PYTHON INTRODUCTION

python™

Python Introduction

In this session

- What is Python?
- What is Anaconda?
- Popularity rank
- Why use Anaconda in Machine Learning?
- Anaconda installation
- PyCharm installation
- Hello world
- Basic calculation
- Variable assignment
- Basic Operator
- Data Structure (Tuple, List, Matrix, Dictionaries)
- If Statement
- For Loop
- Function

Python Introduction

What is Python?

- Computer language
- Interpreter
- Multi-paradigm: (OOP, imperative, functional, procedural)
- Typing: dynamic
- Statistical and graphics
- Linear and nonlinear modelling
- Age 26 (C# 17)
- Free Software (GNU project)
- Linux, Windows and MacOS
- One of the most powerful ML language
- Tool for ML exploration
- Good for building a production model
- Supported in Azure ML Studio

Python Introduction

What is Anaconda?

- Anaconda is a python and R distribution
- Provide everything you need for data science "out of the box"
- The core python language
- 100+ python "packages" (libraries)
- Spyder and Jupyter ready
- conda: Anaconda's own package manager
- Microsoft Azure ML Studio fully supports

Python Introduction

Popularity rank

Language Rank	Types	Spectrum Ranking
1. C		100.0
2. Java		98.1
3. Python		98.0
4. C++		95.9
5. R		87.9
6. C#		86.7
7. PHP		82.8
8. JavaScript		82.2
9. Ruby		74.5
10. Go		71.9

IEEE SPECTRUM

Source: The 2016 Top Programming Languages
http://spectrum.ieee.org/computing/software/the-2016-top-programming-languages

Python Introduction

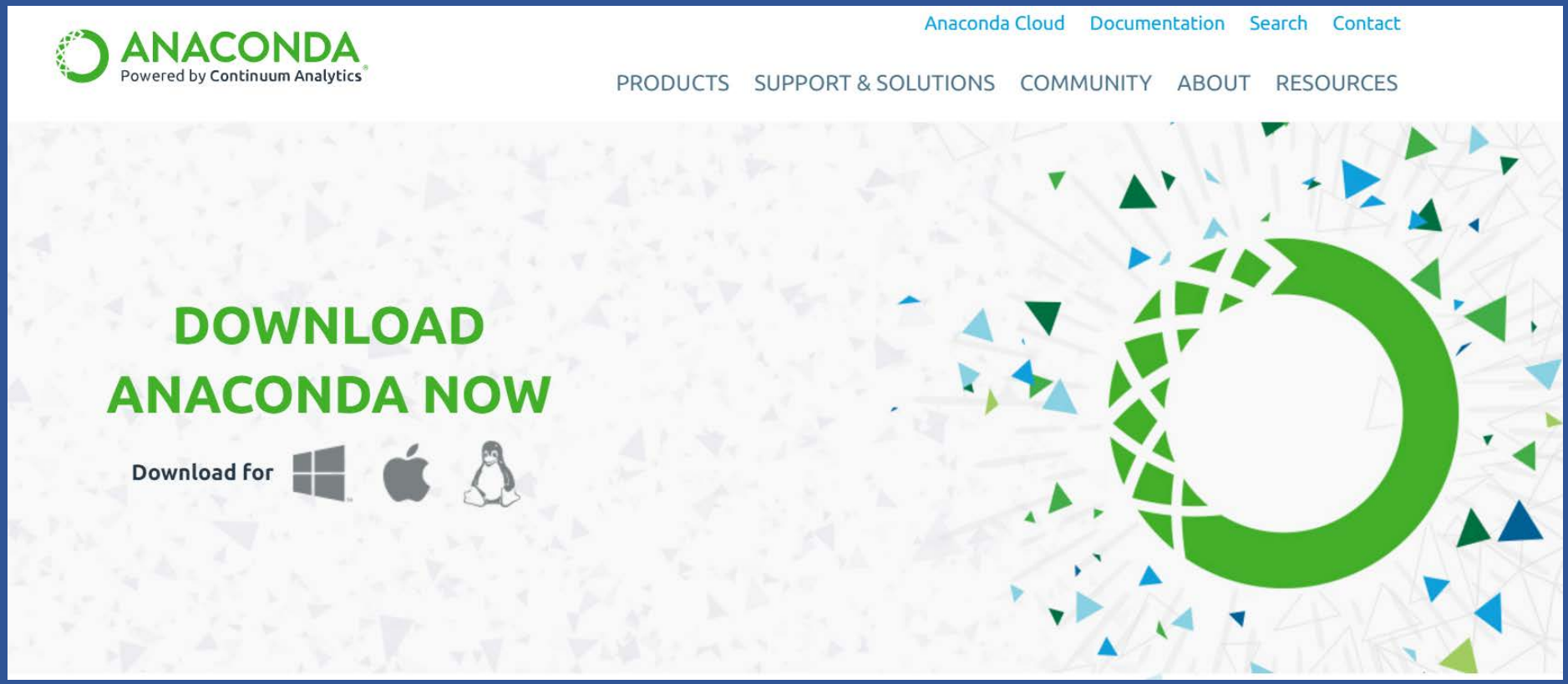
Why use Anaconda in Machine Learning?

