

Python tutorial

[download Python interpreter](#)

[Python IDE - PyCharm](#)

配置环境

[project 添加 Python files](#)

hello world

[drawing a shape](#)

[variables & data types](#)

working with string

[working with numbers](#)

[getting input from users](#)

[building a basic calculator](#)

[mad libs game](#)

Lists

[Lists function](#)

Tuples

[tuple is immutable](#)

[difference between list and tuple](#)

Function

[Return Statement](#)

If statement

[if statement with comparison](#)

[Building a better caculator](#)

Dictionaries

while loop

[building a guessing game](#)

For loop

[Exponent function](#)

[2D Lists & Nested Loops](#)

Build a translator

判断某个元素是否是AEIOU方法

Comments

Try Except

Reading Files

Writing Files

Pycharm Shift+Tab 取消缩进； Tab 缩进

Modules and Pip

Module 含义， 使用

Python 官方 Module 列表

官网列表中 Module 到底在哪里呢？

第一种：buind in modules

build in python language. 在安装Python时，在安装文件夹里。

打开看看里面内置的modules

在电脑中的位置

如选择其他解释器

第二种：安装 third party

python-doc

安装工具--pip

pip 安装 external or third party python modules

pip 安装的包具体位置

pip 卸载第三方包

Classes & Objects

Building a Multiple Choice Quiz

Object Function - 类成员函数

Inheritance

一个类：Chef

另一个类：ChineseChef

如何构成继承关系

继承中同名函数override（覆盖）

主文件

未采用继承

采用继承

Python Interpreter

2022/06/20 1:03 睡不着学习吧

download Python interpreter

查看 windows 系统中 Python 解释器的路径

```
▼
Bash | 复制代码

1 where python
```

```
Microsoft Windows [版本 10.0.19044.1766]
(c) Microsoft Corporation。保留所有权利。

C:\Users\ckBu>where python
D:\Software\Anaconda3\installation\python.exe
C:\Users\ckBu\AppData\Local\Microsoft\WindowsApps\python.exe
E:\my_tools\MSYS2\installation\mingw64\bin\python.exe
E:\my_tools\MSYS2\installation\clang64\bin\python.exe

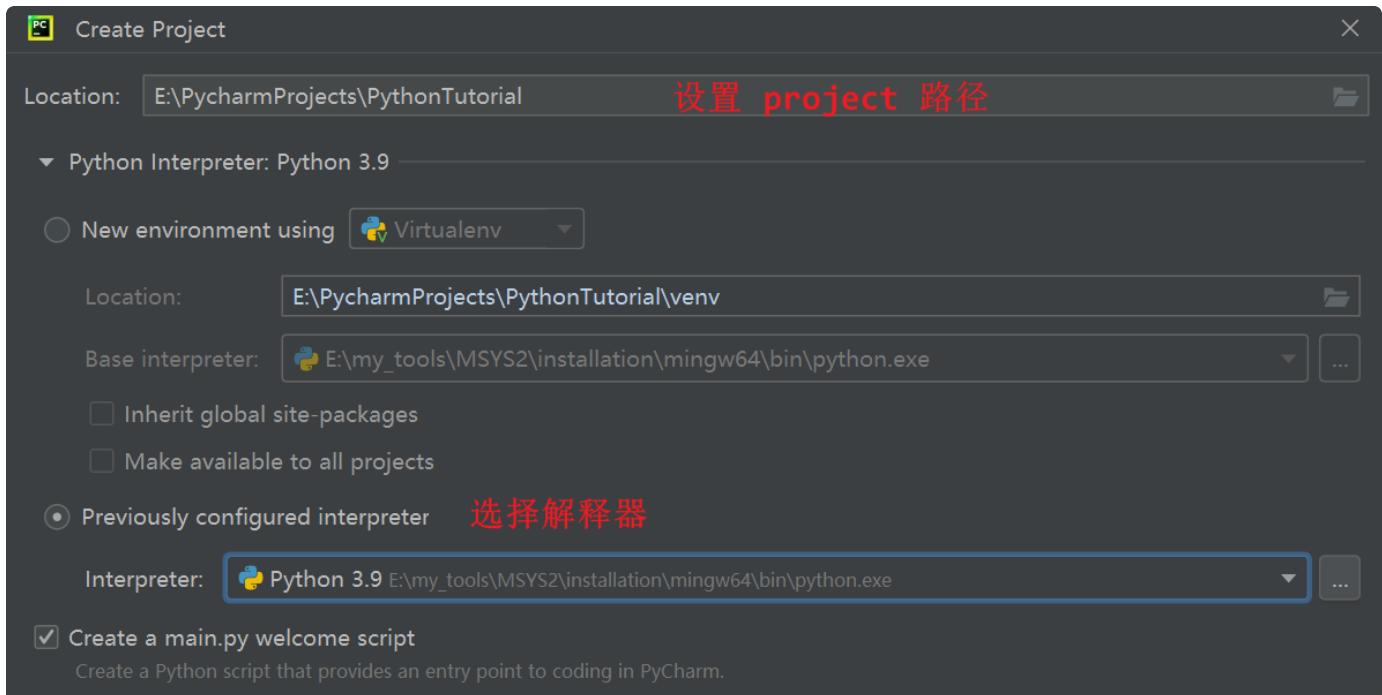
C:\Users\ckBu>
C:\Users\ckBu>
C:\Users\ckBu>python
Python 3.9.7 (default, Sep 16 2021, 16:59:28) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32

Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated. Libraries may fail to load. To activate this environment
please see https://conda.io/activation

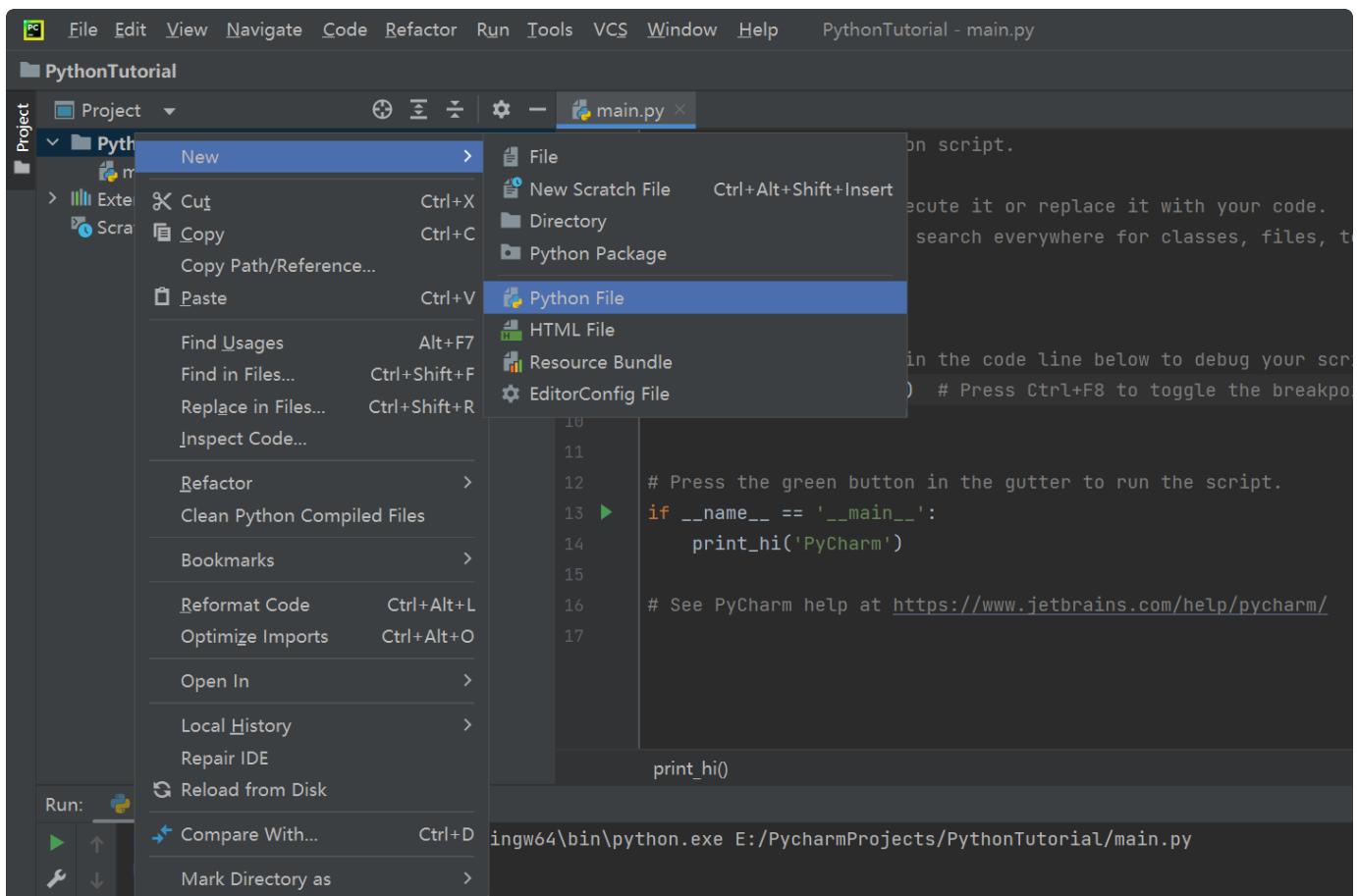
Type "help", "copyright", "credits" or "license" for more information.
>>> -
```

