

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



linkedin.com/in/cs-MohamedAyman



github.com/cs-MohamedAyman



codeforces.com/profile/Mohamed_Ayman



Mohamed Ayman

Experience



- Valeo
 - Deep Learning Researcher
 - Algorithm Software Engineer

[2019-Present]



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

[2019-Present]



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

[2016-2019]



Cairo University

Education

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

[2018-2021]

[2013-2017]



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
-



Let's
STARTUP

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 Features



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.

2 **Free and open-source**

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.

Python Features



4

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 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.

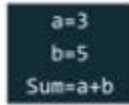
Python Features



- Simple & Easy to Learn



Open Source



High-level



Interpreted



Large community

Java

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

Python

```
print("Hello, world")
```

Python Features



- Portable & Extensible



Python Features



- Testing Frameworks

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

Tools



Splinter

Framework



Python Features



- Web Development



- Develop web applications
- Scrape websites

Frameworks

django



WEB2PY

Python Features



- Computer Graphics

➤ Graphical User Interface

➤ Desktop applications

➤ Game development

Libraries

Tk
Tkinter



Python Features



- Data Science



- Well-suited for data manipulation & analysis
- Deals with tabular data with heterogeneously-typed columns
- Arbitrary matrix data
- Observational/ statistical datasets

Libraries



NumPy

Pandas



matplotlib

seaborn

Python Features



- Big Data

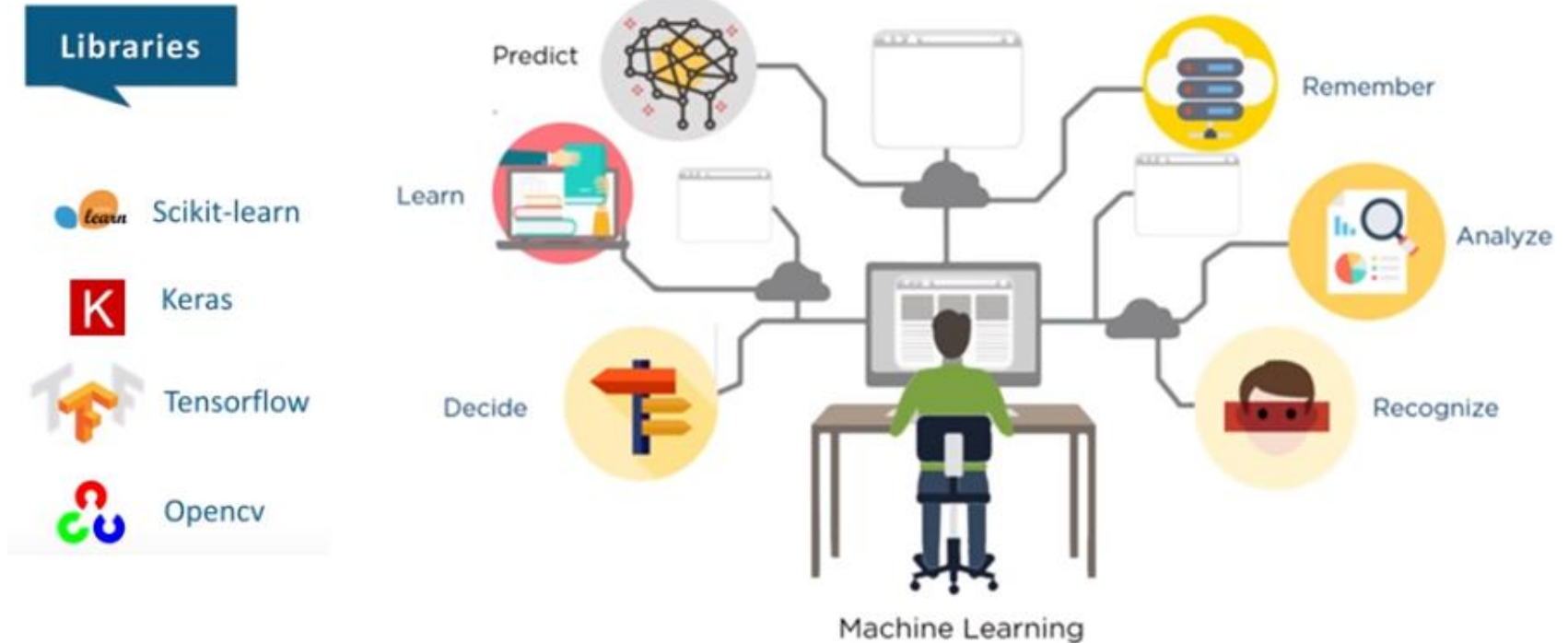
- Python handles **BIG DATA!**
- Python supports **parallel** computing
- You can write **MapReduce** codes in Python

Libraries



Python Features

- 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



Part 1: Python Basics and Functions

Lecture 1: Python Overview

Lecture 2: Variable Types

Lecture 3: Basic Operations

Lecture 4: Conditions

Lecture 5: Loops

Lecture 6: Functions

Part 2: Python Collections and Strings

Lecture 7: Strings

Lecture 8: Lists

Lecture 9: Tuples

Lecture 10: Dictionaries

Lecture 11: Sets

Lecture 12: Numbers

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.



Sponsored by Telegram

Enter | Register

HOME TOP CONTESTS GYM PROBLEMS SET GROUPS RATING API HELP CALENDAR

36th Petrozavodsk Programming Camp. Winter 2019

By SoahinRoman, 7 days ago, translation, 666

From January 29 to February 08, Petrozavodsk State University held the 36th training camp of the strongest student teams in preparation for the ICPC World Finals. The Camp is held since 2001 twice a year (at the end of January and August). This time 53 teams from 33 universities, 21 cities, 10 countries (Russia, Belarus, Kazakhstan, Poland, Latvia, Lithuania, Estonia, Romania, South Korea, Japan) took part in the competitions.

As part of the Camp, the sixteenth open international competitions for the Cup of the Head of the Republic of Karelia in team programming were held. The current world champions won the cup — a team of MSU: Mikhail Igatov, Vladimir Makeev, Grigory Reznikov. And they became the best in the final results for all 9 working days of the camp ([results](#)).



The Camp was held according to the traditional scheme of 9 working days and two days of rest for recuperation ([schedule link](#)). On weekends one could choose different options for outdoor activities: bowling, trampolines, karting.

[Read more...](#)

+235

SoahinRoman 7 days ago

Pay attention

Before contest

[Codeforces Round #543 \(Div. 1, based on Technocup 2019 Final Round\)](#)

21:52:52

[Register now >](#)

Host write registration

Before contest

[Codeforces Round #543 \(Div. 2, based on Technocup 2019 Final Round\)](#)

21:52:52

[Register now >](#)

Host write registration

Like

2 people like this. Be the first of your friends.

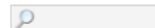


Codeforces Online Judge



Mohamed Ayman | Logout

HOME TOP CONTESTS GYM PROBLEMSET GROUPS RATING API HELP CALENDAR



MAIN ACMSGURU | **PROBLEMS** SUBMIT STATUS STANDINGS CUSTOM TEST

Problems

#	Name				
1294F	Three Paths on a Tree	dfs and similar, dp, trees			2100 x1488
1294E	Obtain a Permutation	greedy, implementation, math			2000 x1886
1294D	MEX maximizing	data structures, math			1600 x4272
1294C	Product of Three Numbers	greedy, math, number theory			1300 x8603
1294B	Collecting Packages	implementation, sortings			1200 x9042
1294A	Collecting Coins	math			900 x12769
1293B	JOE is on TV!	combinatorics, greedy, math			1000 x9324
1293A	ConneR and the A.R.C. Markland-N	binary search, brute force, implementation			1100 x8075
1292F	Nora's Toy Boxes	bitmasks, combinatorics, dp			3400 x24

→ Pay attention

Before contest
[Educational Codeforces Round 81](#)
(Rated for Div. 2)
3 days

Like 137 people like this. Be the first of your friends.

→ Filter Problems

Difficulty: —
[Add tag](#)

→ Settings

☒ Show tags for unsolved problems



AtCoder - Online Judge



Home Contest Ranking

English Mohamed_Ayman

Contest

Permanent Contests

Contest Name

• practice contest

Upcoming Contests

Start Time

Contest Name

5/30(Sat) 14:00

• NOMURA Programming Competition 2020

Recent Contests

Start Time

Contest Name

4/26(Sun) 14:00

• AtCoder Beginner Contest 164

4/19(Sun) 14:00

• AtCoder Beginner Contest 163

4/12(Sun) 14:00

• AtCoder Beginner Contest 162

4/4(Sat) 14:00

• AtCoder Beginner Contest 161

[Detail]

Information

How to get an account / participate in contests?

AtCoder's Contest Format

AtCoder's Testcases - AtCoder's Rating System - AtCoder Race Ranking

AtCoder Beginner Contest 164 Announcement posted: about 17 hours ago

We will hold AtCoder Beginner Contest 164.

- Contest URL: <https://atcoder.jp/contests/abc164>
- Start Time: <http://www.timeanddate.com/worldclock/fixedtime.html?iso=20200426T100&p1=248>
- Duration: TBD (around 2 hours)
- Number of Tasks: 6
- Writer: [kyopro_friends](#), [Kmcode](#), [latte0119](#), [tozangezan](#), [ynymxiaolongbao](#), [wo01](#)
- Rated range: ~ 1999

The point values will be 100-200-300-400-500-600.


We are looking forward to your participation!

last update: about 17 hours ago



AtCoder - Online Judge



 AtCoder Beginner Contest 164 English Mohamed_Ayman (Guest)

Contest Duration: 2020-04-26(Sun) 14:00 ~ 2020-04-26(Sun) 15:40 (local time) (100 minutes) [Back to Home](#)

[Top](#) [Tasks](#) [Clarifications](#) [Submit](#) [Results](#) [Standings](#) [Virtual Standings](#) [Custom Test](#) [Editorial](#) [Discuss](#)

AtCoder Beginner Contest 164

Virtual Participation

Can Participate: All Rated Range: ~ 1999 Penalty: 5 minutes

Contest Information

- Duration: 100 minutes
- Rated Range: 0 - 1999

Point Values

Task	Score
A	100
B	200
C	300
D	400
E	500
F	600



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.

