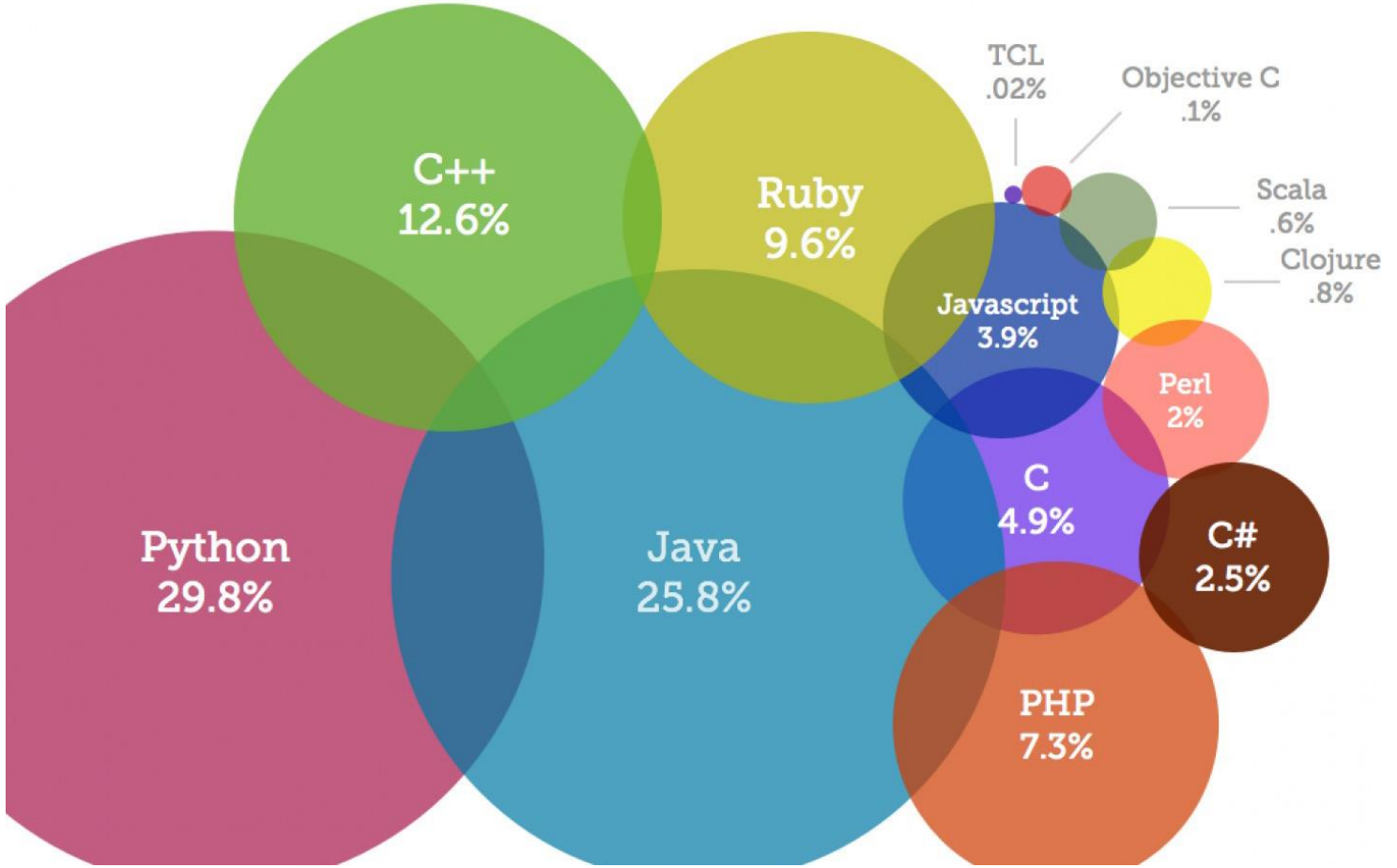
Python for Selenium

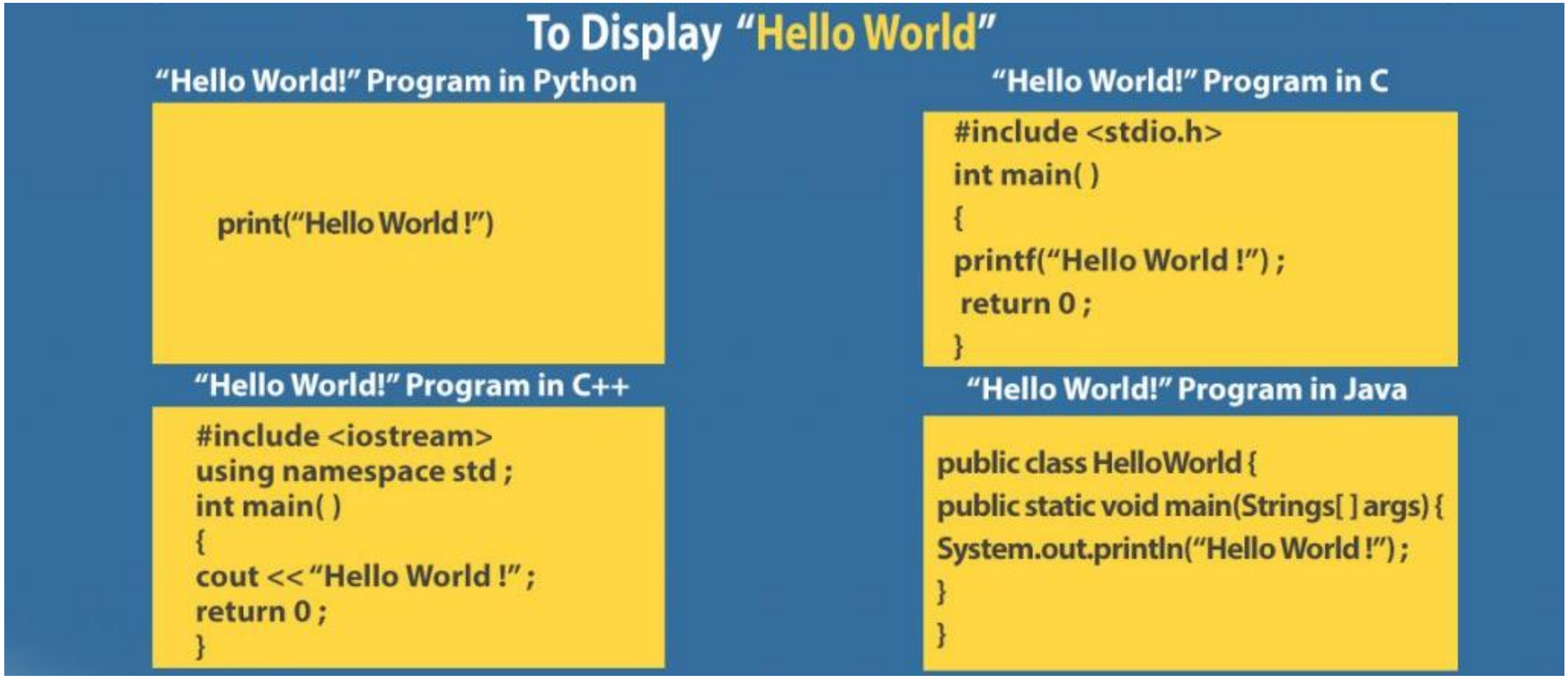
Introduction to Python

- Python is high level object oriented scripting language.
- Python is open source language.
- Platform independent