- 720+ data science packages
  - visualizations
  - machine learning
  - deep learning
  - Big Data
  - tensor calculation
- Native access
  - HDFS
  - Amazon S3
- Distributed computing
- GPU supercharged
- Agile & fast experimentation data science
- Use Microsoft Excel® to perform predictive analytics
- Query and transform Big Data
- Easily deploy production
- Spyder & Jupiter build-in

Python Introduction

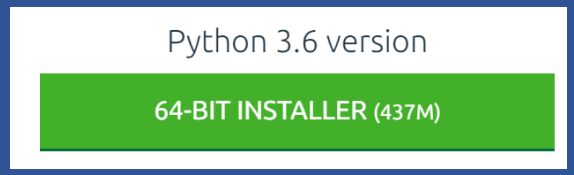
Anaconda installation

https://www.continuum.io/downloads



Python Introduction

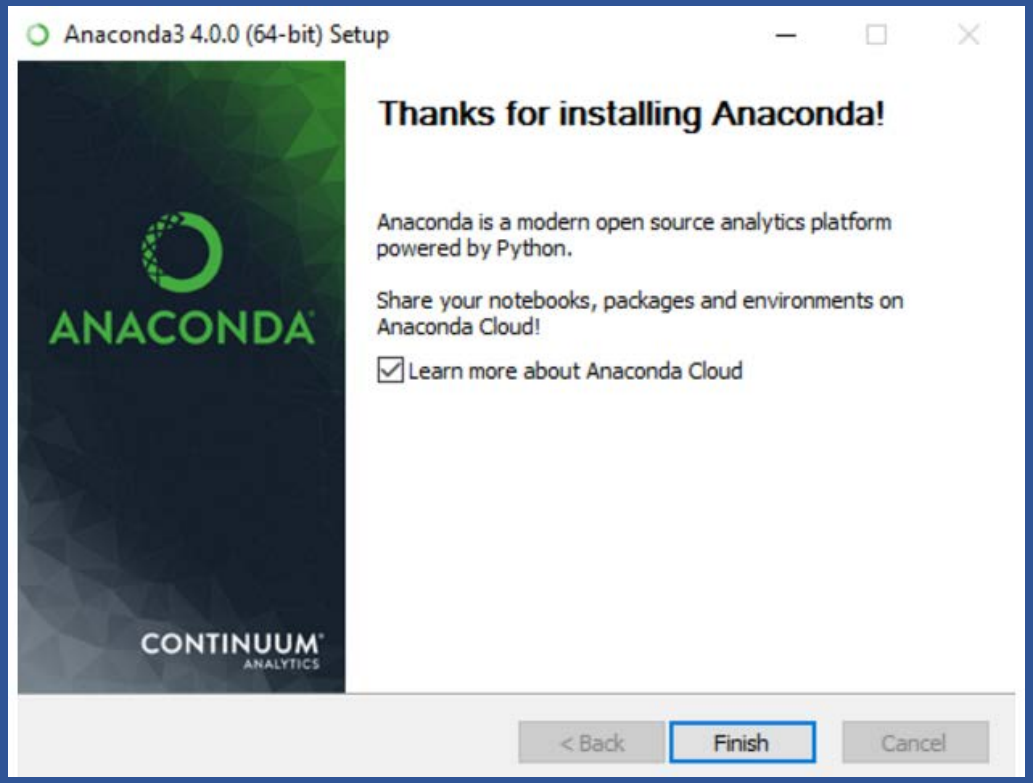
Anaconda installation



1. Click Python 3.6 version
2. Click Run to launch the installer.
3. Click Next.
4. Read the licensing terms and click I Agree.
5. Select an install for “Just Me”
6. Select a destination folder to install Anaconda and click Next.
7. Choose whether to add Anaconda to your PATH environment variable.
8. Choose whether to register Anaconda as your default Python 3.6.
9. Click Install.
10. Click Next.

