

A L G O R I T H M

P R A C T I C E

人人圈开始

+

人之智能

# 深度学习算法与实践

数论子

深度学习

Steven Tang

# 课程安排

Class schedule

三 → 四

• Python 程序语言学习

• Python 程序进阶

• 神经网络基础

• Tensorflow 编程框架学习

• 卷积神经网络学习

• Keras 学习与大型神经网络实践

• 循环神经网络学习

• 自然语言理解

• Pytorch 基础和项目实践

• 目标检测综合运用

高等数学  
线性代数  
概率论

数学

数学

数学

数学

机器学习

人工智能

# 课程体系介绍



The diagram illustrates the relationship between different levels of artificial intelligence and machine learning. It features four nested ellipses representing the following concepts from top to bottom:

- 人工智能 (Artificial Intelligence)**: The outermost ellipse. Handwritten notes include "2012 AI" and "机器学习" (Machine Learning) with an arrow pointing to the next level.
- 机器学习 (Machine Learning)**: The second ellipse. Handwritten notes include "2012 机器学习" and "深度学习" (Deep Learning) with an arrow pointing to the next level.
- 神经网络 (Neural Networks)**: The third ellipse. Handwritten notes include "深度学习" and "深度学习" (Deep Learning) with an arrow pointing to the next level.
- 深度学习 (Deep Learning)**: The innermost ellipse. Handwritten notes include "深度学习" and "深度学习" (Deep Learning).

Additional handwritten notes and symbols include:

- Top left: "50c" with a downward arrow and "2020" circled.
- Top right: "400" circled, with "138" and "0.0" written next to it.
- Bottom left: "深度学习" (Deep Learning) circled, with "深度学习" (Deep Learning) written next to it.
- Bottom right: "深度学习" (Deep Learning) circled, with "深度学习" (Deep Learning) written next to it.



# 1

## Python 程序基础



# 本章节内容



Python语言特点



Python 环境配置和  
Anaconda平台安装



简单程序示例

# Python语言的诞生

4812

Guido van Rossum, Python语言创立者,

20世纪90年代初 诞生, 2002年, Python 2.x, 2008年, Python 3.x



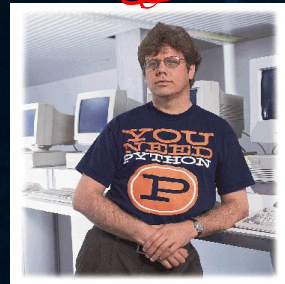
Java



PHP



C++



Python



渡一教育  
Duyi Education

2020

AI →

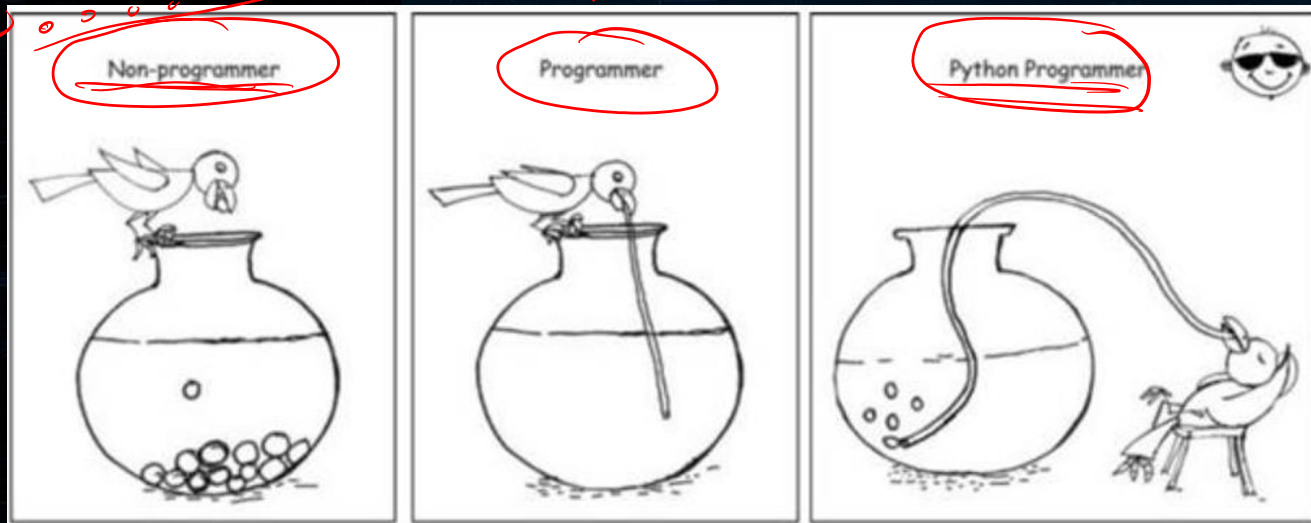
# 为什么选PYTHON作为人工智能语言?

