WenQuanYi Micro Hei [Scale=0.9]WenQuanYi Micro Hei Mono song-WenQuanYi Micro Hei sfWenQuanYi Micro Hei "zh" = 0pt plus 1pt

PythonåřŕÄ¿Ńå■Ř

Release 1.2.378

ä¡lJèĂĚïijŽEmily Guo

Aug 03, 2020

Contents

Contents:

CHAPTER 1

åĽ■èÍĂ

åŚŁåĹńæđŕçĞěïijŇ60çğŠå■ęäijŽäÿĂäÿłåřŔä¿Ńå■ŘïijŇçşżçż§å■ęäźăPythonïijŇäżŐåĚěéŮÍåĹřåď ğåÿĹã

æĎ§åŔŮPythonäźŃç¿Ő | äÿĂãĂĄPythonå§žçąĂ |äžŇãĂĄPythonå■ŮçňęäÿšåŇæ■čåĹŹläÿĽ˜ãĂĄPytho

åŘŐçż■çńăèŁĆåŇĚæŃňïijŽPyQtåĹűä¡IJGUI; FlaskåĽ■çńŕåijĂåŔŚ; PythonæŢřæ■őåĹĘæđŘ

çŻőåĽ■èűĚè£ĞG200äÿłåřŔä¿Ńå■ŘãĂĆ

æĹŚäijŽäÿĂçŻť˜æŕŔåď˙l˜æŻť˜æŰřäÿĂäÿłåřŔä¿Ńå■ŘïijŇæňćè£Őstar.

https://github.com/jackzhenguo/python-small-examples

å¿Ĺè■čåźÿïijŇè£ŹäÿłåžŞèćńAIæİČåĺĄåłŠä¡ŞãĂŁéĞŔå■Řä¡■ãĂŃæŁěéĄŞïijŇäÿŃéİćæŸŕæŁěéĄŞæĹłåŽ

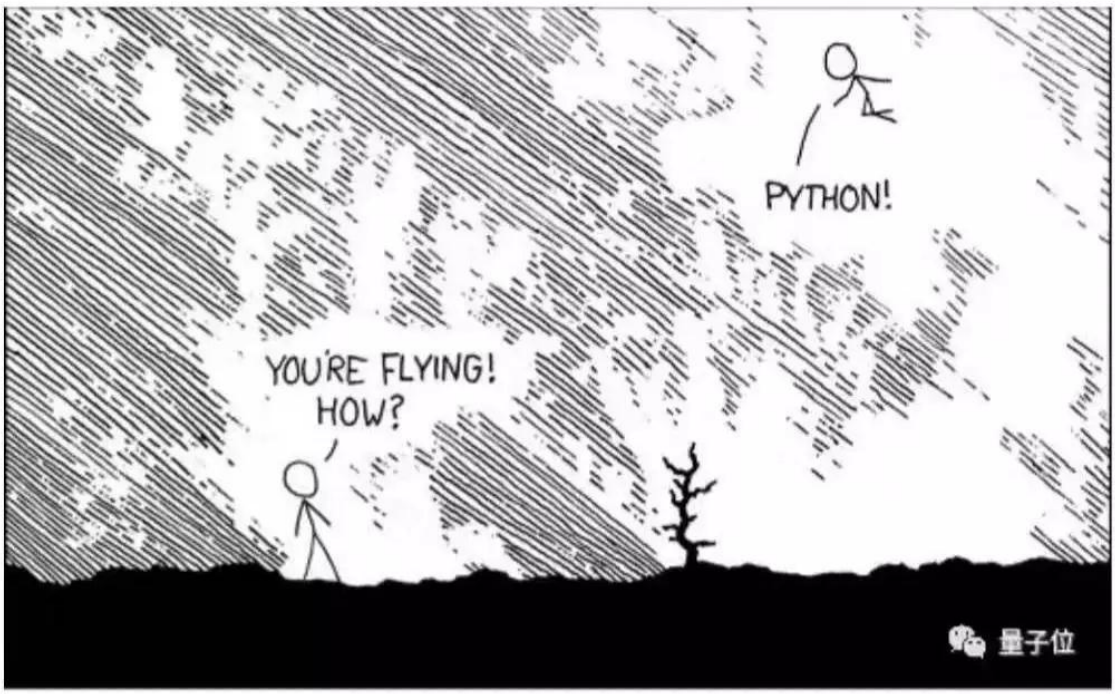
Python趣味打怪：60秒学会一个例子，147段简单代码助你从入门到大师 | 中文资源