Python Introduction

Anaconda installation

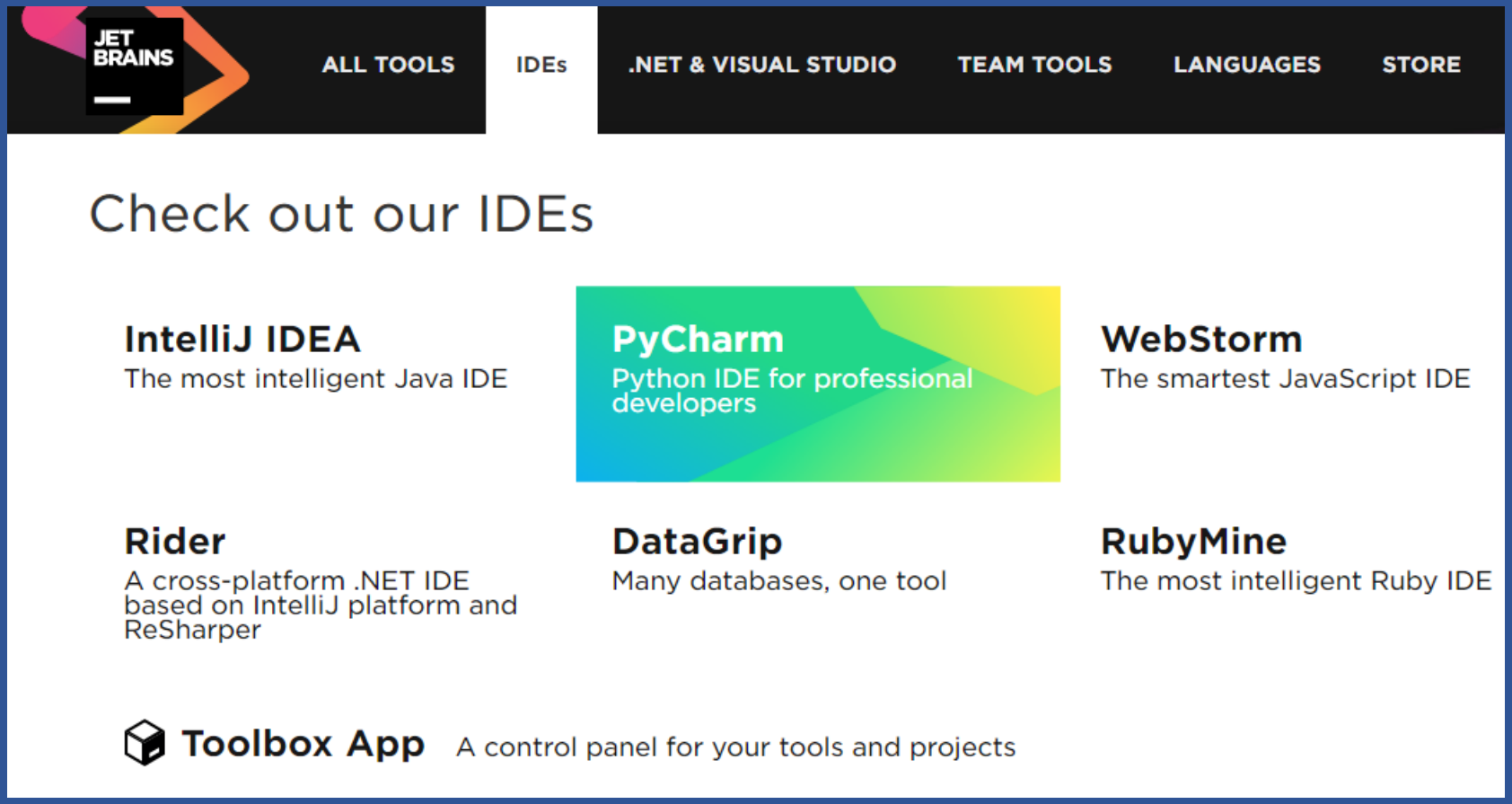


11. After a successful installation you will see the “Thanks for installing Anaconda” image:

Python Introduction

PyCharm installation

