting Coins 900 x12769 Apply 1293B JOE is on TV! combinatorics, greedy, math 1000 x9324 ConneR and the A.R.C. Markland-N 1293A binary search, brute force, implementation 1100 x8075 → Settings 1292F Nora's Toy Boxes bitmasks, combinatorics, do 3400 ₹ x24



### AtCoder - Online Judge

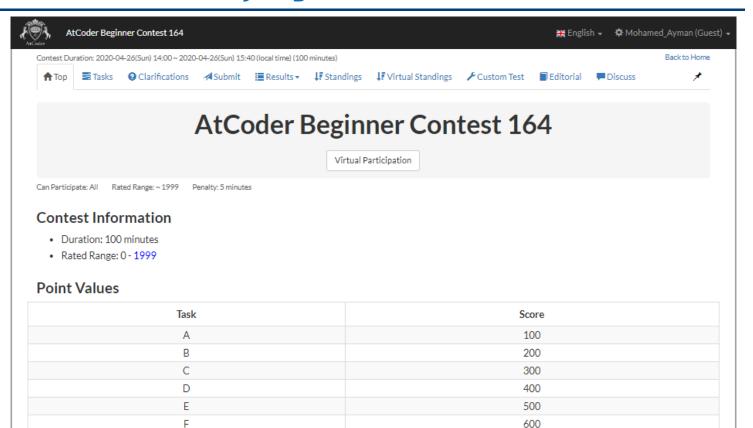






#### AtCoder - Online Judge







### HackerRank - Online Judge



Virtual Event | 11/08 | Learn how to master the art and science of skill assessments | Live streamed from San Francisco

HackerRank

Products Customers Resources Research Blog About Us

Login Sign Up

#### Join over 5 million developers.

Practice coding, prepare for interviews, and get hired.