[Sign Up & Code](#)

Hiring Talent? [Learn more](#)



UBER

vmware



HackerRank - Online Judge



Practice

Dashboard

Explore HackerRank Skills

PROBLEM SOLVING

Algorithms

Data Structures

Mathematics

LANGUAGE PROFICIENCY

C

C++

Java

Python

Ruby

Linux Shell

Functional Programming

SPECIALIZED SKILLS

Artificial Intelligence

SQL

Databases

Distributed Systems

Regex

Security



HackerRank - Online Judge



Practice > Python

Python

Say "Hello, World!" With Python

Easy, Max Score: 5, Success Rate: 97.45%,

[Solve Challenge](#)

Python If-Else

Easy, Max Score: 10, Success Rate: 91.22%,

[Solve Challenge](#)

Arithmetic Operators

Easy, Max Score: 10, Success Rate: 98.69%,

[Solve Challenge](#)

Python: Division

Easy, Max Score: 10, Success Rate: 98.76%,

[Solve Challenge](#)

Loops

Easy, Max Score: 10, Success Rate: 98.44%,

[Solve Challenge](#)

STATUS

- ☐ Solved
- ☐ Unsolved

DIFFICULTY

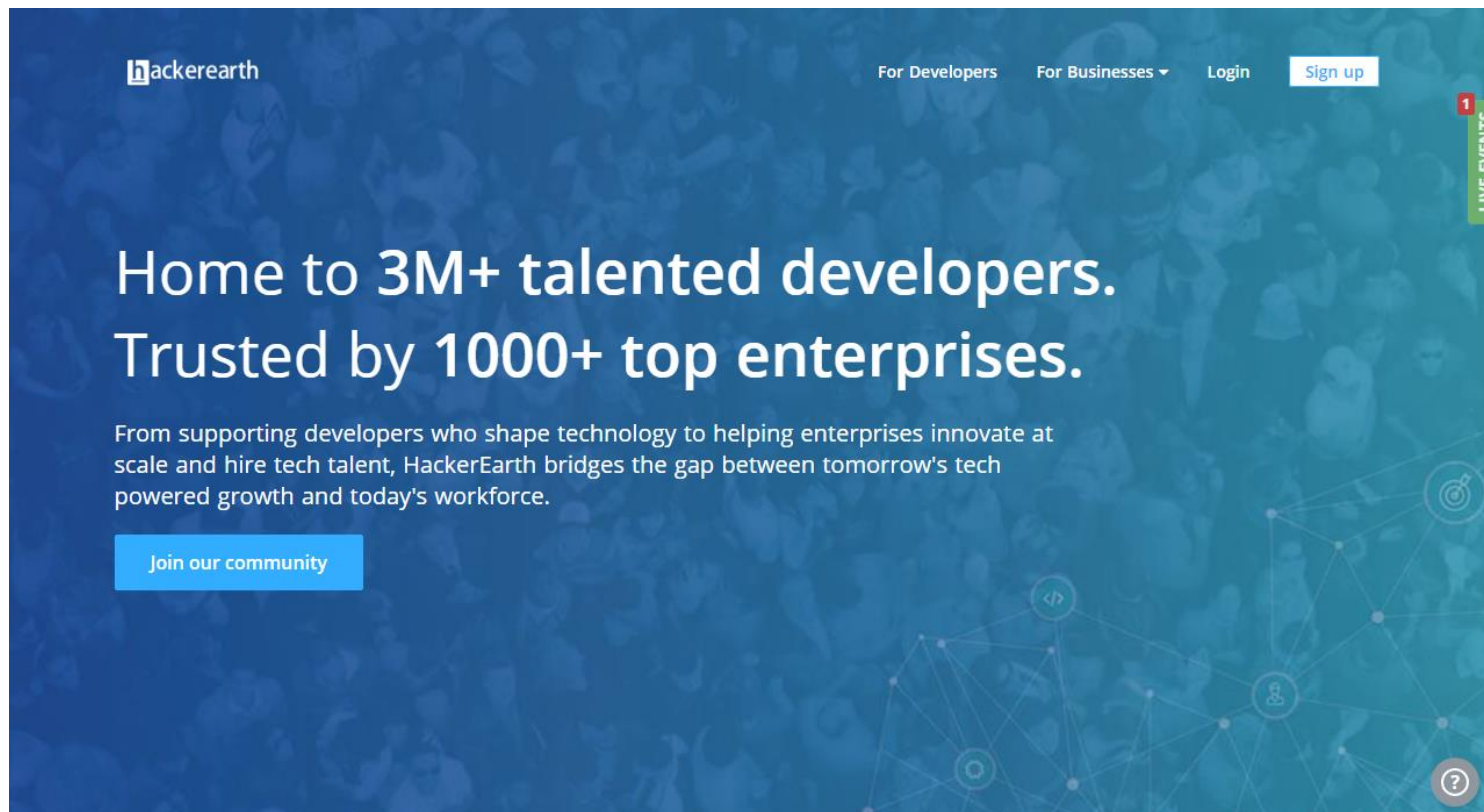
- ☐ Easy
- ☐ Medium
- ☐ Hard

SUBDOMAINS

- ☐ Introduction
- ☐ Basic Data Types
- ☐ Strings
- ☐ Sets
- ☐ Math
- ☐ Itertools
- ☐ Collections
- ☐ Date and Time



HackerEarth - Online Judge



The image is a screenshot of the HackerEarth website. The background is a dark blue with a subtle pattern of light blue faces. In the top left corner is the HackerEarth logo. In the top right corner, there are links for 'For Developers', 'For Businesses' (with a dropdown arrow), 'Login', and a 'Sign up' button. On the far right, there is a vertical green bar with a red '1' and the text 'LIVE EVENTS'. The main content area features the text 'Home to 3M+ talented developers. Trusted by 1000+ top enterprises.' followed by a paragraph: 'From supporting developers who shape technology to helping enterprises innovate at scale and hire tech talent, HackerEarth bridges the gap between tomorrow's tech powered growth and today's workforce.' Below this is a blue button that says 'Join our community'. In the bottom right corner, there is a network diagram with nodes and lines, and several circular icons containing symbols like a target, a person, and a question mark.

hackerearth

For Developers For Businesses Login Sign up

LIVE EVENTS

Home to 3M+ talented developers. Trusted by 1000+ top enterprises.


From supporting developers who shape technology to helping enterprises innovate at scale and hire tech talent, HackerEarth bridges the gap between tomorrow's tech powered growth and today's workforce.

Join our community



HackerEarth - Online Judge



CHALLENGESPRACTICECOMPANIES

Search developers, problems, etc

LOGIN

SIGN UP


Signup and get free access to 100+ Tutorials and Practice Problems

Start Now


1
LIVE EVENTS

Programming Tutorials and Practice Problems

RECOMMENDED


**Start online programming**

Familiarize with online programming in just 7 easy steps


**Code Monk**


Be better at programming, one step at a time

ALL TRACKS


**Basic Programming**


Input/Output, Complexity Analysis, Implementation, etc.

THIS WEEK'S LEADER	POINTS
 Mayank Chaudhary	782.9


**Data Structures**


Arrays, Stacks, Queues, etc.

THIS WEEK'S LEADER	POINTS
 Suraj Jha	406.1


**Algorithms**

Searching, Sorting, Greedy Algorithms, etc.

THIS WEEK'S LEADER	POINTS
 DARK_SHADOW	500.3

**Math**

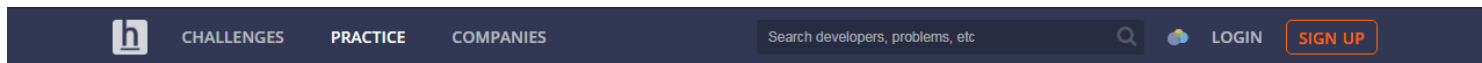
Number Theory, Combinatorics, Geometry

THIS WEEK'S LEADER	POINTS
 Satyam Kumar	330.0

?



HackerEarth - Online Judge



Signup and get free access to 100+ Tutorials and Practice Problems

Start Now

LIVE EVENTS

All Tracks > Basic Programming > Input/Output > Basics of Input/Output



Basic Programming

Solve any problem to achieve a rank
[View Leaderboard](#)

Input/Output

Complexity Analysis

Implementation

Operators

Bit Manipulation

Recursion

Basics of Input/Output

TUTORIAL PROBLEMS

The very first step of getting started with online judge is to understand:

- How to read input data?
- How to output the answer?

At HackerEarth, input data is read from [standard input](#) stream (STDIN) and results are printed to [standard output](#) stream (STDOUT). Most of the questions will deal with either integers or strings.

An example C code to read an integer from STDIN and printing it out to STDOUT is shown below.

```
#include <stdio.h>
int main()
{
    int n;
    scanf("%d",&n); //read input integer from STDIN
    printf("%d",n); //print output integer to STDOUT
    return 0;
}
```

Sample code snippet to read integer for all other languages are given in the code editor below.