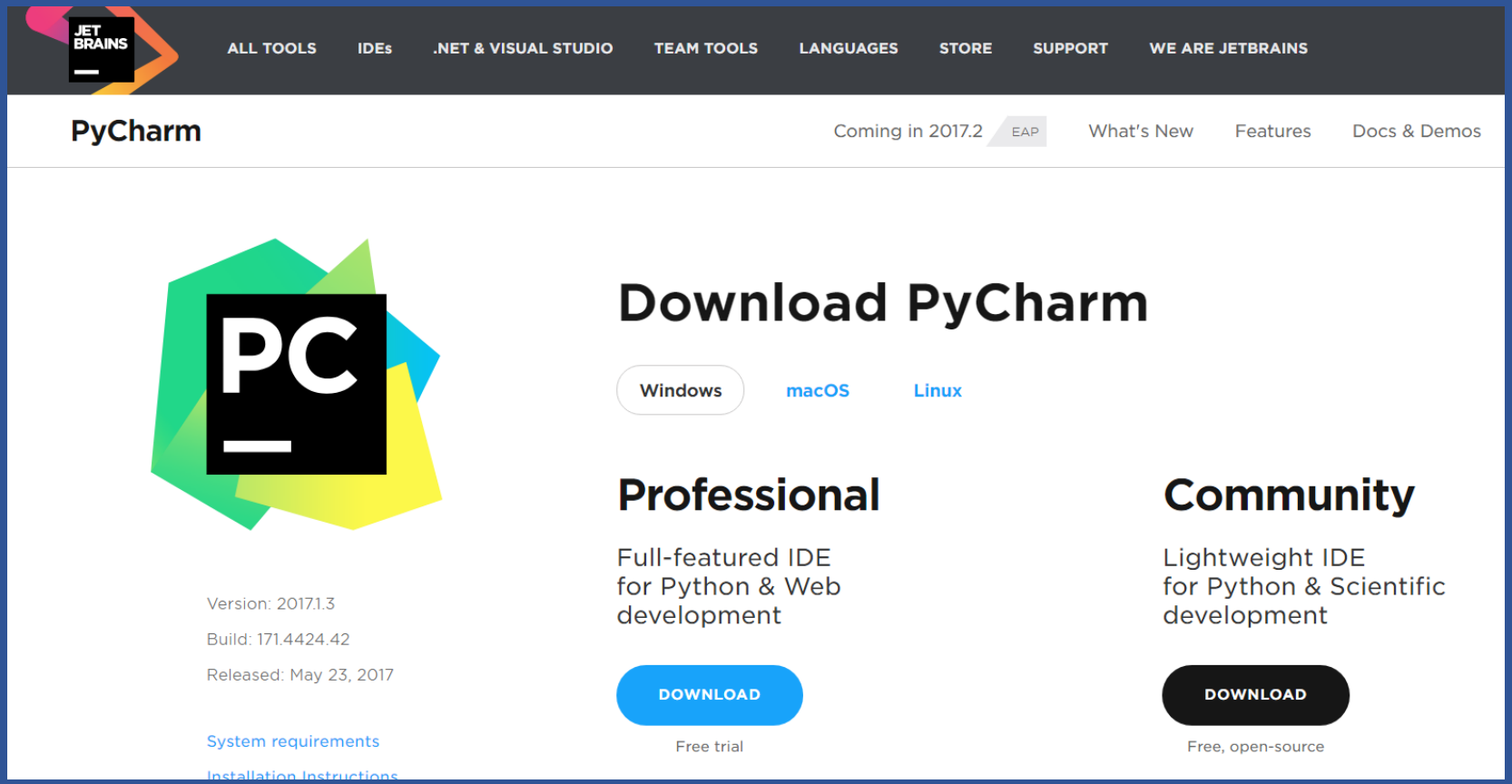
https://www.jetbrains.com/



Python Introduction

PyCharm installation

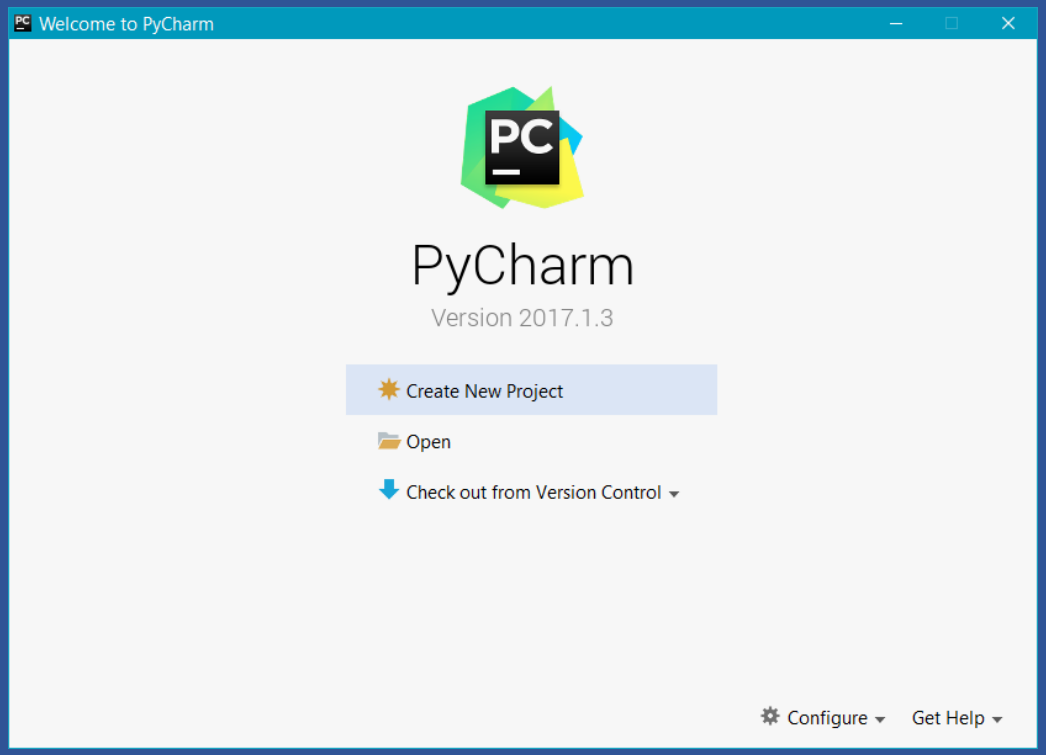
Click Professional download



Python Introduction

PyCharm installation