4/8

2012 — 2018

2020

2012





# 第一个Python程序

使用Python语言编写的Hello程序只有一行代码：print(“Hello World”)

```
>>>print("Hello World")  
Hello World
```

极客

第一行的“>>>”是Python语言运行环境的提示符，第二行是Python语句的执行结果

# C语言的Hello程序

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
printf("Hello World\n");
```

```
return 0;
```

```
}
```

# Java语言的Hello程序

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

# Python语言优势1：代码简洁

抄写

简洁

脚本语言 + 语句执行

例1：

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

例2：

```
sum = 33333 * 44444  
print(sum)
```





# 编译和解释

高级语言按照计算机执行方式的不同可分成两类

- 静态语言

- 脚本语言

这里所说的执行方式指计算机执行一个程序的过程，静态语言采用编译执行，脚本语言采用解释执行。

# 编译和解释

- 编译是将源代码转换成目标代码的过程，通常，源代码是高级语言代码，目标代码是机器语言代码，执行编译的计算机程序称为编译器



# 编译和解释

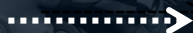
- 解释是将源代码逐条转换成目标代码同时逐条运行目标代码的过程。执行解释的计算机程序称为解释器。



源代码



解释器



输出

# 编译和解释

编译和解释

编译是一次性地翻译，一旦程序被编译，不再需要编译程序或者源代码。

- 对于相同源代码，编译所产生的目标代码执行速度更快。
- 目标代码不需要编译器就可以运行，在同类型操作系统上使用灵活。



# 编译和解释程序

⑦

exe

解释则在每次程序运行时都需要解释器和源代码。



- 解释执行需要保留源代码，程序纠错和维护十分方便。
- 只要存在解释器，源代码可以在任何操作系统上运行，可移植性好

python

## Python语言的优势：可读性好

→ def mean(numbers):

→ s = 0.0

→ for num in numbers:

→ s = s + num

→ return s / len(numbers)

nums = [0,1,2,3,4,5,6,7,8,9]

print(mean(nums))

# Python语言的优势

跨平台 + 开源



PHP

500G

2006

Java

C++

<http://pypi.python.org/>

# 本章节内容



Python语言特点



Python 环境配置和  
Anaconda平台安装



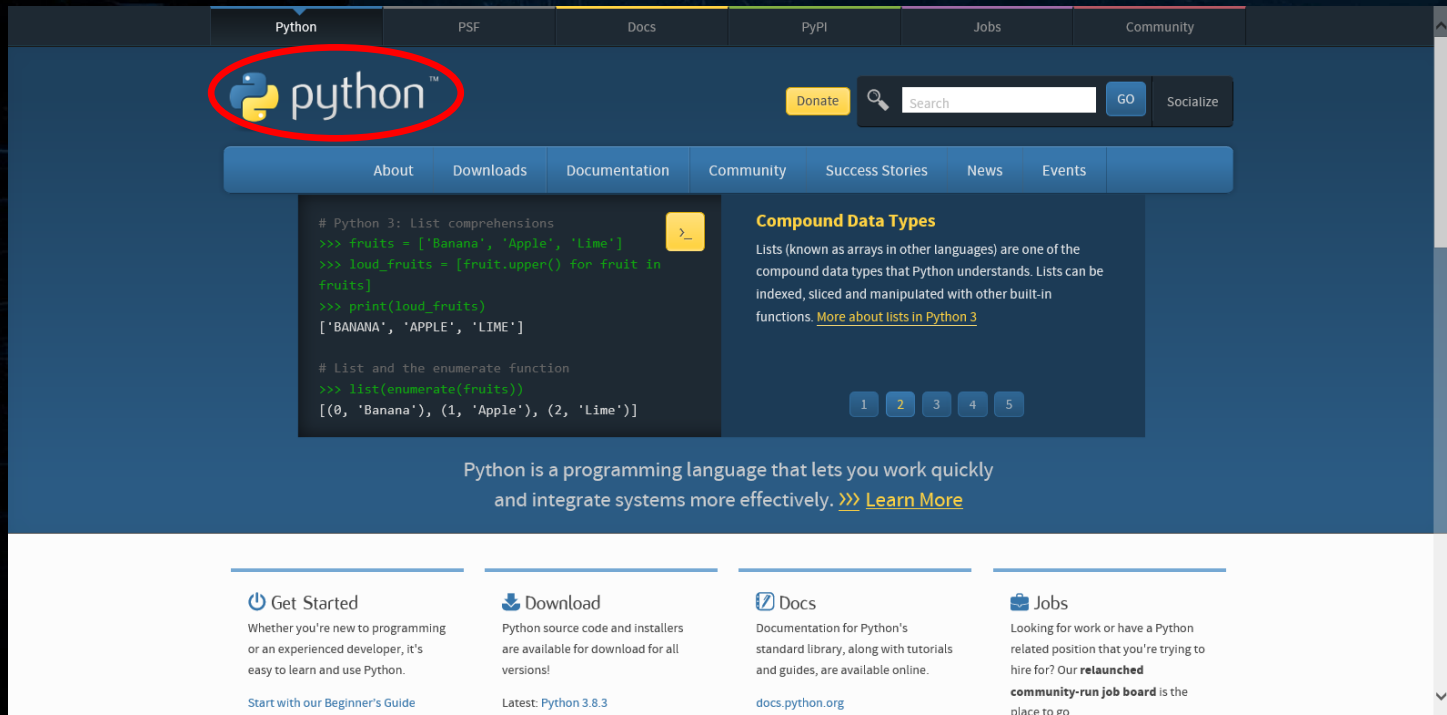
简单程序示例



# 安装

- 到 Python 主页 下载 并 安装 Python 基本 开发 和 运行 环境，网 址：  
