廣東工業大學
Guangdong University of Technology

Python Foundation

GONG Yi (龚怡)

2167570874@qq.com

Link

Slides：
https://pan.baidu.com/s/1uqWdtLXJhJ9TvKN1CsK_9w

Access code ： math

Course objectives and requirements

- Target
  - Master knowledge of Python and numerical methods
- Learning requirements
  - Listen carefully and take notes
  - Classroom engagement and interaction (^_^)
  - Practiced algorithms after class

Course assessment

- Assessment methodology
  - Usual grades 30% (Homework 20%，Attendance 10%)
  - Checking examination (Course reports) 70%

- Grade distribution： Conforms to a normal distribution

- Credit Hours：36（Lecture24＋Lab12）

Course assessment

- **Practical time (tentative)：**
  - **Saturday, March 4, period 2-4 （9:20-12:00）**
  - **Saturday, April 8, period 2-4 （9:20-12:00）**
  - **Saturday, May 6, period 2-4 （9:20-12:00）**
  - **Saturday, May 13, period 2-4 (9:20-12:00）**

Course reference books

- 《数值计算》
- 《计算方法》
- 《数值方法——设计、分析和算法实现》
  - Anne Greenbaum等，机械工业出版社
- 《Python数值计算与模拟》等

- Library, online books and introductions on numerical calculation methods, Absence from class 3 times or

Failing to hand in homework 3 times or

Absence from python practical class 3 times

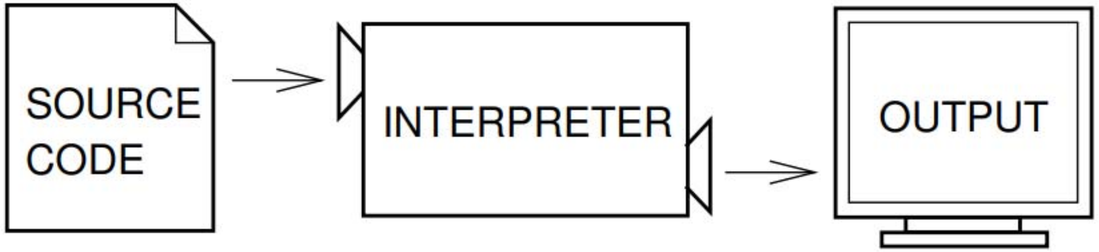
Your grade is zero！

What is Python?

- *Python* is a programming language

- The python interpreter software that reads source code and performs its instructions.

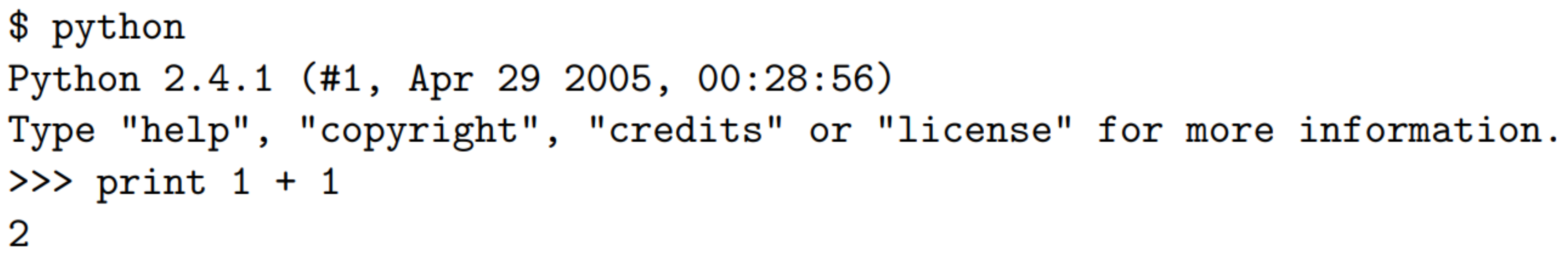
- The
ma
'or Linux,



- You can download the python interpreter for free at https://python.org/

Two ways to use the interpreter:

- command-line mode



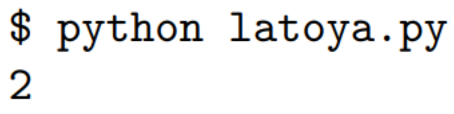
```
$ python
Python 2.4.1 (#1, Apr 29 2005, 00:28:56)
Type "help", "copyright", "credits" or "license" for more information.
>>> print 1 + 1
2
```

- script mode

latoya.py with the following contents:

print 1 + 1

And run on shell



```
$ python latoya.py
2
```

Why is named phthon?