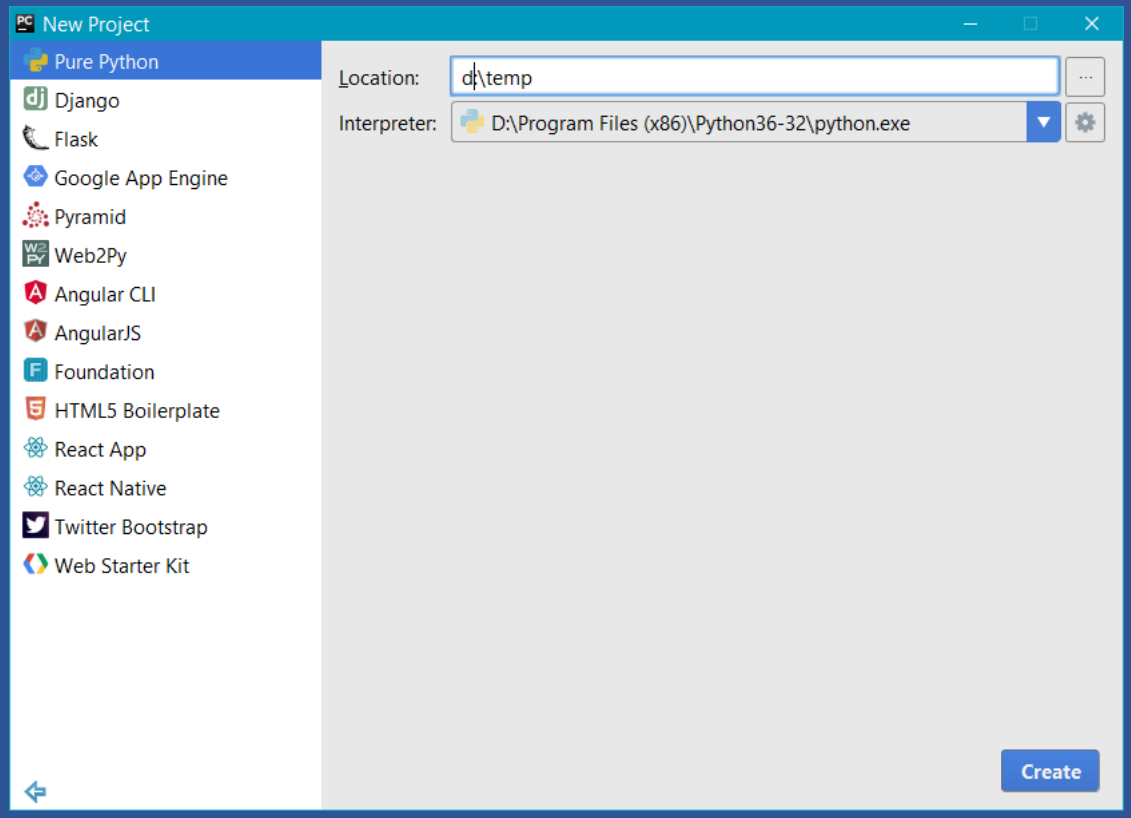
Install with all default setting and run



Python Introduction

Hello world

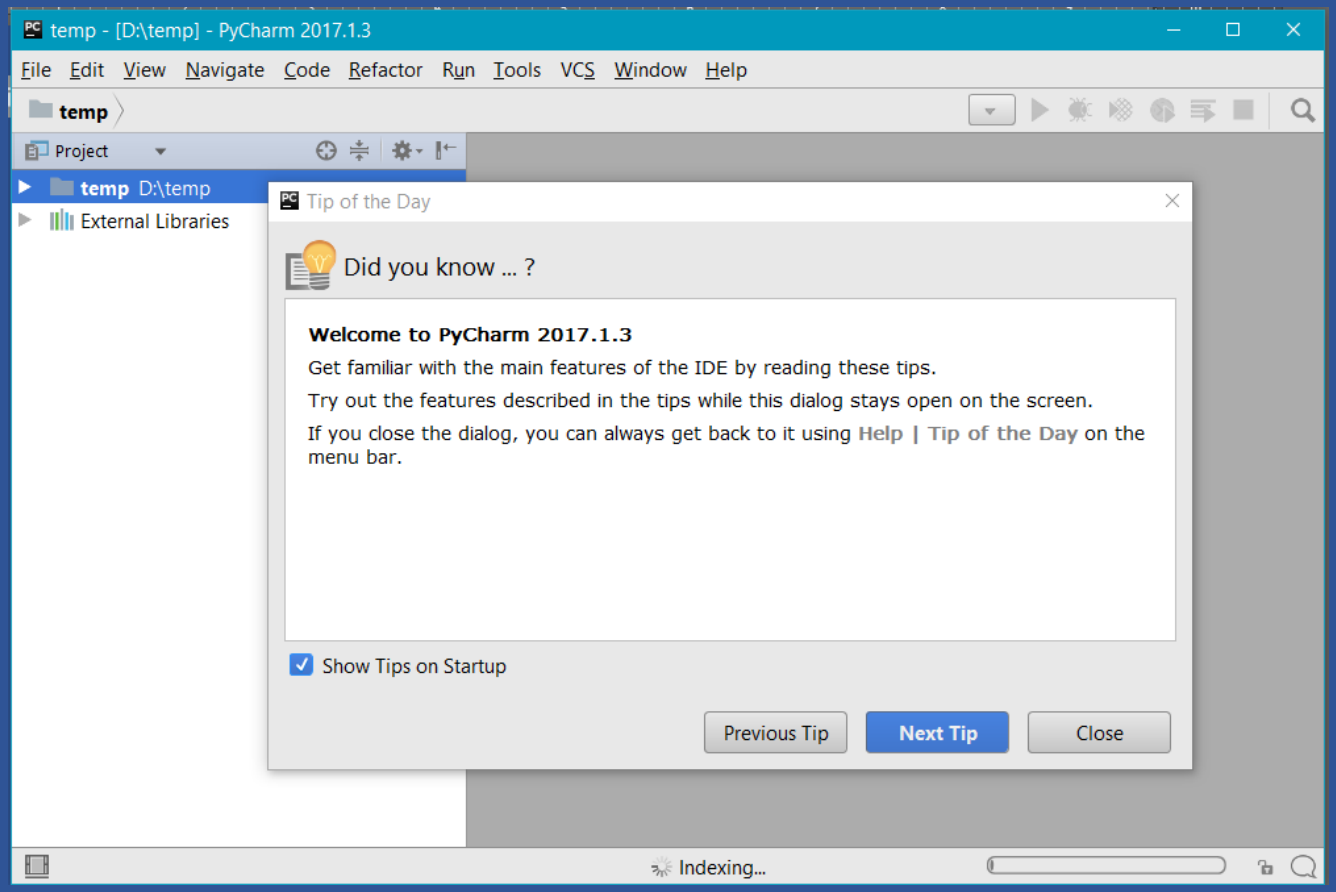
1. Create new folder c:\temp (or d:\temp)
2. In Location box type c:\temp (or d:\temp)