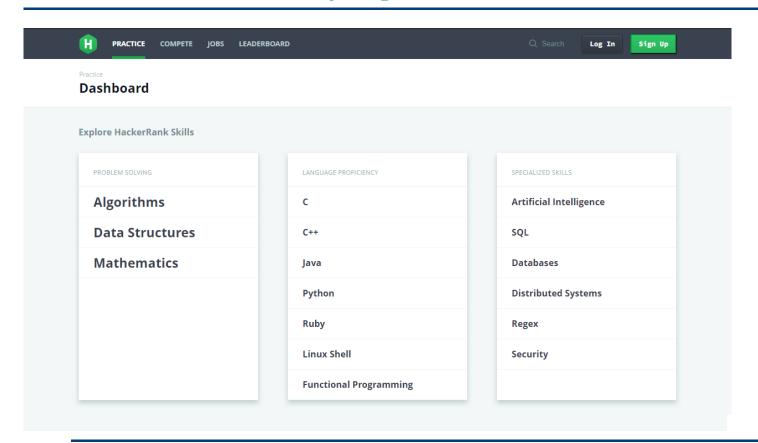






## HackerRank - Online Judge

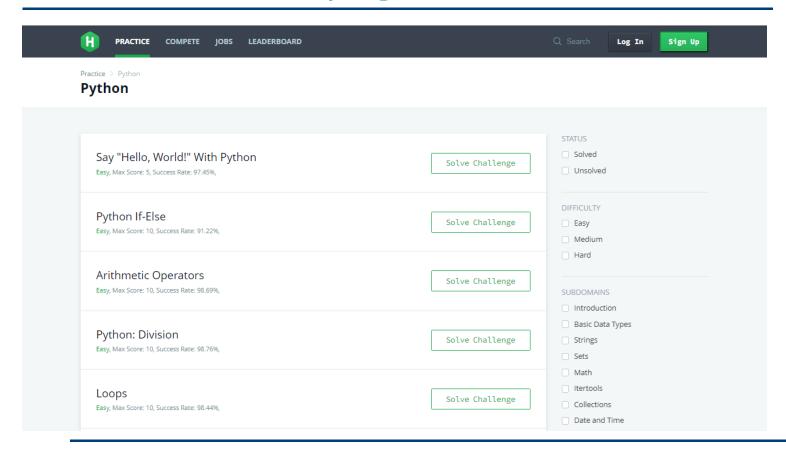






#### HackerRank - Online Judge

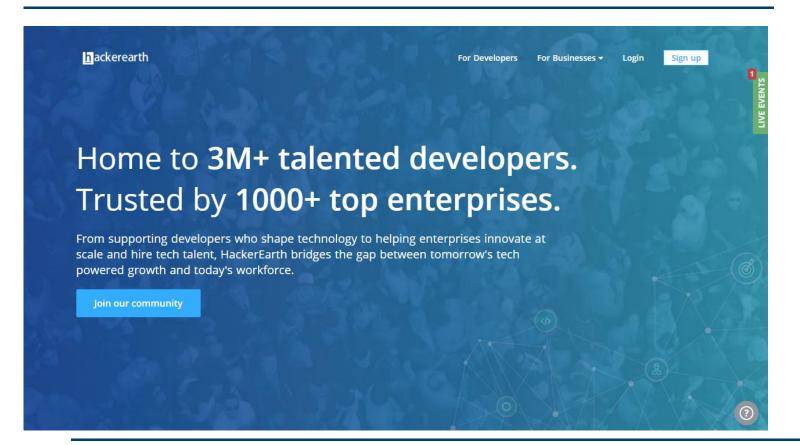






### HackerEarth - Online Judge

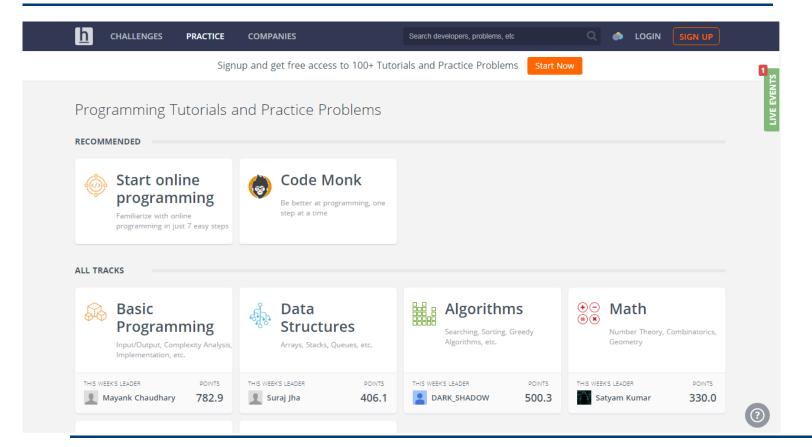






#### HackerEarth - Online Judge

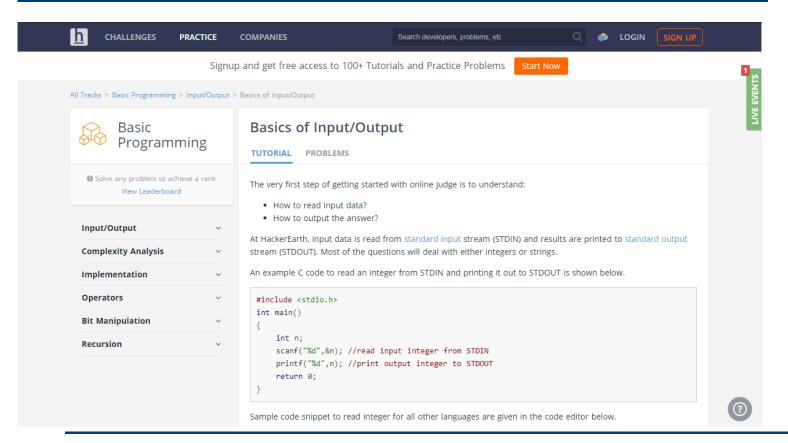






## HackerEarth - Online Judge







# Lecture Agenda



- ✓ Section 1: Python Features
- ✓ Section 2: Python Content
- ✓ Section 3: Practice on Online Judges

Section 4: Tutorials and References

**Section 5: Online Courses** 

Section 6: Python Installation



### **Python Tutorials**





programiz.com/python-programming



docs.python.org/3



geeksforgeeks.org/python-programming-language



tutorialspoint.com/python3

### Python 3 TutorialsPoint



#### PYTHON 3 - BASIC TUTORIAL

[001 - 330]

- Python 3 - What is New?