- The name Python comes from the surreal British comedy group Monty Python, not from the snake.

- Python programmers are affectionately called Pythonistas.

Contents

Introduction

Variables

Sub-branch Statement

Loop Statement

Function

Introduction

Python is the leading language which has its application in every aspect of life which involve technology.

Think of a field and you'll find python as a supporting pillar in that field.

Instagram, YouTube, NASA, Google, Drop box, Machine Learning, Data Science, Big Data, Reddit, Linux, and the list will go on.

Introduction

- All programs use basic instructions as building blocks:
- “Do this; then do that.”
- “If this c n; otherwise,
- “Do this action exactly 27 times.”
- “Keep doing that until this condition is true.”

Python is no exception

Python Feature(1)

- Easy to learn ：Python has relatively few keywords, simple structure, and a well-defined syntax, which makes it easier to learn;
- Easy to read: Python code definition is clearer;
- Easy to maintain: Python's success is that its source code is easy to maintain;
- A wide range of standard libraries: Python is rich in the libraries, which are cross-platform and are compatible with Linux, Windows and MacOS;

Python Feature(2)

- Interactive mode: In interactive mode, you can input the python codes and get the result in the terminal, and test and debug code fragments interactively;

- Portability: Since it is open source, Python has been ported (that is, made to work) to many platforms;

- Extensible: If you need a piece of critical code that runs fast, or you want to write algorithms that don't want to be open, you can use C or C++ to complete that part of the program and call it from your Python program;

Python Feature(3)

- Database: Python provides interfaces to all major commercial databases;
- GUI programming: Python supports GUI creation and migration to many system calls;
- Embedable: You can embed Python into C/C++ programs, giving users of your program the ability to "script";

Python Application

- Scientific computing

With the development of many program libraries such as NumPy, Scipy, Matplotlib, Enthought libraries, etc., Python is more and more suitable for scientific calculations and drawing high-quality 2D and 3D images.

- Artificial Intelligence(AI)

MASA and Google used Python a lot in the early days, accumulating a rich scientific operation library for Python, when the AI era came, most of the artificial intelligence code on the market is written in Python, especially after PyTorch, Python became the first language in the AI era.

Python Application

- Data analysis

In fact, Python has become one of the standard languages and platforms for data analysis and data science. Numpy, pandas, scipy and Matplotlib libraries together form the basis of Python data analysis.

- Web Development

Python has many free data function libraries, free web page template systems, and libraries for interacting with web servers, which can realize web development and build web frameworks, and the most famous Python web framework is Django.

Python Application

- Crawler development

In the field of crawlers, Python is almost dominant. It uses all network data as resources, and use automated programs to the targeted data collection and processing

- Cloud computing development

Python is a programming language for cloud computing. OpenStack, the current popular cloud computing framework, is developed by Python.