Python Introduction

Hello world

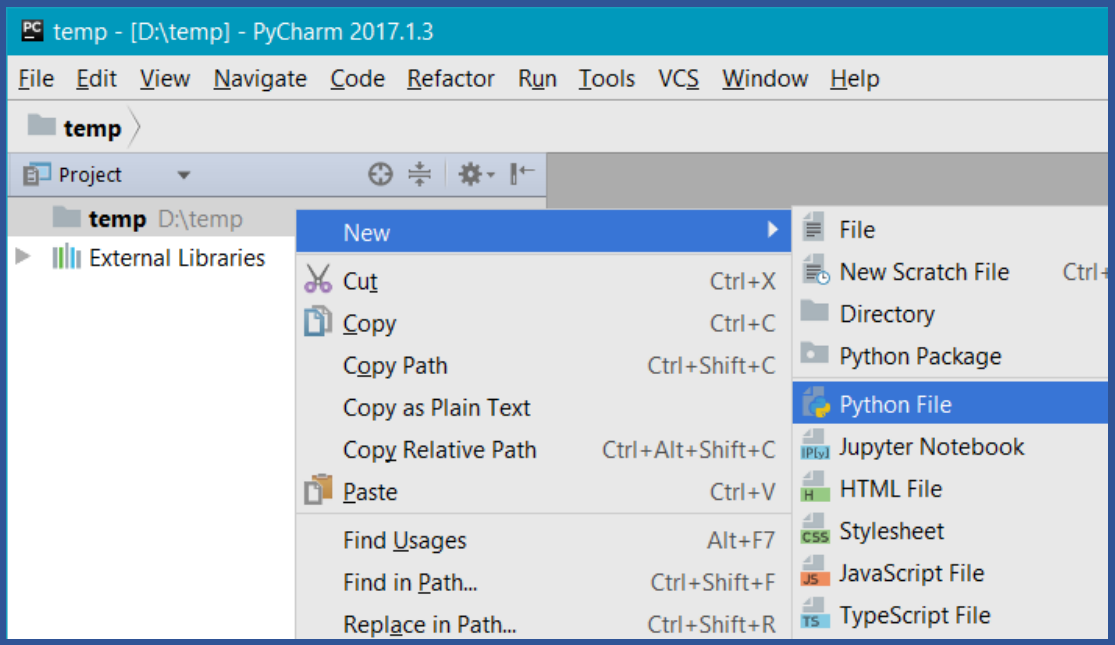
3. Click create

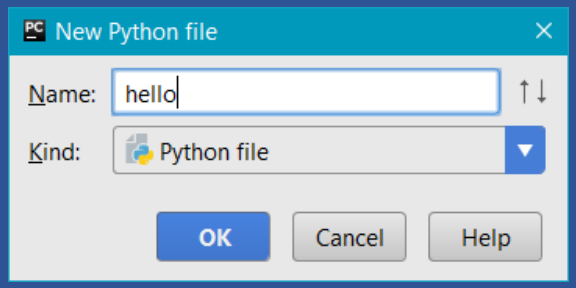


Python Introduction

Hello world

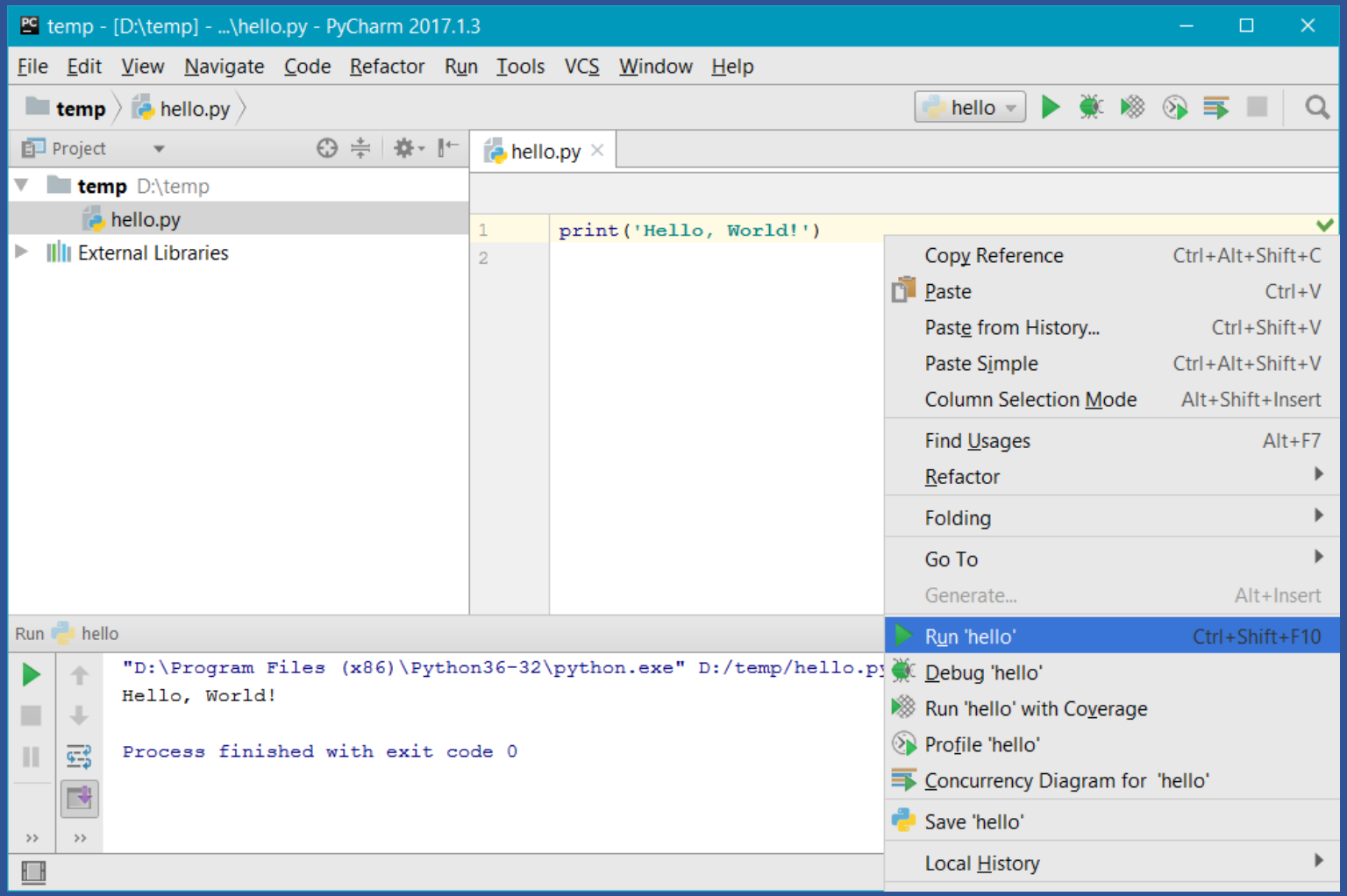
4. Right click at temp D:\temp
5. Click New
6. Click Python File
7. In Name box type hello
8. Click OK





Python Introduction

Hello world



Python Introduction

Basic computation

```
import math

# basic computation in R
a = 2 + 3  a: 5
b = 6/3  b: 2.0
c = (3*8)/(2*3)  c: 4.0
d = math.log(12)  d: 2.4849066497880004
e = math.sqrt(121)  e: 11.0
print('end')
```

Python Introduction

Debugging in PyCharm

Break point

Debugger

Values

Variables

```
import math

# basic computation in R
a = 2 + 3   a: 5
b = 6/3   b: 2.0
c = (3*8)/(2*3)   c: 4.0
d = math.log(12)   d: 2.4849066497880004
e = math.sqrt(121)   e: 11.0
print('end')
```

Python Introduction

Variable declaration

```
# variable assignment & declaration
a = 1
b = 2
c = a + b
k = (a * 2) + (b * 3)
o = 1; p = 'dog'; q = 3
x = y = z = 1        # multiple
l, m, n = 1, 2, 'cat'
print('end')
```