关注前沿科技 量子位 1周前

鱼羊 发自 凹非寺

量子位 报道 | 公众号 QbitAI

人生苦短，编程苦手，不妨学起Python，感受一飞冲天的快乐。



不要害怕学习的过程枯燥无味，这里有程序员jackzhenguo打造的一份中文Python“糖果包”：147个代码小样，60秒一口，营养又好玩，从

CHAPTER 2

PythonäźŃèůŕ

äžžçŤ§èŃęç§■ïijŇæĹŚçŤĺPython

2.1 1 çőĂæťĄäźŃç¿Ő

éĂŽè£ĞäÿĂèąŇäżčçăĄïijŇä¡ŞäijŽPythonèŕ■èĺĂçőĂæťĄäźŃç¿Ő

1) äÿĂèąŇäżčçăĄäžďæ■ća,bïijŽ

```
a, b = b, a
```

2) äÿĂèąŇäżčçăĄåŔ■è¡ňåĹŮèąĺ

```
[1,2,3][::-1] # [3,2,1]
```

3) äÿĂèąŇäżčçăĄåŘĹåźűäÿďäÿłå■ŮåĚÿ

```
{**{'a':1,'b':2}, **{'c':3}} # {'a': 1, 'b': 2, 'c': 3}
```

4) äÿĂèąŇäżčçăĄåĹŮèąĺåŐżéĞ■

```
set([1,2,2,3,3,3]) # {1, 2, 3}
```

5) äÿĂèąŇäżčçăĄæśĆåď'Žäÿłå ĹŮèąĺäÿ■çŽĎæIJĂåď'ğåĂij

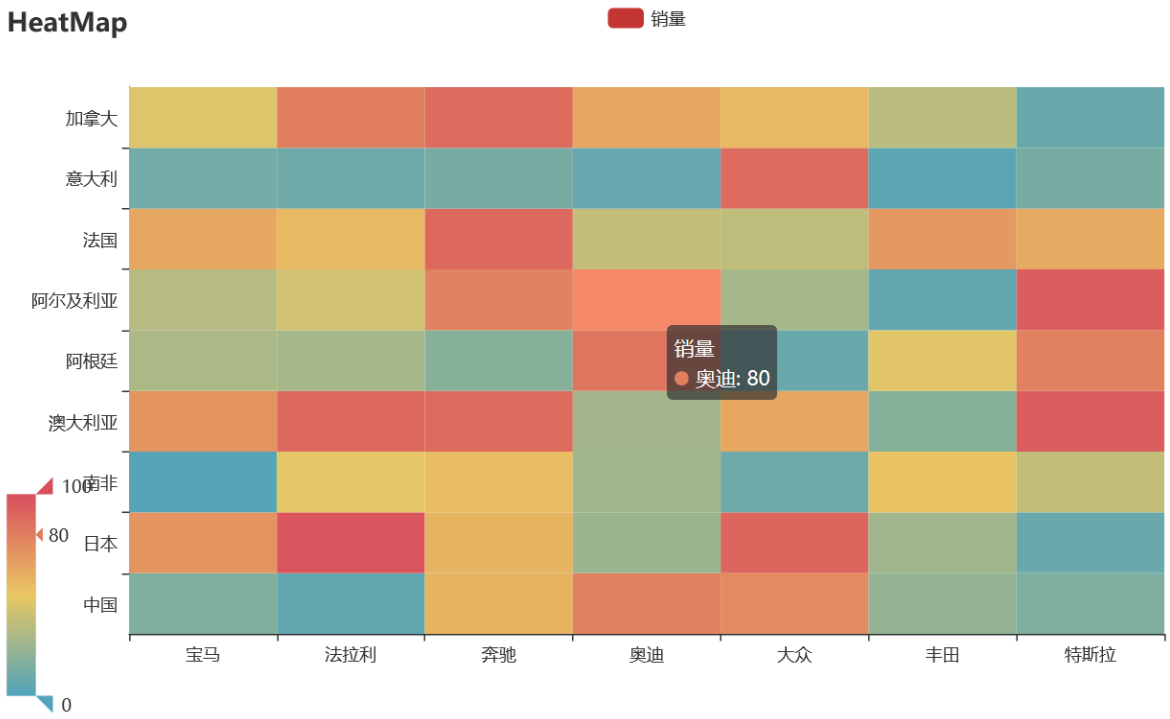
```
max(max([ [1,2,3], [5,1], [4] ], key=lambda v: max(v))) # 5
```