[www.python.org/downloads/](http://www.python.org/downloads/)
- 根据操作系统不同选择不同版本
- 下载相应的Python 3.0系列版本程序

# 安装



The screenshot shows the Python.org homepage. The Python logo, consisting of two interlocking snakes (one blue, one yellow) and the word "python" in lowercase, is circled in red. The navigation bar includes links for Python, PSF, Docs, PyPI, Jobs, and Community. Below the navigation bar, there are links for About, Downloads, Documentation, Community, Success Stories, News, and Events. The main content area features a code snippet on the left and an article titled "Compound Data Types" on the right. The code snippet demonstrates list comprehensions and the enumerate function. The article discusses lists as compound data types and provides a link to "More about lists in Python 3". Below the article, there are five numbered buttons (1, 2, 3, 4, 5). At the bottom of the page, there are four sections: "Get Started", "Download", "Docs", and "Jobs". Each section provides information about getting started with Python, downloading source code and installers, accessing documentation, and finding jobs or hiring opportunities.

Python

PSF

Docs

PyPI

Jobs

Community

python™

Donate

Search

GO

Socialize

About

Downloads

Documentation

Community

Success Stories

News

Events

```
# Python 3: List comprehensions
>>> fruits = ['Banana', 'Apple', 'Lime']
>>> loud_fruits = [fruit.upper() for fruit in fruits]
>>> print(loud_fruits)
['BANANA', 'APPLE', 'LIME']