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



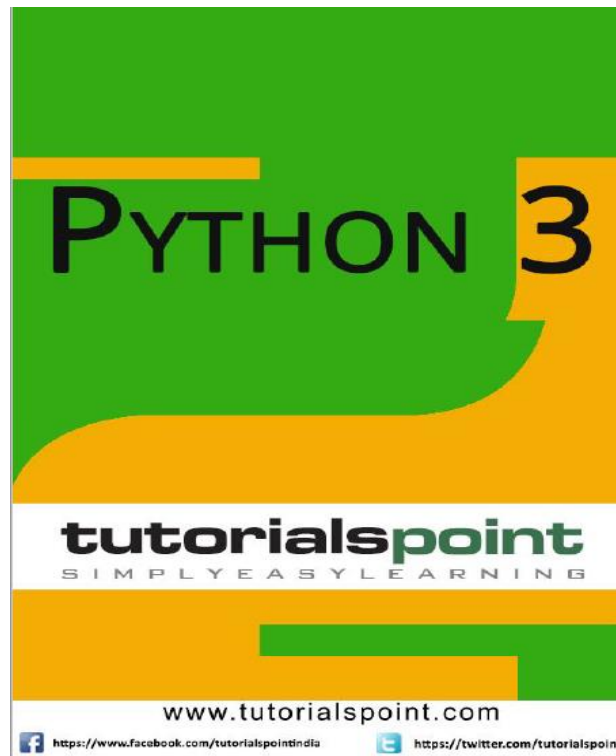
tutorialspoint.com/python3

- **PYTHON 3 - BASIC TUTORIAL** [001 - 330]

- Python 3 - What is New?
- Environment Setup
- Variable Types
- Decision Making
- Numbers
- Lists
- Dictionary
- Functions
- Files I/O
- Overview
- Basic Syntax
- Basic Operators
- Loops
- Strings
- Tuples
- Date & Time
- Modules
- Exceptions Handling

- **PYTHON 3 - ADVANCED TUTORIAL** [330 - 500]

- Object Oriented
- CGI Programming
- Multithreaded Programming
- GUI Programming
- Regular Expressions
- MySQL Database Access
- XML Processing
- Extension Programming with C

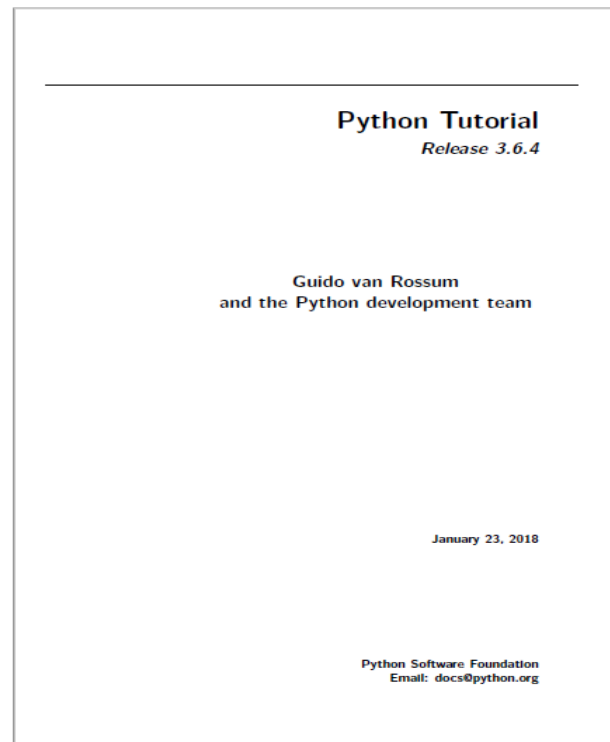


Python 3 TutorialsPoint

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]

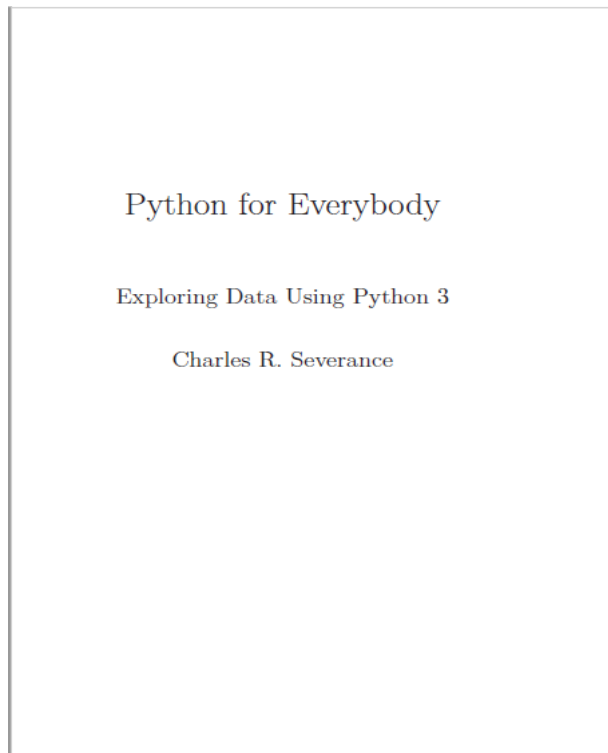


Python Tutorial, Guido van Rossum

Python for Everybody, Charles R. Severance



- Why should you learn to write programs? [001 - 015]
- Variables, expressions, and statements [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, 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

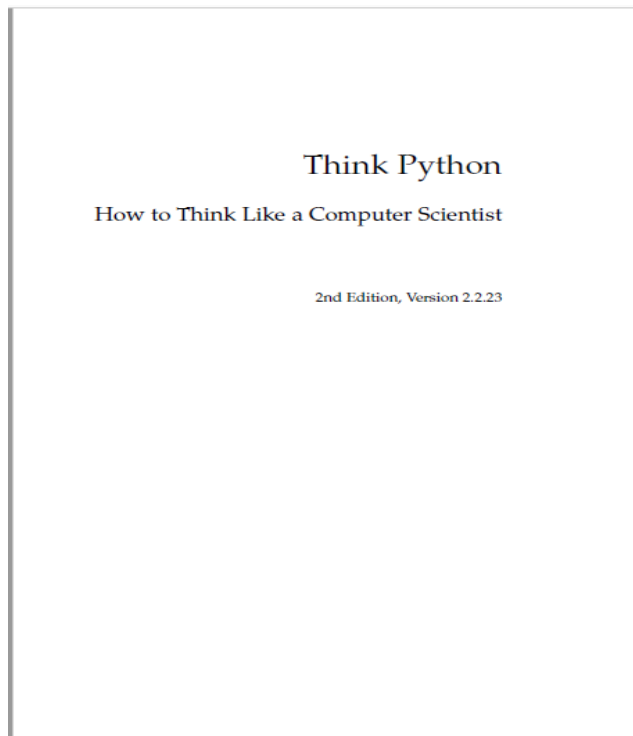
Cody Jackson

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]

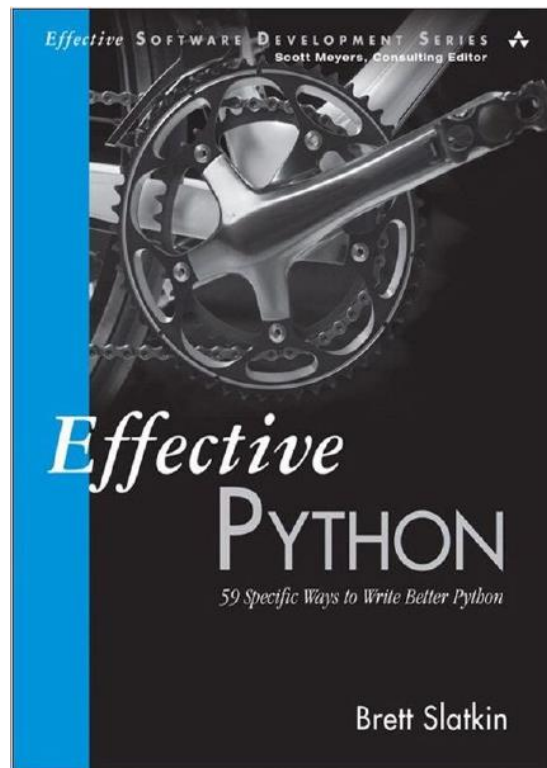


Think Python, Allen B. Downey

Effective Python, Brett Slatkin



- Pythonic Thinking [001 - 025]
- 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]

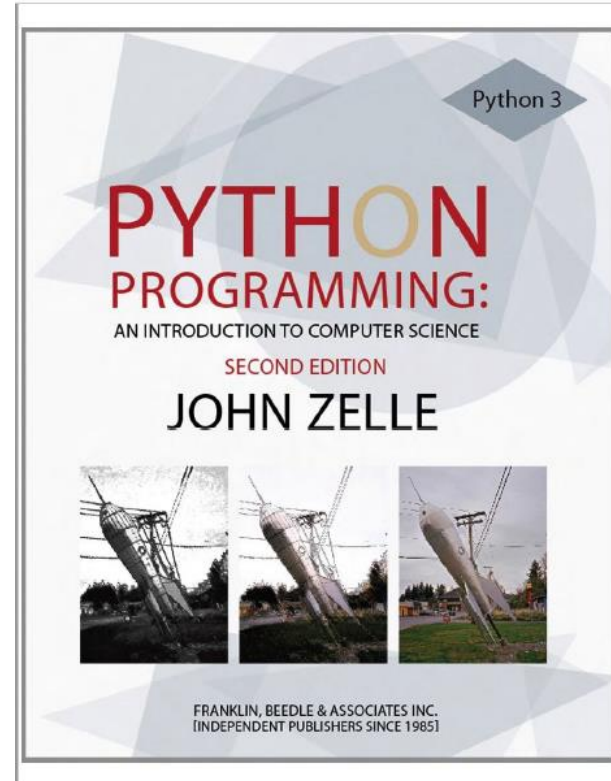


Effective Python, Brett Slatkin

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



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

Python YouTube Playlists



-
- | | | |
|---|--------|--------------------------------------|
| • Python Tutorial Videos Simplilearn
youtube.com/playlist?list=PLEiEAq2VkuUUKoW1o-A-VEmkoGKSC26i-I | [20 H] | Channel: Simplilearn |
| • Python Tutorial For Beginners Edureka
youtube.com/playlist?list=PL9ooVrP1hQOHY-BeYrKHDrHKphsJOyRyu | [45 H] | Channel: edureka! |
| • Python Tutorial for Beginners
youtube.com/playlist?list=PLS1QulWo1RIaJECMeUT4LFwJ-ghgoSH6n | [40 H] | Channel: ProgrammingKnowledge |
| • Python 3 Programming Tutorial
youtube.com/playlist?list=PLeo1K3hjS3usILfyyQlvUBokXkHPSve6S | [10 H] | Channel: codebasics |
| • Learning to program with Python 3
youtube.com/playlist?list=PLQVvva0QuDeAams7fkdcwOGBpGdHpXln | [3 H] | Channel: sentdex |
| • Python 3 Tutorial for Beginners
youtube.com/playlist?list=PLS1QulWo1RIYt4e0WnBp-ZjCNq8X0FX0I | [30 H] | Channel: ProgrammingKnowledge |
| • Complete Python Training Course
youtube.com/playlist?list=PLtbZLf-cj_AWhtJE6Rb5oWf02RC2qVU-I | [3 H] | Channel: Python Programmer |
| • Python Tutorial for Beginners
youtube.com/playlist?list=PLsyeobzWxl7pol9JTVyndKe62ieoN-MZ3 | [20 H] | Channel: Telusko |
| • Complete Python tutorial in Hindi (2020)
youtube.com/playlist?list=PLwgFb6VsUj_IQTpQKDtLXKXEIQychT_Zj | [20 H] | Channel: Harshit vashisth |
-

