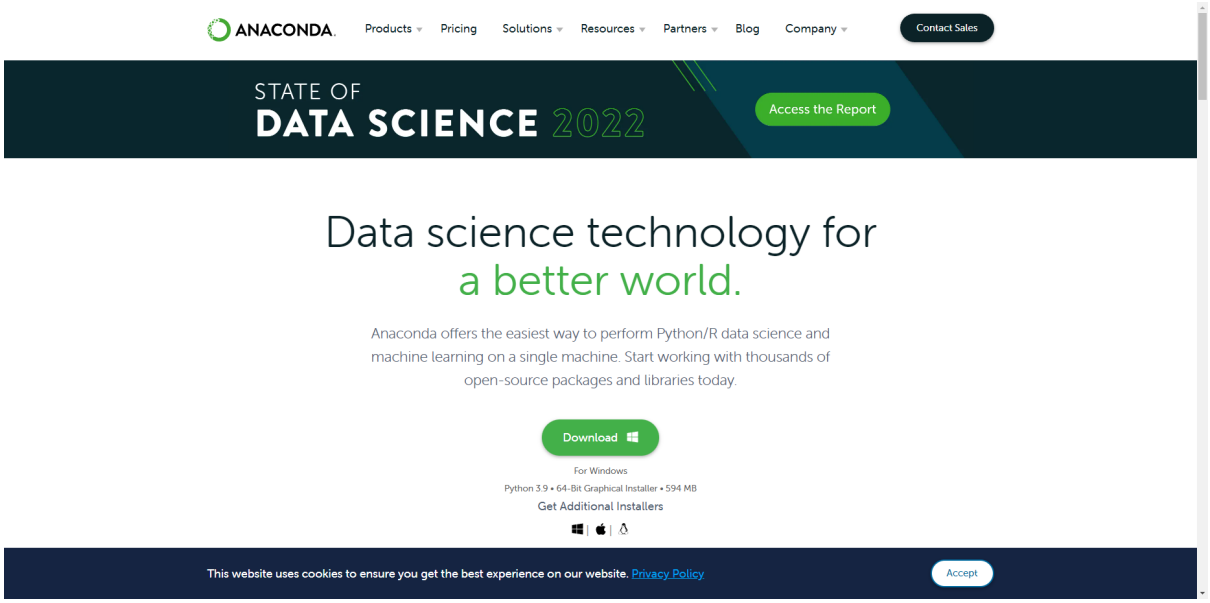
hw2_python基础

- 姓名：黄浩扬
- 学号：2201212800
- 导师：王荣刚

1、安装Anaconda

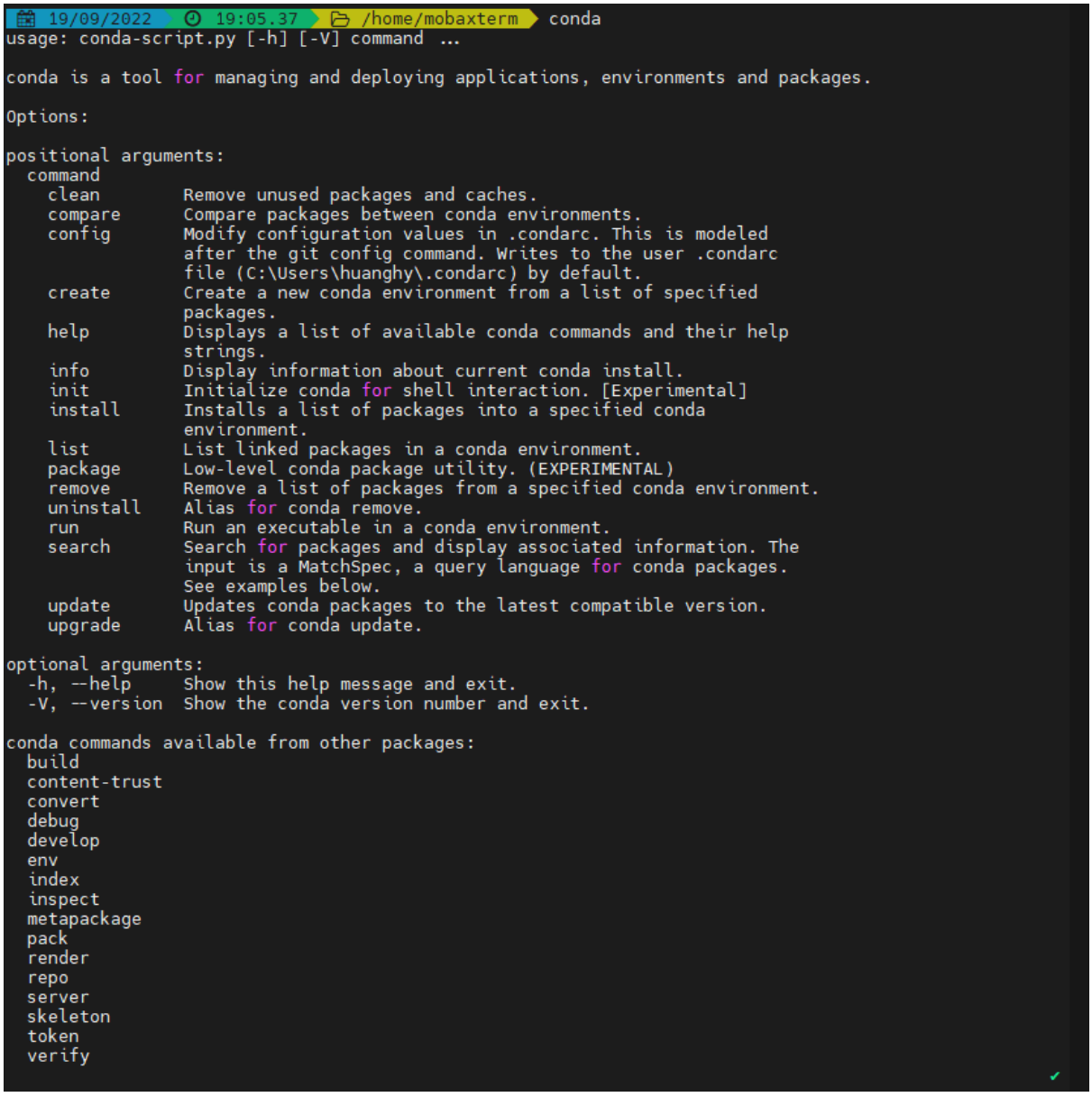
（1）下载Anaconda并安装

前往Anaconda官网https://www.anaconda.com/点击download，下载完安装包后点击安装



（2）验证是否安装成功

在终端输入conda，提示如下界面即为安装成功



输入conda -V可查看conda的版本

2、安装虚拟环境并删除

（1）更新Anaconda至最新版本

在终端输入以下命令，更新Anaconda至最新版本

```
conda update -n base -c defaults conda
```

```
19/09/2022  19:10.20  /home/mobaxterm  conda update -n base -c defaults conda

Collecting package metadata (current_repodata.json):  ...working...  done
Solving environment:  ...working...  done

## Package Plan ##

  environment location: D:\Software\anaconda3

  added / updated specs:
    - conda


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    conda-4.14.0               |   py39haa95532_0         937 KB
    conda-repo-cli-1.0.5       |   py39haa95532_0         122 KB
    pathlib-1.0.1              |     pyhd3eb1b0_1          17 KB
    pyjwt-2.4.0                |   py39haa95532_0          38 KB
    ------------------------------------------------------------
                                           Total:         1.1 MB

The following NEW packages will be INSTALLED:

  pathlib            pkgs/main/noarch::pathlib-1.0.1-pyhd3eb1b0_1

The following packages will be UPDATED:

  conda                               4.12.0-py39haa95532_0 --> 4.14.0-py39haa95532_0
  conda-repo-cli     pkgs/main/noarch::conda-repo-cli-1.0.~ --> pkgs/main/win-64::conda-repo-cli-1.0.
5-py39haa95532_0
  pyjwt                                2.1.0-py39haa95532_0 --> 2.4.0-py39haa95532_0


Proceed ([y]/n)?

Downloading and Extracting Packages
conda-4.14.0          | 937 KB     | ########## | 100%
pyjwt-2.4.0           | 38 KB      | ########## | 100%
pathlib-1.0.1         | 17 KB      | ########## | 100%
conda-repo-cli-1.0.5  | 122 KB     | ########## | 100%
Preparing transaction:  ...working...  done
Verifying transaction:  ...working...  WARNING conda.core.path_actions:verify(957): Unable to create en
vironments file. Path not writable.
  environment location: C:\Users\huanghy\.conda\environments.txt

done
Executing transaction:  ...working...  WARNING conda.core.envs_manager:register_env(50): Unable to regi
ster environment. Path not writable or missing.
  environment location: D:\Software\anaconda3
  registry file: C:\Users\huanghy\.conda\environments.txt
done
```