Python IDE – PyCharm

配置环境

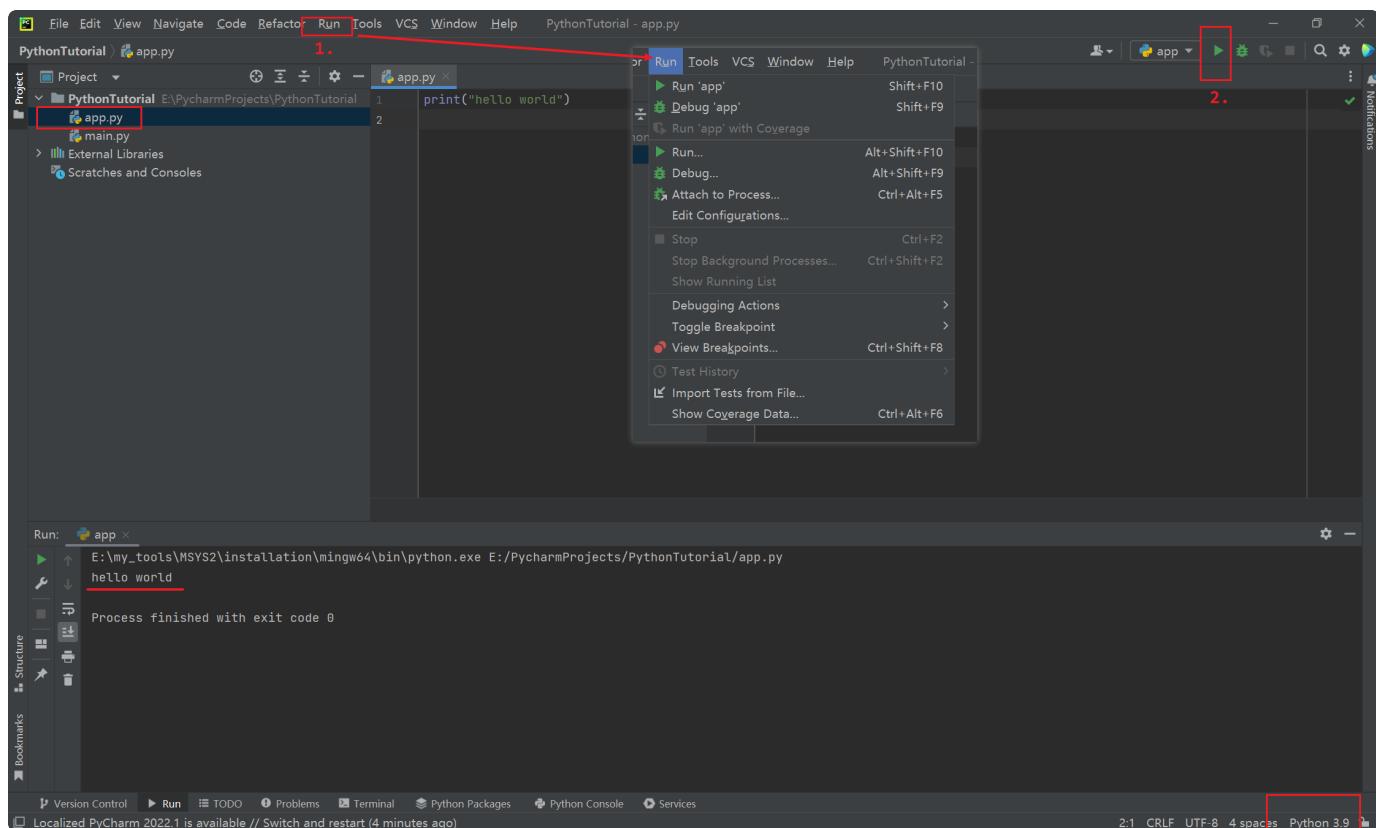


project 添加 Python files



```
1 print(" you can output exerything you want to console. ")
```

hello world



drawing a shape

The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help, and PythonTutorial - app.py. The Project tool window on the left shows a folder named 'PythonTutorial' containing files app.py, main.py, External Libraries, and Scratches and Consoles. The main code editor window displays the following Python code:

```
1 print(" /| ")
2 print(" / |")
3 print(" /  |")
4 print("/___|")
```

The Run tool window at the bottom shows the output of running the script:

```
E:\my_tools\MSYS2\installation\mingw64\bin\python.exe E:/PycharmProjects/PythonTutorial/app.py
/|
/ |
/ |
/___|
Process finished with exit code 0
```

The status bar at the bottom indicates Localized PyCharm 2022.1 is available // Switch and restart (12 minutes ago) and 5:1 CR LF UTF-8 4 spaces Python 3.9.

variables & data types

The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help, and PythonTutorial - Variables&DataTypes.py. The Project tool window on the left shows a folder named 'PythonTutorial' containing files app.py, main.py, Variables&DataTypes.py, External Libraries, and Scratches and Consoles. The main code editor window displays the following Python code:

```
1 character_name = "Tom"      # string
2 character_age = "50"        # 
3 isMale = True               # boolean
4 print("There once was a man named " + character_name + ", ")
5 print("he was " + character_age + " years old. ")

6
7 character_name = "Mike"
8 print("He really liked the name " + character_name + ", ")
9 print("but didn't like being " + character_age + ". ")
```

A red annotation '变量的声明和定义：直接写变量名+定义' is placed above the first four lines of code.

The Run tool window at the bottom shows the output of running the script:

```
E:\my_tools\MSYS2\installation\mingw64\bin\python.exe E:/PycharmProjects/PythonTutorial/Variables&DataTypes.py
There once was a man named Tom,
he was 50 years old.
He really liked the name Mike,
but didn't like being 50.
Process finished with exit code 0
```

The status bar at the bottom indicates Python Debugger Extension Available, Cython extension speeds up Python debugging // Install... How does it work? (13 minutes ago) and 7:24 CR LF UTF-8 4 spaces Python 3.9.

working with string

介绍了 string 自带的一些好用的函数 **string function**

The screenshot shows the PyCharm IDE interface. On the left, the Project tool window displays a file named 'WorkingWithString.py' under the 'PythonTutorial' project. The code in the editor is as follows:

```
1 print("Giraffe\\Academy")
2 print("Giraffe\"Academy")
3 print("Giraffe\Academy")
4 print("Giraffe\\Academy")
5
6 phrase = "Giraffe Academy"
7 print(phrase + " is cool.")
8 print(phrase.lower())
9 print(phrase.islower())
10 print(phrase.upper())
11 print(phrase.upper().isupper())
12 print(len(phrase)) # len() 求解string length
13 print(phrase[0]) # string[] 读取index处元素
14 print(phrase[len(phrase)-1])
15 print(phrase.index("a")) # string.index(" ")字符串中查找某个单词，并输出index;查找不到不输出
16 print(phrase.index("aff")) # string.index(" ")字符串中查找连续字符
17 # print(phrase.index("z")) # throw an error
18 print(phrase.replace("ffe", "ttr")) # string.replace(old, new) 字符串中替换
```