Most popular languages in 2019



Python Comparison with other languages

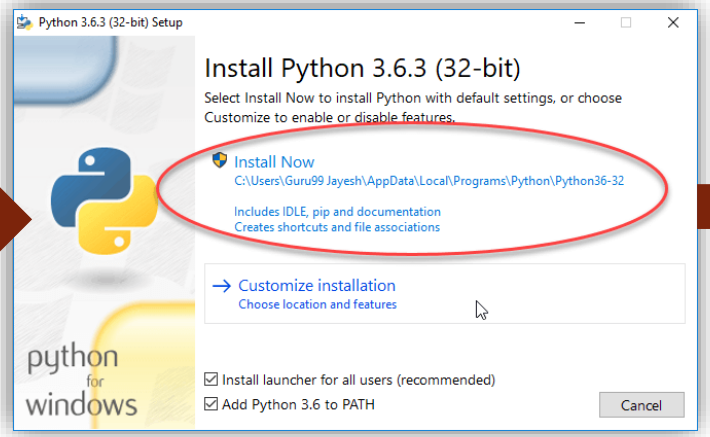


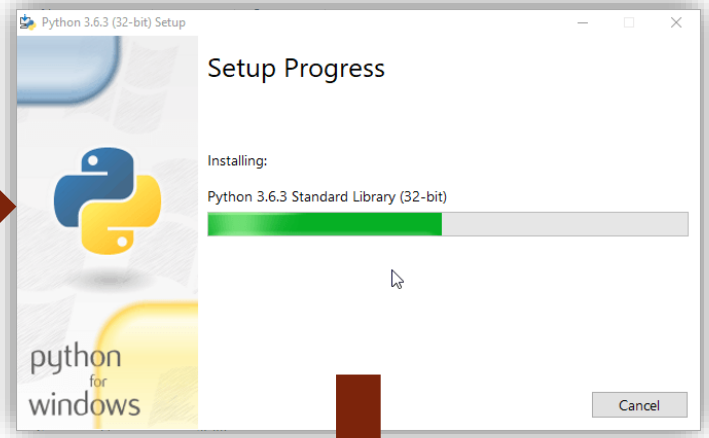
History

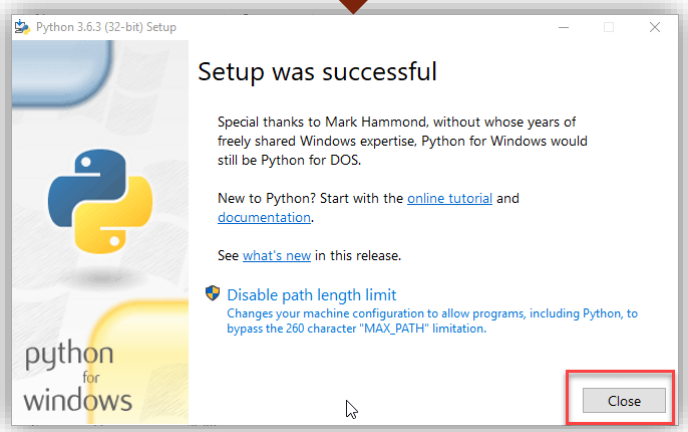
- The implementation of Python was started in the December 1989 by **Guido Van Rossum** at CWI in Netherland.
- In February 1991, van Rossum published the code (labeled version 0.9.0).
- In 1994, Python 1.0 was released with new features like: lambda, map, filter, and reduce.
- Python 2.0 added new features like: list comprehensions, garbage collection system.
- On December 3, 2008, Python 3.0 (also called "Py3K") was released.
- Python is influenced by :
  - ABC language.
  - Modula-3

Download and install Python on Windows

- https://www.python.org/downloads/



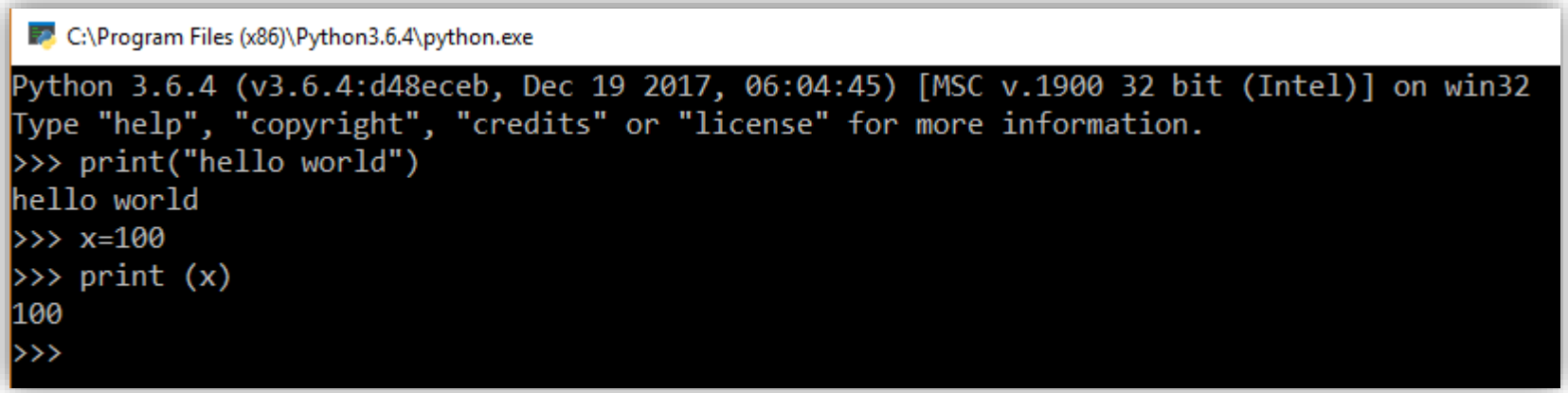




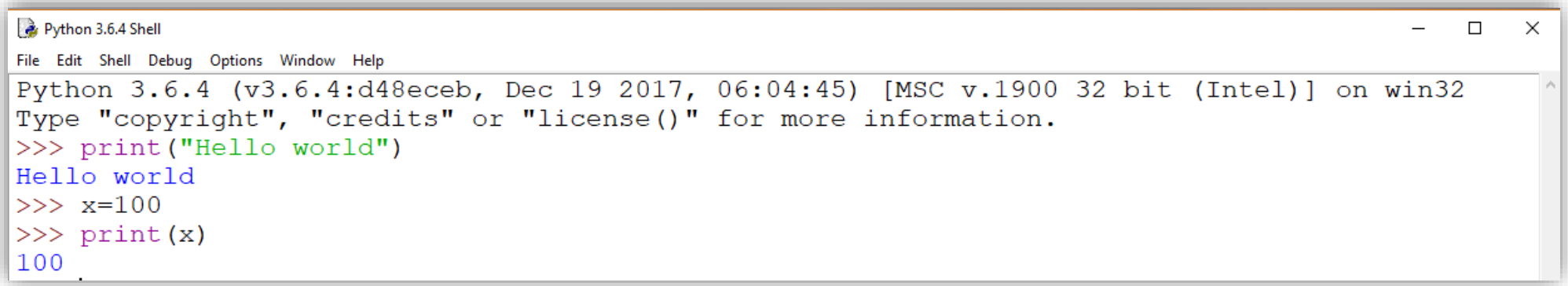

```
C:\Users\admin>python
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
```