- Overview

- Environment Setup

- Basic Syntax

- Variable Types

- Basic Operators

- Decision Making

- Loops

- Numbers

- Strings

- Lists

- Tuples

- Dictionary

- Date & Time

- Functions

- Modules

- Files I/O

- Exceptions Handling

#### PYTHON 3 - ADVANCED TUTORIAL

[330 - 500]

- Object Oriented

- Regular Expressions

- CGI Programming

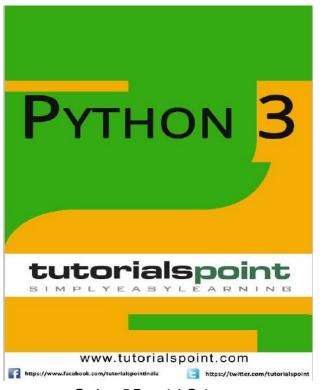
- MySQL Database Access

- Multithreaded Programming

- XML Processing

- GUI Programming

- Extension Programming with C



Python 3 TutorialsPoint

### Python Tutorial, Guido van Rossum



•	Using the Python Interpreter	[005 - 010]
•	An Informal Introduction to Python	[010 - 020]
•	More Control Flow Tools	[020 - 030]
•	Data Structures	[030 - 040]
•	Modules	[040 - 050]
•	Input and Output	[050 - 055]
•	Errors and Exceptions	[060 - 065]
•	Classes	[065 - 080]
•	Brief Tour of the Standard Library	[080 - 095]
•	Virtual Environments and Packages	[095 - 100]

Python Tutorial Release 3.6.4
Guido van Rossum and the Python development team
January 23, 2018
Python Software Foundation Email: docs@python.org

Python Tutorial, Guido van Rossum

# Python for Everybody, Charles R. Severance



•	Why should you learn to write programs? Variables, expressions, and statements	[001 - 015] [020 - 030]
•	Conditional execution	[030 - 040]
•	Functions	[040 - 055]
•	Iteration	[055 - 065]
•	Strings	[065 - 075]
•	Files	[080 - 090]
•	Lists	[090 - 105]
•	Dictionaries	[105 - 115]
•	Tuples	[115 - 125]
•	Regular expressions	[125 - 140]
•	Object-oriented programming	[170 - 185]
•	Using Databases and SQL	[185 - 210]

Python for Everybody

Exploring Data Using Python 3

Charles R. Severance

Python for Everybody, Charles R. Severance

# Learning to Program Using Python, Cody Jackson



•	Introduction	[010 - 015]
•	How is Python Different?	[015 - 020]
•	Comparison of Programming Languages	[020 - 030]
•	The Python Interpreter	[030 - 035]
•	Types and Operators	[035 - 040]
•	Strings	[040 - 050]
•	Lists	[050 - 055]
•	Dictionaries	[055 - 060]
•	Tuples	[060 - 070]
•	Files	[070 - 075]
•	Statements	[075 - 090]
•	Making a Program	[100 - 105]
•	Exceptions	[105 - 110]
•	Object Oriented Programming	[110 - 125]
•	Databases	[125 - 135]

Learning to Program Using Python

Cody Jackson

Learning to Program Using Python, Cody Jackson

# Think Python, Allen B. Downey



•	The way of the program	[001 - 010]
•	Variables, expressions and statements	[010 - 015]
•	Functions	[015 - 025]
•	Conditionals and recursion	[040 - 050]
•	Fruitful functions	[050 - 060]
•	Iteration	[060 - 070]
•	Strings	[070 - 080]
•	Lists	[090 - 100]
•	Dictionaries	[100 - 110]
•	Tuples	[110 - 120]
•	Files	[135 - 145]
•	Classes and objects	[145 - 155]
•	Classes and functions	[155 - 160]
•	Classes and methods	[160 - 170]
•	Inheritance	[170 - 180]

#### Think Python

How to Think Like a Computer Scientist

2nd Edition, Version 2.2.23

Think Python, Allen B. Downey

### Effective Python, Brett Slatkin



•	Pythonic Thinking	[001 - 025]
	i yenome immining	[007 053]

• Functions [025 - 050]

• Classes and Inheritance [050 - 075]

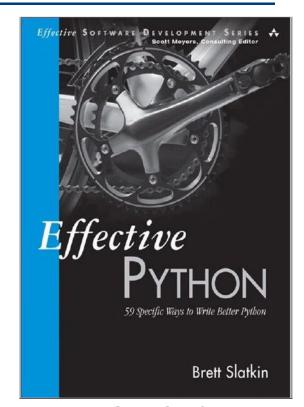
Metaclasses and Attributes [075 - 100]

• Concurrency and Parallelism [100 - 125]