The right side of the interface shows the Python Console tab, which displays the output of the code execution. The output is:

```
E:\my_tools\MSYS2\installation\ming
Giraffe
Academy
Giraffe\"Academy
Giraffe\Academy
Giraffe\Academy
Giraffe Academy is cool.
giraffe academy
False
GIRAFFE ACADEMY
True
15
6
y
3
3
Girattr Academy
```

working with numbers

from math import * # 从 math module 中导入所有的函数

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonTutorial - WorkingWithNumbers.py
PythonTutorial > WorkingWithNumbers.py
Project External Libraries Scratches and Consoles
PythonTutorial E:\PycharmPr...
app.py main.py Variables&DataTypes.py WorkingWithNumbers.py WorkingWithString.py
1 print(2)
2 print(2.35)
3 print(-3.2)
4 print(3 + 4.5 + 3*3 + (1 + 2)/3)
5 print(10 % 3) # mod
6
7 my_number = -5
8 print(my_number)
9 print(str(my_number) + " is my number.") # str() print number to string "-5"
10 print(abs(my_number)) # abs() absolute value of negative number = |-5|
11 print(pow(9, 2)) # pow(底数, 指数) 9^2 = 81
12 print(max(4, 6)) # max()/min()
13 print(round(3.3)) # round() 四舍五入取整 = 3
14 # print(floor(4.3)) # python 库里没有这个函数
15
16 from math import *
17 print(floor(4.6)) # 从 math module 里面导入所有的函数库, 供我使用。 floor() 向下取整 = 4
18 print(ceil(4.2)) # ceil() 向上取整 = 5
19 print(sqrt(49)) # sqrt() = 7
20
```

```
E:\my_tools\MSYS2\installation\mingw64\bin\python ^
2
2.35
-3.2
17.5
1
-5
-5 is my number.
5
81
6
3
4
5
7.0
```

getting input from users

Python | 复制代码

```
variable = input() # 从终端读取输入存储在变量中
读取出来的数字字符串, 两种方法转换为数字 int() float() ✓ 推荐使用
```

The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help, and PythonTutorial - GettingInputFromUsers.py. The Project tool window on the left shows a file tree with PythonTutorial, app.py, GettingInputFromUsers.py, main.py, Variables&DataTypes.py, WorkingWithNumbers.py, and WorkingWithString.py. The GettingInputFromUsers.py file is open in the editor, containing the following code:

```
name = input("Enter your name: ") # input() 从终端读入数据, 存到变量中
age = input("Enter your age: ")
print(type(age))
print("Hello " + name + "! You are " + age + " years old.")
```

The Run tool window at the bottom shows the output of running the script. It prompts for 'name' and 'age', and then prints the result: 'Hello Jone! You are 34\ years old.' The status bar at the bottom right indicates 8:1 CRLF UTF-8 4 spaces Python 3.9.

building a basic calculator

The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help, and PythonTutorial - BuildingABasicCalculator.py. The Project tool window on the left shows a file tree with PythonTutorial, app.py, BuildingABasicCalculator.py, GettingInputFromUsers.py, main.py, Variables&DataTypes.py, WorkingWithNumbers.py, and WorkingWithString.py. The BuildingABasicCalculator.py file is open in the editor, containing the following code:

```
number1 = input("Enter a number: ")
number2 = input("Enter another number: ")

result = int(number1) + float(number2) # number1 and number2 are string. so plus is concat two numbers not add.
# 两种方式将 input 读取进来的字符串转化为数字, int() float() 我们建议都使用 float() 做转换
print(result)
```

The Run tool window at the bottom shows the output of running the script. It prompts for 'number1' and 'number2', and then prints the result: '8.7'. The status bar at the bottom right indicates 7:1 CRLF UTF-8 4 spaces Python 3.9.

2022/06/20 玩好了学习

mad libs game

The screenshot shows the PyCharm IDE interface. The project navigation bar at the top lists files like app.py, BuildingABasicCalculator.py, GettingInputFromUsers.py, MadLibsGame.py, main.py, Variables&DataTypes.py, WorkingWithNumbers.py, and WorkingWithString.py. Below the navigation bar, the main editor window displays the Python code for the Mad Libs game:

```
color = input("Enter a color: ")
plural_noun = input("Enter a plural noun: ")
celebrity = input("Enter a celebrity: ")

print("Roses are " + color)
print(plural_noun + " are blue")
print("I love " + celebrity)
```

Underneath the code editor, the 'Run' tab is selected, showing the command E:/my_tools/MSYS2/installation/mingw64/bin/python.exe E:/PycharmProjects/PythonTutorial/MadLibsGame.py. The run output window shows the program's interaction with the user:

```
E:/my_tools/MSYS2/installation/mingw64/bin/python.exe E:/PycharmProjects/PythonTutorial/MadLibsGame.py
Enter a color: Red
Enter a plural noun: tree
Enter a celebrity: Talye
Roses are Red
tree are blue
I love Talye

Process finished with exit code 0
```

At the bottom of the interface, there are tabs for Version Control, Run, Python Packages, TODO, Python Console, Problems, Terminal, and Services. A status bar at the very bottom indicates the file is Localized and shows the Python version as Python 3.9.

Lists

The screenshot shows the PyCharm IDE interface with the following details:

- File Menu:** File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help.
- Project:** PythonTutorial E:\PycharmProjects\PythonTutorial
- Code Editor:** The file `Lists.py` contains the following Python code:

```
friends = ["Kevin", "Karen", "Jim", 50, True]      # list []
#       0       1       2       3       4
#     -5      -4      -3      -2      -1
print(friends)    # 输出 list 中所有元素
print(friends[3])  # 索引输出指定元素
print(friends[-1]) # back of the lists output 使用负数
print(friends[1:]) # 输出1~最后
print(friends[1:3]) # [1, 3) = [1, 2]

friends[1] = "Mike"      # 通过索引修改list中某个位置的值
Lists
friends: list[Union[str, int, bool]] = ["Kevin", "Karen", "Jim", 50, True]
```
- Run Tab:** Shows the run history for the script `Lists.py`. The output is:

```
E:\my_tools\MSYS2\installation\mingw64\bin\python.exe E:/PycharmProjects/PythonTutorial/Lists.py
['Kevin', 'Karen', 'Jim', 50, True]
50
True
['Karen', 'Jim', 50, True]
['Karen', 'Jim']
['Kevin', 'Mike', 'Jim', 50, True]

Process finished with exit code 0
```

Lists function

The screenshot shows the PyCharm IDE interface with the following details:

- Project:** PythonTutorial
- File:** ListsFunction.py
- Code Content:**

```
lucky_numbers = [4, 6, 15, 16, 23, 24]
friends = ["kevin", "karen", "Jim", "Oscar", "Toby", "Jim"]
# 增
# friends.extend(lucky_numbers) # list.extend() 在list后面添加 iterator,也就是添加其他 list 的每个元素
# friends.append("Mike") # list.append() 在 list 后面添加一个 object
# friends.insert(1, "Vicky") # list.insert(index, object) 在list index索引处插入一个 object
# friends.insert(2, lucky_numbers) # 把整个list对象当作一个object插入到list中
# 删
# friends.remove(friends[1]) # list.remove() 删除一个object
# friends.clear() # list.clear() 清空list
# friends.pop(2) # list.pop(index) pop the element in index / last element
# 查
print(friends.index("Oscar")) # list.index(element) find the index of the element if it exists.
print(friends.count("Jim"))
# 排序
friends.sort() # list.sort() 默认ascend, reverse=True descend
print(friends)
lucky_numbers.sort(reverse=True)
print(lucky_numbers)
friends.reverse() # list.reverse() list倒序输出
print(friends)
# 以上的所有操作都会修改 list 中 element
# 拷贝
friends_copy = friends.copy() # list.copy() 拷贝一份给新的list
print(friends_copy)
```

- Bottom Status Bar:** Version Control, Run, Python Packages, TODO, Python Console, Problems, Terminal, Services. Localized PyCharm 2022.1 is available // Switch and restart (59 minutes ago). 22:29 CRLF UTF-8 4 spaces Python 3.9.

Tuples

container, can store different types of values.

open and close parentheses = ()

open and close square bracket = []

tuple is immutable

cannot be changed or modified;

once you create a tuple and you can't change it.

```
# coordinates[0] = 10    # tuple is immutable
print(coordinates)
```

```
coordinates[0] = 10
TypeError: 'tuple' object does not support item assignment
(4, 5)
5
```

```
coordinates = (4, 5) # immutable; cannot be changed or modified; once you create a tuple and you c ↵1 ^ v
print(coordinates) # access element in tuple. tuple[]
print(coordinates[1])
# coordinates[0] = 10 # tuple is immutable
print(coordinates)
```

```
E:\my_tools\MSYS2\installation\mingw64\bin\python.exe E:/PycharmProjects/PythonTutorial/Tuple.py
Traceback (most recent call last):
  File "E:/PycharmProjects/PythonTutorial/Tuple.py", line 4, in <module>
    coordinates[0] = 10
TypeError: 'tuple' object does not support item assignment
(4, 5)
5

Process finished with exit code 1
```

difference between list and tuple

大部分时候使用的是 List 数据结构；只要特殊场景下使用 tuple 数据结构。

一般我不想修改数据结构内的值，使用 tuple，因为它具有 immutable 属性。

Function

collection of code.

a bunch of lines of code that are basically doing one thing.