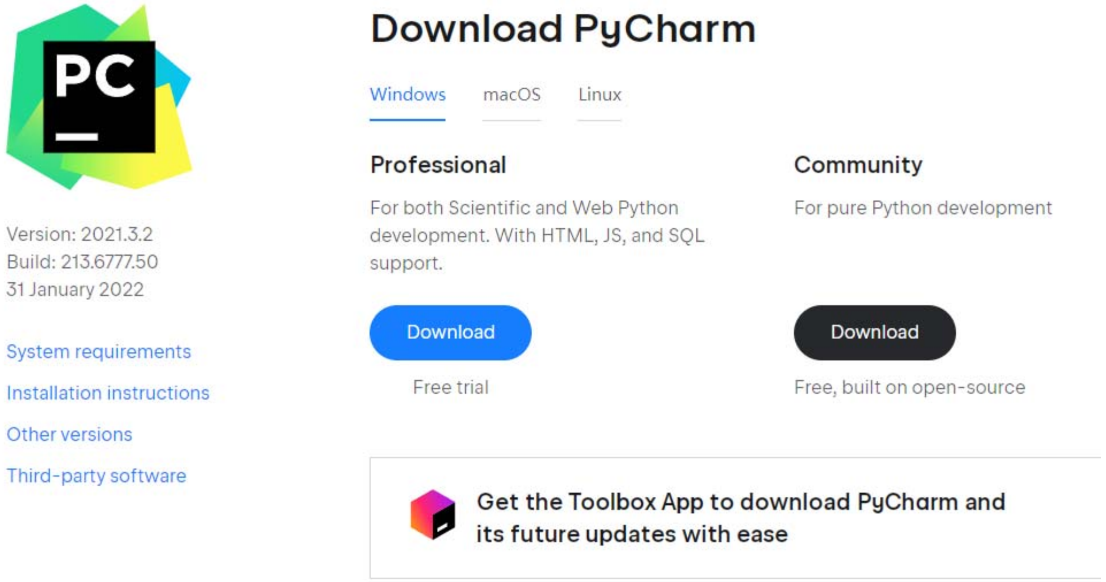
- Automatic operation and maintenance

Python is a comprehensive language that can meet most of the automated operation and maintenance needs, and can be done on both the front-end and back-end.

Installation

This Course use PyCharm As IDE development Enviroment, download and installation Community version.

https://www.jetbrains.com/pycharm/download/#section=windows



Version: 2021.3.2
Build: 213.6777.50
31 January 2022

System requirements

Installation instructions

Other versions

Third-party software

Download PyCharm

Professional

For both Scientific and Web Python development. With HTML, JS, and SQL support.