• Built-in Modules [125 - 150]

• Collaboration [150 - 175]

• Production [175 - 200]

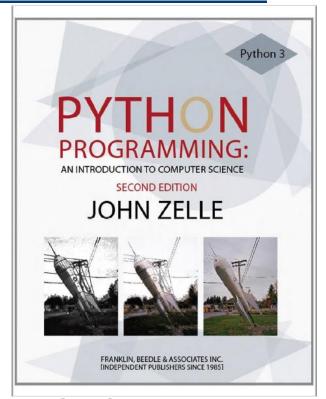


Effective Python, Brett Slatkin

# Python Programming, John Zelle



•	Computers and Programs	[001 - 020]
•	Writing Simple Programs	[020 - 045]
•	Computing with Numbers	[045 - 065]
•	Objects and Graphics	[065 - 100]
•	Sequences: Strings, Lists, and Files	[100 - 140]
•	Defining Functions	[140 - 165]
•	Decision Structures	[165 - 200]
•	Loop Structures and Booleans	[200 - 225]
•	Simulation and Design	[225 - 245]
•	Defining Classes	[245 - 280]
•	Data Collections	[280 - 325]
•	Object-Oriented Design	[325 - 360]
•	Algorithm Design and Recursion	[360 - 385]



Python Programming, John Zelle

# Lecture Agenda



- ✓ Section 1: Python Features
- ✓ Section 2: Python Content
- ✓ Section 3: Practice on Online Judges
- ✓ Section 4: Tutorials and References

**Section 5: Online Courses** 

Section 6: Python Installation



## Python Coursera Specializations





<ul> <li>Introduction to Scripting in Python Specialization by Rice University</li> </ul>	[40 H]
https://www.coursera.org/specializations/introduction-scripting-in-python	
<ul> <li>Python 3 Programming Specialization by University of Michigan</li> </ul>	[135 H]
https://www.coursera.org/specializations/python-3-programming	
<ul> <li>Fundamentals of Computing Specialization by Rice University</li> </ul>	[115 H]
https://www.coursera.org/specializations/computer-fundamentals	
<ul> <li>Learn to Program: Crafting Quality Code by University of Toronto</li> </ul>	[15 H]
https://www.coursera.org/learn/program-code	

# Python YouTube Playlists





Python Tutorial Videos   Simplilearn	[20 H]	Channel: Simplilearn
<pre>youtube.com/playlist?list=PLEiEAq2VkUUKoW1o-A-VEmkoGKSC26i_I</pre>		
Python Tutorial For Beginners   Edureka	[45 H]	Channel: edureka!
youtube.com/playlist?list=PL9ooVrP1hQ0HY-BeYrKHDrHKphsJ0yRyu		
Python Tutorial for Beginners	[40 H]	Channel: ProgrammingKnowledge
<pre>youtube.com/playlist?list=PLS1QulWo1RlaJECMeUT4LFwJ-ghgoSH6n</pre>		
Python 3 Programming Tutorial	[10 H]	Channel: codebasics
<pre>youtube.com/playlist?list=PLeo1K3hjS3usILfyvQlvUBokXkHPSve6S</pre>		
<ul> <li>Learning to program with Python 3</li> </ul>	[3 H]	Channel: sentdex
youtube.com/playlist?list=PLQVvvaa0QuDeAams7fkdcwOGBpGdHpXln		
Python 3 Tutorial for Beginners	[30 H]	Channel: ProgrammingKnowledge
<pre>youtube.com/playlist?list=PLS1QulWo1RIYt4e0WnBp-ZjCNq8X0FX0]</pre>		
Complete Python Training Course	[3 H]	Channel: Python Programmer
<pre>youtube.com/playlist?list=PLtb2Lf-cj_AWhtjE6Rb5oWf02RC2qVU-j</pre>		
Python Tutorial for Beginners	[20 H]	Channel: Telusko
<pre>youtube.com/playlist?list=PLsyeobzWxl7poL9JTVyndKe62ieoN-MZ3</pre>		
<ul> <li>Complete Python tutorial in Hindi (2020)</li> </ul>	[20 H]	Channel: Harshit vashisth
<pre>youtube.com/playlist?list=PLwgFb6VsUj_IQTpQKDtLXKXEIQychT_2j</pre>		