Different ways to Run Python code

- 1) C:\Program Files (x86)\Python3.6.4\python.exe

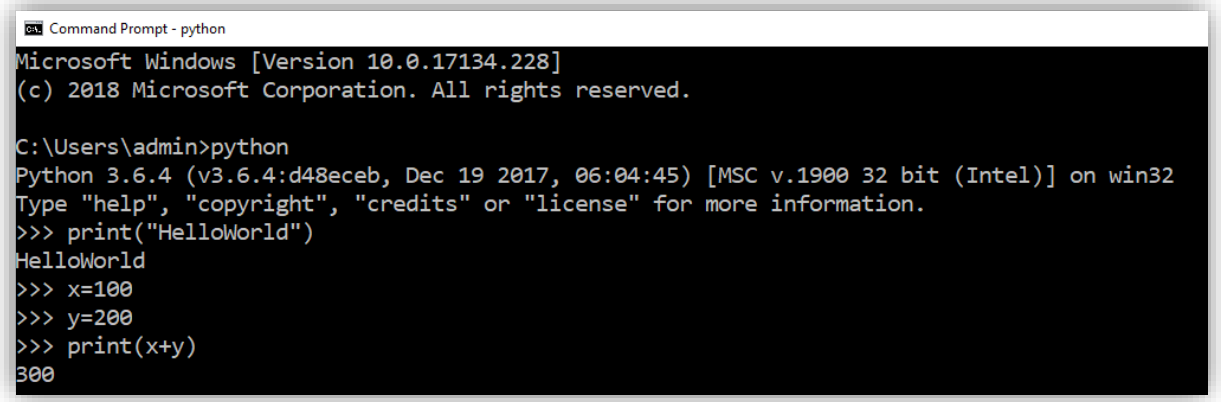


- 2) IDLE (Python shell) & New File

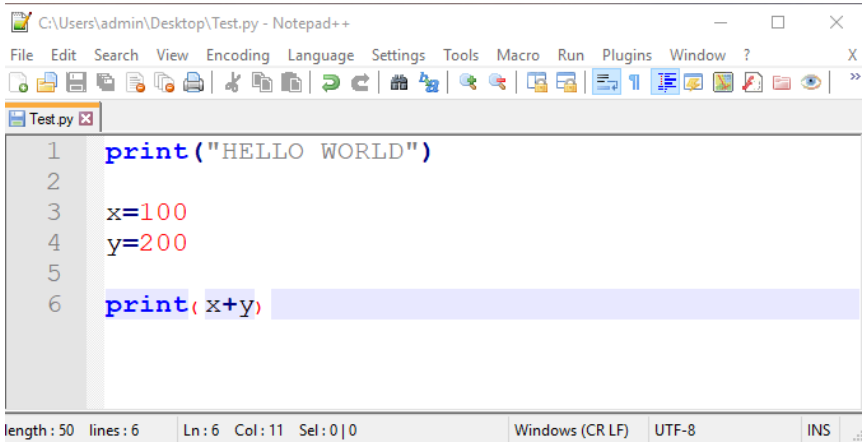


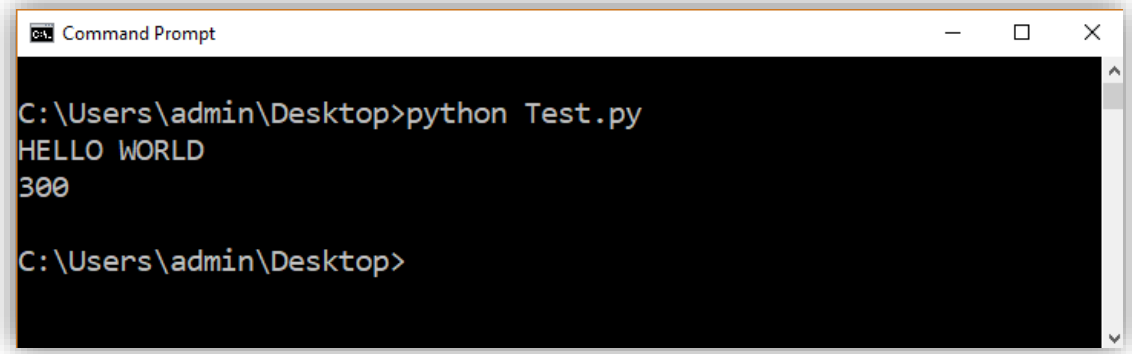
Different ways to Run Python code

- 3) Windows command prompt



- 4) Notepad++ (or) EditPlus & cmd



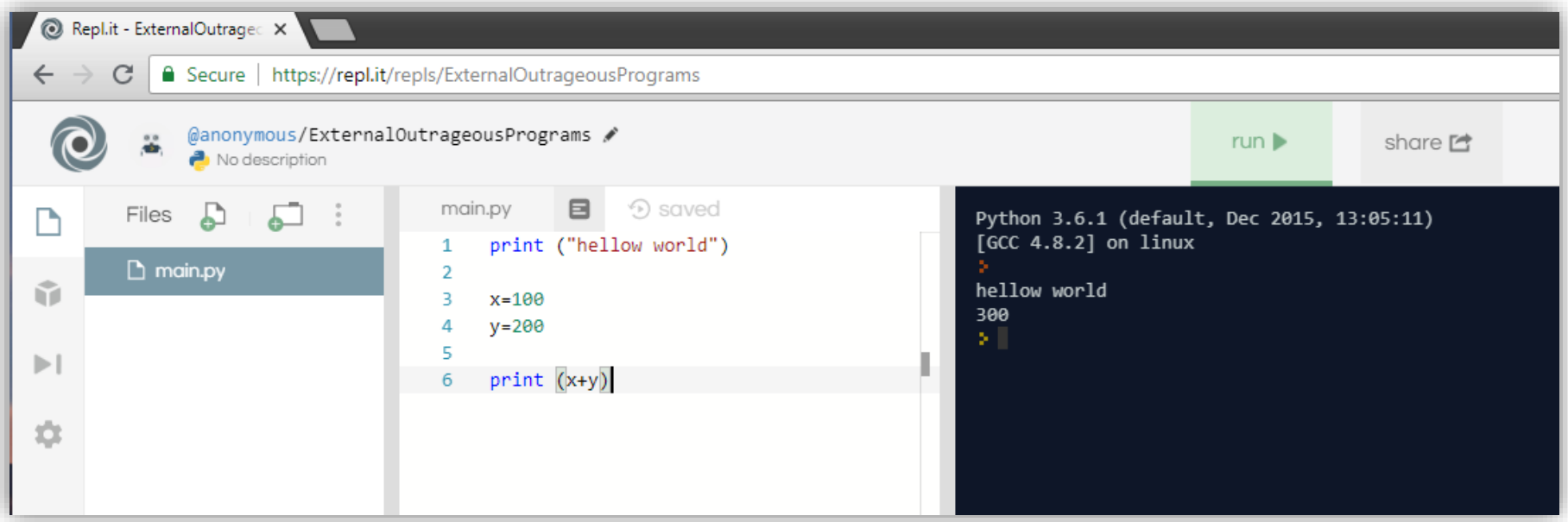


Different ways to Run Python code

- 5) Online compilers

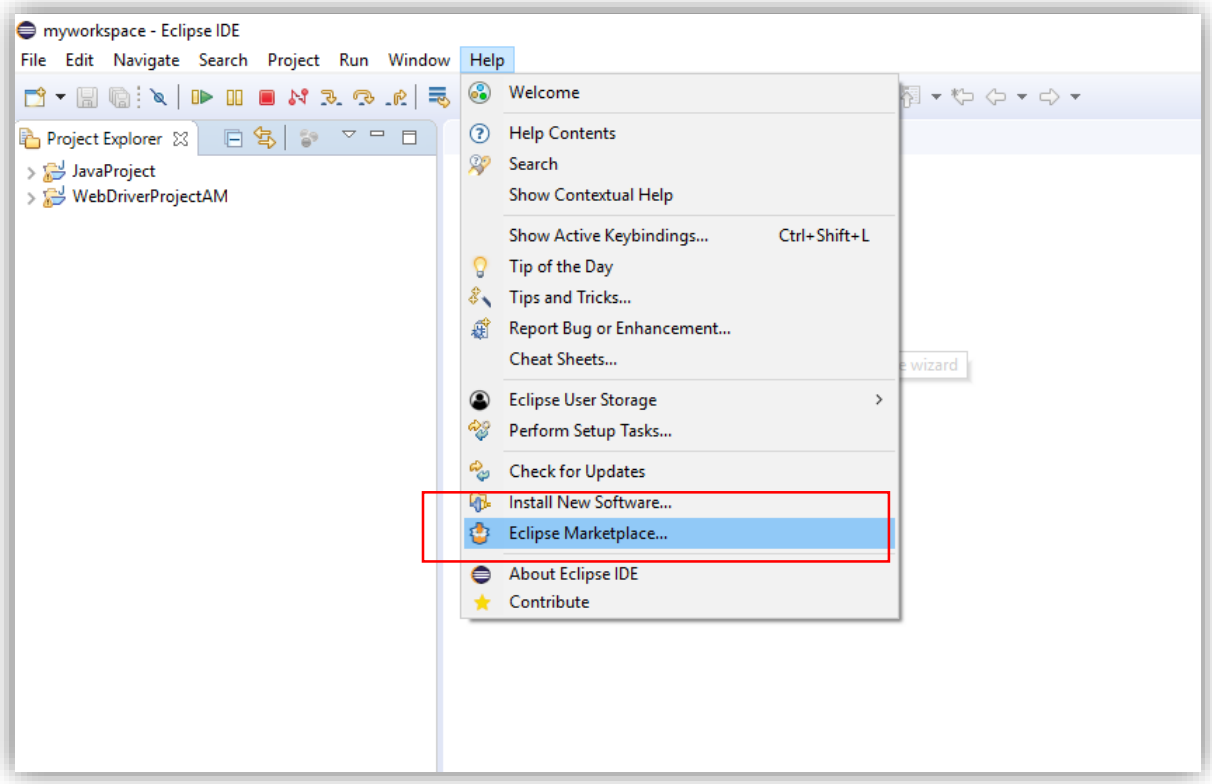
https://repl.it/repls/ExternalOutrageousPrograms