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



jetbrains.com/pycharm

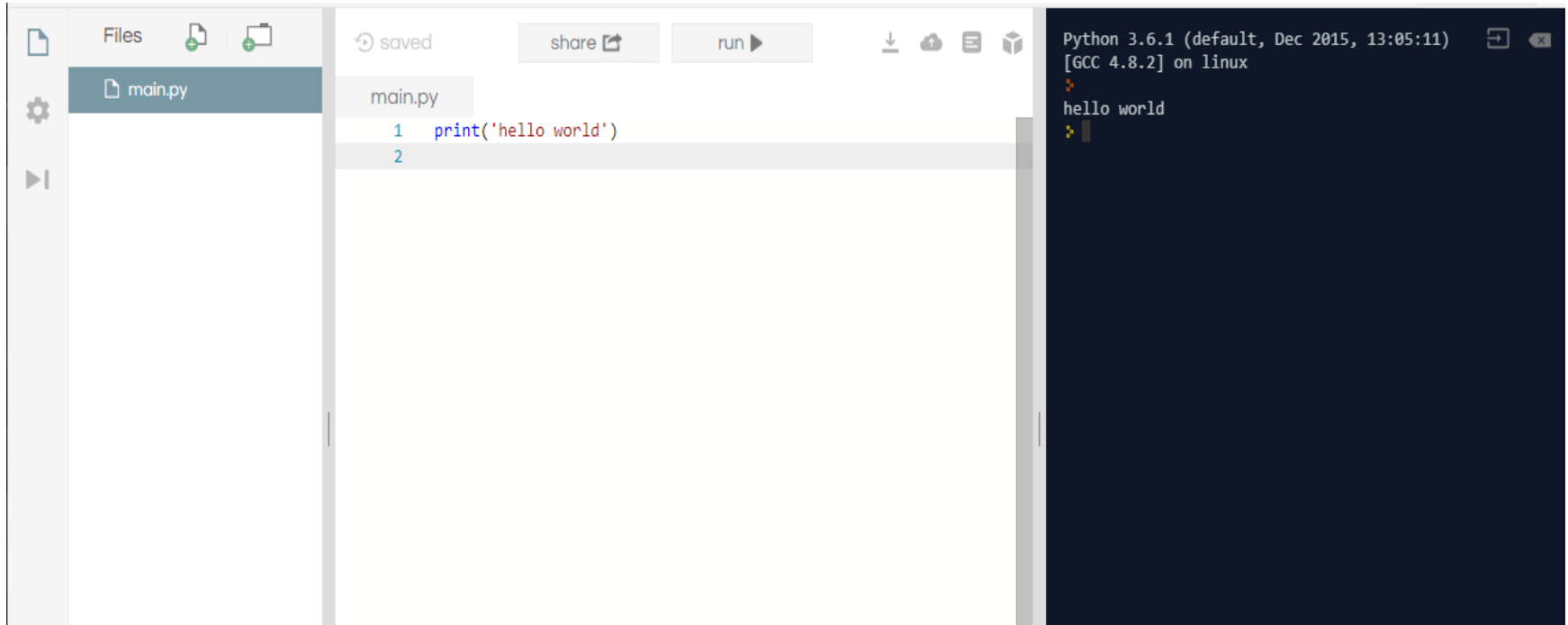


spyder-ide.org

Python Online Interpreters



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

A screenshot of the repl.it online Python interpreter interface. The interface is divided into three main sections. On the left is a sidebar with a 'Files' tab, a gear icon for settings, and a play button icon. The 'Files' tab shows a file named 'main.py'. The middle section is a code editor with a light gray background, showing the code '1 print('hello world')' and '2'. Above the code editor are buttons for 'saved', 'share', and 'run'. On the right is a dark-themed terminal window. The terminal shows the prompt 'Python 3.6.1 (default, Dec 2015, 13:05:11) [GCC 4.8.2] on linux', followed by the command 'hello world' and its output 'hello world'.

```
Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
>
hello world
>
```

Python Online Interpreters



ideone online interpreter: ideone.com

ideone.com

</> source code close shortcuts fullscreen ↗

```
1 print('hello world')
2
```

input Output clear the output ☒ syntax highlight

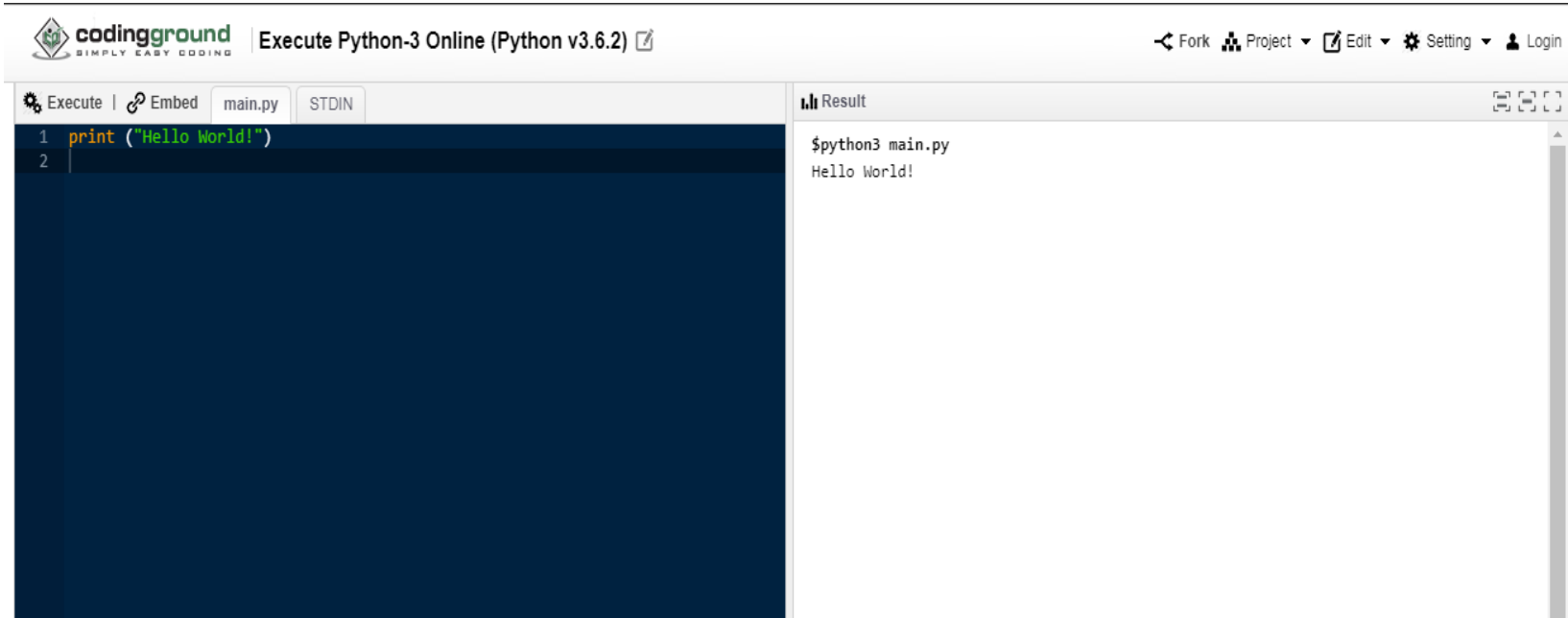
Success #stdin #stdout 0.04s 9216KB
hello world

save ideone it!

Python Online Interpreters



Tutorials point online interpreter: [tutorialspoint.com/execute_python3_online.php](https://www.tutorialspoint.com/execute_python3_online.php)

A screenshot of the CodingGround online Python interpreter interface. The header shows the "codingground" logo and the text "Execute Python-3 Online (Python v3.6.2)". On the right, there are links for "Fork", "Project", "Edit", "Setting", and "Login". Below the header, there are tabs for "Execute", "Embed", "main.py", and "STDIN". The "main.py" tab is active, showing a code editor with two lines of Python code:

```
1 print ("Hello World!")
2
```

 To the right of the code editor is a "Result" panel showing the output of the code:

```
$python3 main.py
Hello World!
```

Python IDE Installation



python.org

- 1 Go to [Download Python](#) page on the official site and click **Download Python 3.6.0** (You may see different version name).
- 2 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.
- 3 Open IDLE, copy the following code below and press enter.