出现以上界面即为更新成功

（2）创建虚拟环境

在终端输入以下命令创建一个虚拟环境

```
conda create -n <your_env_name> python=x.x
```

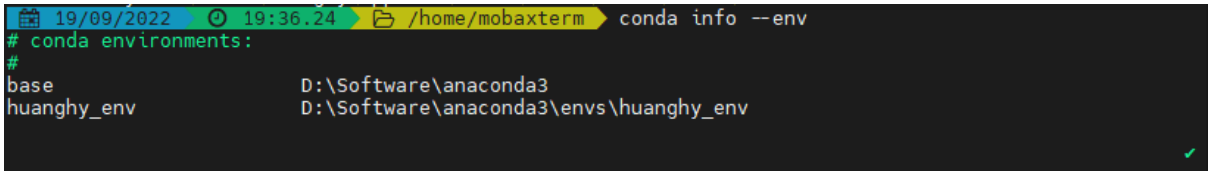
```
 19/09/2022  19:11.46  /home/mobaxterm  conda create -n huanghy_env python=3.10
Collecting package metadata (current_repodata.json):  ...working...  done
Solving environment:  ...working...  done

## Package Plan ##

  environment location: D:\Software\anaconda3\envs\huanghy_env

  added / updated specs:
    - python=3.10


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    ca-certificates-2022.07.19 |       haa95532_0         123 KB
    certifi-2022.6.15          |  py310haa95532_0         153 KB
    libffi-3.4.2               |       hd77b12b_4         107 KB
    openssl-1.1.1q             |       h2bbff1b_0         4.8 MB
    pip-22.1.2                 |  py310haa95532_0         2.5 MB
    python-3.10.4              |       hbb2ffb3_0        15.9 MB
    setuptools-63.4.1          |  py310haa95532_0         1.1 MB
    sqlite-3.39.2              |       h2bbff1b_0         805 KB
    tk-8.6.12                  |       h2bbff1b_0         3.1 MB
    tzdata-2022c               |       h04d1e81_0         107 KB
    wincertstore-0.2           |  py310haa95532_2          15 KB
    xz-5.2.5                   |       h8cc25b3_1         246 KB
    zlib-1.2.12                |       h8cc25b3_3         112 KB
    ------------------------------------------------------------
                                           Total:        29.1 MB

The following NEW packages will be INSTALLED:

  bzip2              pkgs/main/win-64::bzip2-1.0.8-he774522_0
  ca-certificates    pkgs/main/win-64::ca-certificates-2022.07.19-haa95532_0
  certifi            pkgs/main/win-64::certifi-2022.6.15-py310haa95532_0
  libffi             pkgs/main/win-64::libffi-3.4.2-hd77b12b_4
  openssl            pkgs/main/win-64::openssl-1.1.1q-h2bbff1b_0
  pip                pkgs/main/win-64::pip-22.1.2-py310haa95532_0
  python             pkgs/main/win-64::python-3.10.4-hbb2ffb3_0
  setuptools         pkgs/main/win-64::setuptools-63.4.1-py310haa95532_0
  sqlite             pkgs/main/win-64::sqlite-3.39.2-h2bbff1b_0
  tk                 pkgs/main/win-64::tk-8.6.12-h2bbff1b_0
  tzdata             pkgs/main/noarch::tzdata-2022c-h04d1e81_0
  vc                 pkgs/main/win-64::vc-14.2-h21ff451_1
  vs2015_runtime     pkgs/main/win-64::vs2015_runtime-14.27.29016-h5e58377_2
  wheel              pkgs/main/noarch::wheel-0.37.1-pyhd3eb1b0_0
  wincertstore       pkgs/main/win-64::wincertstore-0.2-py310haa95532_2
  xz                 pkgs/main/win-64::xz-5.2.5-h8cc25b3_1
  zlib               pkgs/main/win-64::zlib-1.2.12-h8cc25b3_3


Proceed ([y]/n)? y


Downloading and Extracting Packages
certifi-2022.6.15    | 153 KB    | ########## | 100%
xz-5.2.5             | 246 KB    | ########## | 100%
tzdata-2022c         | 107 KB    | ########## | 100%
ca-certificates-2022 | 123 KB    | ########## | 100%
setuptools-63.4.1    | 1.1 MB    | ########## | 100%
libffi-3.4.2         | 107 KB    | ########## | 100%
python-3.10.4        | 15.9 MB   | ########## | 100%
openssl-1.1.1q       | 4.8 MB    | ########## | 100%
pip-22.1.2           | 2.5 MB    | ########## | 100%
wincertstore-0.2     | 15 KB     | ########## | 100%
sqlite-3.39.2        | 805 KB    | ########## | 100%
tk-8.6.12            | 3.1 MB    | ########## | 100%
zlib-1.2.12          | 112 KB    | ########## | 100%
Preparing transaction:  ...working...  done
Verifying transaction:  ...working...  WARNING conda.core.path_actions:verify(958): Unable to create en
vironments file. Path not writable.
  environment location: C:\Users\huanghy\.conda\environments.txt

done
Executing transaction:  ...working...  WARNING conda.core.envs_manager:register_env(51): Unable to regi
ster environment. Path not writable or missing.
  environment location: D:\Software\anaconda3\envs\huanghy_env
  registry file: C:\Users\huanghy\.conda\environments.txt
done
#
# To activate this environment, use
#
#     $ conda activate huanghy_env
#
# To deactivate an active environment, use
#
#     $ conda deactivate
```

出现以上界面即创建成功

（3）查看已创建的虚拟环境

在终端输入

```
conda info --env
```



可以看见虚拟环境成功创建

（4）删除创建的虚拟环境

在终端输入以下命令

```
conda remove -n <your_env_name> --all
```

```
19/09/2022  19:43.33  /home/mobaxterm  conda env list
# conda environments:
#
base                  D:\Software\anaconda3
huanghy_env           D:\Software\anaconda3\envs\huanghy_env

19/09/2022  19:44.41  /home/mobaxterm  conda remove -n huanghy_env --all

Remove all packages in environment D:\Software\anaconda3\envs\huanghy_env:


## Package Plan ##

  environment location: D:\Software\anaconda3\envs\huanghy_env


The following packages will be REMOVED:

  bzip2-1.0.8-he774522_0
  ca-certificates-2022.07.19-haa95532_0
  certifi-2022.6.15-py310haa95532_0
  libffi-3.4.2-hd77b12b_4
  openssl-1.1.1q-h2bbff1b_0
  pip-22.1.2-py310haa95532_0
  python-3.10.4-hbb2ffb3_0
  setuptools-63.4.1-py310haa95532_0
  sqlite-3.39.2-h2bbff1b_0
  tk-8.6.12-h2bbff1b_0
  tzdata-2022c-h04d1e81_0
  vc-14.2-h21ff451_1
  vs2015_runtime-14.27.29016-h5e58377_2
  wheel-0.37.1-pyhd3eb1b0_0
  wincertstore-0.2-py310haa95532_2
  xz-5.2.5-h8cc25b3_1
  zlib-1.2.12-h8cc25b3_3


Proceed ([y]/n)? y

Preparing transaction: ...working... done
Verifying transaction: ...working... a3\envs\huanghWARNING conda.core.path_actions:verify(958): Unabl
e to create environments file. Path not writable.
  environment location: C:\Users\huanghy\.conda\environments.txt

done
Executing transaction: ...working... WARNING conda.core.envs_manager:register_env(51): Unable to regi
ster environment. Path not writable or missing.
  environment location: D:\Software\anaconda3\envs\huanghy_env
  registry file: C:\Users\huanghy\.conda\environments.txt
done
```