```
1 # keyword: def      function name: say_hi      parameters: none or lots 冒号
2         indented(缩进)
3
4 print(say_hi("World"))
5
6 # output: Hello, World!
```

```
2 { def say_hi():
3     name = input("Enter your name: ")
4     print("Hello User " + name)
5 }
```

The screenshot shows the PyCharm IDE interface with a Python project named 'PythonTutorial'. The 'Functions.py' file is open in the editor. The code contains three functions: `say_hi`, `say_age`, and `say_information`. The `say_hi` function uses `input` to get a name and `print` to greet the user. The `say_age` function takes an age parameter and prints a message. The `say_information` function takes a name and age, and prints both. A red curly brace highlights the opening brace of the `say_hi` function. The right panel shows the terminal output of the script execution, which includes the prompt 'Enter your name:', the user input 'Nio', the greeting 'Hello User Nio', the call to `say_age(25)`, the output 'Hello 25', and the final call to `say_information("Mike", 22)` with the output 'Hello Mike, you are 22'. The terminal also shows the path 'E:\my_tools\MSYS2\installation\mingw64\bin\python.exe' and the exit code 'Process finished with exit code 0'.

Return Statement

return keyword. It just break back out and never be reached.

```
def cube(number):
    return number * number * number

result = cube(4)
print(result)
```

Run: ReturnStatement
E:\my_tools\MSYS2\installation\mingw64\bin\python.exe E:/PycharmProjects/PythonTutorial/ReturnStatement.py
64
Process finished with exit code 0

If statement

```
if is_male and is_tall: # if statement with multi-conditions using
# keywords: or / and
print("You are a tall male.")
elif is_male and not is_tall:    # elif 多条件分支    not 非
```

The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help, and PythonTutorial - IfStatement.py. The left sidebar has a Project view with files like app.py, BuildingABasicCalculator.py, Functions.py, GettingInputFromUsers.py, IfStatement.py, IfStatementComparisons.py, Lists.py, ListsFunction.py, MadLibsGame.py, main.py, ReturnStatement.py, Tuple.py, Variables&DataTypes.py, WorkingWithNumbers.py, and WorkingWithString.py. The main editor window contains the following Python code:

```
is_male = True
is_tall = False

if is_male or is_tall:    # if statement with multi-conditions using keywords: or / and
    print("You are a male or tall or both.")
else:
    print("You neither male nor tall.")

if is_male and is_tall:
    print("You are a tall male.")
elif is_male and not is_tall:
    print("You are a short male.")
elif not is_male and is_tall:
    print("You are not a male but a tall.")
else:
    print("You are short female.")

elif is_male and not is_tall
```

The Run tab shows the output of the script:

```
E:\my_tools\MSYS2\installation\mingw64\bin\python.exe E:/PycharmProjects/PythonTutorial/IfStatement.py
You are a male or tall or both.
You are a short male.

Process finished with exit code 0
```

if statement with comparison

求三个数中的最大值

The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help, and PythonTutorial - IfStatementComparisons.py. The left sidebar has a Project view with files like app.py, BuildingABasicCalculator.py, Functions.py, GettingInputFromUsers.py, IfStatement.py, IfStatementComparisons.py, Lists.py, ListsFunction.py, MadLibsGame.py, main.py, ReturnStatement.py, Tuple.py, Variables&DataTypes.py, WorkingWithNumbers.py, and WorkingWithString.py. The main editor window contains the following Python code:

```
def max_num(number1, number2, number3):
    if number1 > number2:
        if number1 >= number3:
            max_number = number1
        else:
            max_number = number3
    else:
        if number2 >= number3:
            max_number = number2
        else:
            max_number = number3
    return max_number

def max_num_three(num1, num2, num3):
    if num1 >= num2 and num1 >= num3:
        max_number = num1
    elif num2 >= num1 and num2 >= num3:
        max_number = num2
    elif num3 >= num1 and num3 >= num2:
        max_number = num3
    return max_number

def max_num_three_list(num1, num2, num3):
    list = [num1, num2, num3]
    return max(list)

print(max_num(2, 2, 2))
print(max_num_three(3, 3, 3))
print(max_num_three_list(3, 1, 5))
```

Building a better calculator

Python | 复制代码

```
1 Python中判断两个变量类型是否相同:  
2     isinstance(operator, int/str/float/...)
```

The screenshot shows the PyCharm IDE interface with the following details:

- File Menu:** File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help.
- Project:** PythonTutorial (E:\PycharmPro).
- Code Editor:** The file AdvancedCalculator.py is open, displaying the following Python code:

```
1 def advanced_caculator(number1, operator, number2):  
2     if not isinstance(operator, str):  
3         print("The operator must be str datatype.")  
4     if operator == "+":  
5         answer = number1 + number2  
6     elif operator == "-":  
7         answer = number1 - number2  
8     elif operator == "*":  
9         answer = number1 * number2  
10    elif operator == "/":  
11        if number2 == 0:  
12            answer = 100000  
13        else:  
14            answer = 1.0 * number1 / number2  
15    else:  
16        print("Invalid operator.")  
17        answer = -1.1111  
18    return answer  
19  
20  
21  
22    num1 = input("Enter first number:")  
23    op = input("Enter operator:")  
24    num2 = input("Enter second number:")  
25    print(advanced_caculator(float(num1), op, float(num2)))
```

- Run Tab:** Run: AdvancedCalculator.
- Bottom Status Bar:** Process finished with exit code 0, Version Control, Run, Python Packages, TODO, Python Console, Problems, Terminal, Services, 26:1 CRLF UTF-8 4 spaces Python 3.9.

Dictionaries

```

1 open and closed curly bracket = { }
2
3 Dictionary {key: value, }
4
5 Dictionary keys are unique! 字典的 key 必须不同。
6
7 访问: dictionary[key] -> get value
8         dictionary.get(key) -> get value
9

```

The screenshot shows the PyCharm IDE interface with two code snippets side-by-side.

Left Snippet:

```

monthConversionNumerical = {
    1: "January",
    2: "February",
    3: "March",
    4: "April",
    5: "May",
    6: "June",
    7: "July",
    8: "August",
    9: "September",
    10: "October",
    11: "November",
    12: "December",
}
print(monthConversionNumerical[1])
print(monthConversionNumerical.get("Oct"))

print(monthConversionNumerical[1])
print(monthConversionNumerical.get(6))

```

Right Snippet:

```

monthConversion = {
    "Jan": "January",
    "Feb": "February",
    "Mar": "March",
    "Apr": "April",
    "May": "May",
    "Jun": "June",
    "Jul": "July",
    "Aug": "August",
    "Sep": "September",
    "Oct": "October",
    "Nov": "November",
    "Dec": "December",
}

```

The Project sidebar on the left lists various Python files in the 'PythonTutorial' project. The bottom navigation bar includes tabs for Version Control, Run, Python Packages, TODO, Python Console, Problems, Terminal, and Services. A status bar at the bottom right shows the file is 33.1_CRLF_UTF-8_4 spaces_Python 3.9.

while loop

The screenshot shows the PyCharm IDE interface. The project tree on the left lists files like AdvancedCalculator.py, app.py, etc. The main editor window displays a Python script named WhileLoop.py with the following code:

```
i = 1
while i <= 10:
    print(i)
    i += 1
print("Done with loop.")
print("now the i is " + str(i))
```

The output window on the right shows the execution results:

```
E:\my_tools\MSYS2\installation\m:
1
2
3
4
5
6
7
8
9
10
Done with loop.
now the i is 11
```

building a guessing game

The screenshot shows the PyCharm IDE interface. The project tree on the left lists files like AdvancedCalculator.py, app.py, etc. The main editor window displays a Python script named BuildingAGuessingGame.py with the following code:

```
secret_word = "giraffe"
guess = ""  
定义全局变量

def guess_word(secretword):...
def guessWord(secret_word_):
    # global guess
    while guess != secret_word_: 函数中使用了全局变量。
        guess = input("Enter guess:")

# guess_word(secret_word)
guessWord(secret_word)

guessWord() > while guess != secret_word_
```

The code contains a syntax error at line 20 where it tries to use the global variable 'guess' without the 'global' keyword. The PyCharm editor highlights this with a red box and a warning message: "函数中使用了全局变量" (A global variable was used in the function).

The run tab at the bottom shows the command line and the resulting traceback:

```
E:\my_tools\MSYS2\installation\mingw64\bin\python.exe E:/PycharmProjects/PythonTutorial/BuildingAGuessingGame.py
Traceback (most recent call last):
  File "E:\PycharmProjects\PythonTutorial\BuildingAGuessingGame.py", line 25, in <module>
    guessWord(secret_word)
  File "E:\PycharmProjects\PythonTutorial\BuildingAGuessingGame.py", line 20, in guessWord
    while guess != secret_word_:
      UnboundLocalError: local variable 'guess' referenced before assignment 报错：在使用变量“guess”之前没有赋值
```

Python | 复制代码

- 1 函数体内使用了全局变量，需要明确指出：使用 `global` 关键字。
- 2 不然默认就是函数里能使用的局部变量。

```
secret_word = "giraffe"
guess = ""
guess_count = 0    # 玩家当前猜的次数
guess_limit = 3    # 玩家总共可以猜的次数
out_of_guess = False  # 表示玩家次数已尽，结束游戏

def guess_word(secretword):...

def guessWord(secret_word_):...

def guessWordOfficial(secret_word_guess):
    global guess
    global guess_count
    global guess_limit
    global out_of_guess

    while guess != secret_word_guess and not out_of_guess:
        if guess_count < guess_limit:
            guess = input("Enter guess: ")
            guess_count += 1
        else:
            out_of_guess = True

    # guess_word(secret_word)
    # guessWord(secret_word)
    guessWordOfficial(secret_word)
    if out_of_guess:
        print("Out of guesses, YOU LOSE!")
    else:
        print("You Win!")


```

For loop

```

1 # 输出不换行, 对参数执行 end="" 即可; 如果 end=" "则输出后面空一格。
2 print(letter, end="")
3
4 # range 使用。
5 range(a) # [0, a-1)
6 range(a, b) # [a, b-1) ; b > a
7
8 # 除就是 1 / 2 = 0.5
9 # 除法运算
10 result = 1 / result

```

```

# for loop in str
for letter in "Giraffe Academy":
    if letter != ' ':
        print(letter, end="") # 输出不换行, 对参数执行 end="" 即可; 如果 end=" "则输出后面空一格。
    else:
        print(letter)

# for loop in array
friends = ["Jam", "Mike", "Nico", "Ram"]
for name in friends:
    print(name)

for index in range(10): # 遍历 [0, 10)
    print(index)

for index in range(5, 10): # 遍历 [5, 10)
    print(index)

for index in range(len(friends)): # 遍历 [0, 4]
    print(friends[index])

```

Exponent function

$\star\star$ 表示指数

```

print(2*4) # 2 multiply 4 = 8
print(2**4) # 2 exponential 4 = 2^4 = 16

```

```
Project BuildingAGuessingGame.py ForLoops.py ExponentFunction.py
PythonTutorial E:\PycharmPr
AdvancedCalculator.py
app.py
BuildingABasicCalculator.py
BuildingAGuessingGame.py
Dictionaries.py
ExponentFunction.py
ForLoops.py
Functions.py
GettingInputFromUsers.py
IfStatement.py
IfStatementComparisons.py
Lists.py
ListsFunction.py
MadLibsGame.py
main.py
ReturnStatement.py
Tuple.py
Variables&DataTypes.py
WhileLoop.py
WorkingWithNumbers.py
WorkingWithString.py

> External Libraries
Scratches and Consoles

1 print(2*4) # 2 multiply 4 = 8
2 print(2**4) # 2 exponential 4 = 2^4 = 16
3
4
5 def raise_to_power(base_num, pow_num): # base^pow 指数函数。
6     result = 1
7     for pow_ in range(abs(pow_num)): # 控制循环几次，也就相应的 base_num 自己乘了多少次
8         result = result * base_num
9
10    if pow_num < 0:
11        result = 1 / result # 除就是 1 / 2 = 0.5
12
13    return result
14
15
16 print(raise_to_power(2, -3))
17
```

2D Lists & Nested Loops

```
Project BuildingAGuessingGame.py ForLoops.py ExponentFunction.py 2DListsandNestedLoop.py
PythonTutorial E:\PycharmPr
2DListsandNestedLoop.py
AdvancedCalculator.py
app.py
BuildingABasicCalculator.py
BuildingAGuessingGame.py
Dictionaries.py
ExponentFunction.py
ForLoops.py
Functions.py
GettingInputFromUsers.py
IfStatement.py
IfStatementComparisons.py
Lists.py
ListsFunction.py
MadLibsGame.py
main.py
ReturnStatement.py
Tuple.py
Variables&DataTypes.py
WhileLoop.py
WorkingWithNumbers.py
WorkingWithString.py

> External Libraries
Scratches and Consoles

1 number_grid =
2     [1, 2, 3, 4, 5],
3     [11, 22, 33, 44],
4     [111, 222, 333],
5     [0, 0]
6
7
8 # nested loop = two for loop
9 for row in number_grid: # row 变量就是行，也就是 2D list 中的 list
10    for col in row: # col 变量就是行中的元素，也就是 list 中的元素
11        print(col, end=" ") # 表示不换行，后面接空格
12    print() # 表示换行
13
```

Run: 2DListsandNestedLoop

E:\my_tools\MSYS2\installation\mingw64\bin\python.exe E:/PycharmProjects/PythonTutorial/2DListsandNestedLoop.py

1 2 3 4 5
11 22 33 44
111 222 333
0 0

Process finished with exit code 0

Build a translator

判断某个元素是否是AEIOU方法

Python | 复制代码

```
1 # 判断某个元素是否为元音, if letter in "AEIOUaeiou": 即可
2 if letter in "AEIOUaeiou":
3 # if letter.lower() in "aeiou":
```

4

```
1 # Giraffe Language 用字母 g 替换单词中的元音字母
2 # vowels(a e i o u) -> g
3 #
4 # dog -> dg
5
6 # str 可以相加; 单引号字符可以和双引号字符串一起相加。
7 letter1 = 'a'
8 letter2 = "b"
9 trans = ""
10 print(trans + letter1 + letter2)
11
12
13 def translate(phrase):
14     translation = ""
15     for letter in phrase:
16         # if letter == 'a' or letter == 'e' or letter == 'i' or letter == 'o' or \
17         #     letter == 'A' or letter == 'E' or letter == 'I' or letter == 'O' or letter == 'U':
18         if letter in "AEIOUaeiou": # 判断某个元素是否为元音, if letter in "AEIOUaeiou": 即可
19             # if letter.lower() in "aeiou":
20                 letter = 'g'
21             translation += letter
22
23
24
25 print(translate("HelloWorld"))
26 print(translate(input("Enter a phrase: ")))
27
```

Comments

Python | 复制代码

```
1 hashtag # use for comments
2
```

Try Except

```

1 except ZeroDivisionError as zeroErr: # 把错误类型存储为变量，打印变量即可具体显示错误原因
2     print(zeroErr)

```

The screenshot shows the PyCharm IDE interface. The code editor displays the following Python script:

```

1 # catch 的异常过于笼统
2 try:
3     number = int(input("Enter a number: "))
4     print(number)
5 except:
6     print("Invalid input. Please enter a number.")
7
8
9 # 通过在 except 后面指定错误类型来精细化定位异常类型
10 try:
11     number = int(input("Enter a number: "))
12     print(number)
13     num = 10 / 0
14 except ZeroDivisionError as zeroErr:
15     print(zeroErr)
16     # except ZeroDivisionError:
17     #     print("Divided by zero.")
18 except ValueError:
19     print("Invalid Error.")
20

```

The project navigation sidebar on the left lists several Python files under the 'PythonTutorial' folder, including '2DListsandNestedLoop.py', 'AdvancedCalculator.py', 'app.py', etc.

Reading Files

```

1 # 1. 读取文件
2 employee_file = open("ReadingFiles_employee.txt", "r")
3
4 # 2. 判断是否读取成功
5 if employee_file.readable():    # employee_file.readable()/writeable() 文
件是够可读/可写 返回 boolean
6 # 3. 三种方式读取整个文件内容
7     # 方式一:
8     # print(employee_file.read())  # employee_file.read() 读取
9     print("-----")
10    # 方式二:
11    # print(employee_file.readline())  # 读取第一行
12    # print(employee_file.readline())  # 紧接着读取第二行等等
13    # 方式三:
14    employee_list = employee_file.readlines()    # readlines 把所有行存储在
list 中
15    for line in employee_list:
16        print(line)
17
18 # 4. 关闭文件
19 employee_file.close()    # 关闭文件
20

```