# List and the enumerate function
>>> list(enumerate(fruits))
[(0, 'Banana'), (1, 'Apple'), (2, 'Lime')]
```

### Compound Data Types

Lists (known as arrays in other languages) are one of the compound data types that Python understands. Lists can be indexed, sliced and manipulated with other built-in functions. [More about lists 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](#)

#### Get Started

Whether you're new to programming or an experienced developer, it's easy to learn and use Python.

[Start with our Beginner's Guide](#)

#### Download

Python source code and installers are available for download for all versions!

Latest: Python 3.8.3

#### Docs

Documentation for Python's standard library, along with tutorials and guides, are available online.

[docs.python.org](#)

#### Jobs

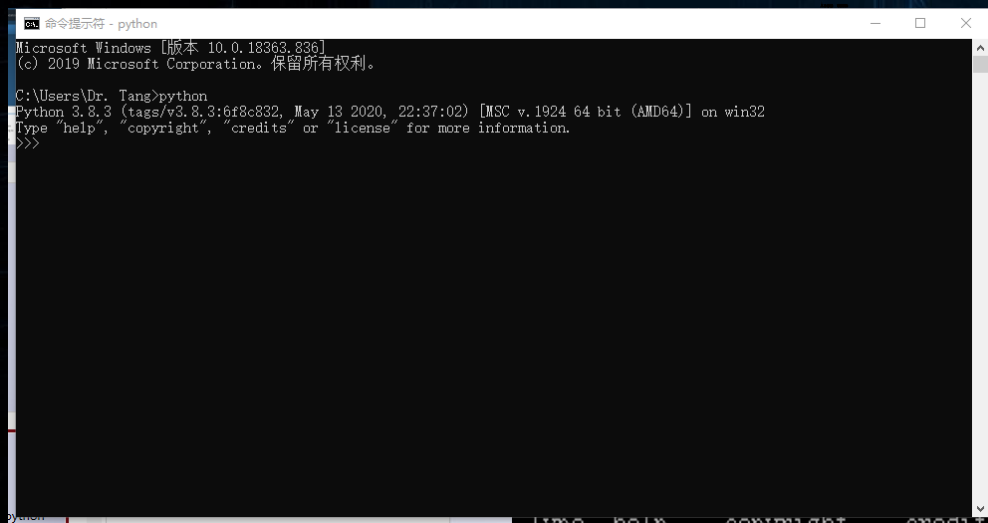
Looking for work or have a Python related position that you're trying to hire for? Our **relaunched** **community-run job board** is the place to go.

# 安装



# 启动

- 方法1：启动Windows命令行工具，输入python



```
命令提示符 - python
Microsoft Windows [版本 10.0.18363.836]
(c) 2019 Microsoft Corporation. 保留所有权利。

C:\Users\Dr. Tang>python
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type 'help', 'copyright', 'credits' or 'license' for more information.
>>>
```

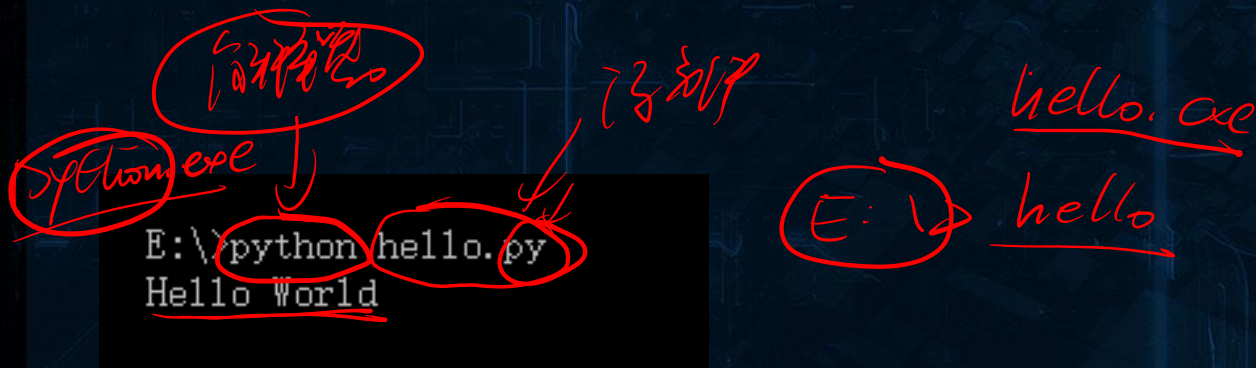


# 启动

- 方法2：调用IDLE来启动Python图形化运行环境

# 启动

- 方法3：按照语法格式编写代码，编写可以用任何文本编辑器，保存为文件。



# 启动

- 方法4：打开IDLE，点击Ctrl+N打开一个新窗口，输入语句并保存，使用快捷键F5即可运行该程序

# Hello程序

- 采用上述某个方法，执行：