# Lecture Agenda



- ✓ Section 1: Python Features
- ✓ Section 2: Python Content
- ✓ Section 3: Practice on Online Judges
- ✓ Section 4: Tutorials and References
- ✓ Section 5: Online Courses

Section 6: Python Installation



#### Python Interpreters Installation





python.org





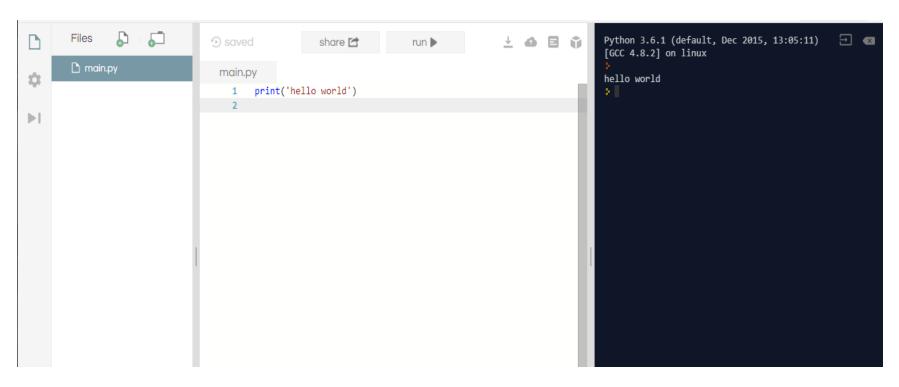
anaconda.com/distribution



# Python Online Interpreters



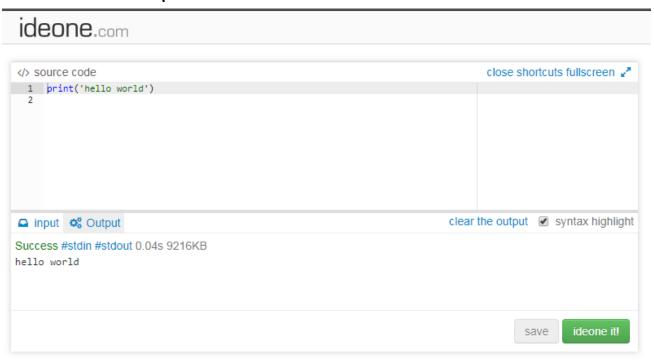
#### repl.it online interpreter: repl.it/languages/python3



# Python Online Interpreters



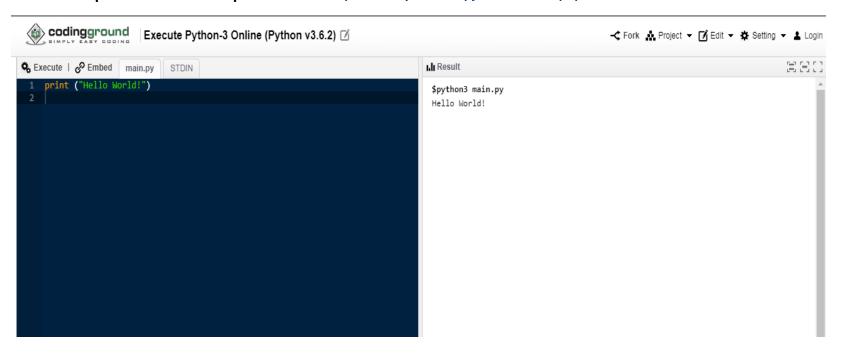
#### ideone online interpreter: <u>ideone.com</u>



## Python Online Interpreters



#### Tutorials point online interpreter: tutorialspoint.com/execute python3 online.php





- Go to Download Python page on the official site and click **Download Python 3.6.0** (You may see different version name).
- When the download is completed, double-click the file and follow the instructions to install it.

  When Python is installed, a program called IDLE is also installed along with it. It provides graphical user interface to work with Python.
- Open IDLE, copy the following code below and press enter.

  print("Hello, World!")
- To create a file in IDLE, go to File > New Window (Shortcut: Ctrl+N).
- Write Python code (you can copy the code below for now) and save (Shortcut: Ctrl+S) with .py file extension like: hello.py or your-first-program.py