```

# txt, csv, html and so on.
employee_file = open("ReadingFiles_employee.txt", "r")    # 读取文件

if employee_file.readable():    # employee_file.readable()/writeable() 文件是够可读/可写 返回 boolean
    # 方式一:
    # print(employee_file.read())  # employee_file.read() 读取
    print("-----")
    # 方式二:
    # print(employee_file.readline())  # 读取第一行
    # print(employee_file.readline())  # 紧接着读取第二行等等
    # 方式三:
    employee_list = employee_file.readlines()    # readlines 把所有行存储在 list 中
    for line in employee_list:
        print(line)

employee_file.close()    # 关闭文件

```

Writing Files

Pycharm Shift+Tab 取消缩进； Tab 缩进

```
1     "w"  文件写操作
2
3 1. 已有文件写，内容会覆盖原内容
4 2. 将内容写进一个新的文件中
5
6 "a" append操作，在文件末尾添加内容
7 1. 执行一次，就会在文件末尾添加一次，"\n"写的时候注意换行
```

The screenshot shows a PyCharm interface with three tabs: 'ReadingFiles.py', 'WritingFiles.py', and 'ReadingFiles_employee.txt'. The 'WritingFiles.py' tab contains Python code demonstrating file writing modes ('w' vs 'a'). The 'ReadingFiles_employee.txt' tab shows the resulting file content.

Code in WritingFiles.py:

```
1 import os.path
2
3 # employee_file = open("ReadingFiles_employee.txt", "r")
4 # employee_file = open("ReadingFiles_employee.txt", "a")      # "a" means append behind
5 employee_file = open("ReadingFiles_employee.txt", "w")      # only "w" means override
6
7 try:
8     print("-----")
9     employee_file.write("\nToby - Human Resources")           文件写状态下，会覆盖原有文件内容
10    employee_file.write("\nKelly - Customer Servers")
11
12 except ValueError as readErr:
13     print(readErr)
14
15 employee_file.close()
```

Content of ReadingFiles_employee.txt:

```
1
2 Toby - Human Resources
3 Kelly - Customer Servers
```

```
main.py 16
ReadingFiles.py 17 # write in a new file
ReadingFiles_employee.txt 18 employee_new_file = open("WritingNewFiles.txt", "w")
ReturnStatement.py 19
TryExcept.py 20 employee_new_file.write("Taiwan is China.")
Tuple.py 21
Variables&DataTypes.py 22 employee_new_file.close()
WhileLoop.py 23
WorkingWithNumbers.py
WorkingWithString.py
WritingFiles.py
WritingNewFiles.txt 23
```

将内容写进一个新的文件中。

Modules and Pip

Module 含义，使用

Python | 复制代码

- 1 Module is just python file that we can import into our current python file.
- 2 Module contains useful functions or so on.
- 3 import functionality from external python files.(so you don't need to copy any functions over into this file)
- 4 You can write something once and import into other files.
- 5 usage: import Module and you can use all of those stuff from that in your current python file.
- 6 meaning: Modules make python awesome!

Python 官方 Module 列表

Python | 复制代码

- 1 python modules
- 2
- 3 <https://docs.python.org/3/py-modindex.html>

The screenshot shows the Python Module Index page from the official Python documentation. The URL is <https://docs.python.org/3/py-modindex.html#cap-m>. The page title is "Python Module Index". A red banner at the top right reads "版本3.10.5的Python中可以使用的所有modules". Below the title is a navigation bar with links to modules: _ | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | z. The main content area is organized by letter. The 'a' section contains entries like abc, aifc, argparse, array, ast, asynchat, asyncio, asyncore, atexit, and audiooop. The 'b' section contains base64, bdb, binascii, binhex, and bisect. Each entry includes a brief description of its purpose.

	Description
future	Future statement definitions
main	The environment where top-level code is run. Covers command-line interfaces, import-time behavior, and ``__name__ == '__main__'``.
_thread	Low-level threading API.
a	Abstract base classes according to :pep:`3119`. Deprecated: Read and write audio files in AIFF or AIFC format. Command-line option and argument parsing library. Space efficient arrays of uniformly typed numeric values. Abstract Syntax Tree classes and manipulation. Deprecated: Support for asynchronous command/response protocols. Asynchronous I/O. Deprecated: A base class for developing asynchronous socket handling services. Register and execute cleanup functions. Deprecated: Manipulate raw audio data.
b	RFC 4648: Base16, Base32, Base64 Data Encodings; Base85 and Ascii85 Debugger framework. Tools for converting between binary and various ASCII-encoded binary representations. Encode and decode files in binhex4 format. Array bisection algorithms for binary searching.

官网列表中 Module 到底在哪里呢？

第一种：buind in modules

build in python language. 在安装Python时，在安装文件夹里。

The screenshot shows the PyCharm interface with the 'PythonTutorial - ModulesAndPip.py' file open. In the Project tool window, the 'External Libraries' section is expanded, showing the 'Lib' directory under 'Python 3.8 (OpencvPractise)'. A red arrow points to this 'Lib' directory. To the right, the code editor displays a Python script named 'UsefulTools.py' with the following content:

```
Module is just python file that we can import into our current
Module contains useful functions or so on.
import functionality from external python files.(so you don't have to write everything)
You can write something once and import into other files.
usage: import Module and you can use all of those stuff from that module
meaning: Modules make python awesome!

"""

import UsefulTools

print(UsefulTools.beatles)
```

Below the code editor, the Run tool window shows the output of running the script: 'E:/my_tools/MSYS2/installation/mingw64/bin/python.exe E:/PycharmProjects/PythonTutorial/ModulesAndPip.py' followed by the list: ['John Lennon', 'Paul McCartney', 'George Harrison', 'Ringo Star']. The status bar at the bottom indicates 'Python 3.8 (OpencvPractise)'.

打开看看里面内置的modules

The screenshot shows the PyCharm interface with the 'PythonTutorial - ModulesAndPip.py' file open. In the Project tool window, the 'External Libraries' section is expanded, showing the 'Lib' directory under 'Python 3.8 (OpencvPractise)'. A large red box highlights the 'Lib' directory. To the right, the code editor displays the same 'UsefulTools.py' script as in the previous screenshot. Below the code editor, the Run tool window shows the output of running the script: 'E:/my_tools/MSYS2/installation/mingw64/bin/python.exe E:/PycharmProjects/PythonTutorial/ModulesAndPip.py' followed by the list: ['John Lennon', 'Paul McCartney', 'George Harrison', 'Ringo Star']. The status bar at the bottom indicates 'Python 3.8 (OpencvPractise)'.

在电脑中的位置

__pycache__	_compat_pickle.py	contextvars.py	keyword.py	random.py	textwrap.py
asyncio	_compression.py	copy.py	linecache.py	re.py	this.py
collections	_dummy_thread.py	copyreg.py	locale.py	reprlib.py	threading.py
concurrent	_markupbase.py	cProfile.py	lzma.py	rlcompleter.py	timeit.py
ctypes	_osx_support.py	crypt.py	mailbox.py	runpy.py	token.py
curses	_py_abc.py	csv.py	mailcap.py	sched.py	tokenize.py
dbm	_pydecimal.py	dataclasses.py	mimetypes.py	secrets.py	trace.py
distutils	_pyio.py	datetime.py	modulefinder.py	selectors.py	traceback.py
email	_sitebuiltins.py	decimal.py	netrc.py	shelve.py	tracemalloc.py
encodings	_strptime.py	difflib.py	nntplib.py	shlex.py	tty.py
ensurepip	_threading_local.py	dis.py	ntpath.py	shutil.py	turtle.py
html	_weakrefset.py	doctest.py	nturl2path.py	signal.py	types.py
http	abc.py	dummy_threading.py	numbers.py	site.py	typing.py
idlelib	aifc.py	enum.py	opcode.py	smtpd.py	uu.py
importlib	antigravity.py	filecmp.py	operator.py	smtplib.py	uuid.py
json	argparse.py	fileinput.py	optparse.py	snphdr.py	warnings.py
lib2to3	ast.py	fnmatch.py	os.py	socket.py	wave.py
logging	asynchat.py	formatter.py	pathlib.py	socketserver.py	weakref.py
msilib	asyncore.py	fractions.py	pdb.py	sre_compile.py	webbrowser.py
multiprocessing	base64.py	ftplib.py	pickle.py	sre_constants.py	xdrlib.py
pydoc_data	bdb.py	functools.py	pickletools.py	sre_parse.py	zipapp.py
site-packages	binhex.py	genericpath.py	pipes.py	ssl.py	zipfile.py
sqlite3	bisect.py	getopt.py	pkgutil.py	stat.py	zipimport.py
test	bz2.py	getpass.py	platform.py	statistics.py	
tkinter	calendar.py	gettext.py	plistlib.py	string.py	
turtledemo	cgi.py	glob.py	poplib.py	stringprep.py	
unittest	cgitb.py	gzip.py	posixpath.py	struct.py	
urllib	chunk.py	hashlib.py	pprint.py	subprocess.py	
venv	cmd.py	heapq.py	profile.py	sunau.py	
wsgiref	code.py	hmac.py	pstats.py	symbol.py	
xml	codecs.py	imaplib.py	pty.py	symtable.py	
xmlrpc	codeop.py	imghdr.py	py_compile.py	sysconfig.py	
future.py	colors.py	imp.py	pyclbr.py	tabnanny.py	
phello.foo.py	compileall.py	inspect.py	pydoc.py	tarfile.py	
_bootlocale.py	configparser.py	io.py	queue.py	telnetlib.py	
_collections_abc.py	contextlib.py	ipaddress.py	quopri.py	tempfile.py	

少量的python
build in modules
存放在安装文件夹的
Lib下

当然选择不同解释器，相应的内置modules位置也不同。

如选择其他解释器

The screenshot shows the PyCharm IDE interface. On the left, the Project tool window displays the 'External Libraries' section, which lists various Python modules. A red box highlights the 'python3.9 > E:\my_tools\MSYS2\installation\mingw64\bin\python.exe' entry. To the right, the code editor shows a Python script named 'ModulesAndPip.py'. The code imports a module named 'UsefulTools' and prints its 'beatles' attribute. A red box also highlights the 'python3.9' entry in the project's external libraries.