```
>>> print("Hello World")
Hello World
>>> print("世界，你好")
世界，你好
>>>
```

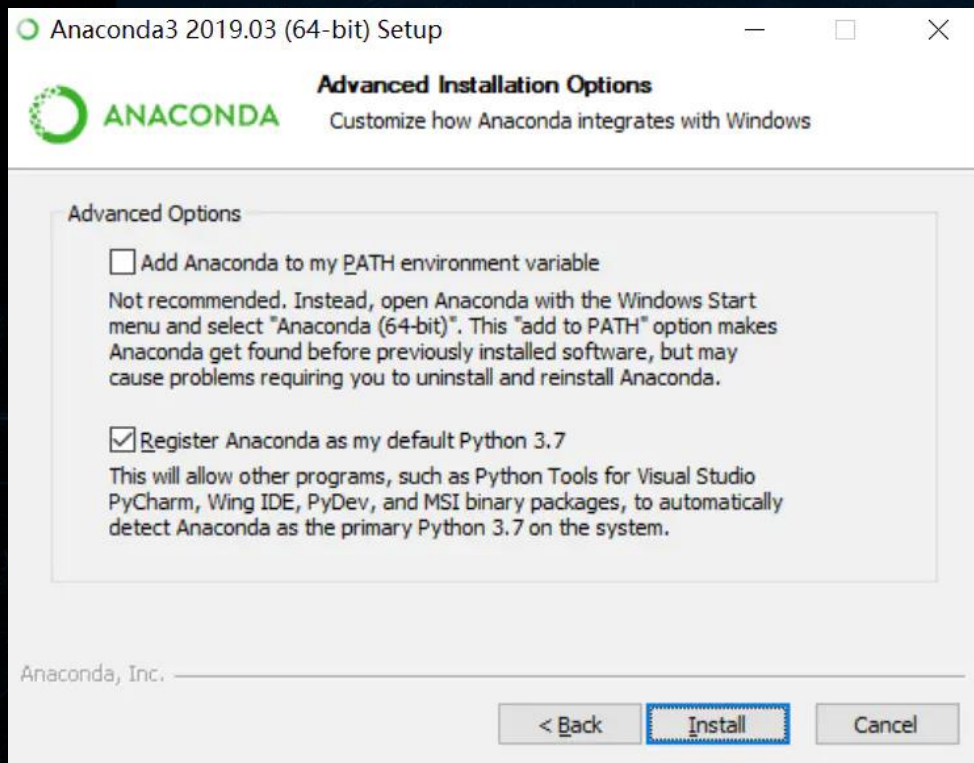


# Anaconda环境安装

# 下载

- <https://www.anaconda.com/distribution/>

# 安装



# Anaconda中添加环境

- `conda create -n duyì python=3.7`
- `activate duyì`

```
(base) C:\Users\Dr. Tang>conda create -n duyì python=3.7
WARNING conda.base.context:use_only_tar_bz2(632): Conda is constrained to only using the old .tar.bz2 file format because you have conda-build installed, and it is <3.
Collecting package metadata (repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 4.7.5
  latest version: 4.8.3
Please update conda by running

  $ conda update -n base conda

## Package Plan ##

  environment location: D:\Anaconda3\envs\duyì_python
  added / updated specs:
    - python=3.7

The following packages will be downloaded:



| package           | build  | size   | url                                                     |
|-------------------|--------|--------|---------------------------------------------------------|
| setuptools-47.1.1 | py37_0 | 673 KB | https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/win/ |
| Total:            |        | 673 KB |                                                         |