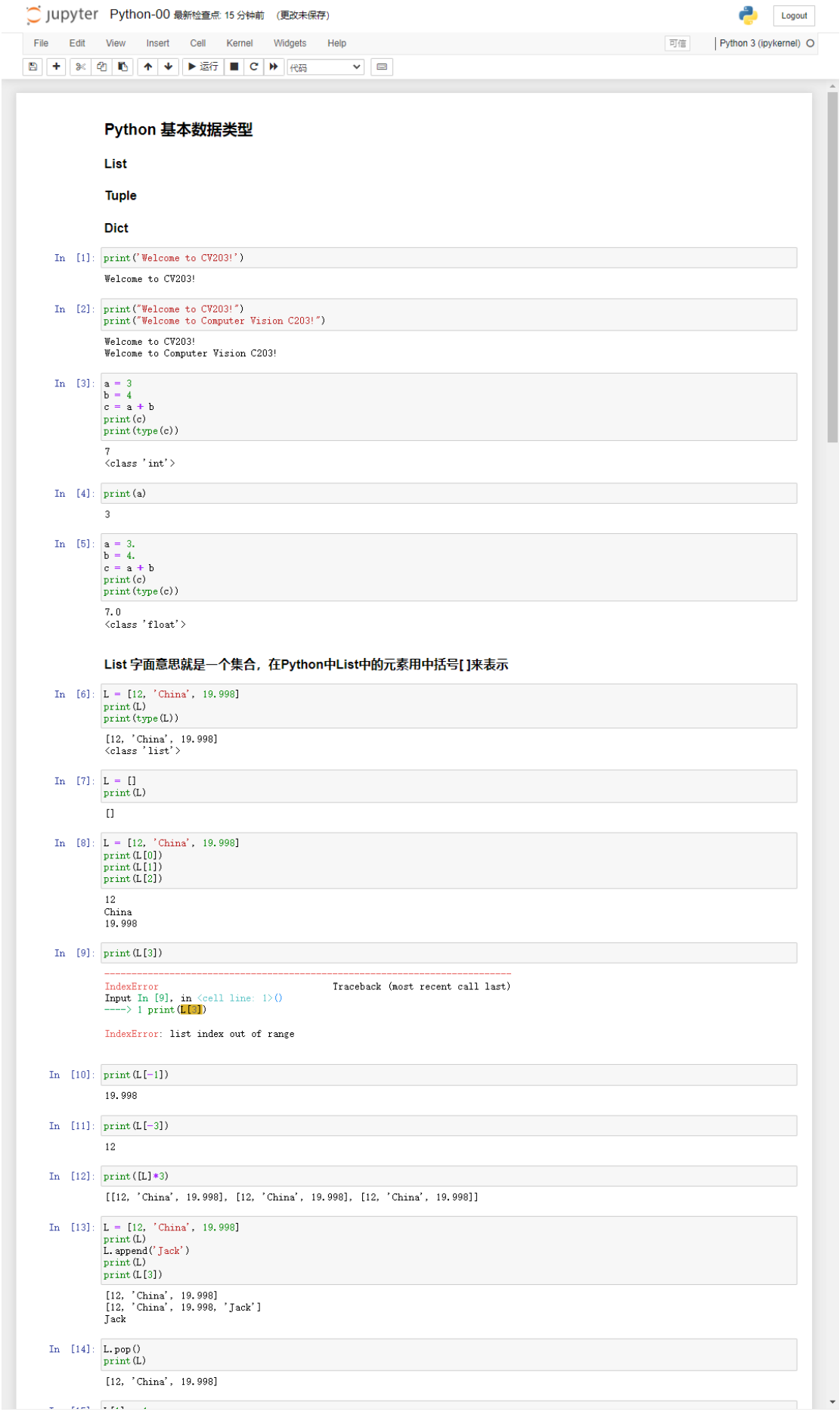
删除后查看虚拟环境列表，可见虚拟环境已被删除

```
19/09/2022  19:46.36  /home/mobaxterm  conda env list
# conda environments:
#
base                  D:\Software\anaconda3
```

3、运行给定四个.ipynb文件

（1）Python-00.ipynb

由于单个文件所有运行结果太多，只截取了部分结果，之后的几个文件同理


jupyter Python-00 最新检查点: 15 分钟前 (更改未保存)
Logout
File Edit View Insert Cell Kernel Widgets Help
可信
Python 3 (ipykernel)
运行
代码
Python 基本数据类型
List
Tuple
Dict
In [1]:
print('Welcome to CV203!')
Welcome to CV203!
In [2]:
print("Welcome to CV203!")
print("Welcome to Computer Vision C203!")
Welcome to CV203!
Welcome to Computer Vision C203!
In [3]:
a = 3
b = 4
c = a + b
print(c)
print(type(c))
7
<class 'int'>
In [4]:
print(a)
3
In [5]:
a = 3.
b = 4.
c = a + b
print(c)
print(type(c))
7.0
<class 'float'>
List 字面意思就是一个集合，在Python中List中的元素用中括号[]来表示
In [6]:
L = [12, 'China', 19.998]
print(L)
print(type(L))
[12, 'China', 19.998]
<class 'list'>
In [7]:
L = []
print(L)
[]
In [8]:
L = [12, 'China', 19.998]
print(L[0])
print(L[1])
print(L[2])
12
China
19.998
In [9]:
print(L[3])
IndexError Traceback (most recent call last)
Input In [9], in <cell line: 1>()
----> 1 print(L[3])
IndexError: list index out of range
In [10]:
print(L[-1])
19.998
In [11]:
print(L[-3])
12
In [12]:
print([L]*3)
[[12, 'China', 19.998], [12, 'China', 19.998], [12, 'China', 19.998]]
In [13]:
L = [12, 'China', 19.998]
print(L)
L.append('Jack')
print(L)
print(L[3])
[12, 'China', 19.998]
[12, 'China', 19.998, 'Jack']
Jack
In [14]:
L.pop()
print(L)
[12, 'China', 19.998]

jupyter Python-00 最新检查点: 8 分钟前 (已自动保存)

Logout

File Edit View Insert Cell Kernel Widgets Help 可信 Python 3 (ipykernel)

运行 Markdown

Tuple 可以看做是一种“不变”的List，访问也是通过下标，用小括号 () 表示

In [19]:
```
t = (3.14, 'China', 'Jason')
print(t)
```

```
(3.14, 'China', 'Jason')
```

In [20]:
```
t[1] = 'America'
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [20], in <cell line: 1>()
----> 1 t[1] = 'America'

TypeError: 'tuple' object does not support item assignment
```

In [21]:
```
t = ()
print(t)
```

```
()
```

In [22]:
```
t = (3.14, )
print(t)
```

```
(3.14,)
```

In [23]:
```
a = [1,2,3,4]
print(a.insert(0,8))
print(a)
```

```
None
[8, 1, 2, 3, 4]
```

Dict 是Python中非常重要的数据类型，就像它的字面意思一样，它是个活字典，其实就是Key-Value键值对，可以用花括号{ }定义

In [24]:
```
d = {
    'Adam': 95,
    'Lisa': 85,
    'Bart': 59,
    'Paul': 75
}
print(d)
```

```
{'Adam': 95, 'Lisa': 85, 'Bart': 59, 'Paul': 75}
```