```
1  """
2      Module is just python file that we can import into our current python file.
3      Module contains useful functions or so on.
4      import functionality from external python files.(so you don't need to copy any
5      You can write something once and import into other files.
6      usage: import Module and you can use all of those stuff from that in your current
7      meaning: Modules make python awesome!
8  """
9
10 import UsefulTools
11
12 print(UsefulTools.beatles)
13
```

有很多开发人员，他们开发了优秀的module，那我们怎么才能使用到它们呢？

将他们作为第三方库，安装进来。

第二种：安装 third party

这些库不会在安装python时自动安装进来。根据你需求，手动针对性安装使用。

1. 找到你想要安装的第三方库，下载它们。

python-docx 列举了第三方python modules

python-doc

A screenshot of a code block. It contains a single line of code: '1 https://python-docx.readthedocs.io/en/latest/'. Above the code, there is a 'Python' language icon and a '复制代码' (Copy code) button.

python-docx.readthedocs.io/en/latest/

已导入 HoloLens 3D 资料 AR-OpenCV 消遣 科研 ISP SLAM Work Tools nnvl-邱锡鹏 The latest in mac... 花花刷题列表 convolution

python-docx 0.8.11 documentation »

Table of Contents

- python-docx
- What it can do
- User Guide
- API Documentation
- Contributor Guide

Next topic

- Installing

Useful Links

- python-docx @ GitHub
- python-docx @ PyPI
- Issue Tracker

Quick search

Go

python-docx

Release v0.8.11 (Installation)

python-docx is a Python library for creating and updating Microsoft Word (.docx) files.

What it can do

Here's an example of what *python-docx* can do:

```
from docx import Document
from docx.shared import Inches

document = Document()

document.add_heading('Document Title', 0)

p = document.add_paragraph('A plain paragraph having some ')
p.add_run('bold').bold = True
p.add_run(' and some ')
p.add_run('italic.').italic = True

document.add_heading('Heading, level 1', level=1)
document.add_paragraph('Intense quote', style='Intense Quote')

document.add_paragraph(
    'first item in unordered list', style='List Bullet'
)
document.add_paragraph(
    'first item in ordered list', style='List Number'
)

document.add_picture('monty-truth.png', width=Inches(1.25))

records = (
    (3, '101', 'Spam'),
    (2, '42', 'Eggs'),
    (1, '631', 'Spam, spam, eggs, and spam')
)
```

安装工具--pip

在python3版本中，集成安装了 pip 工具。

pip 是一个 package manager (install、manage、update、uninstall 不同的python modules)，可以用来安装 pyhton modules

检查验证 pip 已经安装好。

▼

Python | 复制代码

```
1 pip --version
```

```
C:\Users\ckBu>pip --version
pip 21.2.4 from D:\Software\Anaconda3\installation\lib\site-packages\pip (python 3.9)
C:\Users\ckBu>
C:\Users\ckBu>
C:\Users\ckBu>python
Python 3.9.7 (default, Sep 16 2021, 16:59:28) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated. Libraries may fail to load. To activate this environment
please see https://conda.io/activation
Type "help", "copyright", "credits" or "license" for more information.
>>> -
```

pip 安装 external or third party python modules

▼

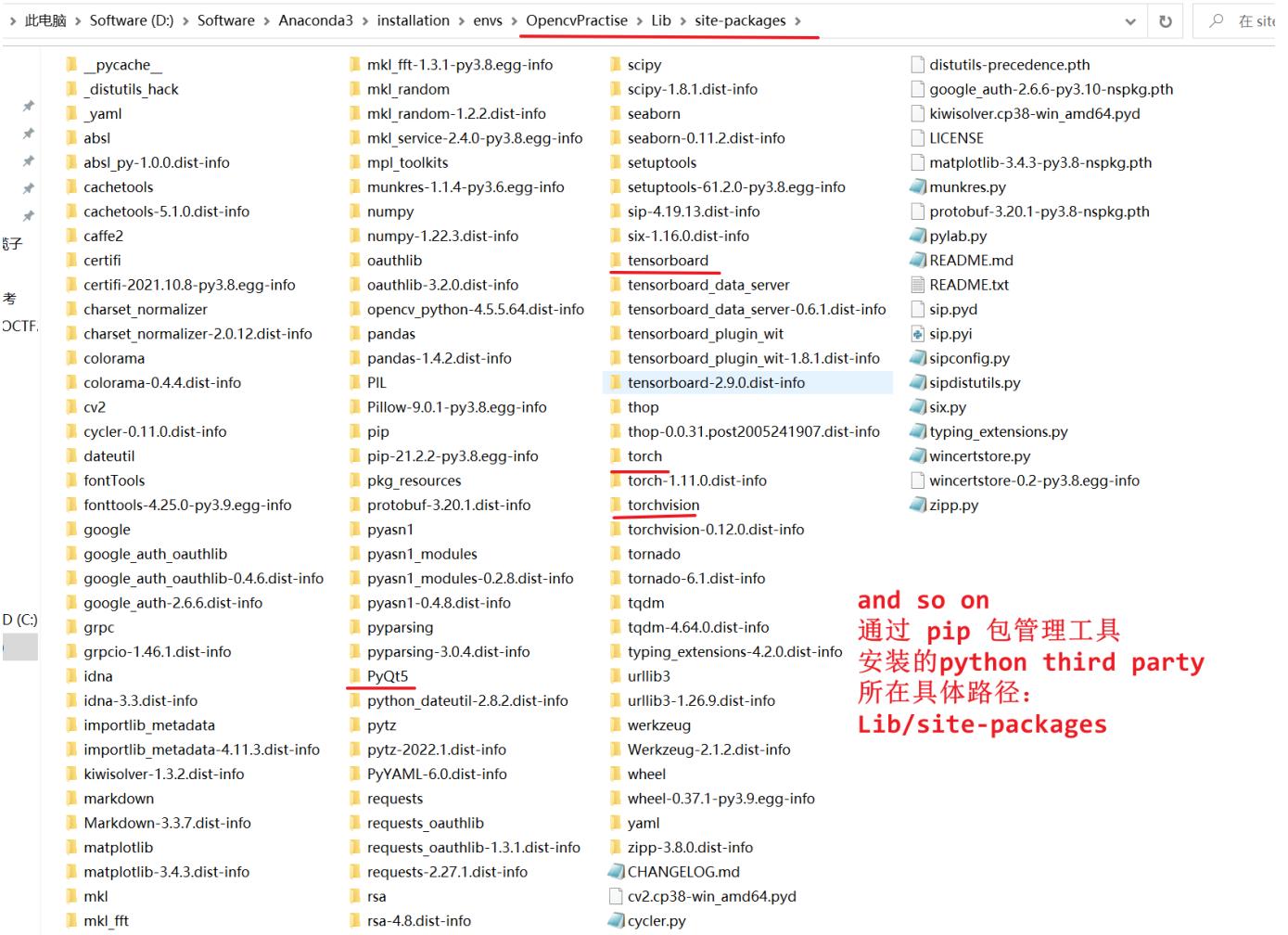
Python | 复制代码