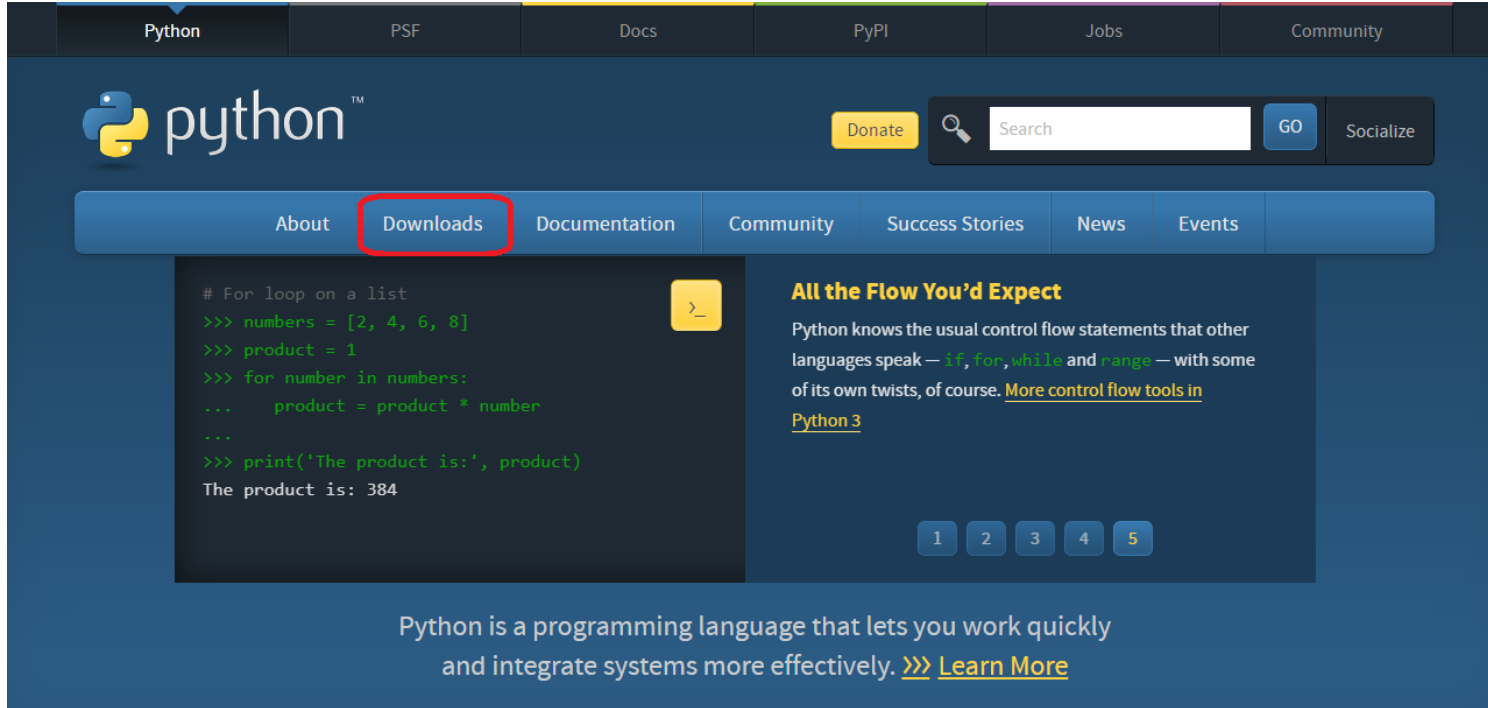
```
print("Hello, World!")
```

- 4 To create a file in IDLE, go to **File > New Window** (Shortcut: **Ctrl+N**).
- 5 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!")
```

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

Python IDE Installation

A screenshot of the Python.org homepage. The top navigation bar includes links for Python, PSF, Docs, PyPI, Jobs, and Community. Below this is a dark blue header with the Python logo, a "Donate" button, a search bar with a "GO" button, and a "Socialize" button. A secondary navigation bar contains links for About, Downloads (highlighted with a red rectangle), Documentation, Community, Success Stories, News, and Events. The main content area features a code snippet on the left, a yellow terminal icon, and an article titled "All the Flow You'd Expect" on the right. At the bottom, a blue banner contains the text "Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)".

Python

PSF

Docs

PyPI

Jobs

Community

python™

Donate

Search

GO

Socialize

About

Downloads

Documentation

Community

Success Stories

News

Events