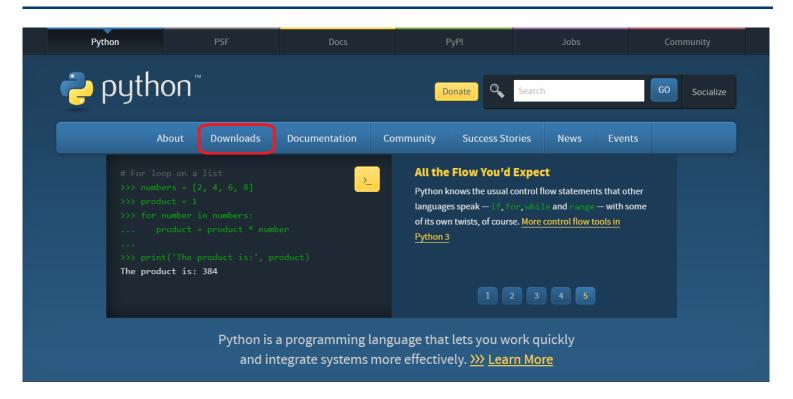
```
1 print("Hello, World!")
```

Go to Run > Run module (Shortcut: F5) and you can see the output. Congratulations, you've successfully run your first Python program.

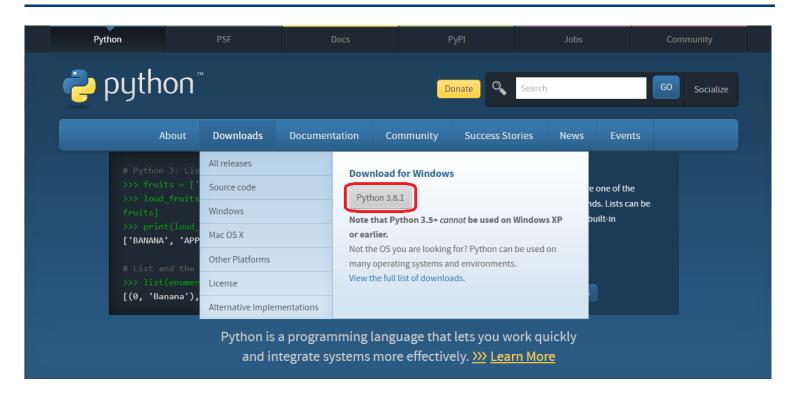


python.org

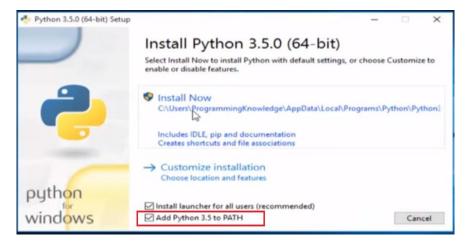


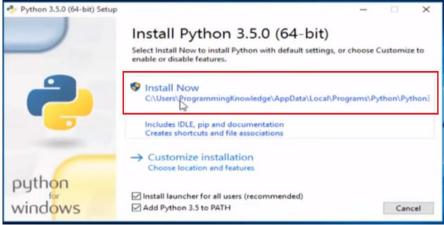




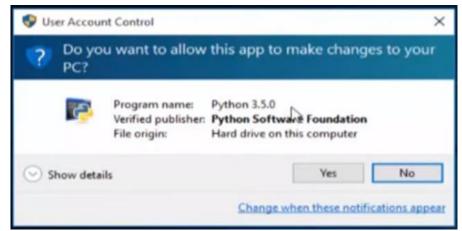


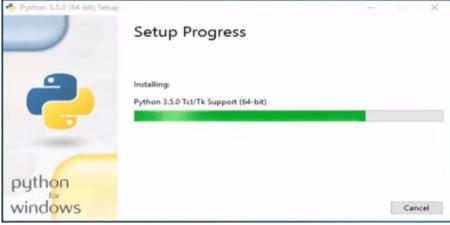




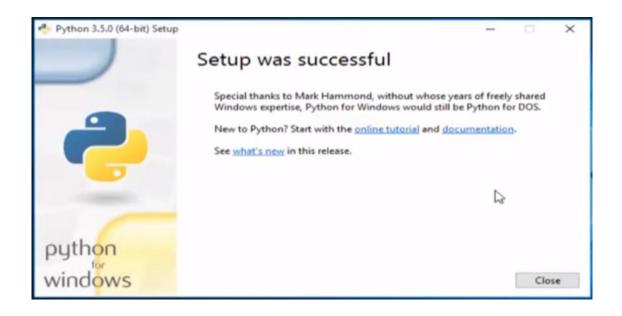






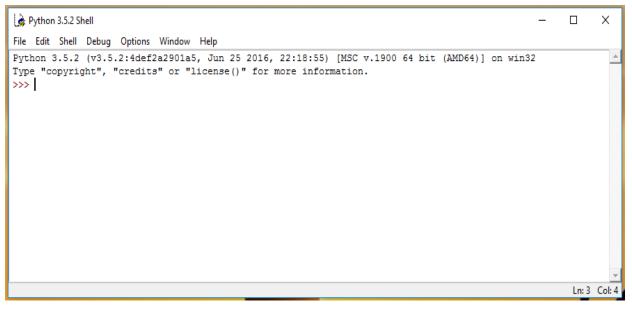




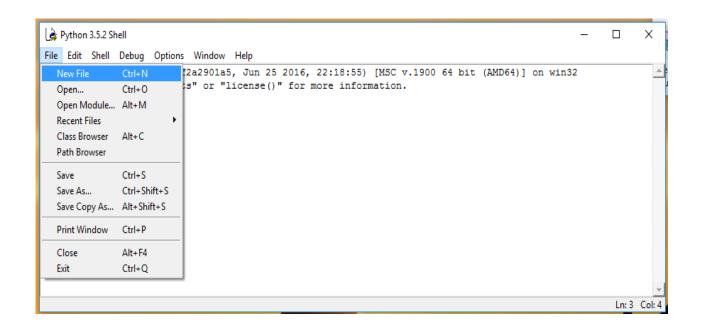




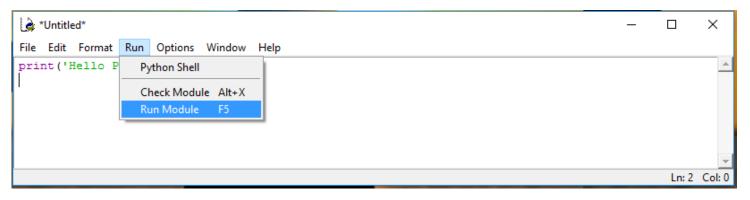


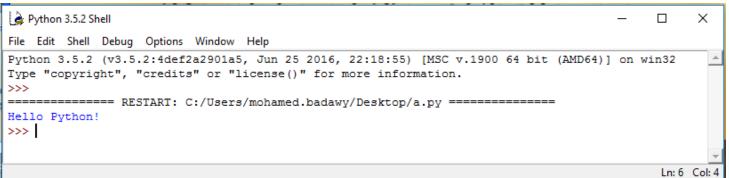












# First Python Program



#### There are two mods of programming to execute the programs

1. Interactive Mode Programming