Free trial

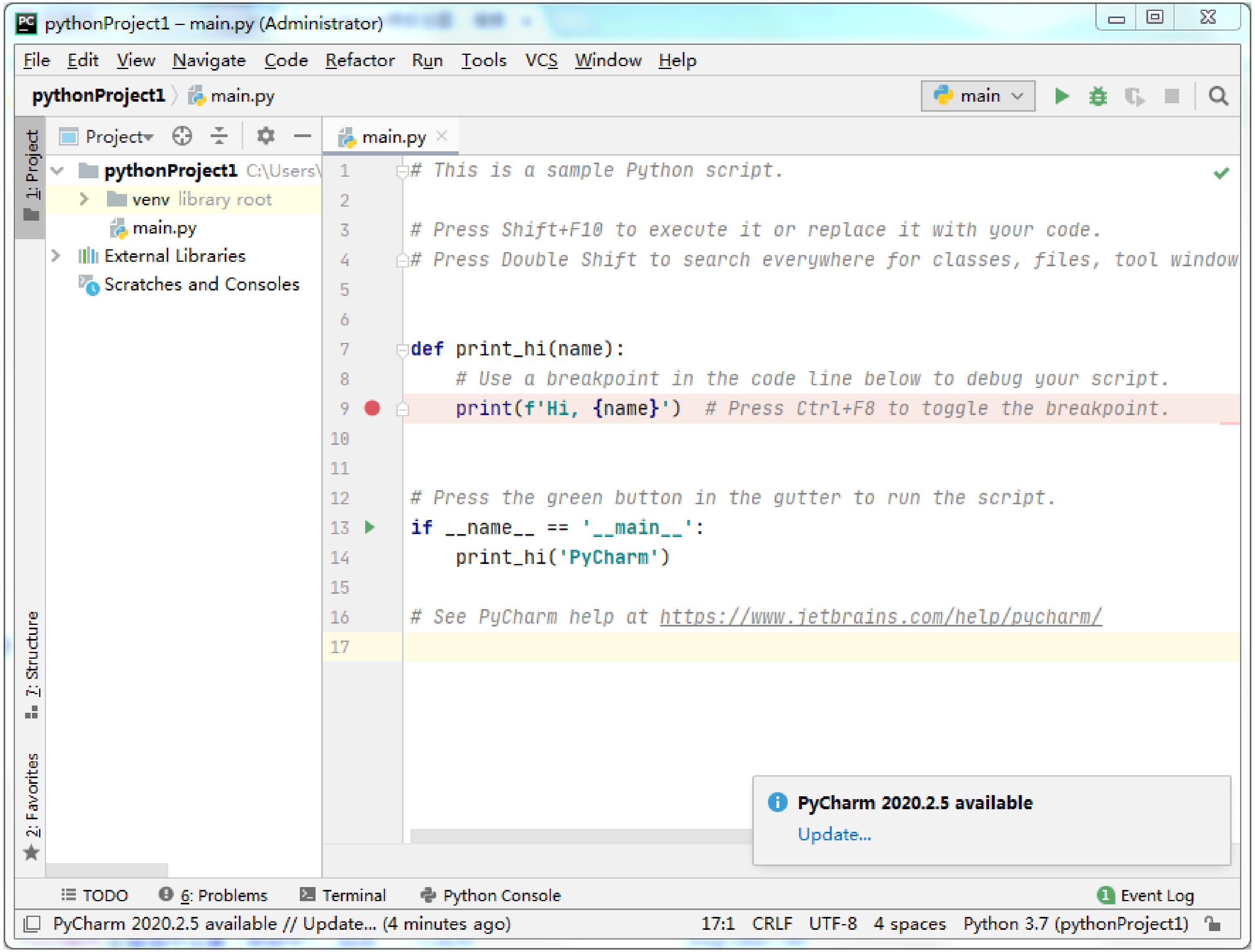
Community

For pure Python development

Free, built on open-source

Get the Toolbox App to download PyCharm and its future updates with ease

PyCharm Main Screen



```
# This is a sample Python script.

# Press Shift+F10 to execute it or replace it with your code.
# Press Double Shift to search everywhere for classes, files, tool window


def print_hi(name):
    # Use a breakpoint in the code line below to debug your script.
    print(f'Hi, {name}')  # Press Ctrl+F8 to toggle the breakpoint.


# Press the green button in the gutter to run the script.
if __name__ == '__main__':
    print_hi('PyCharm')

# See PyCharm help at https://www.jetbrains.com/help/pycharm/

```

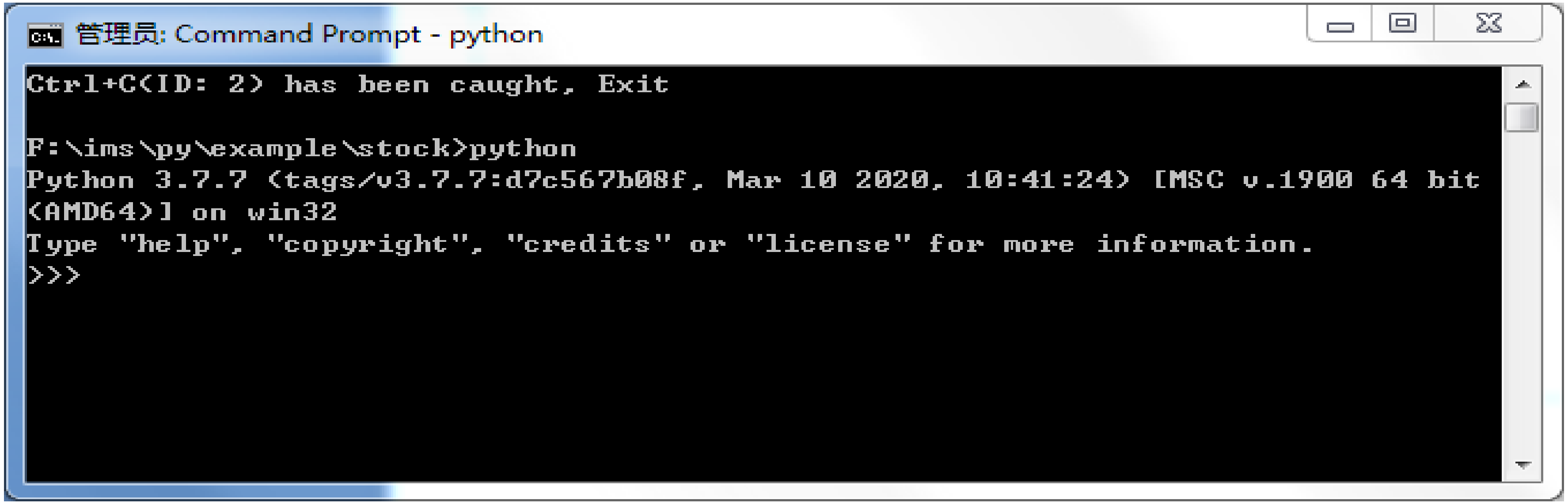
PyCharm 2020.2.5 available
Update...

PyCharm 2020.2.5 available // Update... (4 minutes ago) 17:1 CRLF UTF-8 4 spaces Python 3.7 (pythonProject1)

Interactive Programming

Interactive programming does not need to create script files, but uses the interaction mode of the Python interpreter to write code.