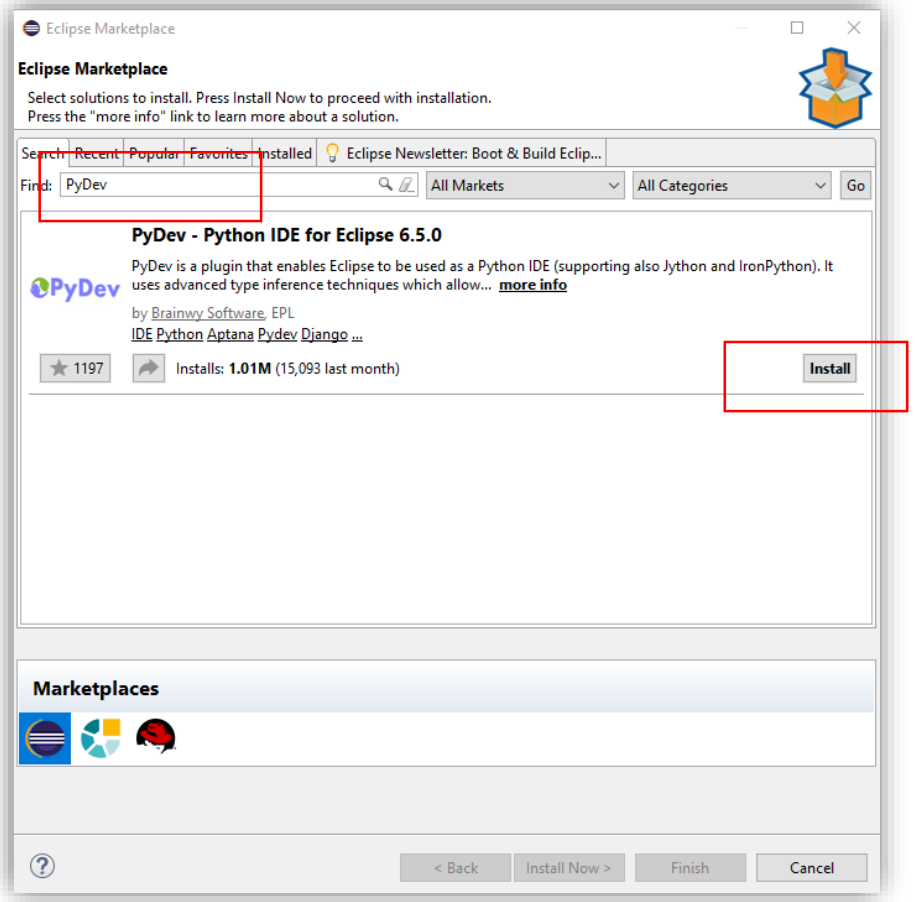
Many more….