Python Introduction

Basic data type

```
# Basic data type
name = 'laploy'        # character
who = name + ' v.'     # string concat
price = 1500           # integer
kilo = 1.5             # floating point
x = price + kilo       # automatic type casting
yes = True             # bool
gen = 123              # integer
gen = 'hello'          # change to string
print('end')
```

Python Introduction

Basic Operator

```
# Basic Operators
# Arithmetic
a = (2 + 3) * 2  a: 10
x, y = 2, 3  x: 2  y: 3
b = (x + y) * 2  b: 10
c = (2 * 3) / 2  c: 3.0
d = 2 / (3 * 4)  d: 0.16666666666666666
e = 3 * (4/2)  e: 6.0
# tuple and operator +
v = (1, 2, 3)  v: <class 'tuple'>: (1, 2, 3)
t = (2, 2, 1)  t: <class 'tuple'>: (2, 2, 1)
f = v + t  f: <class 'tuple'>: (1, 2, 3, 2, 2, 1)
# relation operator
x1 = a > c  x1: True
x2 = a < d  x2: False
x3 = a == b  x3: True
x4 = a != b  x4: False
print('end')
```

Python Introduction

Data Structure : List

```
L1 = []  # An empty list  L1: <class 'list'>: []
# Four items: indexes 0..3
L2 = [123, 'abc', 1.23, {}]  L2: <class 'list'>: [123, 'abc', 1.23, {}]
# Nested sublist
L3 = ['Bob', 40.0, ['dev', 'mgr']]  L3: <class 'list'>: ['Bob', 40.0, ['dev', 'mgr']]
L4 = list('spam')  L4: <class 'list'>: ['s', 'p', 'a', 'm']
L5 = list(range(-4, 4))  L5: <class 'list'>: [-4, -3, -2, -1, 0, 1, 2, 3]
x = [1, 2, 3]  x: <class 'list'>: [1, 2, 3]
y = [1, 2, 3]  y: <class 'list'>: [1, 2, 3]
z = x + y  z: <class 'list'>: [1, 2, 3, 1, 2, 3]
# List comprehensions
a1 = [v for v in 'SPAM']  a1: <class 'list'>: ['S', 'P', 'A', 'M']
a2 = [v * 4 for v in 'cat']  a2: <class 'list'>: ['cccc', 'aaaa', 'tttt']
print('end')
```

Python Introduction

Data Structure : Matrix

```
# Matrix
matrix = [[1, 2, 3],  matrix: <class 'list'>: [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
          [4, 5, 6],
          [7, 8, 9]]
a = matrix[1]   # get a row  a: <class 'list'>: [4, 5, 6]
b = matrix[1][2]    # get row 1 column 2  b: 6
c = [[1, 2, 3],  c: <class 'list'>: [[1, 2, 3], [12, 13, 14], [7, 8, 9]]
          [4, 5, 6],
          [7, 8, 9]]
c[1] = [12, 13, 14]
d = [[1, 2, 3],  d: <class 'list'>: [[1, 2, 3], [], [7, 8, 9]]
          [4, 5, 6],
          [7, 8, 9]]
d[1] = []
print('end')
```

Python Introduction

Data structure: Dictionary

```
# Dictionary
a = {'name': 'loy', 'age': 20, 'gender': 'M'}  a: {'name'
b = a['name']  b: 'loy'
c = len(a)  c: 3
d = 'name' in a  d: True
e = list(a.keys())  e: <class 'list'>: ['name', 'age', 'g
f = list(a.values())  f: <class 'list'>: ['loy', 20, 'M']
g = {'name': 'loy', 'age': 20, 'gender': 'M'}  g: {'name'
h = g['name'] = 'lap'  h: 'lap'
i = {'name': 'loy', 'age': 20, 'gender': 'M'}  i: {'age':
# delete entry
del i['name']
j = {'name': 'loy', 'age': 20, 'gender': 'M'}  j: {'name'
# add entry
l = j['year'] = 3  l: 3
print('end')
```

Python Introduction

If statement

```
# If Statement
x = 1
if x == 1:
    print('same')
elif x > 1:
    print('bigger')
else:
    print('smaller')

print('end')
```

Python Introduction

Loop statement

```
# For Loop statement
string = "Hello World"
s = ''
for x in string:
    s += x + ' '
# while loop
start, end = 0, 5
while start <= end:
    print('*')
    start += 1
print('end')
```

Python Introduction

Function Basic

```python
# Function
def add1(a1, b1):
    r1 = a1 + b1
    return r1

a = add1(2, 4)  a: 6

# anonymous function
add2 = lambda a2, b2: a2 + b2

b = add2(10, 20)  b: 30

print('end')
```

Python Introduction

More information

Official Python tutorial

https://docs.python.org/3/tutorial/

PyCharm 2017 Quick Start Guide

https://www.jetbrains.com/help/pycharm/quick-start-guide.html