Just enter Python command in the command line to start interactive programming. The prompt window is as follows ：



```
Ctrl+C(ID: 2) has been caught, Exit

F:\ims\py\example\stock>python
Python 3.7.7 (tags/v3.7.7:d7c567b08f, Mar 10 2020, 10:41:24) [MSC v.1900 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Script Programming

- Call the interpreter through the script parameters to start executing the script until the script is completed. When the script execution is completed, the interpreter is no longer valid.

- Let‘s write a simple Python script program. All Python files are suffixed with .py. Copy the following source code to the test.py file：

```
a = 10;b=20;print(a+b)
```

- Run script: c:\>python test.py

- Run statement: c:\>python -c "a = 10;b=20;print(a+b)"

Print Output

- Python2.x using print “abc”,
- python 3.x using print() function
- If the print() function of Python 3. x is used under the Python 2. x, you can import future Package, which disables the print statement of Python 2. x and uses the print function of Python 3. x

Output Chinese garbled

If no encoding is specified in the Python file, an error will be reported during execution , for example:

```
#!/usr/bin/python
print (“hello, world")
```

Output error result :

```
  File "test.py", line 2
SyntaxError: Non-ASCII character '\xe4' in file test.py on line 2, but no encoding declared; see http://www.python.org/peps/pep-0263.html for details
```

Output Chinese garbled

The default encoding in Python is ASCII. Chinese characters cannot be printed correctly without modifying the encoding, so an error will be reported when reading Chinese characters.

The solution is to add following statement:

 # -*- coding: UTF-8 -*-

or

# coding=utf-8

Note: No spaces are allowed around the sign = of # coding=utf-8.

Code comment

- Single-Line Comments

# is often used as a single-line comment symbol. When # is used in the code, any data on its right will be ignored during program execution and treated as a comment.

```
>>> print(‘Hello world.’)  # output “Hello world”.
Hello world.
```

Code comment

- Python multiline comments

In Python, When there are multiple lines that require comments, you need to use multiple line comments to comment on multiple lines. Multiline comments are enclosed by three single quotes ''' or three double quotes " " " ，

For example ：

```
'''
This is a multiline comment with three single quotes
This is a multiline comment with three single quotes
This is a multiline comment with three single quotes
'''
```

Identifier

- The identifier consists of letters, numbers and underscores, but cannot start with a number;

- Identifiers in Python are case-sensitive;

- Identifiers that begin with an underscore have special meaning. Those beginning with a single underscore (_foo) represent class properties that cannot be accessed directly. They need to be accessed through the interface provided by the class. They cannot be imported with "from xxx import *"　;

Identifier

- The (__foo) beginning with a double underscore represents the private member of the class ；

- The (__foo__) beginning and ending with a double underscore represents the special method identification in python, such as__ init__ () represents the constructor fuction of the class.

- Python can display multiple statements on the same line by using a semicolon apart

Reserved Characters

assert
break
class
continue
def
del
elif
else
except
finally
for
from
global
if
import
in
is
lambda
or
pass
print
raise
return
try
while
with
yield

Standard Data Type

- Numbers
- Strings
- List
- Tuple
- Dictionary
- Set

Four Types Of Numerical

- int（signed int type）
- long（ Long integer, It can also represent octal and hexadecimal）
- float（ float type）
- complex（ complex type）

Boolean Type

- Either True or False

NULL value

- In Python language, there is a special value None, which represents null value, which is different from the logical value False, value 0, and empty string ' ‘, which means that there is no value, and it compares with any other value to be False.

String Type

Strings in Python are immutable sequences of characters enclosed in delimiters such as single quotes ('), double quotes ("), triple single quotes ('''), or triple double quotes (''''').

Create a string

A string can be created by assigning a variable a sequence of characters enclosed in string delimiters:

```
var1 = 'Hello World!'

var2 = "Hello Word!"

var3 = '''Hello Word!'''
```

String Type

Operator	Description
+	String connection
*	Repeat output string
[]	Get the characters in the string through the index
[:]	Intercept a part of a string
in	Member operator -- Returns True if the string contains a given string
not in	Member operator -- Returns True if the string does not contain the given string
r/R	Original string - precedes the first quotation mark of the string with the letter r or R, all characters in the string are used directly as they are, and no longer escaped to special or unprintable characters
%	format string

String type

```
>>> str1= 'Python'
>>> str2= ' good'
>>> str3=str1+str2      #strint cat
>>> print(str3)
Python good

>>> print (str1 * 2)    # Output string twice
PythonPython

>>> print(2* str1)
PythonPython
```

List Data Types

A list is a comma-separated list of elements written between square brackets []. The size of the list is variable, and it can increase or decrease depending on demand. The elements in a list can be of different types, and objects of data types such as numbers, strings, tuples, dictionaries, collections, etc., and can even contain lists (that is, nested).

The following are legal list objects：

- ['Google', 'Baidu', 1997, 2008]
- [1, 2, 3, 4, 5]
- ["a", "b", "c", "d"]
- [123, ["das", "aaa"], 234]

List Data Types

(1) List creation and deletion