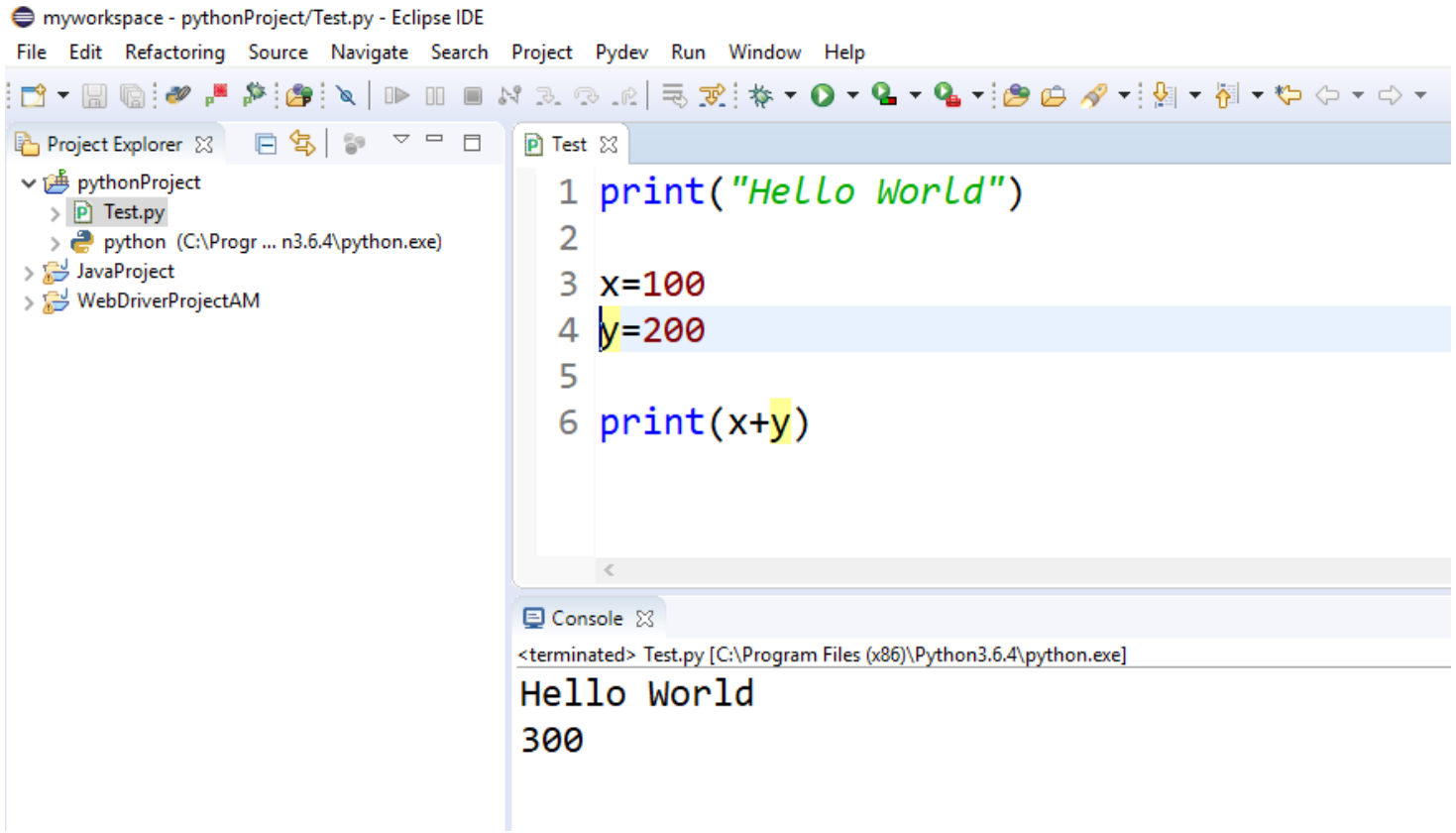


Different ways to Run Python code

- 7) Eclipse PyDev plug-in

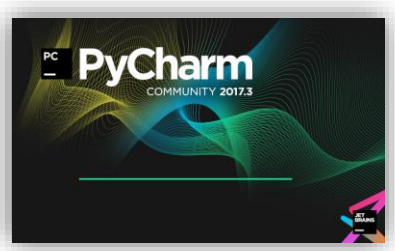


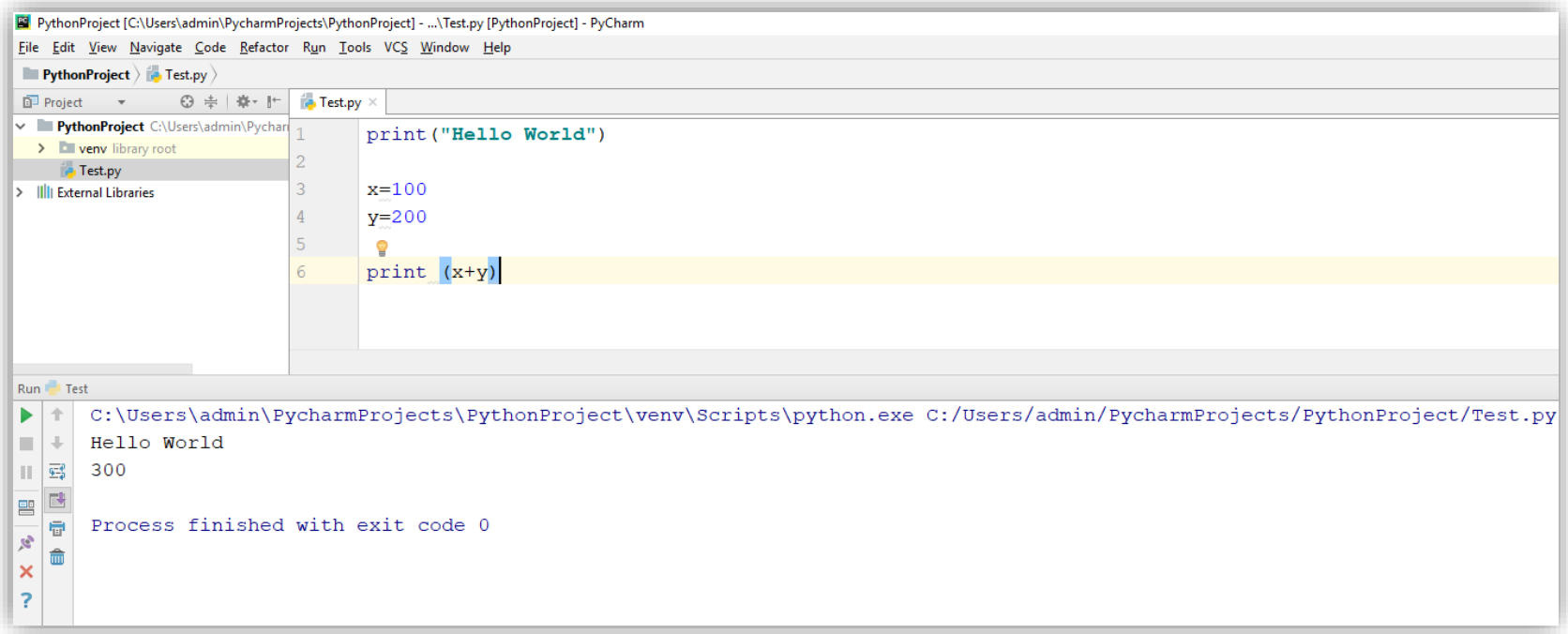



myworkspace - pythonProject/Test.py - Eclipse IDE
File Edit Refactoring Source Navigate Search Project Pydev Run Window Help
Project Explorer
pythonProject
Test.py
python (C:\Progr ... n3.6.4\python.exe)
JavaProject
WebDriverProjectAM
Test
1 print("Hello World")
2
3 x=100
4 y=200
5
6 print(x+y)
Console
<terminated> Test.py [C:\Program Files (x86)\Python3.6.4\python.exe]
Hello World
300