6) äÿĂèąŇäżčçăĄçŤ§æĹŘéĂĘåžŔåžŘåĹŮ

```
list(range(10,-1,-1)) # [10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
```

2.2 2 PythonçżŸåŻ¿

PythonçżŸåŻ¿æŰźä¿£ãĂĄæijĆäžőïijŇçŤżåŻ¿çěđåŹĺpyechartsåĞäèąŇäżčçăĄåřśèČ¡çżŸåĹűåĞžçČ■åŁ



çĆńéĚůçŽĎæřťçŘČåŻ¿ïijŽ

çżŔåÿÿä¡£çŤĺçŽĎèŕ■äžŚåŻ¿ïijŽ

2.3 3 PythonåŁĺçŤż

äžĚéĂĆçŤĺPythonçŽĎåÿÿçŤĺçżŸåŻ¿åžŞïijŽMatplotlibïijŇåřśèČ¡åĹűä¡IJåĞžåŁĺçŤżïijŇè¿ĚåŁľçőŮæşŢ

å¡Šåźűæ0ŠåžŔåŁĺçŤżåśŢçď žïijŽ

ä¡£çŤĺturtuleçżŸåĹűçŽĎæijńåď l'éŻłèŁśïijŽ

timelineæŮűéŮť è¡őæŠ■åŻ¿ïijŽ

2.4 4 PythonæŢřæ■őåĹĘæđŘ

PythonéİđåÿÿéĂĆåŘĹåĄŽæŢřåĂijèőąçőŮãĂĄæŢřæ■őåĹĘæđŘïijŇäÿĂèąŇäżčçăĄåőŇæĹŘæŢřæ■őéĂ


```
pd.pivot_table(df, index=['Manager', 'Rep'], values=['Price'],␣
↪aggfunc=np.sum)
```

2.5 5 PythonælJžåŹĺå■ęäźă

PythonæIJžåŹĺå■ęäźăåžŞSklearnåŁ§èČ¡åijžåď ğïijŇæŐěåŔčæŸŞçŤĺïijŇåŇĚæŃňæŢřæ■őéćĎåď ĎçŘ

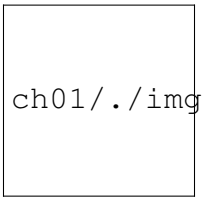
```
from sklearn.cluster import KMeans
KMeans( n_clusters=3 )
```

2.6 6 Python-GUI

PyQtèő¿èőąåŹĺåijĂåŔŚGUIïijŇèČ¡åď §è£ĚéĂ§éĂŽè£ĞæŃŰåŁĺçżĎåžžæŘ■åžžåĞžæİěïijŇä¡£çŤĺæŰźä¿

éŹď æ■d'äźŃåď ŰïijŇä¡£çŤĺPythonçŽĎFlaskæąĘæđűæŘ■åžžWebæąĘæđűïijŇäź§éİđåÿÿæŰźä¿£ãĂĆ

æĂżäźŃïijŇåIJĺè£ŹäÿłPythonåřŔä¿Ńå■ŘïijŇä¡ăéČ¡èČ¡å■ęåĹřåĚşäžŐä¡£çŤĺPythonåźšæť žçŽĎæŰźä


ch01/./img/kmeans.png

CHAPTER 3

äÿĂãĂĄPythonå§žçąĂ

Pythonå§žçąĂäÿżèęĄæĂżçżŞPythonåÿÿçŤĺåĘĚç¡őåĞ¡æŢřïijŻPythonçŃňæIJL’çŽĎèŕ■æşŢçL’źæĂğã globalç■L’ïijŻåĘĚç¡őæŢřæ■őçżŞæđĎåŇĚæŃňïijŽåĹŮèąĺ(list), å■ŮåĚÿ(dict), éŻĘåŘĹ(set), åĚČçżĎ(tuple) äżěåŔŁçŻÿåĚşçŽĎéńŸçžğæĺąåİŮcollectionsäÿ■çŽĎCounter, namedtuple, defaultdictïijŇheapqæĺąåİŮãĂĆçŻőåL’■åĚśæIJL’86äÿłářŔä¿Ńå■Ř