In [25]:
```
len(d)
```

Out[25]:
```
4
```

In [26]:
```
print(d)
d['Jone'] = 59
print(d)
```

```
{'Adam': 95, 'Lisa': 85, 'Bart': 59, 'Paul': 75}
{'Adam': 95, 'Lisa': 85, 'Bart': 59, 'Paul': 75, 'Jone': 59}
```

In [27]:
```
print(d['Adam'])
```

```
95
```

In [28]:
```
d['Jack'] = 80
print(d)
```

```
{'Adam': 95, 'Lisa': 85, 'Bart': 59, 'Paul': 75, 'Jone': 59, 'Jack': 80}
```

In [29]:
```
d[0] = 'PKU'
print(d)
```

```
{'Adam': 95, 'Lisa': 85, 'Bart': 59, 'Paul': 75, 'Jone': 59, 'Jack': 80, 0: 'PKU'}
```

In [30]:
```
d['0']
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Input In [30], in <cell line: 1>()
----> 1 d['0']

KeyError: '0'
```

In [31]:
```
if 'Adam' in d :
    print('exist key')
else:
    print('No')
```

```
exist key
```

In [32]:
```
print(d.get('Adam'))
```

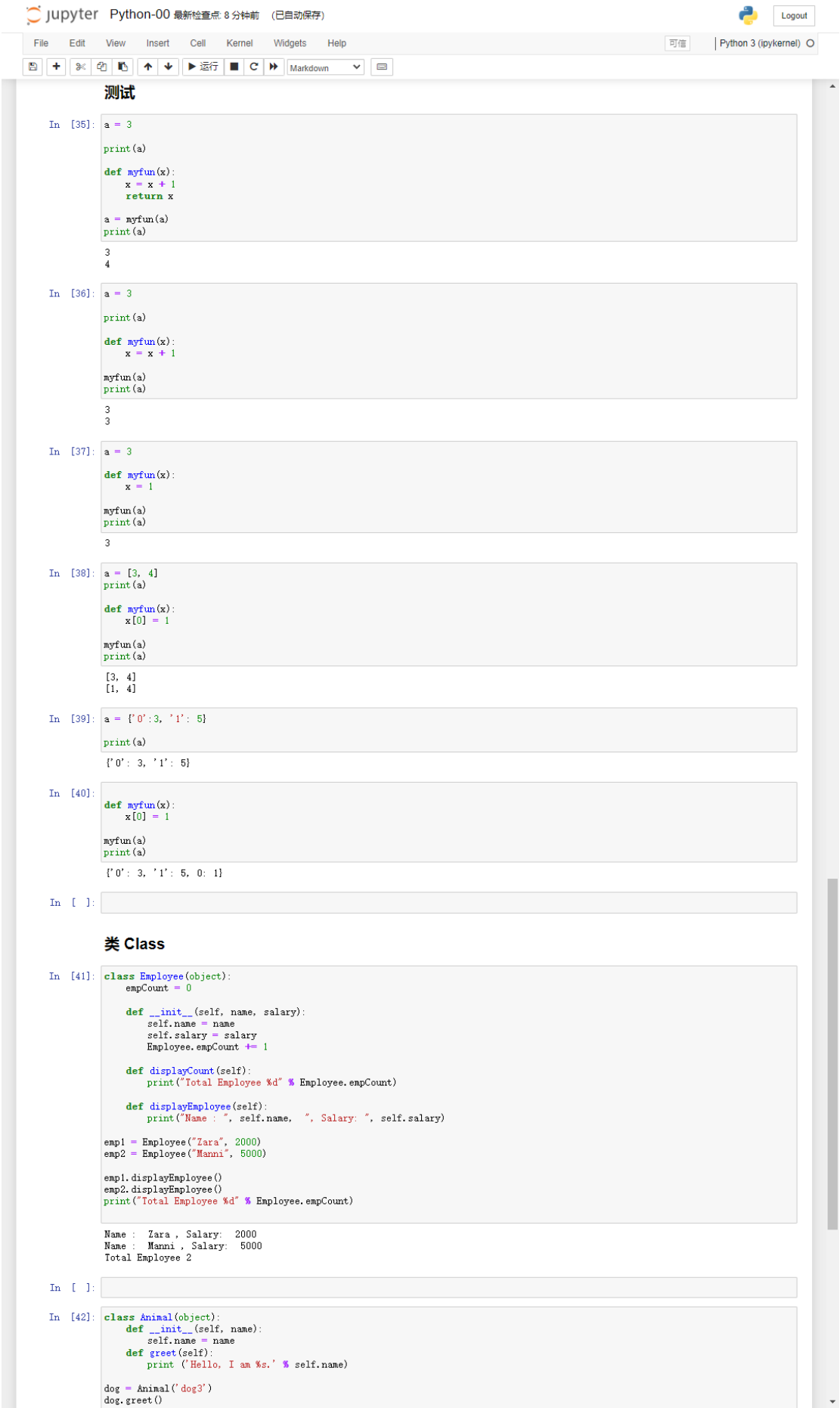
```
95
```

In [33]:
```
print(d.get('Jack'))
```

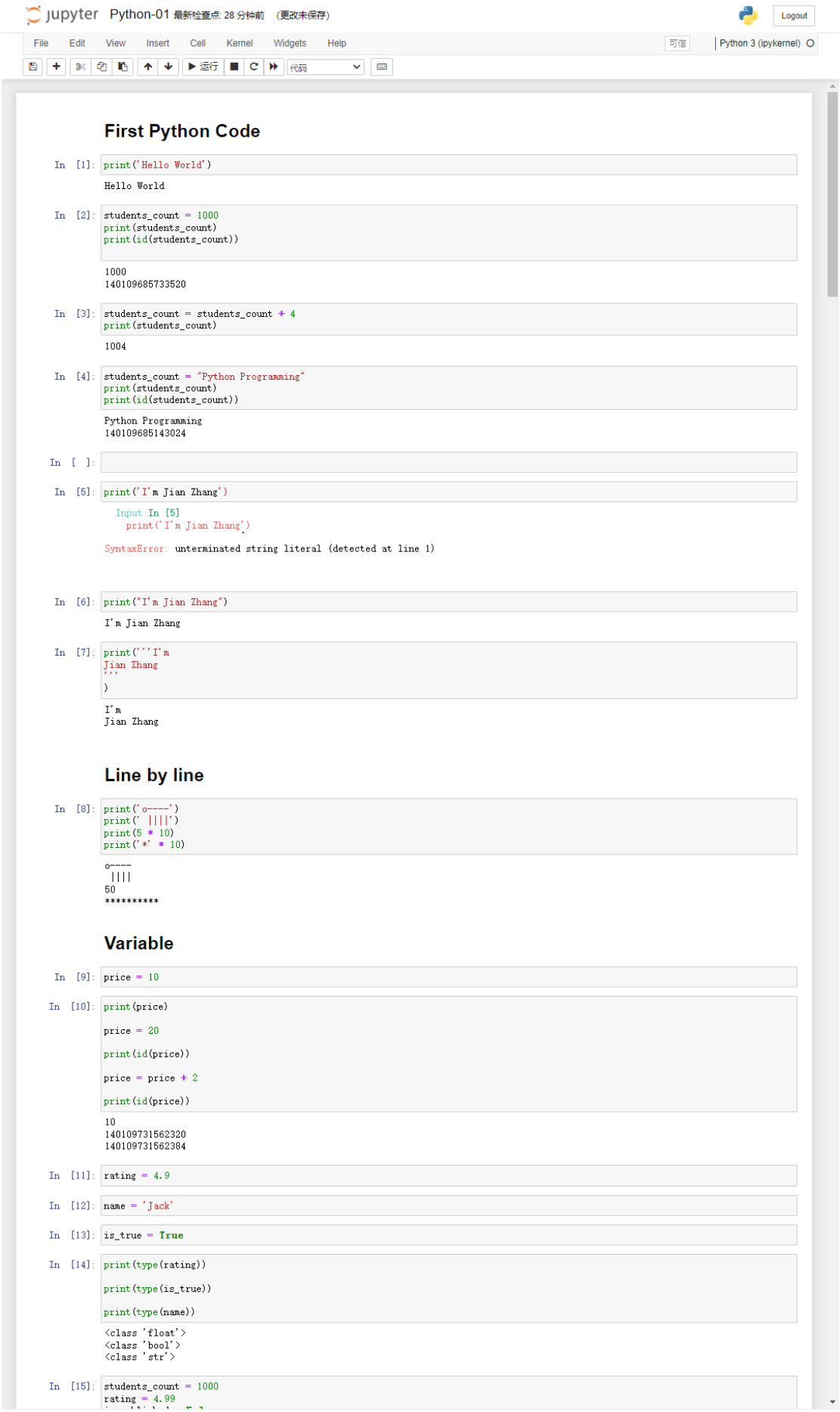
```
80
```

In [34]:
```
for key in d:
    print(key, ':', d.get(key))
```

```
Adam : 95
Lisa : 85
Bart : 59
Paul : 75
Jone : 59
```


jupyter Python-00 最新检查点: 8 分钟前 (已自动保存)
Logout
File Edit View Insert Cell Kernel Widgets Help
可信
Python 3 (ipykernel)
运行
Markdown
测试
In [35]:
a = 3
print(a)
def myfun(x):
x = x + 1
return x
a = myfun(a)
print(a)
3
4
In [36]:
a = 3
print(a)
def myfun(x):
x = x + 1
myfun(a)
print(a)
3
3
In [37]:
a = 3
def myfun(x):
x = 1
myfun(a)
print(a)
3
In [38]:
a = [3, 4]
print(a)
def myfun(x):
x[0] = 1
myfun(a)
print(a)
[3, 4]
[1, 4]
In [39]:
a = {'0':3, '1': 5}
print(a)
{'0': 3, '1': 5}
In [40]:
def myfun(x):
x[0] = 1
myfun(a)
print(a)
{'0': 3, '1': 5, 0: 1}
In [ ]:
类 Class
In [41]:
class Employee(object):
empCount = 0
def __init__(self, name, salary):
self.name = name
self.salary = salary
Employee.empCount += 1
def displayCount(self):
print("Total Employee %d" % Employee.empCount)
def displayEmployee(self):
print("Name : ", self.name, ", Salary: ", self.salary)
emp1 = Employee("Zara", 2000)
emp2 = Employee("Manni", 5000)
emp1.displayEmployee()
emp2.displayEmployee()
print("Total Employee %d" % Employee.empCount)
Name : Zara , Salary: 2000
Name : Manni , Salary: 5000
Total Employee 2
In [ ]:
In [42]:
class Animal(object):
def __init__(self, name):
self.name = name
def greet(self):
print ('Hello, I am %s.' % self.name)
dog = Animal('dog3')
dog.greet()

（2）Python-01.ipynb


jupyter Python-01 最新检查点: 28 分钟前 (更改未保存)
Logout
File Edit View Insert Cell Kernel Widgets Help
可信