Different ways to Run Python code

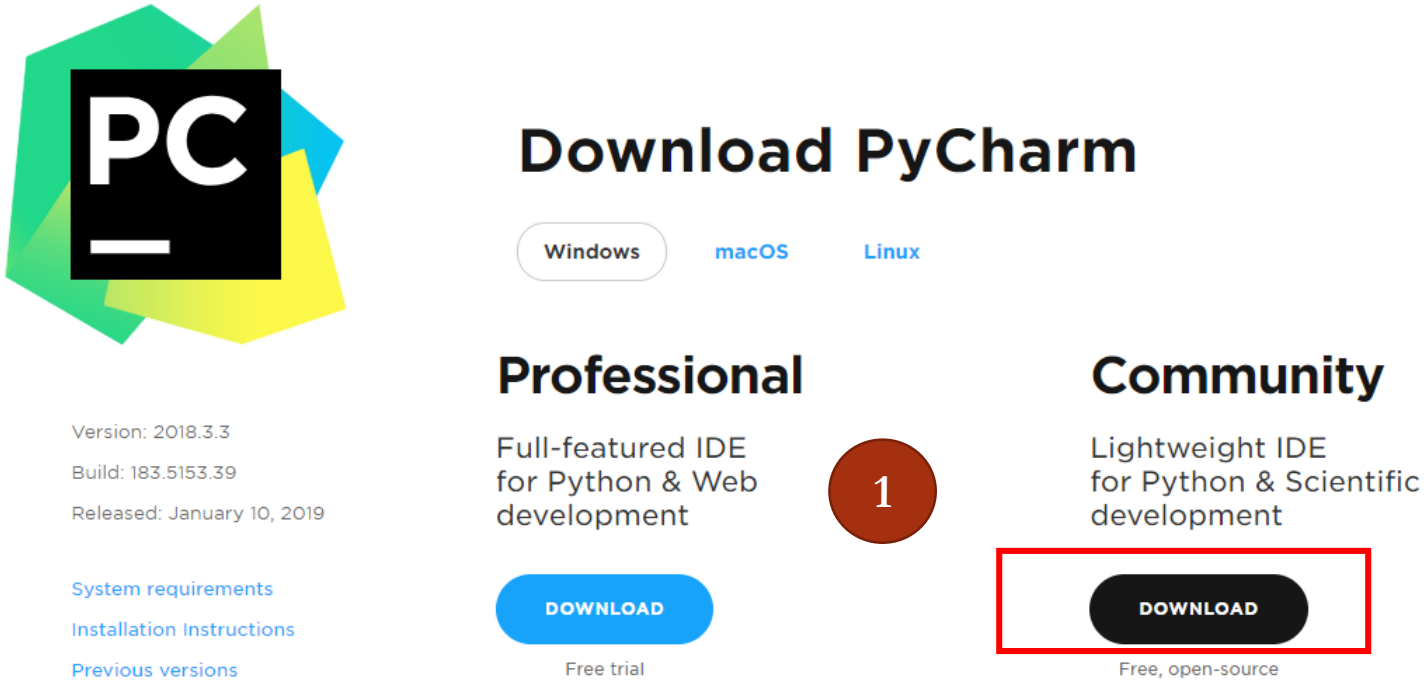
- 6) PyCharm IDE

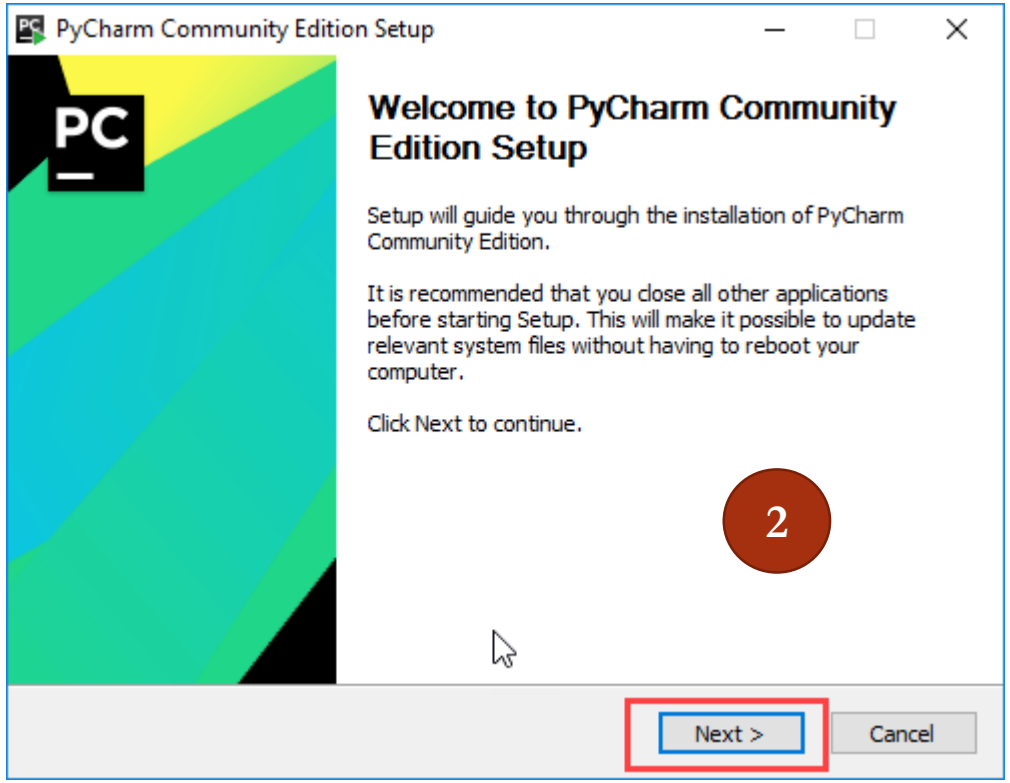


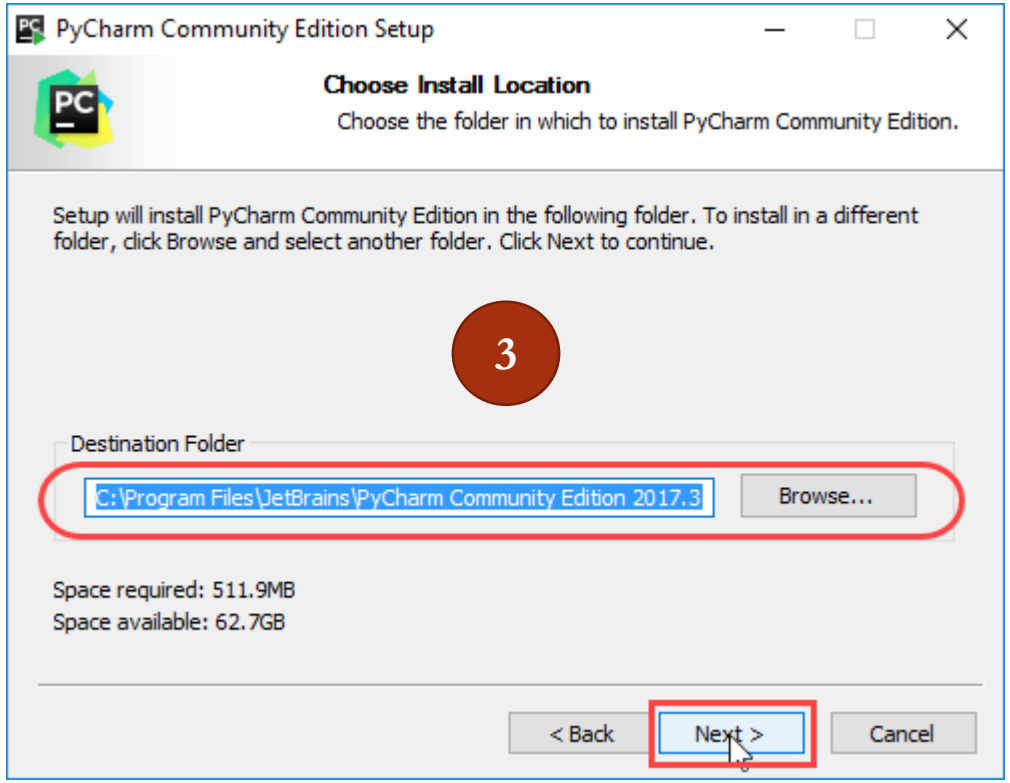


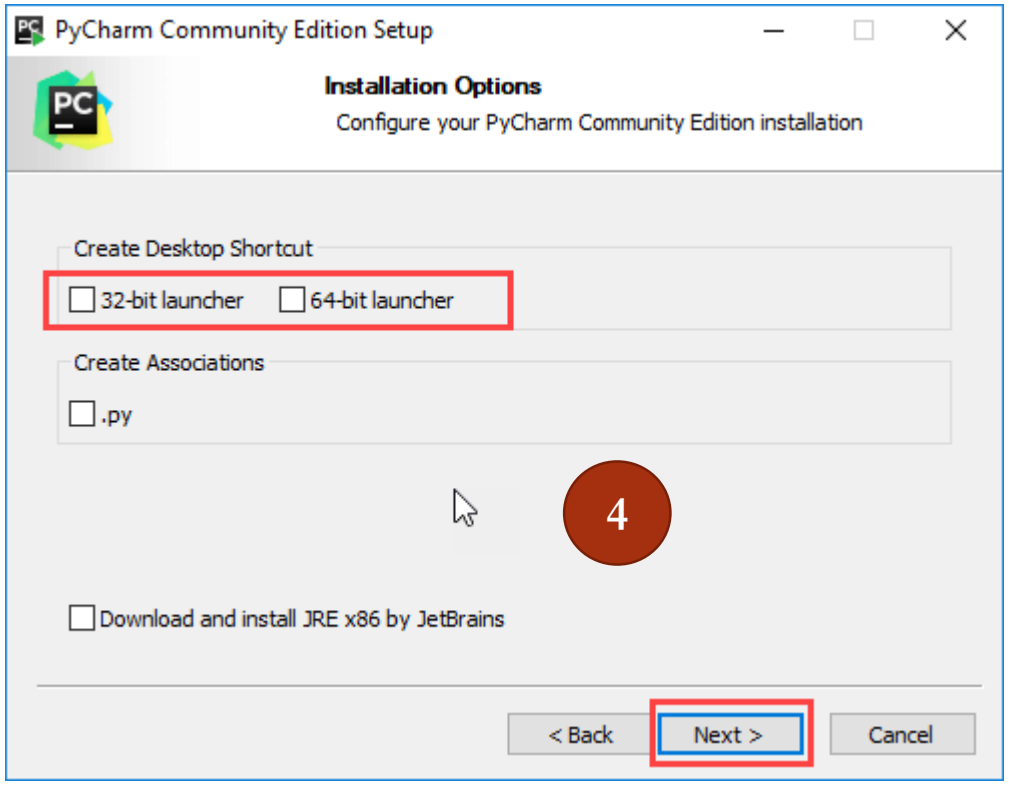
Install PyCharm IDE

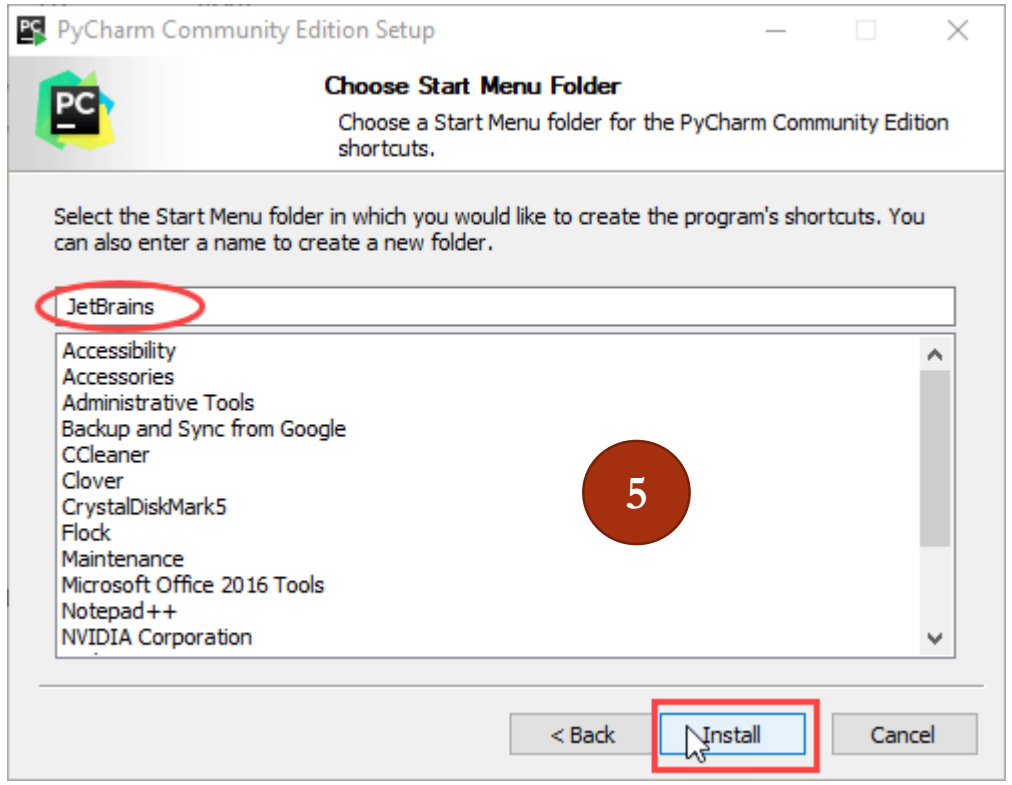
- https://www.jetbrains.com/