```
# For loop on a list
>>> numbers = [2, 4, 6, 8]
>>> product = 1
>>> for number in numbers:
...     product = product * number
...
>>> print('The product is:', product)
The product is: 384
```

All the Flow You'd Expect

Python knows the usual control flow statements that other languages speak — *if*, *for*, *while* and *range* — with some of its own twists, of course. [More control flow tools in Python 3](#)

1 2 3 4 5

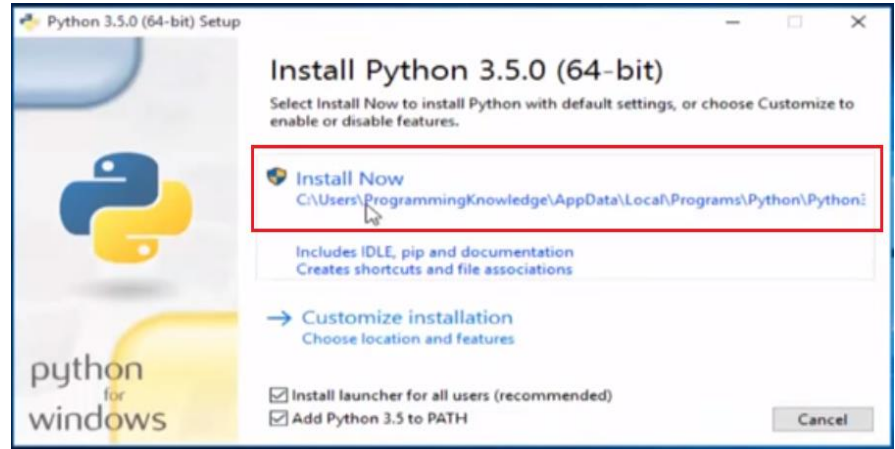
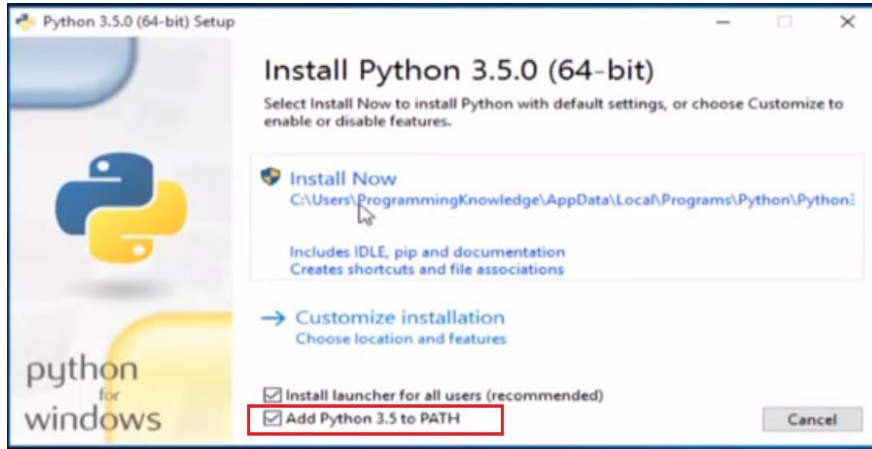
Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)

Python IDE Installation

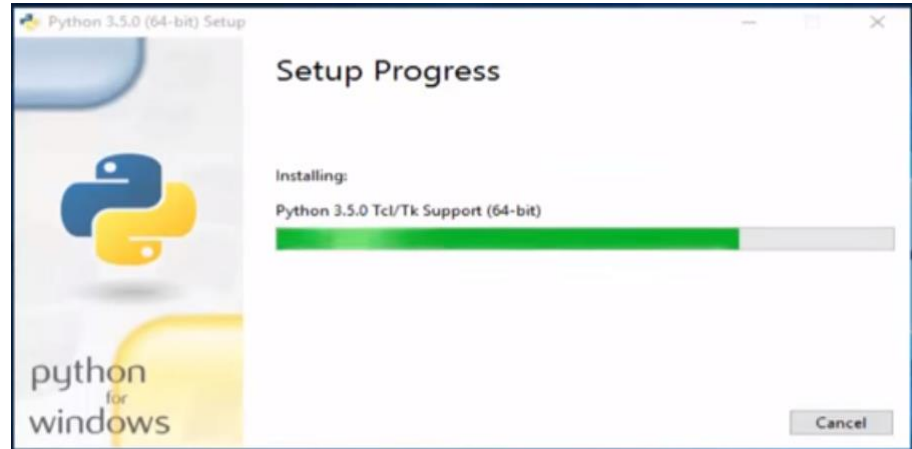
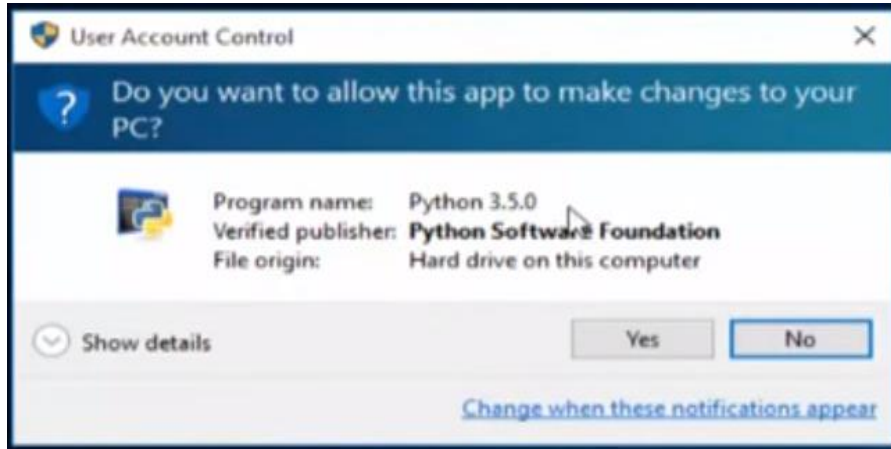


The screenshot shows the Python.org homepage. The top navigation bar includes links for Python, PSF, Docs, PyPI, Jobs, and Community. Below this is a secondary navigation bar with links for About, Downloads, Documentation, Community, Success Stories, News, and Events. The 'Downloads' menu is open, showing a list of options: All releases, Source code, Windows, Mac OS X, Other Platforms, License, and Alternative Implementations. The 'Windows' option is highlighted, and a sub-menu is displayed showing 'Python 3.8.1' as the selected version, which is circled in red. To the right of the sub-menu, there is a note stating: 'Note that Python 3.5+ cannot be used on Windows XP or earlier. Not the OS you are looking for? Python can be used on many operating systems and environments. View the full list of downloads.' The background of the website features a dark blue gradient with the Python logo and the text 'python™'. At the bottom, there is a statement: 'Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)'.

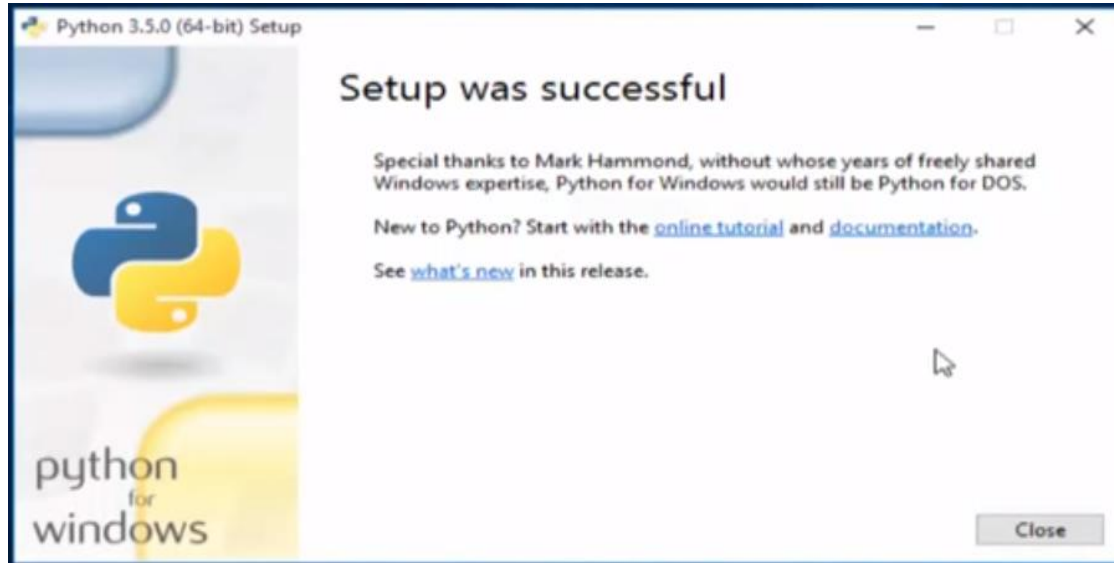
Python IDE Installation



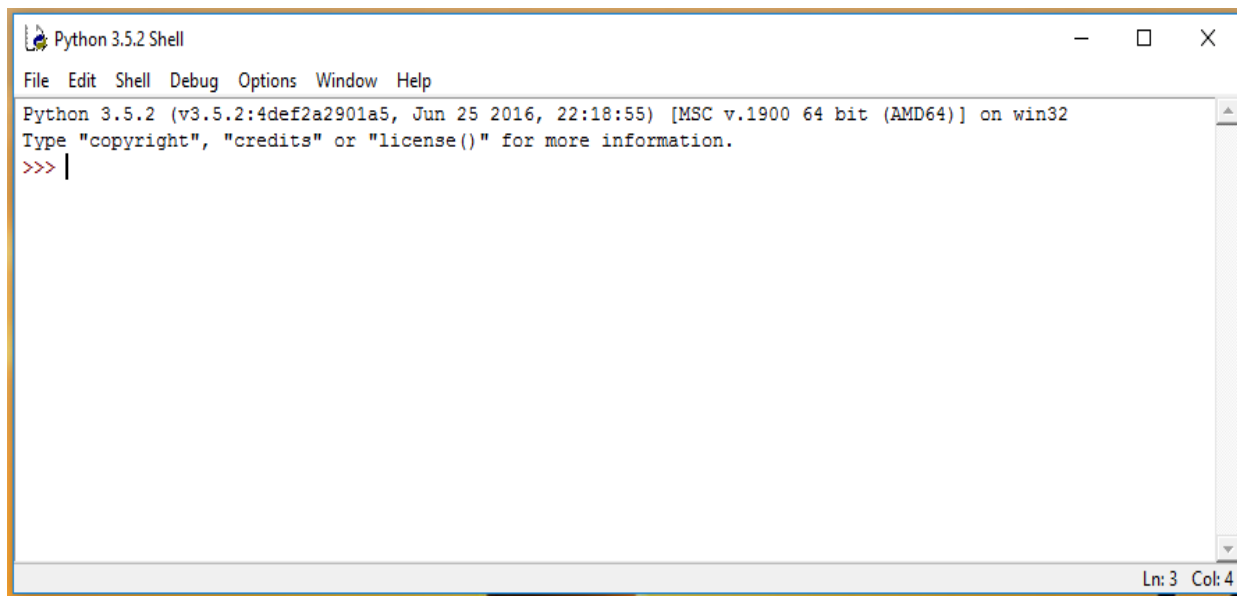
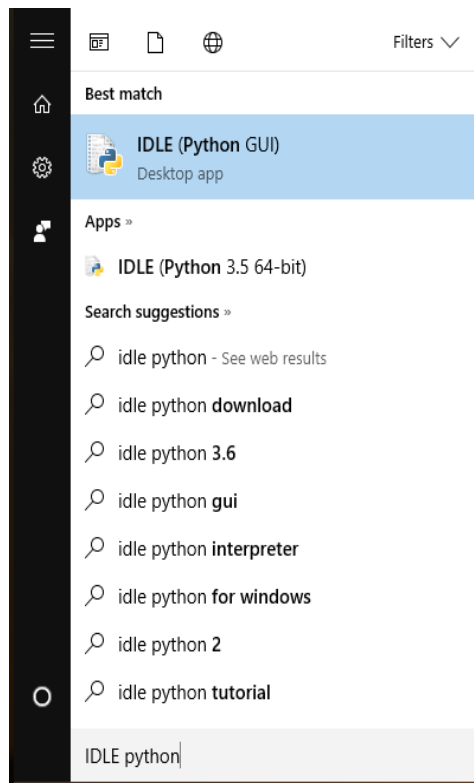
Python IDE Installation



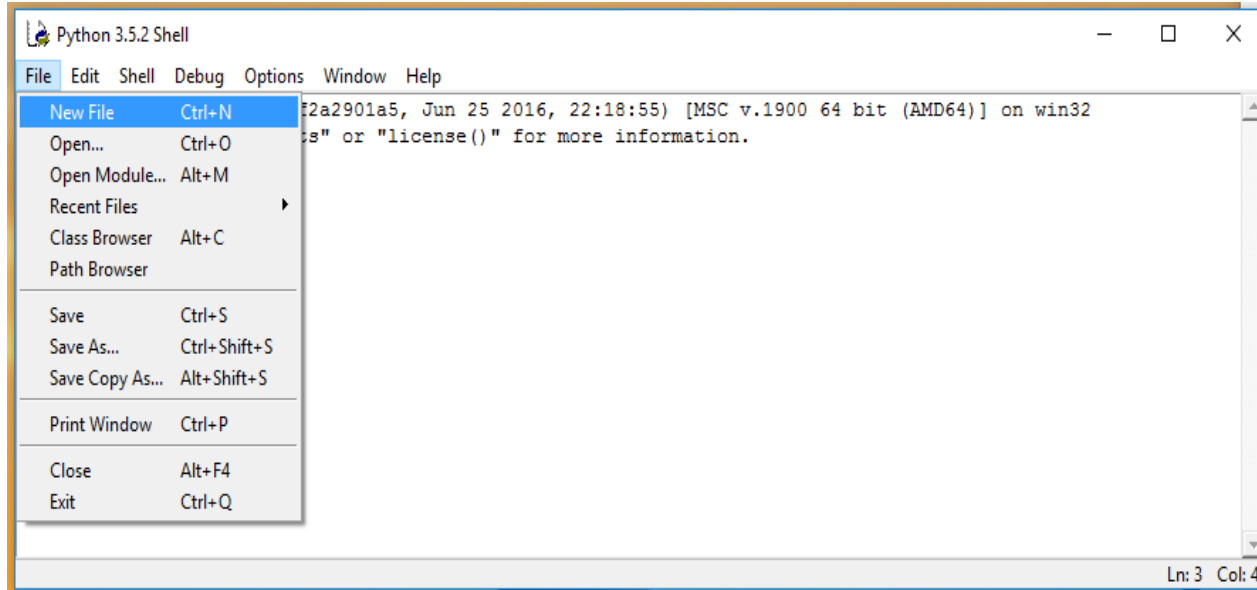
Python IDE Installation



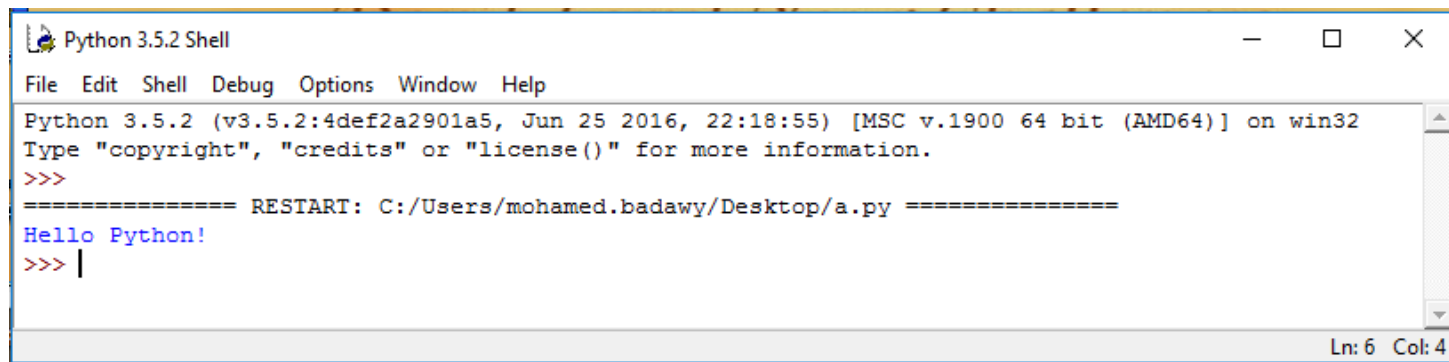
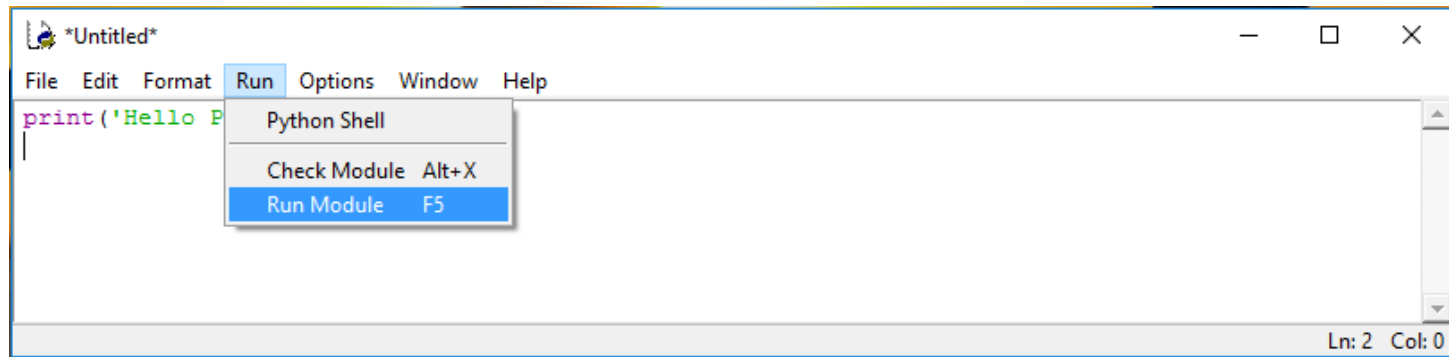
Python IDE Installation



Python IDE Installation



Python IDE Installation



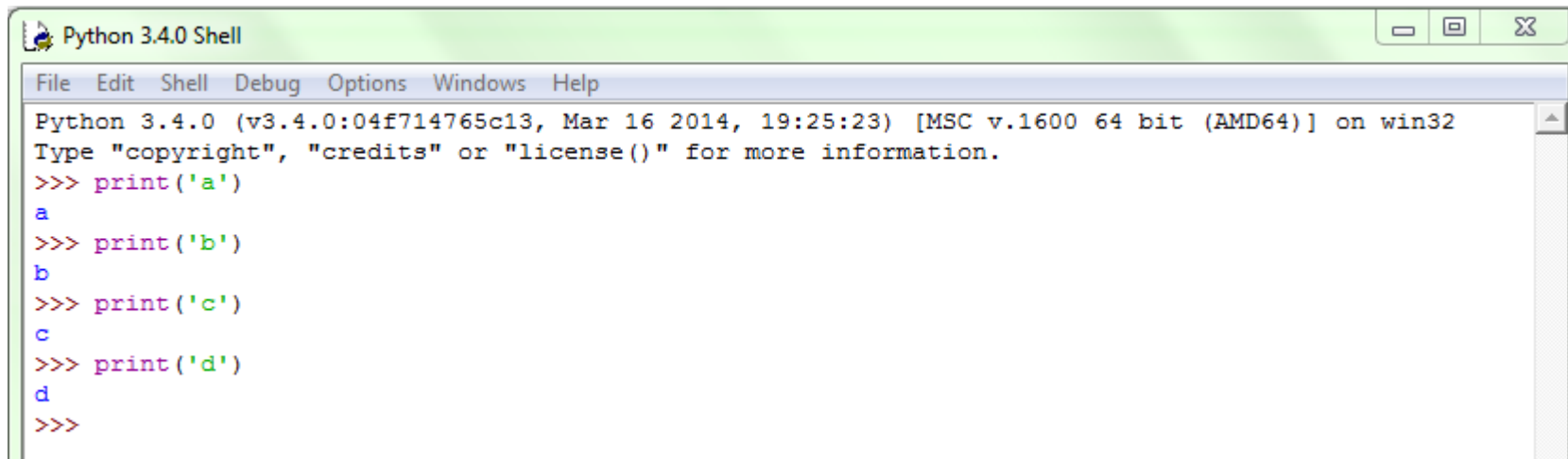
First Python Program



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