Python 3 (ipykernel)
运行
代码
First Python Code
In [1]:
print('Hello World')
Hello World
In [2]:
students_count = 1000
print(students_count)
print(id(students_count))
1000
140109685733520
In [3]:
students_count = students_count + 4
print(students_count)
1004
In [4]:
students_count = "Python Programming"
print(students_count)
print(id(students_count))
Python Programming
140109685143024
In [ ]:
In [5]:
print('I'm Jian Zhang')
Input In [5]
print('I'm Jian Zhang')
SyntaxError: unterminated string literal (detected at line 1)
In [6]:
print("I'm Jian Zhang")
I'm Jian Zhang
In [7]:
print('''I'm
Jian Zhang
'''
)
I'm
Jian Zhang
Line by line
In [8]:
print('o----')
print(' ||||')
print(5 * 10)
print('*' * 10)
o----
||||
50
****
Variable
In [9]:
price = 10
In [10]:
print(price)
price = 20
print(id(price))
price = price + 2
print(id(price))
10
140109731562320
140109731562384
In [11]:
rating = 4.9
In [12]:
name = 'Jack'
In [13]:
is_true = True
In [14]:
print(type(rating))
print(type(is_true))
print(type(name))
<class 'float'>
<class 'bool'>
<class 'str'>
In [15]:
students_count = 1000
rating = 4.99

jupyter Python-01 最新检查点: 29 分钟前 (更改未保存)

Logout

File Edit View Insert Cell Kernel Widgets Help 可信 Python 3 (ipykernel)

运行 代码

Patient Example: Name, age

In [16]:
```
name = input('What is your name? ')
print('Hi ' + name)
```

```
What is your name? huanghy
Hi huanghy
```

In [17]:
```
name = input('What is your name? ')
age = input('What is your age? ')
print(name + ' is ' + age + ' years old.')
```

```
What is your name? huanghy
What is your age? 21
huanghy is 21 years old.
```

In [18]:
```
price = 1_000_000
```

In [19]:
```
print(price)
```

```
1000000
```

In [20]:
```
print(type(price))
```

```
<class 'int'>
```

Formatted String

In [21]:
```
first_name = "Jian"
last_name = "Zhang"
full_name = first_name + " " + last_name
print(full_name)
full_name_formated = f"{first_name} {last_name}"
print(full_name_formated)
```

```
Jian Zhang
Jian Zhang
```

Escape Character

In [22]:
```
course_full_name = "Python Programing with \"Mosh\""
print(course_full_name)
course_full_name = "Python Programing with \'Mosh\'"
print(course_full_name)
course_full_name = "Python Programing with \\Mosh\\"
print(course_full_name)
course_full_name = "Python Programing with \nMosh"
print(course_full_name)
```

```
Python Programing with "Mosh"
Python Programing with 'Mosh'
Python Programing with \Mosh\
Python Programing with
Mosh
```