Invoking the interpreter without passing a script file as a parameter brings up the following prompt

# First Python Program



There are two mods of programming to execute the programs

#### 2. Script Mode Programming

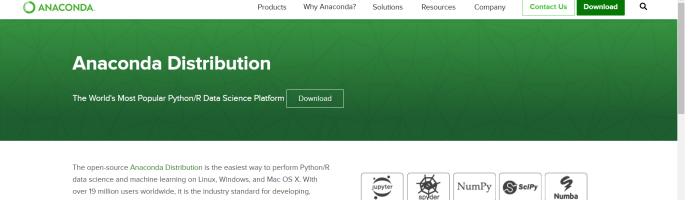
Invoking the interpreter with a script parameter begins execution of the script and continues until the script is finished. When the script is finish, the interpreter is no longer active.

```
Python 3.4.0 Shell

File Edit Shell Debug Options Windows Help

print('a')
print('b')
print('c')
print('d')
```







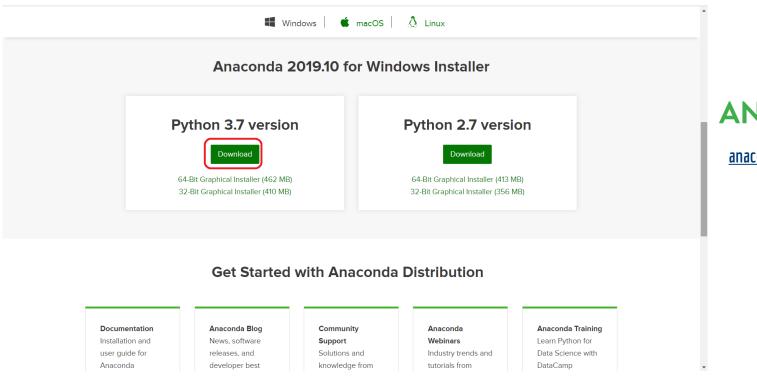
anaconda.com/distribution

- testing, and training on a single machine, enabling *individual data scientists* to:

   Quickly download 7,500+ Python/R data science packages
  - Manage libraries, dependencies, and environments with Conda
  - Develop and train machine learning and deep learning models with scikitlearn, TensorFlow, and Theano
  - Analyze data with scalability and performance with Dask, NumPy, pandas, and Numba
  - · Visualize results with Matplotlib, Bokeh, Datashader, and Holoviews



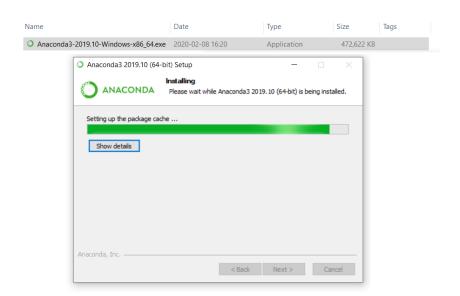


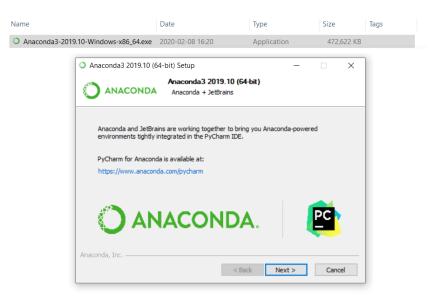




anaconda.com/distribution

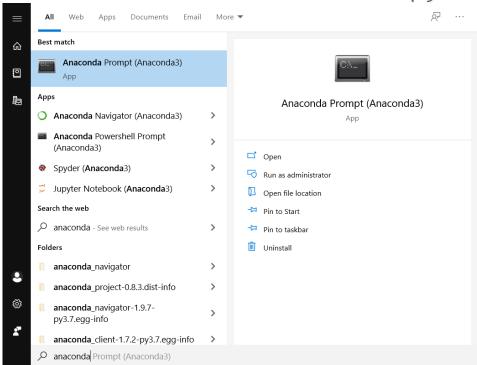












C:\Users\LenovoY540\Anaconda3\python.exe



#### Ensure that all steps done successfully, you should run the following command lines:

```
Anaconda Prompt (Anaconda3)
(base) C:\Users\LenovoY540>conda list
 packages in environment at C:\Users\LenovoY540\Anaconda3:
                         Version
                                                   Build Channel
# Name
ipyw jlab nb ext conf
                         0.1.0
                                                  py37 0
alabaster
                         0.7.12
                                                  py37 0
anaconda
                         2019.10
                                                 py37 0
anaconda-client
                         1.7.2
                                                  py37 0
anaconda-navigator
                         1.9.7
                                                  py37 0
anaconda-project
                         0.8.3
                                                    py_0
asn1crypto
                         1.0.1
                                                  py37 0
astroid
                         2.3.1
                                                  py37 0
astropy
                         3.2.1
                                          py37he774522 0
atomicwrites
                         1.3.0
                                                  py37 1
(base) C:\Users\LenovoY540>python --version
Python 3.7.4
(base) C:\Users\LenovoY540>where python
```

## First Python Program



There are two mods of programming to execute the programs

1. Interactive Mode Programming

Invoking the interpreter without passing a script file as a parameter brings up the following prompt

```
Anaconda Prompt (Anaconda3) - python

(base) C:\Users\LenovoY540>python

Python 3.7.4 (default, Aug 9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32

Type "help", "copyright", "credits" or "license" for more information.

>>> print('a')

a

>>> print('b')

b

>>> print('c')

c

>>> print('d')

d

>>>
```

# First Python Program



There are two mods of programming to execute the programs

#### 2. Script Mode Programming

Invoking the interpreter with a script parameter begins execution of the script and continues until the script is finished. When the script is finish, the interpreter is no longer active.

```
Anaconda Prompt (Anaconda3)

(base) C:\Users\LenovoY540>python E:/scripts/test.py
a
b
c
d
```

# Lecture Agenda



- ✓ Section 1: Python Features
- ✓ Section 2: Python Content
- ✓ Section 3: Practice on Online Judges
- ✓ Section 4: Tutorials and References
- ✓ Section 5: Online Courses
- ✓ Section 6: Python Installation