```
1 pip install module-name
```

pip 安装的包具体位置

Lib/site-packages

以anaconda中虚拟环境下的pip所安装的第三方包路径为例。



and so on
通过 pip 包管理工具
安装的 python third party
所在具体路径:
Lib/site-packages

pip 卸载第三方包

```
1 pip uninstall modules-name
```

Classes & Objects

```
1  """
2      创建类时初始化列表中的参数会传递给类内的 __init__ 函数，完成该类的一些属性的初始化。
3
4      The name of the StudentClass's name, is going be equal to the name that we passed in.
5      The name of the StudentClass's major, is going be equal to the major that we passed in.
6      The name of the StudentClass's gpa, is going be equal to the gpa that we passed in.
7      The name of the StudentClass's is_to_probation, is going be equal to the is_to_probation that we passed in.
8  """
9
10
11 class StudentClass:
12     def __init__(self, name, major, gpa, is_to_probation):
13         self.name = name
14         self.major = major
15         self.gpa = gpa
16         self.is_to_probation = is_to_probation
17
18
```

file 文件定义类

Project E:\PycharmProject PythonTutorial

- 2DListsandNestedLoop.py
- AdvancedCalculator.py
- app.py
- BuildingABasicCalculator.py
- BuildingAGuessingGame.py
- BuildTranslator.py
- ClassesAndObjects.py
- Comments.py
- Dictionaries.py
- ExponentFunction.py
- ForLoops.py
- Functions.py
- GettingInputFromUsers.py
- IfStatement.py
- IfStatementComparisons.py
- Lists.py
- ListsFunction.py
- MadLibsGame.py
- main.py
- ModulesAndPip.py
- ReadingFiles.py
- ReadingFiles_employee.txt
- ReturnStatement.py
- StudentFile.py
- TryExcept.py
- Tuple.py
- UsefulTools.py
- Variables&DataTypes.py

Run: ClassesAndObjects

```
E:\my_tools\MSYS2\installation\mingw64\bin\python.exe E:/PycharmProjects/PythonTutorial/ClassesAndObjects.py
Jim
Process finished with exit code 0
```

Building a Multiple Choice Quiz

```
1 class Question:
2     def __init__(self, prompt, answer):
3         self.prompt = prompt
4         self.answer = answer
```

```
1 from Question import Question 导入类
2
3 question_prompts = [
4     "What color are Apples?\n(a) Red/Green\n(b) Purple\n(c) Orange\n\n",
5     "What color are Bananas?\n(a) Teal\n(b) Magenta\n(c) Yellow\n\n",
6     "What color are Strawberries?\n(a) Yellow\n(b) Red\n(c) Blue\n\n",
7 ]
8
9 questions = [
10     Question(question_prompts[0], "a"),
11     Question(question_prompts[1], "c"),
12     Question(question_prompts[2], "b"),
13 ]
14
15 def run_test(questions):
16     score = 0
17     for question in questions:
18         answer = input(question.prompt)
19         if answer == question.answer:
20             score += 1
21     print("You got " + str(score) + "/" + str(len(questions)) + " correct.")
22
23 run_test(questions)
```

Object Function – 类成员函数

```
1  class Student:
2      def __init__(self, name, major, gpa):
3          self.name = name
4          self.major = major
5          self.gpa = gpa
6
7      def on_honor_roll(self):
8          if self.gpa >= 3.5:
9              return True
10         else:
11             return False
12
13     def more_excellent(self, Student): # 自己定义的比较两个类分数
14         return self.gpa >= Student.gpa
15
16
17     def same_major(self, Student): # 自己定义的判断是否同专业
18         result = ""
19         if self.major == Student.major:
20             result = self.name + " and " + Student.name + " are classmate."
21         else:
22             result = self.name + " and " + Student.name + " are not classmate."
23         return result
24
```

```
Project E:\PycharmProject PythonTutorial
1  from Object_Funcntion_Students import Student
2
3  student1 = Student("Oscar", "Accountant", 3.2)
4  student2 = Student("Mike", "Law", 3.5)
5  student3 = Student("Nicolas", "Law", 2.8)
6
7  print(student1.on_honor_roll())
8  print(student2.on_honor_roll())
9
10 print(student1.more_excellent(student2))
11 print(student1.same_major(student2))
12 print(student2.same_major(student3))
13
```

Run: Object_Function

E:\my_tools\MSYS2\installation\mingw64\bin\python.exe E:/PycharmProjects/PythonTutorial/Object_Function.py

True
False
Mike and Nicolas are classmate.
Oscar and Mike are not classmate.

Process finished with exit code 0

Inheritance

一个类：Chef

Python | 复制代码

```
1 class Chef:  
2  
3     def make_chicken(self):  
4         print("The chef makes a chicken.")  
5  
6     def make_salad(self):  
7         print("The chef makes a salad.")  
8  
9     def make_special_dish(self):  
10        print("The chef makes bbq ride.")
```

另一个类：ChineseChef

如何构成继承关系

1. 把父类中的 def 拷贝过来，写到子类中

Python | 复制代码

```
1  
2 class ChineseChef:  
3  
4     def make_chicken(self):  
5         print("The chef makes a chicken.")  
6  
7     def make_salad(self):  
8         print("The chef makes a salad.")  
9  
10    def make_special_dish(self): # 子类和父类同名函数，但做的事情不同  
11        print("The chef makes orange chicken.")  
12  
13    def make_traditional(self): # 子类特有的属性  
14        print("The chef makes fried rice.")  
15
```

2. 使用继承机制

chef.py 父类 子类 主函数

Inheritance_Chef.py Inheritance_Chinese_Chef.py Inheritance_demo.py

```
1 from Inheritance_Chef import Chef    导入父类
2
3
4
5 class ChineseChef(Chef):    继承
6
7     def make_special_dish(self):    # 子类和父类同名函数, 但做的事情不同, override
8         print("The chef makes orange chicken.")
9
10    def make_traditional(self):    # 子类特有的属性
11        print("The chef makes fried rice.")
12
```

表示同名函数, 子类**override**父类

继承中同名函数**override** (覆盖)

主文件

Python | 复制代码

```
1 from Inheritance_Chef import Chef
2
3 myChef = Chef()
4 myChef.make_special_dish()
```

未采用继承

```
from Inheritance_Chef import Chef
from Inheritance_Chinese_Chef import ChineseChef

myChef = Chef()
myChef.make_special_dish() 未采用继承, Chef 类

myChineseChef = ChineseChef() ChineseChef 类
myChineseChef.make_special_dish() 两个类中make_special_dish()不同

myChineseChef.make_traditional() ChineseChef 类独有的函数
```

Run: Inheritance_demo

```
E:\my_tools\MSYS2\installation\mingw64\bin\python.exe E:/PycharmProjects/PythonTutorial/Inheritance_demo.py
The chef makes bbq ride.
The chef makes orange chicken.
The chef makes fried rice.
```

Process finished with exit code 0

采用继承

```
from Inheritance_Chef import Chef
from Inheritance_Chinese_Chef import ChineseChef

myChef = Chef()
myChef.make_special_dish()

myChineseChef = ChineseChef()
myChineseChef.make_special_dish()

myChineseChef.make_traditional()
```

Run: Inheritance_demo

```
E:\my_tools\MSYS2\installation\mingw64\bin\python.exe E:/PycharmProjects/PythonTutorial/Inheritance_demo.py
The chef makes bbq ride.
The chef makes orange chicken.
The chef makes fried rice.
```

Process finished with exit code 0

Python Interpreter

1 python interpreter is a environment that we can use to execute python commands.

```
C:\Users\ckBu>python
Python 3.9.7 (default, Sep 16 2021, 16:59:28) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated. Libraries may fail to load. To activate this environment
please see https://conda.io/activation
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>>
>>> -
```

这个就是 **python interpreter**

```
C:\Users\ckBu>python
Python 3.9.7 (default, Sep 16 2021, 16:59:28) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated. Libraries may fail to load. To activate this environment
please see https://conda.io/activation
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>>
>>> print("Hello World")
Hello World
>>>
>>>
>>> num1 = 10
>>> num2 = 90
>>> print(num1 + num2)
100
>>>
>>>
>>> def say_hi(name):
...     print("Hello " + name);
...
>>> say_hi("Mike")
Hello Mike
>>>
```

在**python interpreter**里，你可以做任何**python**相关事情
但不太方便，所以写**python** 脚本推荐使用 **IDE**（集成开发环境）
这样的编辑器+解释器，更智能更方便。

2022.06.28 16:00 学完。视频看完，代码敲完，笔记记录完。