String Methods

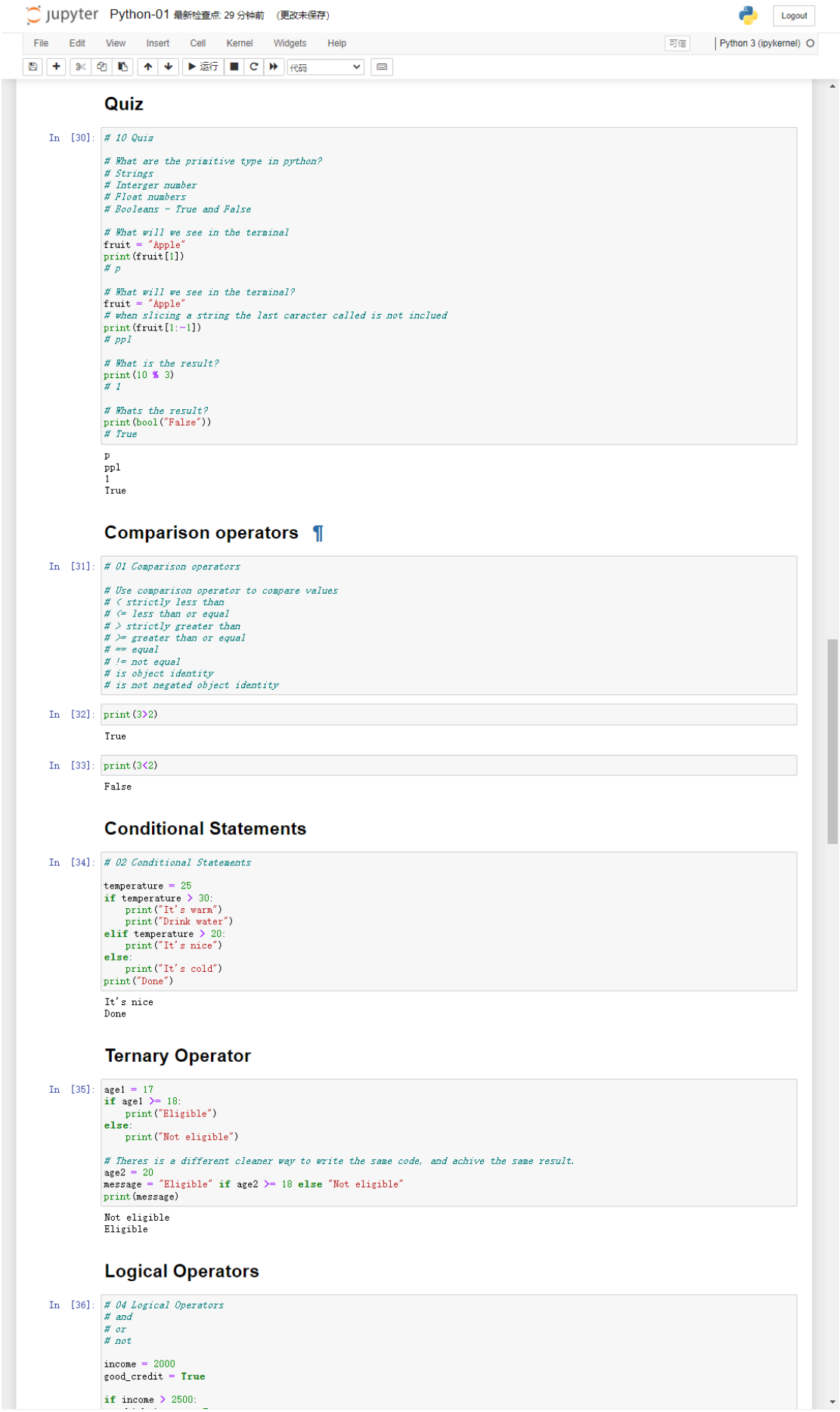
In [23]:
```
course_name_python = "  python programming for beginners    "
print(course_name_python.upper())
print(course_name_python.lower())
print(course_name_python.title())
print(course_name_python.find("gram"))
print(course_name_python.replace("p", "P"))
print("pro" in course_name_python)
print("abc" in course_name_python)
```

```
  PYTHON PROGRAMMING FOR BEGINNERS
  python programming for beginners
  Python Programming For Beginners
12
  Python Programming for beginners
True
False
```

Numbers

In [24]:
```
x = 3
y = 3.1
z = 3 + 2j
print(10 + 3)
print(10 - 3)
print(10 * 3)
print(10 / 3)
print(10 // 3)
print(10 % 3)
print(10 ** 3)
x = x + 3
print(x)
y += 3
print(y)
print(z)
```

```
13
7
30
3.3333333333333335
3
1
1000
```


jupyter Python-01 最新检查点: 29 分钟前 (更改未保存)
Logout
File Edit View Insert Cell Kernel Widgets Help
可信
Python 3 (ipykernel)
运行
代码
Quiz
In [30]:
# 10 Quiz
# What are the primitive type in python?
# Strings
# Interger number
# Float numbers
# Booleans - True and False
# What will we see in the terminal
fruit = "Apple"
print(fruit[1])
# p
# What will we see in the terminal?
fruit = "Apple"
# when slicing a string the last caracter called is not inclued
print(fruit[1:-1])
# ppl
# What is the result?
print(10 % 3)
# 1
# Whats the result?
print(bool("False"))
# True
p
ppl
1
True
Comparison operators ¶
In [31]:
01 Comparison operators
Use comparison operator to compare values
< strictly less than
<= less than or equal
> strictly greater than
>= greater than or equal
== equal
!= not equal
is object identity
is not negated object identity
In [32]:
print(3>2)
True
In [33]:
print(3<2)
False
Conditional Statements
In [34]:
# 02 Conditional Statements
temperature = 25
if temperature > 30:
print("It's warm")
print("Drink water")
elif temperature > 20:
print("It's nice")
else:
print("It's cold")
print("Done")
It's nice
Done
Ternary Operator
In [35]:
age1 = 17
if age1 >= 18:
print("Eligible")
else:
print("Not eligible")
# Theres is a different cleaner way to write the same code, and achive the same result.
age2 = 20
message = "Eligible" if age2 >= 18 else "Not eligible"
print(message)
Not eligible
Eligible
Logical Operators
In [36]:
# 04 Logical Operators
# and
# or
# not
income = 2000
good_credit = True
if income > 2500:

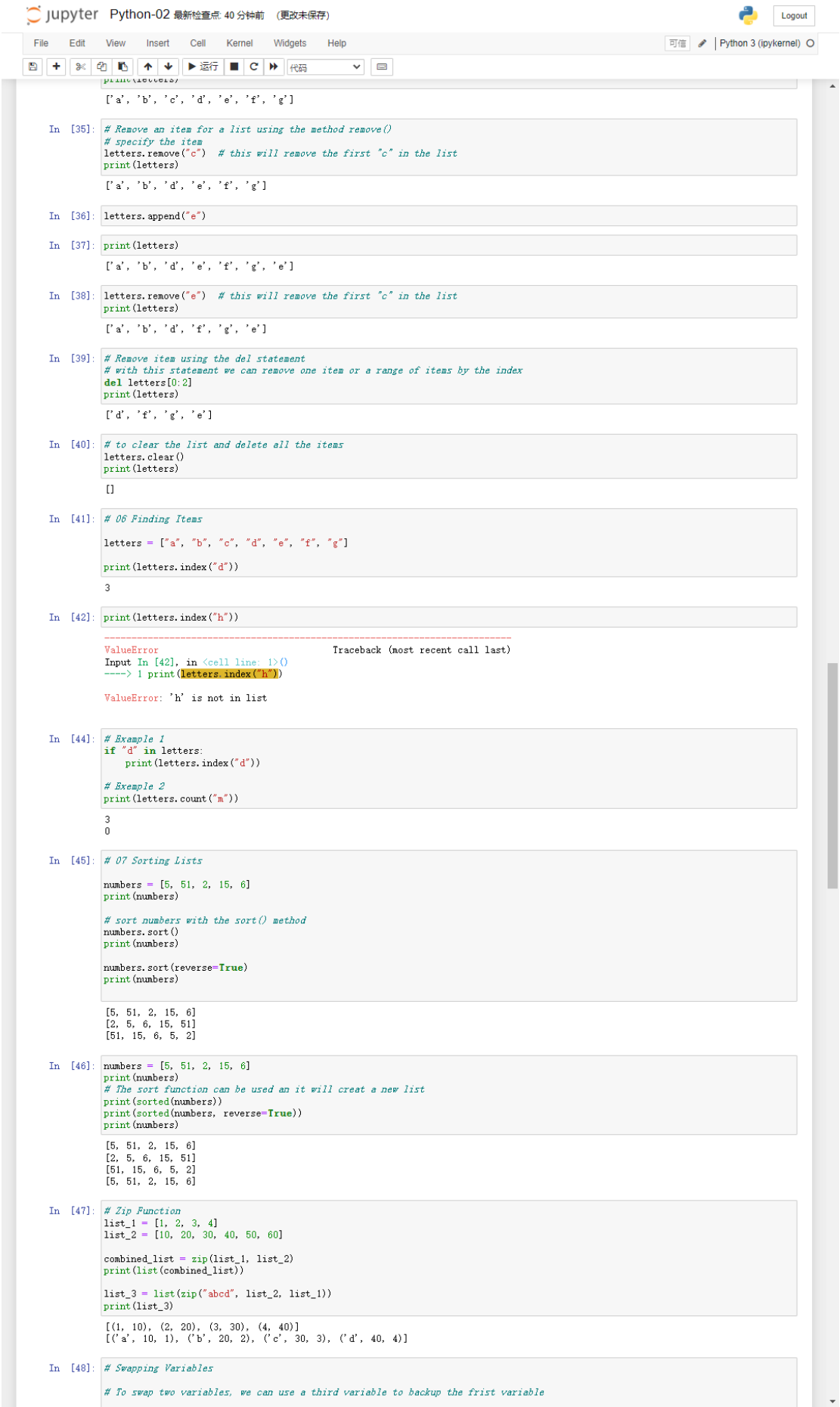

jupyter Python-01 最新检查点: 29 分钟前 (更改未保存)
Logout
File Edit View Insert Cell Kernel Widgets Help
可信
Python 3 (ipykernel)
运行
代码
While Loops
In [43]:
# 12 While Loops
number = 100
while number > 0:
print(number)
number = number // 2
command = ""
while command.lower() != "quit":
command = input(">")
print("Echo", command)
100
50
25
12
6
3
1
>quit
Echo quit
Infinite Loops
In [44]:
# 13 Infinite Loops
while True:
command = input(">")
print("Echo", command)
if command.lower() == "quit":
break
>quit
Echo quit
In [ ]:
In [45]:
# 11 Exercise
# Display the even number (2 4 6 8) followed by this message "We have 4 even numbers"
# 2
# 4
# 6
# 8
# We have 4 even numbers
# My answer
for number_even in range(1, 10):
if number_even % 2 == 0:
print(number_even)
elif number_even > 8:
print("We have 4 even numbers")
break
# Mosh answer
count = 0
for number in range(1, 10):
if number % 2 == 0:
count = count + 1
print(number)
print(f"we have {count} even number")
# In my answer I did not count the numbers
2
4
6
8
We have 4 even numbers
2
4
6
8
we have 4 even number
In [ ]:
In [ ]:
In [ ]:
In [ ]:
In [46]:
for
for
Input In [46]
for
^
SyntaxError: invalid syntax
In [ ]:
In [ ]:
In [ ]:
In [ ]:

（3）Python-02.ipynb


jupyter Python-02 最新检查点: 39 分钟前 (更改未保存)
Logout
File Edit View Insert Cell Kernel Widgets Help
可信 Python 3 (ipykernel)
运行 代码
In [1]:
import math
a = 0.36
b = math.sqrt(a)
print(b)
0.6
In [5]:
pip install MyQR
Collecting MyQR
Downloading MyQR-2.3.1.zip (16 kB)
Preparing metadata (setup.py) ... done
Collecting imageio>=1.5
Downloading imageio-2.22.0-py3-none-any.whl (3.4 MB)
3.4/3.4 MB 2.9 MB/s eta 0:00:00
Collecting numpy>=1.11.1
Downloading numpy-1.23.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (17.1 MB)
17.1/17.1 MB 10.6 MB/s eta 0:00:00
Collecting Pillow>=3.3.1
Downloading Pillow-9.2.0-cp310-cp310-manylinux_2_28_x86_64.whl (3.2 MB)
3.2/3.2 MB 11.2 MB/s eta 0:00:00
Building wheels for collected packages: MyQR
Building wheel for MyQR (setup.py) ... done
Created wheel for MyQR: filename=MyQR-2.3.1-py3-none-any.whl size=16690 sha256=c850834ee5f2a06814435c246cf17302b63750bdf7178641319fa7ba41507646
Stored in directory: /home/huanghy/.cache/pip/wheels/5c/80/ec/154a3dd72ec9170370b6984f6bd6e7c3194b45abf8e6351aea
Successfully built MyQR
Installing collected packages: Pillow, numpy, imageio, MyQR
Successfully installed MyQR-2.3.1 Pillow-9.2.0 imageio-2.22.0 numpy-1.23.3
Note: you may need to restart the kernel to use updated packages.
In [9]:
import MyQR
from MyQR import myqr
"""Generate a QR code"""
myqr.run(words='https://villa.jianzhang.tech/',
save_name='001.png',
)
"""Generate a QR code with a background picture"""
myqr.run(words='https://villa.jianzhang.tech/',
picture='/media/yan1/huanghy/notebook/001.png',
colorized=True, # True: Color, False: Gray
save_name='002.png')
line 16: mode: byte
line 16: mode: byte
Out[9]:
(4, 'H', '/media/yan1/huanghy/notebook/002.png')
In [10]:
Functions
In [11]:
def my_function():
print("Hello from a function")
In [12]:
my_function()
Hello from a function
In [13]:
def my_function(fname):
print(fname + " Refsnes")
my_function("Emil")
my_function("Tobias")
my_function("Linus")
Emil Refsnes
Tobias Refsnes
Linus Refsnes
In [14]:
def add(a,b):
return a+b
In [15]:
def lessthan(a,b):
return a<=b
In [16]:
print(add(5,4))
9
In [17]:
print(lessthan(5,4))
False
In [ ]:
In [18]:
# 01 Lists
# Example of a list with strings
letters = ["a", "b", "c"]
print(letters)
# Example of a list with interger
numbers = [1, 2, 3, 4]
print(numbers)
['a', 'b', 'c']
[1, 2, 3, 4]
In [19]:
# Each item in this list is a list it self.
matrix = [[0, 1], [2, 3]]