```
Python 3.4.0 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.0 (v3.4.0:04f714765c13, Mar 16 2014, 19:25:23) [MSC v.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print('a')
a
>>> print('b')
b
>>> print('c')
c
>>> print('d')
d
>>>
```

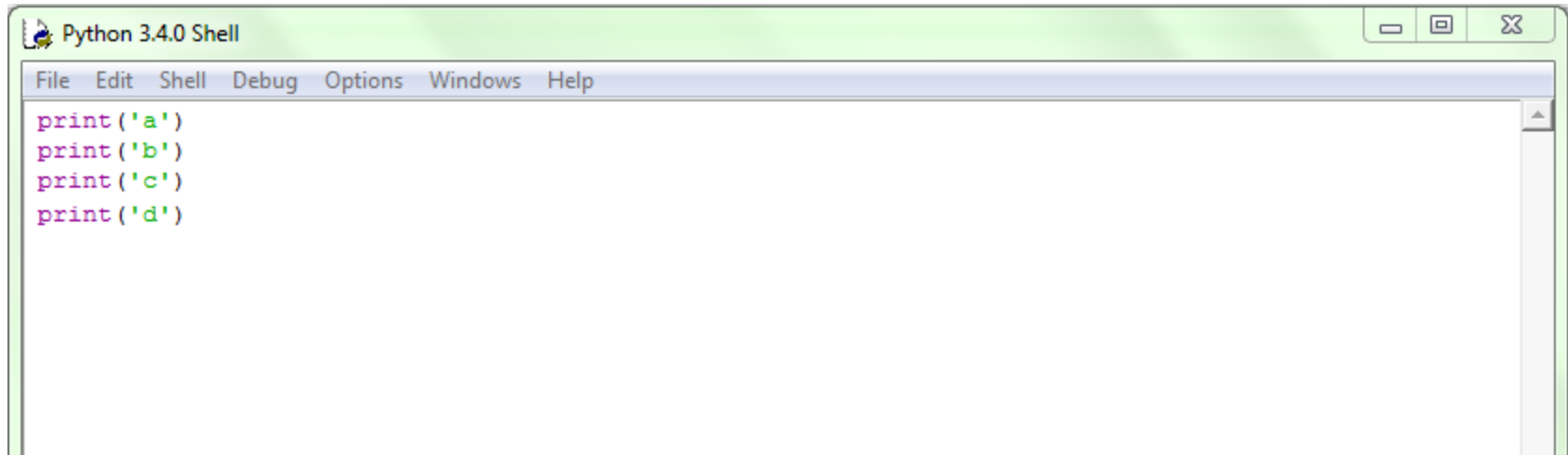
First Python Program



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.

A screenshot of a Python 3.4.0 Shell window. The window has a green title bar with the text "Python 3.4.0 Shell" and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with the following items: File, Edit, Shell, Debug, Options, Windows, and Help. The main area of the window is a text editor with a white background and a vertical scrollbar on the right. It contains four lines of Python code, each on a new line:

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

The code is color-coded: 'print' is in purple, and the string literals are in green.

Python Anaconda Installation

[Products](#)[Why Anaconda?](#)[Solutions](#)[Resources](#)[Company](#)[Contact Us](#)[Download](#)

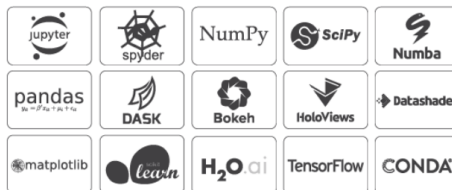
Anaconda Distribution

The World's Most Popular Python/R Data Science Platform

[Download](#)

The open-source [Anaconda Distribution](#) is the easiest way to perform Python/R data science and machine learning on Linux, Windows, and Mac OS X. With over 19 million users worldwide, it is the industry standard for developing, 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 [scikit-learn](#), [TensorFlow](#), and [Theano](#)
- Analyze data with scalability and performance with [Dask](#), [NumPy](#), [pandas](#), and [Numba](#)
- Visualize results with [Matplotlib](#), [Bokeh](#), [Datashader](#), and [HoloViews](#)






ANACONDA®

anaconda.com/distribution

Python Anaconda Installation



 Windows |  macOS |  Linux

Anaconda 2019.10 for Windows Installer

Python 3.7 version

Download

64-Bit Graphical Installer (462 MB)
32-Bit Graphical Installer (410 MB)

Python 2.7 version

Download

64-Bit Graphical Installer (413 MB)
32-Bit Graphical Installer (356 MB)

Get Started with Anaconda Distribution

Documentation
Installation and user guide for Anaconda

Anaconda Blog
News, software releases, and developer best

Community Support
Solutions and knowledge from

Anaconda Webinars
Industry trends and tutorials from

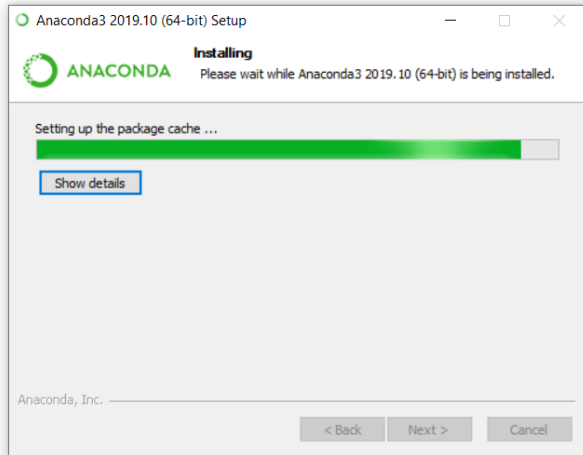
Anaconda Training
Learn Python for Data Science with DataCamp



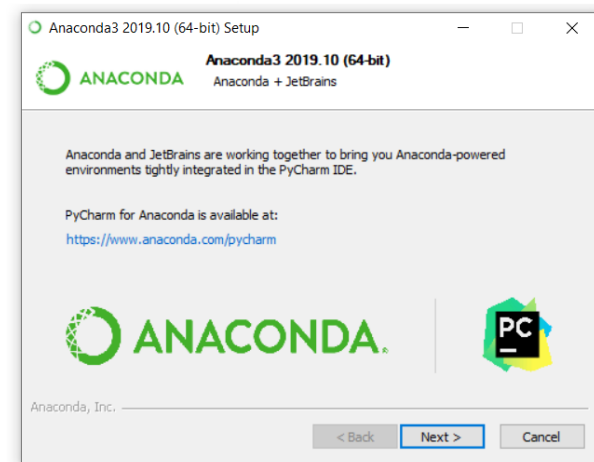
Python Anaconda Installation



Name	Date	Type	Size	Tags
Anaconda3-2019.10-Windows-x86_64.exe	2020-02-08 16:20	Application	472,622 KB	



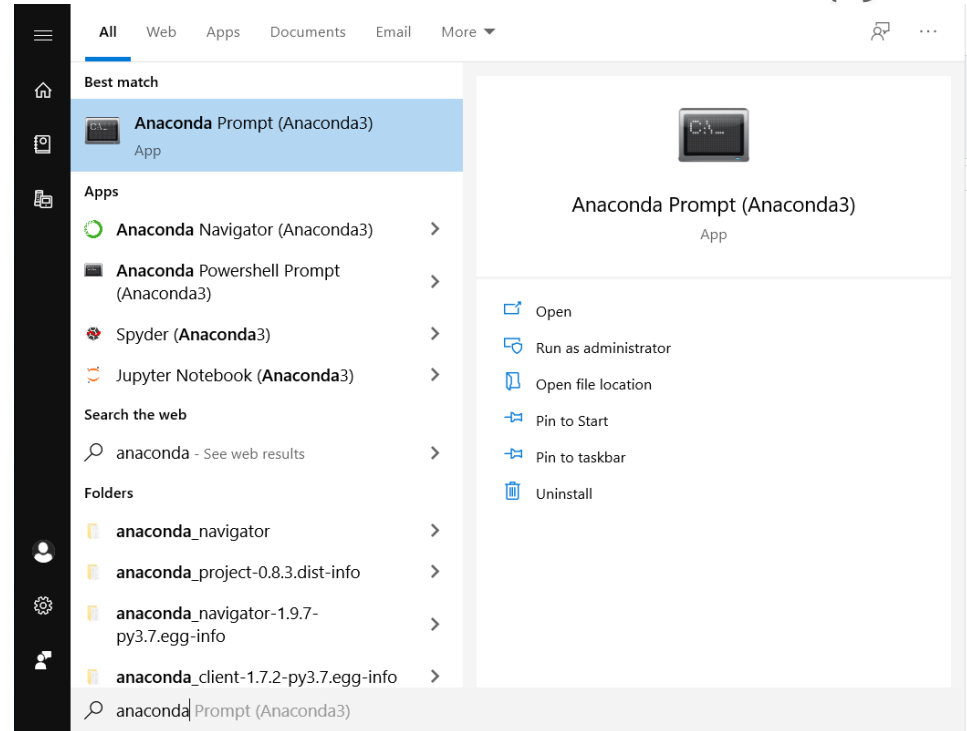
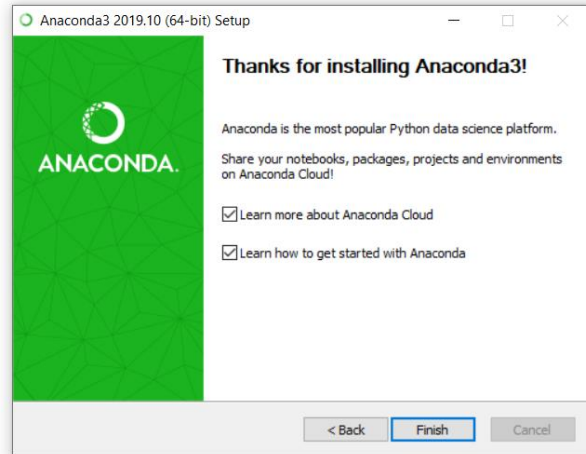
Name	Date	Type	Size	Tags
Anaconda3-2019.10-Windows-x86_64.exe	2020-02-08 16:20	Application	472,622 KB	



Python Anaconda Installation



Name	Date	Type	Size	Tags
Anaconda3-2019.10-Windows-x86_64.exe	2020-02-08 16:20	Application	472,622 KB	



Python Anaconda Installation



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:
```

```
#
```