The following NEW packages will be INSTALLED:



| package        | location                                                         |
|----------------|------------------------------------------------------------------|
| certifi        | anaconda/pkgs/main/win-64::certifi-2020.4.5.1-py37_0             |
| pip            | anaconda/pkgs/main/win-64::pip-20.0.2-py37_0                     |
| python         | anaconda/pkgs/main/win-64::python-3.7.0-ha74fb7_0                |
| setuptools     | anaconda/pkgs/main/win-64::setuptools-47.1.1-py37_0              |
| vc             | anaconda/pkgs/main/win-64::vc-14.1-hb6101f6_4                    |
| vs2015_runtime | anaconda/pkgs/main/win-64::vs2015_runtime-14.16.27012-hf0eaf9b_2 |
| wheel          | anaconda/pkgs/main/win-64::wheel-0.34.2-py37_0                   |
| wincertstore   | anaconda/pkgs/main/win-64::wincertstore-0.2-py37_0               |

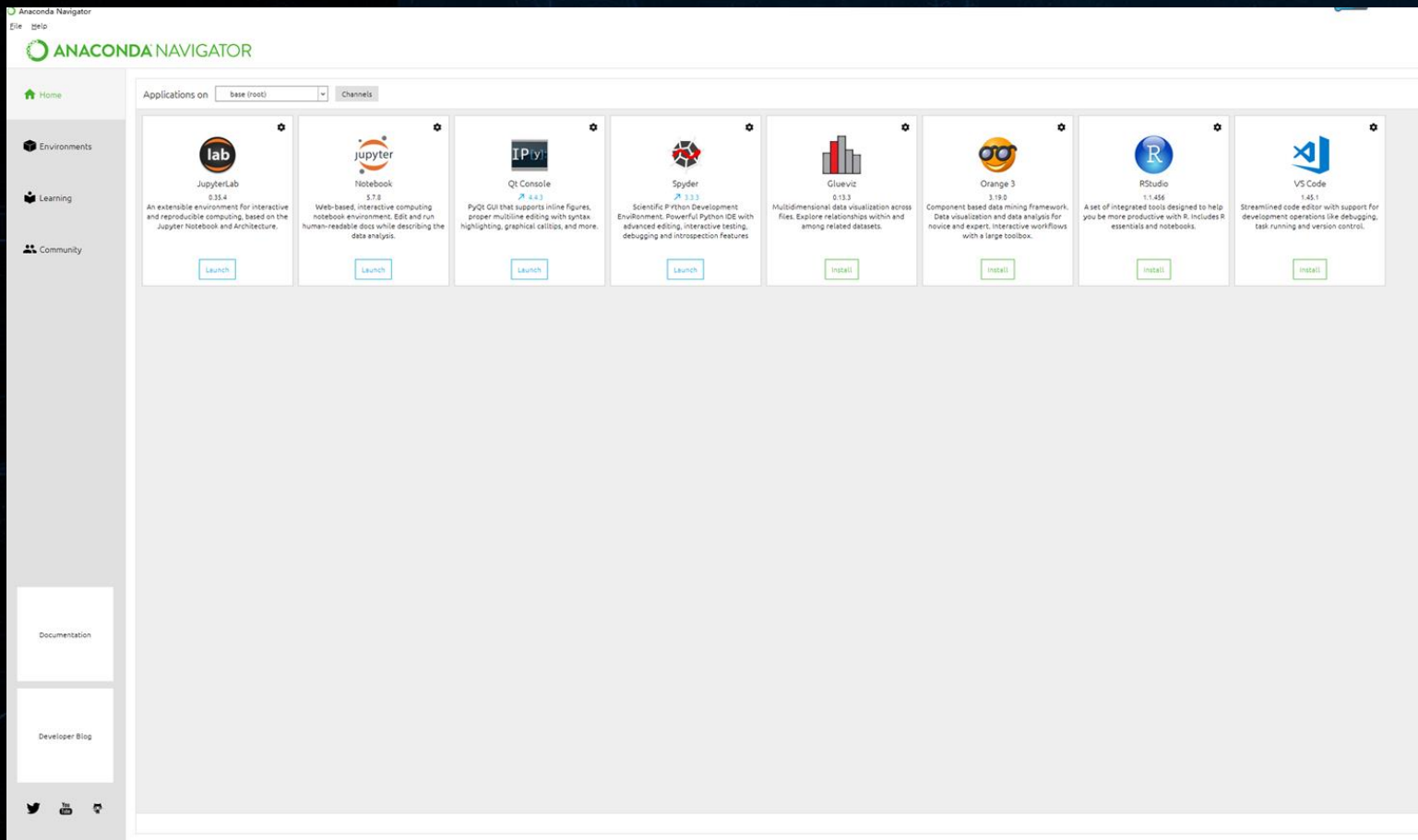


Proceed ([y]/n)?
```

假设E:\Anaconda3是安装目录。  
把E:\Anaconda3\Scripts;  
E:\Anaconda3两个目录添加到系统环境变量。

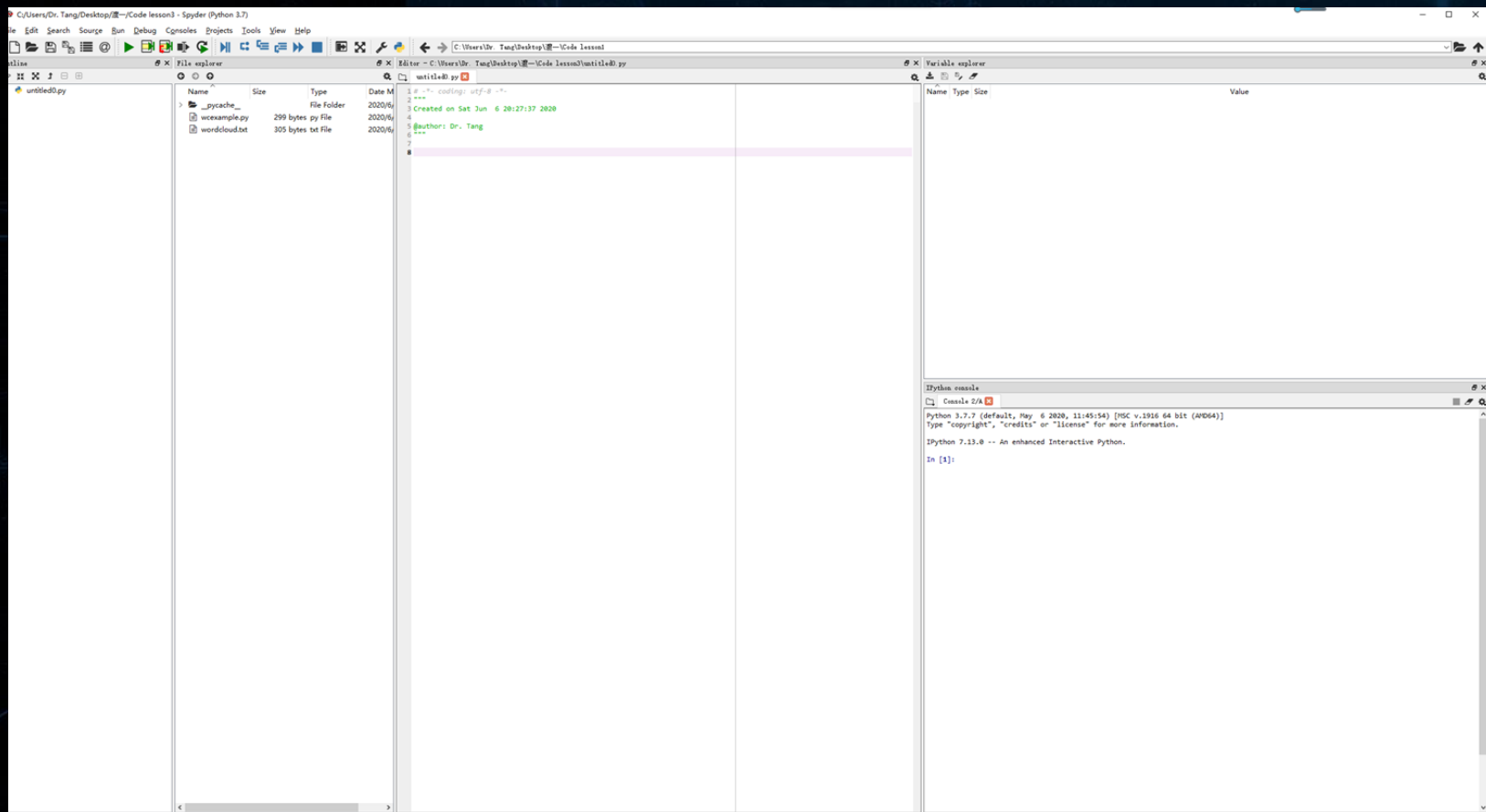


# Anaconda Navigator使用



# Spyder 安装

# Spyder 使用



# 本章节内容



Python语言特点



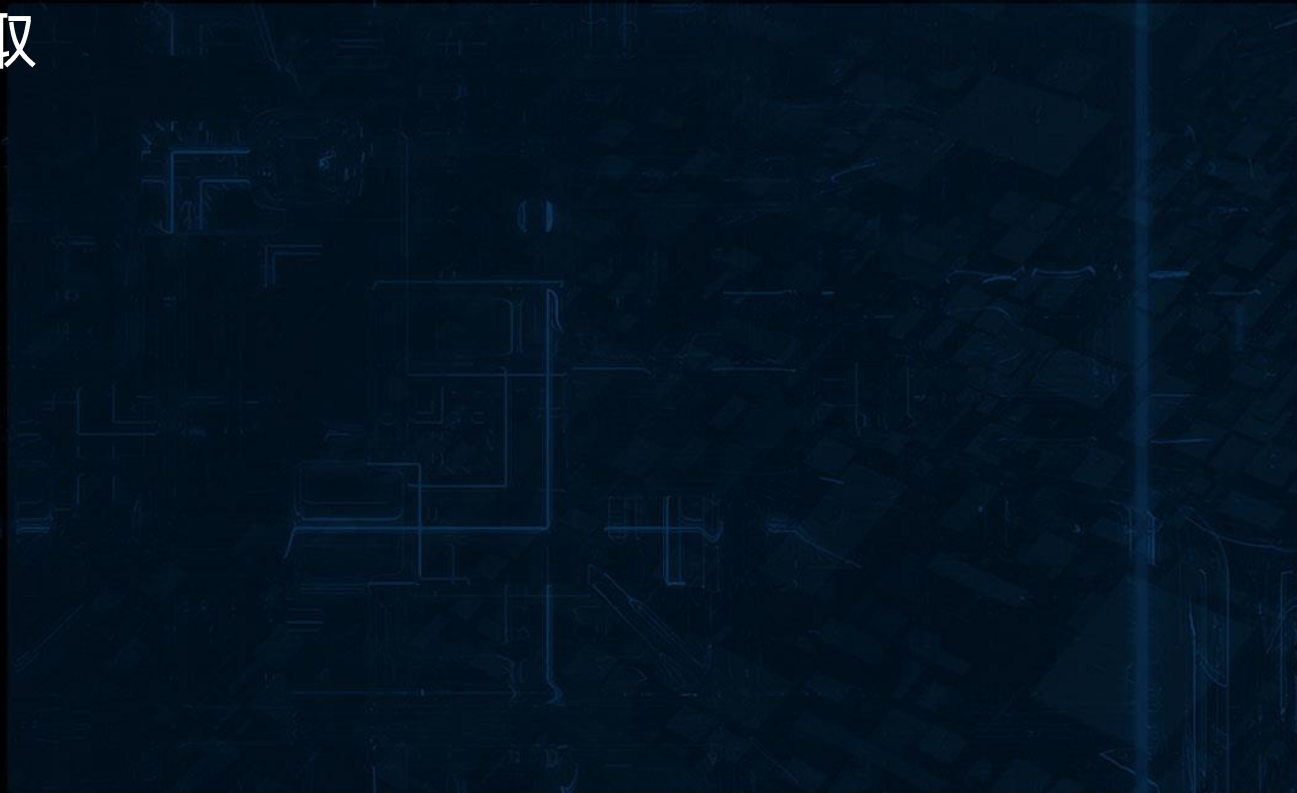
Python 环境配置和  
Anaconda平台安装



简单程序示例



## 案例一：轮廓提取





## 案例三：人脸识别