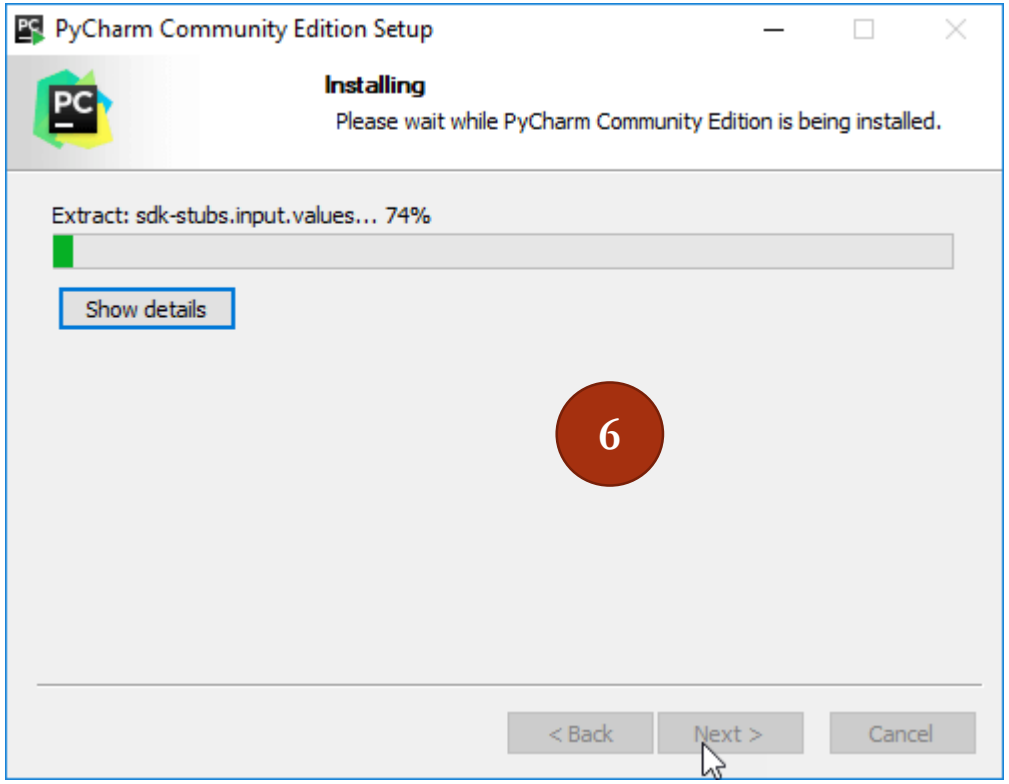


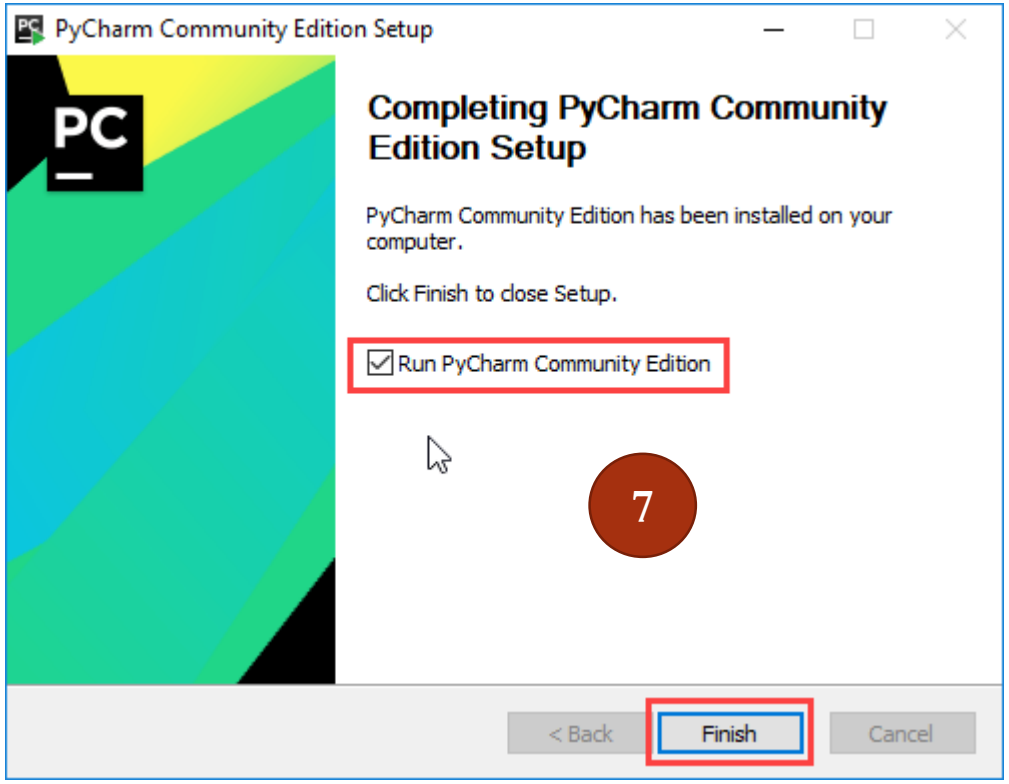

PyCharm Community Edition Setup
Welcome to PyCharm Community Edition Setup
Setup will guide you through the installation of PyCharm Community Edition.
It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.
Click Next to continue.
2
Next >
Cancel