#	Name	Version	Build	Channel
	_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	
	attrs	19.2.0	py_0	

```
(base) C:\Users\LenovoY540>python --version
```

```
Python 3.7.4
```

```
(base) C:\Users\LenovoY540>where python
```

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

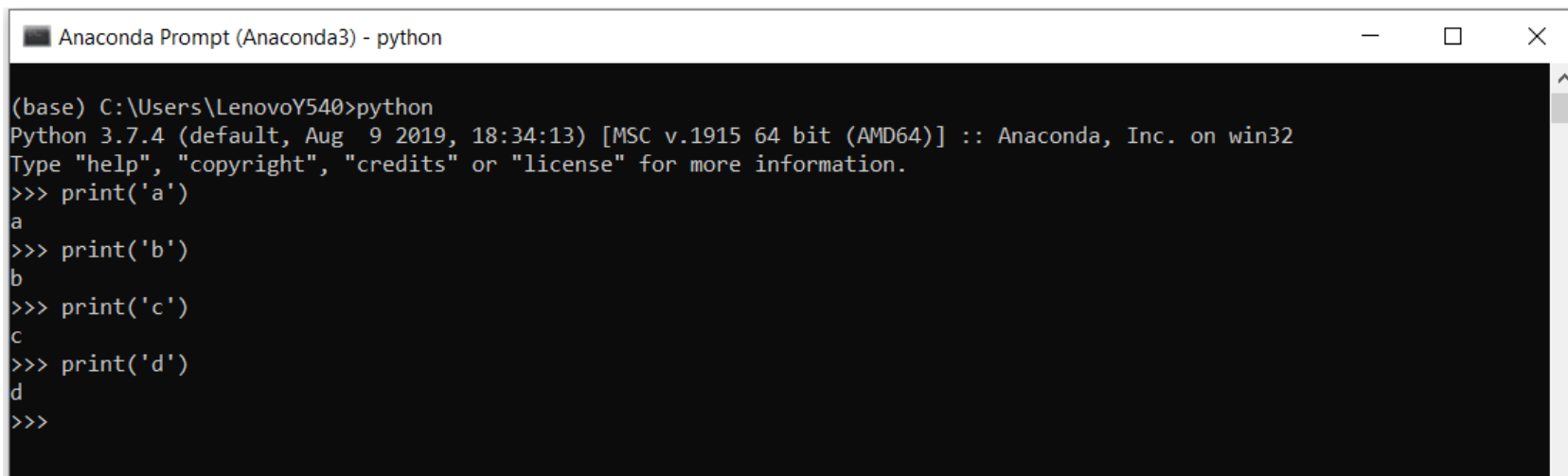
First Python Program



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
>>>
```

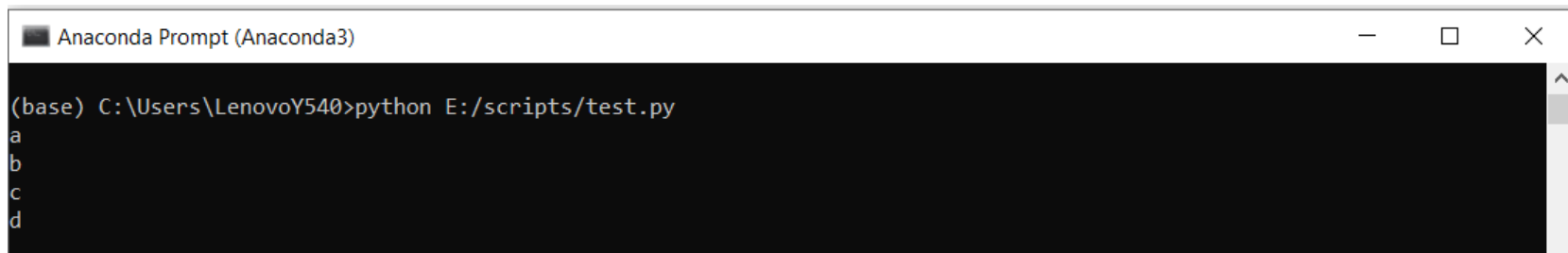
First Python Program



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





DO
MORE.