```
>>> list1 = list()     # Create an empty list
>>> list2 = list ('chemistry')
>>> list2
['c', 'h', 'e', 'm', 'i', 's', 't', 'r', 'y']
```

You can also use "=" to create a list object by assigning a list directly to a variable：

```
>>> lista= []
>>> listb = [ 'good', 123 , 2.2, 'best', 70.2 ]
```

List Data Types

(2) List interception (also known as sharding, slicing)

- The elements in the list can use the subscript operator list [index] to access the elements in the list with the subscript index.
- List subscripts start from 0, and subscripts range from 0 to len(list)-1, len(list) gets the length of the list.
- list[index] can be used like a variable and is called a subscript variable .

List Data Types

(2) List interception (also known as sharding, slicing)

- Python allows the use of negative numbers as subscripts to refer to positions relative to the end of the list, and the list length and negative subscripts can be added to get the actual position.

For example：

```
>>> list1=[1,2,3,4,5]
>>> list1[-1]
5
```

List Data Types

（2） List interception (also known as sharding, slicing ）

- List interception operations use the syntax list[start:end] to return a fragment of the list. This fragment is a sublist of elements with subscripts from start to end-1. The list is intercepted and returns a new list containing the specified elements.

```
 >>> list1 = [ 'good', 123 , 2.2, 'best', 70.2 ]
 >>> print (list1[1:3])
[123, 2.2]
```

List Data Types

（3） Change List

- List element changes：

```
>>> x = [1,1,3,4]
>>> x[1] = 2          #Change the second 1 in the list to 2
```

- Segments alter list elements

```
>>>name = list('Perl')
>>>name[1:] = list('ython')
>>>name
['P', 'y', 't', 'h', 'o', 'n']
```

List Data Types

（3） Change List

- Insert a sequence in the list：

```
>>> number=[1,6]
>>> number[1:1]=[2,3,4,5]
>>> number
[1, 2, 3, 4, 5, 6]
```

List Data Types

(3)　Change List

- Delete an element from the list：

```
 >>> names = ['one', 'two', 'three', 'four', 'five', 'six']
 >>> del names[1]    # Delete the 2nd element of names
>>> names[1:4]=[]   # Delete the 2nd to 4th elements of names
['one', 'six']
```

List Data Types

(4) List is a sequence type

In Python, strings, lists, and tuples are sequence types.

- A sequence is an ordered arrangement of members and one or more of its members can be accessed by an offset.
- Each element in the sequence is assigned a number – its position, also known as an index, the first index is 0, the second is 1, and so on.

List Data Types

(4) List is a sequence type

- Sequences can perform operations such as indexing, slicing, adding, multiplying, and checking members.
- Python already has built-in methods for determining the length of sequences and determining the largest and smallest elements.

List Data Types

(4) List is a kind of sequence type and Common operations for sequences：

operate	description
x in s	Returns True if element x is in sequence s
x not in s	Returns True if element x is not in sequence s
s1+s2	Connect two sequences s1 and s2 to get a new sequence
s*n, n*s	Sequence s is copied n times to obtain a new sequence
s[i]	Get the ith element of sequence s
s[i:j]	Obtain the fragment of sequence s from subscript i to j-1
len(s)	Returns the number of elements contained in sequence s

List Data Types

(4) List is a kind of sequence type and Common operations for sequences：

operate	**description**
max(s)	Returns the largest element of sequence s
min(s)	Returns the smallest element of sequence s
sum(x)	Returns the sum of all elements in the sequence s
<、<=、>、>=、==、!=	Compare two sequences

List object constant functions

method	description
list.append(x)	Add a new object x at the end of the list
list.count(x)	Returns the number of times x appears in the list
list.extend(seq)	Append all elements in the seq sequence at the end of the list at one time
list.index(x)	Returns the subscript of the first element with the value of x in the list, and throws an exception if it does not exist
list.insert(index, x)	Add the element x at the index position in the list
list.pop([index])	Delete and return the element at the specified position in the list. The default is the last element
list.remove(x)	Remove the first match of x in the list
list.reverse()	Reverse the elements in the list
list. sort(key=None, reverse=None)	Sort the list
list.clear()	Delete all elements in the list, but keep the list object
list.copy()	Used to copy the list and return the new list after copying

Homework

Install PyCharm

Write your helloworld.py to show “Hello World” in the terminal