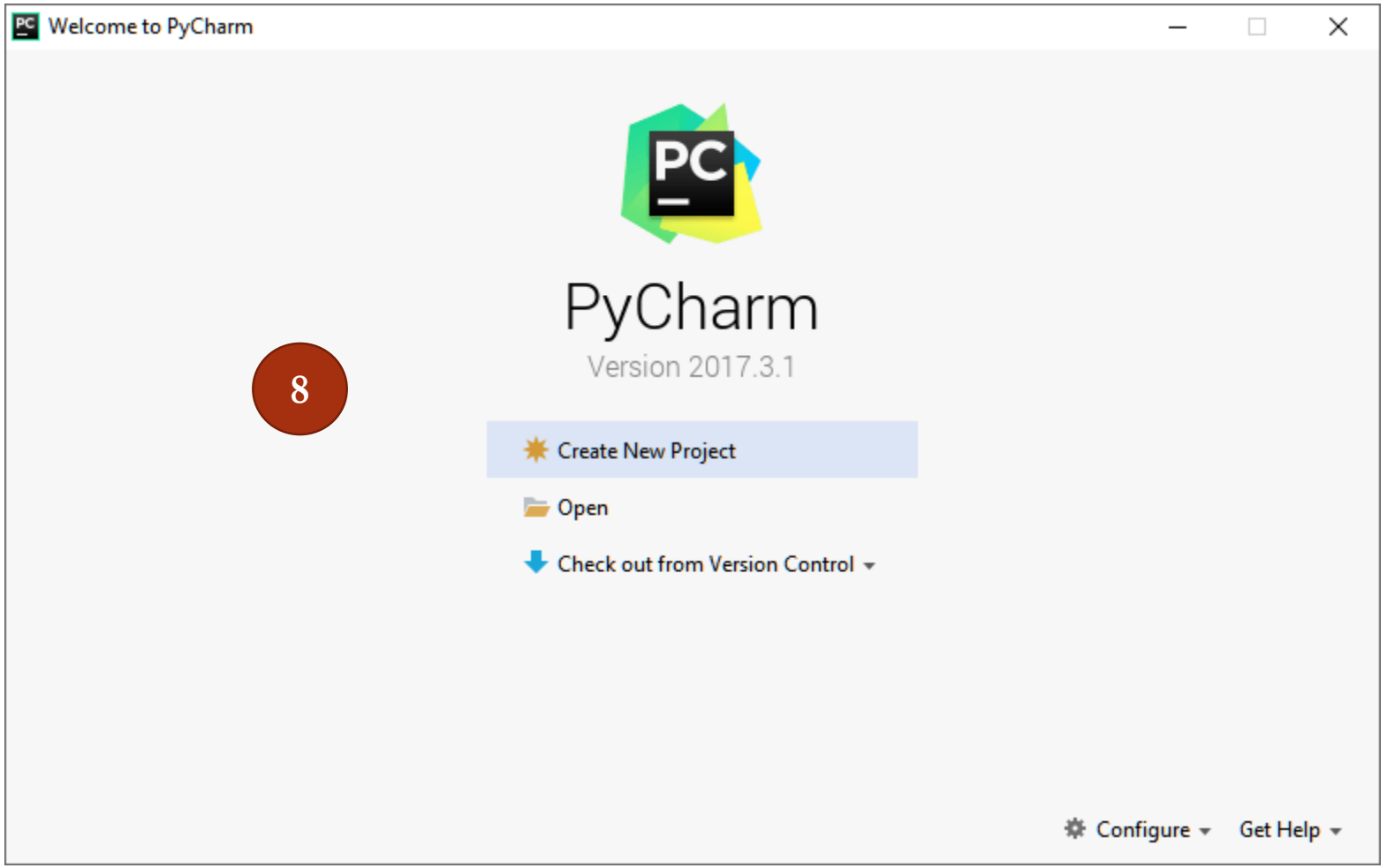

PyCharm Community Edition Setup
Choose Install Location
Choose the folder in which to install PyCharm Community Edition.
Setup will install PyCharm Community Edition in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.
3
Destination Folder
C:\Program Files\JetBrains\PyCharm Community Edition 2017.3
Browse...
Space required: 511.9MB
Space available: 62.7GB
< Back
Next >
Cancel


PyCharm Community Edition Setup
Installation Options
Configure your PyCharm Community Edition installation
Create Desktop Shortcut
32-bit launcher
64-bit launcher
Create Associations
.py
4
Download and install JRE x86 by JetBrains
< Back
Next >
Cancel


PyCharm Community Edition Setup
Choose Start Menu Folder
Choose a Start Menu folder for the PyCharm Community Edition shortcuts.
Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.
JetBrains
Accessibility
Accessories
Administrative Tools
Backup and Sync from Google
CCleaner
Clover
CrystalDiskMark5
Flock
Maintenance
Microsoft Office 2016 Tools
Notepad++
NVIDIA Corporation
5
< Back
Install
Cancel


PyCharm Community Edition Setup
Installing
Please wait while PyCharm Community Edition is being installed.
Extract: sdk-stubs.input.values... 74%
Show details
6
< Back
Next >
Cancel


PyCharm Community Edition Setup
Completing PyCharm Community Edition Setup
PyCharm Community Edition has been installed on your computer.
Click Finish to close Setup.
Run PyCharm Community Edition
7
< Back
Finish
Cancel


Welcome to PyCharm
PyCharm
Version 2017.3.1
Create New Project
Open
Check out from Version Control
Configure
Get Help

8

Comments

- Comments are used to write description about application logics to understand the logics easily.
- The main objective comments the code maintenance will become easy.
- The comments are non-executable code.
- Python will ignore comments in run time.

2 Types of comments

- Single line comments : write the description in single line & it starts with #
  - Syntax: # statement
- Multiline comments:
  - write the description in more than one line starts with “”” ends with : “””””(triple quotes)
  - in python while writing the comments we can write double quote or single quote (“) or (‘)

Keywords

- Python 2.x

```
import keyword
print keyword.kwlist
```

- Python 3.x

```
import keyword
print(keyword.kwlist)
```

- In 2.x parenthesis are optional for print
- In 3.x parenthesis are mandatory for print
- Removed keywords from 3.x : *print*
- Newly added keyords in 3.x : *False, None, True,*

- ['and', 'as', 'assert', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'exec', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'not', 'or', 'pass', 'print', 'raise', 'return', 'try', 'while', 'with', 'yield']

- ['False', 'None', 'True', 'and', 'as', 'assert', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global'

-

Variables

- A variable is nothing but a reserved memory location to store values.
- Variables are used to store the data.
- Memory allocated when the values are stored in variables.
- Every variable must have some type.

Data Types

- Numbers
- String
- List
- Tuple
- Dictionary
- Boolean

Concatenation

- We can contact 2 same types, but we cannot concat 2 different types.

```
print(10+10) # valid
print(10.5+10.5) # valid
print("welcome"+"python") # valid
print (10+15.5) # valid

print(True+5) # valid
print(10+False) # valid
print(True+True)# valid

print(10+"welcome")  #Not valid  -TypeError: unsupported operand type(s)
print(10.5+"welcome")  #Not valid -TypeError: unsupported operand type(s)
print(True+"welcome")  #Not valid -TypeError: unsupported operand type(s)
```

Swapping, Re-declaring & Deleting variables

```
# swapping
x=10
y=5

print("Before swapping values are:",x,y)

x,y=y,x

print("After swapping values are:",x,y)
```

```
#Re-declaring

a=10
print(a)

a=100
print(a)
```

```
a=10
print(a)

del a     # deletes the variable

print(a)   #NameError: name 'a' is not defined
```

Input() & raw_input()

- Getting input from the end-user python2 vs python3:
- Python 2.x
  - input function : Takes any type of data
  - raw_input function : Takes only string data
- Python 3.x
  - Input function : Takes only string data

- Note:
  - In Python 3.x, 'raw_input()' is changed to 'input()'. Thus, 'input()' of Python 3 will behave as 'raw_input()' of Python 2.x

Type conversion

- Getting data from user by using input function in the form of String then converting,
- String to int
  - num1 = int(input('Enter first number: '))
- String to float
  - num2 = float(input('Enter second number: '))

Formatting Output

- Formatting data with the % & { }
- %d int
- %s string
- %f (or) %g float

Example

```
name,age,sal = "John",24,10000.35

#Approach1
print(name,age,sal)

#Approach2
print("Name is:", name)
print("Age is:", age)
print("Sal is:", sal)

#Approach3 : using  %  Here type is imp
print("Name:%s  age:%d  salary:%g" %(name, age, sal))

#Approach3 : using  {} Here value is imp
print("Name:{}  age:{}  salary:{}".format(name, age, sal))

#Approach4 : using  {} Here value is imp
print("Name:{1}  age:{2}  salary:{2}".format(name, age, sal))
```