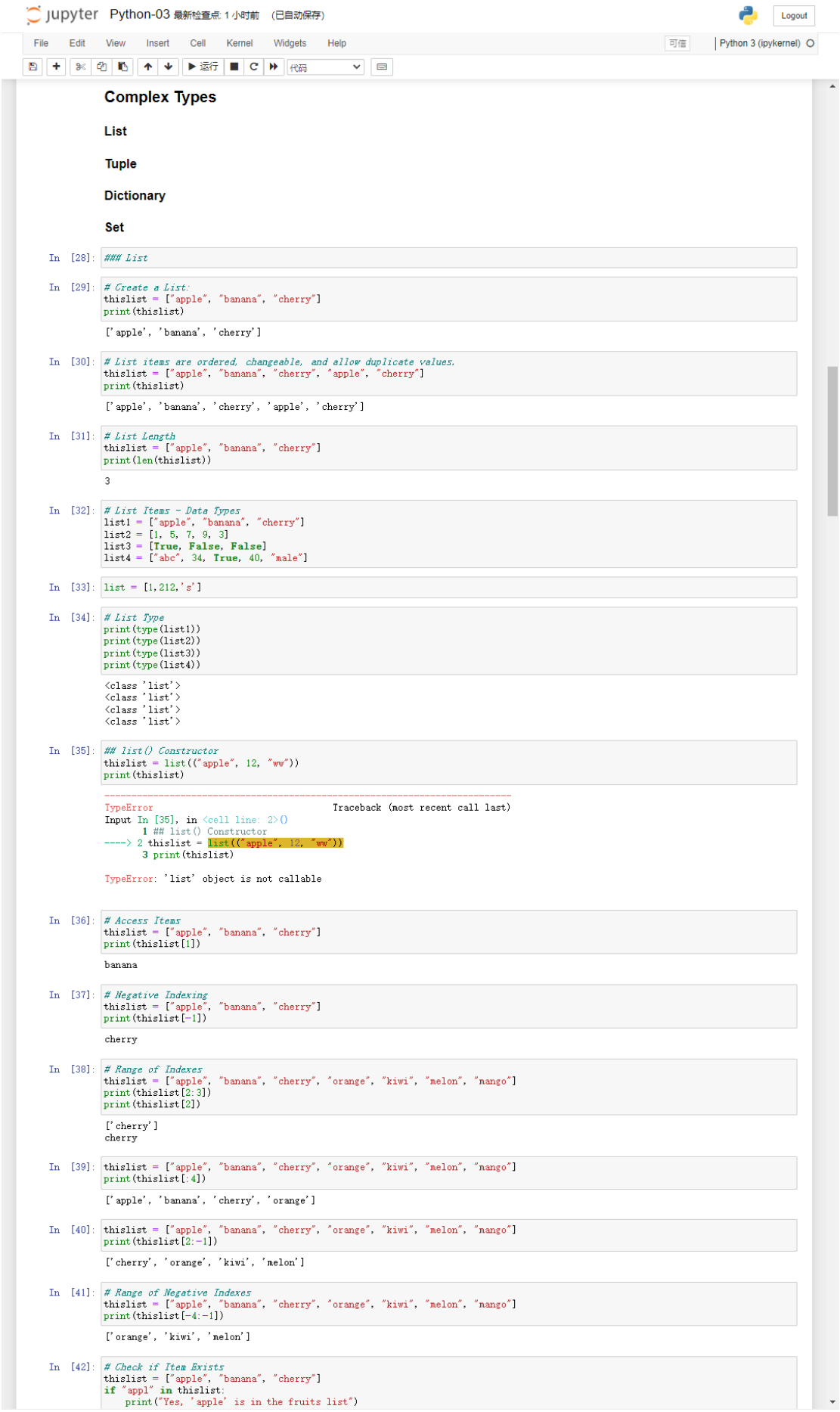
生成的二维码如下：

带有背景的二维码如下：


jupyter Python-02 最新检查点: 40 分钟前 (更改未保存)
Logout
File Edit View Insert Cell Kernel Widgets Help
可信 Python 3 (ipykernel)
运行 代码
['a', 'b', 'c', 'd', 'e', 'f', 'g']
In [35]:
Remove an item for a list using the method remove()
specify the item
letters.remove("c") # this will remove the first "c" in the list
print(letters)
['a', 'b', 'd', 'e', 'f', 'g']
In [36]:
letters.append("e")
In [37]:
print(letters)
['a', 'b', 'd', 'e', 'f', 'g', 'e']
In [38]:
letters.remove("e") # this will remove the first "c" in the list
print(letters)
['a', 'b', 'd', 'f', 'g', 'e']
In [39]:
Remove item using the del statement
with this statement we can remove one item or a range of items by the index
del letters[0:2]
print(letters)
['d', 'f', 'g', 'e']
In [40]:
# to clear the list and delete all the items
letters.clear()
print(letters)
[]
In [41]:
# 06 Finding Items
letters = ["a", "b", "c", "d", "e", "f", "g"]
print(letters.index("d"))
3
In [42]:
print(letters.index("h"))
ValueError Traceback (most recent call last)
Input In [42], in <cell line: 1>()
----> 1 print(letters.index("h"))
ValueError: 'h' is not in list
In [44]:
# Example 1
if "d" in letters:
    print(letters.index("d"))
# Example 2
print(letters.count("m"))
3
0
In [45]:
# 07 Sorting Lists
numbers = [5, 51, 2, 15, 6]
print(numbers)
# sort numbers with the sort() method
numbers.sort()
print(numbers)
numbers.sort(reverse=True)
print(numbers)
[5, 51, 2, 15, 6]
[2, 5, 6, 15, 51]
[51, 15, 6, 5, 2]
In [46]:
numbers = [5, 51, 2, 15, 6]
print(numbers)
# The sort function can be used an it will creat a new list
print(sorted(numbers))
print(sorted(numbers, reverse=True))
print(numbers)
[5, 51, 2, 15, 6]
[2, 5, 6, 15, 51]
[51, 15, 6, 5, 2]
[5, 51, 2, 15, 6]
In [47]:
# Zip Function
list_1 = [1, 2, 3, 4]
list_2 = [10, 20, 30, 40, 50, 60]
combined_list = zip(list_1, list_2)
print(list(combined_list))
list_3 = list(zip("abcd", list_2, list_1))
print(list_3)
[(1, 10), (2, 20), (3, 30), (4, 40)]
[('a', 10, 1), ('b', 20, 2), ('c', 30, 3), ('d', 40, 4)]
In [48]:
Swapping Variables
To swap two variables, we can use a third variable to backup the frist variable

（4）Python-03.ipynb


jupyter Python-03 最新检查点: 1 小时前 (更改未保存)
Logout
File Edit View Insert Cell Kernel Widgets Help
可信
Python 3 (ipykernel)
运行
代码
In [ ]:
Comments
Comments can be used to explain Python code.
Comments can be used to make the code more readable.
Comments can be used to prevent execution when testing code.
In [1]:
#This is a comment
print("Hello, World!")
Hello, World!
In [2]:
print("Hello, World!") #This is a comment
Hello, World!
In [3]:
#This is a comment
#written in
#more than just one line
print("Hello, World!")
Hello, World!
In [4]:
"""
This is a comment
written in
more than just one line
"""
print("Hello, World!")
Hello, World!
Syntax
In [5]:
if 5 > 2:
print("Five is greater than two!")
Input In [5]
print("Five is greater than two!")
^
IndentationError: expected an indented block after 'if' statement on line 1
In [6]:
if 5 > 2:
print("Five is greater than two!")
if 5 > 2:
print("Five is greater than two!")
Five is greater than two!
Five is greater than two!
In [7]:
if 5 > 2:
print("Five is greater than two!")
Five is greater than two!
In [8]:
Type
In [9]:
x = "Hello World"
print(type(x))
<class 'str'>
In [10]:
x = "Hello World"
print(type(x))
<class 'str'>
In [11]:
x = 20.5
print(type(x))
<class 'float'>
In [12]:
x = 20
print(type(x))
<class 'int'>
In [13]:
x = ["apple", "banana", "cherry"]
print(type(x))
<class 'list'>
In [14]:
x = ("apple", "banana", "cherry")
print(type(x))
<class 'tuple'>
In [15]:
x = {"name" : "John", "age" : 36}
print(type(x))
<class 'dict'>
In [16]:
x = range(6)
print(type(x))


jupyter Python-03 最新检查点: 1 小时前 (已自动保存)
Logout
File Edit View Insert Cell Kernel Widgets Help
可信
Python 3 (ipykernel)
运行
代码
Complex Types
List
Tuple
Dictionary
Set
In [28]: ### List
In [29]: # Create a List:
thislist = ["apple", "banana", "cherry"]
print(thislist)
['apple', 'banana', 'cherry']
In [30]: # List items are ordered, changeable, and allow duplicate values.
thislist = ["apple", "banana", "cherry", "apple", "cherry"]
print(thislist)
['apple', 'banana', 'cherry', 'apple', 'cherry']
In [31]: # List Length
thislist = ["apple", "banana", "cherry"]
print(len(thislist))
3
In [32]: # List Items - Data Types
list1 = ["apple", "banana", "cherry"]
list2 = [1, 5, 7, 9, 3]
list3 = [True, False, False]
list4 = ["abc", 34, True, 40, "male"]
In [33]: list = [1,212,'s']
In [34]: # List Type
print(type(list1))
print(type(list2))
print(type(list3))
print(type(list4))
<class 'list'>
<class 'list'>
<class 'list'>
<class 'list'>
In [35]: ## list() Constructor
thislist = list(("apple", 12, "ww"))
print(thislist)
TypeError Traceback (most recent call last)
Input In [35], in <cell line: 2>()
1 ## list() Constructor
----> 2 thislist = list(("apple", 12, "ww"))
3 print(thislist)
TypeError: 'list' object is not callable
In [36]: # Access Items
thislist = ["apple", "banana", "cherry"]
print(thislist[1])
banana
In [37]: # Negative Indexing
thislist = ["apple", "banana", "cherry"]
print(thislist[-1])
cherry
In [38]: # Range of Indexes
thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[2:3])
print(thislist[2])
['cherry']
cherry
In [39]: thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[:4])
['apple', 'banana', 'cherry', 'orange']
In [40]: thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[2:-1])
['cherry', 'orange', 'kiwi', 'melon']
In [41]: # Range of Negative Indexes
thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[-4:-1])
['orange', 'kiwi', 'melon']
In [42]: # Check if Item Exists
thislist = ["apple", "banana", "cherry"]
if "appl" in thislist:
print("Yes, 'apple' is in the fruits list")