3.1 1 æśĆçżİåŕźåĂij

çżİåŕźåĂijæĹŰåď■æŢřçŽĎæĺą

```
In [1]: abs(-6)
Out[1]: 6
```

3.2 2 åĚČçťăéČ¡äÿžçIJ§

æŐěåŔŮäÿĂäÿłè£■äżčåŹĺïijŇåęĆæđIJè£■äżčåŹĺçŽĎæL’ĂæIJL’åĚČçt’ăéČ¡äÿžçIJ§ïijŇéĆčäźĹè£Ť

```
In [2]: all([1,0,3,6])
Out[2]: False

In [3]: all([1,2,3])
Out[3]: True
```

3.3 3 åĚČçťäèĞşåřŚäÿĂäÿłäÿžçIJ§

æŐěåŔŮäÿĂäÿłè£■äžčåŹĺïijŇåęĆæđIJè£■äžčåŹĺéĞŇèĞşåřŚæIJL’äÿĂäÿłåĚČçť’ääÿžçIJ§ïijŇéĆčäź

```
In [4]: any([0,0,0,[]])
Out[4]: False

In [5]: any([0,0,1])
Out[5]: True
```

3.4 4 asciiåśŢçď’žåŕźèšą

èřČçŤĺåŕźèšąçŽĎ**repr**() æŰźæşŢïijŇèŐůå¿ŮèřěæŰźæşŢçŽĎè£ŤåŻđåĂijïijŇåęĆäÿŃä¿Ňå■Řè£ŤåŻđåĂi

```
In [1]: class Student():
   ...:     def __init__(self,id,name):
   ...:         self.id = id
   ...:         self.name = name
   ...:     def __repr__(self):
   ...:         return 'id = '+self.id +', name = '+self.name
   ...:
   ...:

In [2]: xiaoming = Student(id='001',name='xiaoming')

In [3]: print(xiaoming)
id = 001, name = xiaoming

In [4]: ascii(xiaoming)
Out[4]: 'id = 001, name = xiaoming'
```

3.5 5 å■Ąè¡ňäžŇ

åřĘå■Ąè£ŻåĹűè¡ňæ■ćäÿžäžŇè£ŻåĹű

```
In [1]: bin(10)
Out[1]: '0b1010'
```

3.6 6 å■Ąè¡ňåĚń

åřĘå■Ąè£ŻåĹűè¡ňæ■ćäÿžåĚńè£ŻåĹű

```
In [1]: oct(9)
Out[1]: '0o11'
```

3.7 7 å■Ąè¡ňå■ĄåĚ■

åřĘå■Ąè£ŻåĹűè¡ňæ■ćäÿžå■ĄåĚ■è£ŻåĹű

```
In [1]: hex(15)
Out[1]: '0xf'
```

3.8 8 åĹďæŰ■æŸŕçIJ§æŸŕåĄĞ

æţŃèŕŢäÿĂäÿłåŕźèśąæŸŕTrue, è£ŸæŸŕFalse.

```
In [1]: bool([0,0,0])
Out[1]: True

In [2]: bool([])
Out[2]: False

In [3]: bool([1,0,1])
Out[3]: True
```

3.9 9 å■Ůçňęäÿšè¡ňå■ÙèŁĆ

åřĘäÿĂäÿłå■Ůçňęäÿšè¡ňæ■ćæĹŘå■ÙèŁĆçśżåđŃ

```
In [1]: s = "apple"

In [2]: bytes(s,encoding='utf-8')
Out[2]: b'apple'
```

3.10 10 è¡ňäÿžå■Ůçňęäÿš

åřĘå■ŮçňęçśżåđŃãĂĄæŢřåĂijçśżåđŃç■L'è¡ňæ■ćäÿžå■ŮçňęäÿšçśżåđŃ

```
In [1]: i = 100

In [2]: str(i)
Out[2]: '100'
```

3.11 11 æŸŕåŘęåŔŕèřČçŤÍ

åĹďæŰ■åŕźèśąæŸŕåŘęåŔŕèćńèřČçŤĺïijŇèČ¡èćńèřČçŤĺçŽĎåŕźèśąåřśæŸŕäÿĂäÿłcallable åŕźèśąïijŇærŤåęĆåĞ¡æŢř str, int ç■L'éČ¡æŸŕåŔŕèćńèřČçŤĺçŽĎïijŇä¡ĘæŸŕä¿Ńå■Ř4 äÿ■xiaomingåőđä¿ŃæŸŕäÿ■åŔŕèćńèřČçŤĺçŽĎïijŽ

```
In [1]: callable(str)
Out[1]: True

In [2]: callable(int)
Out[2]: True

In [3]: xiaoming
Out[3]: id = 001, name = xiaoming

In [4]: callable(xiaoming)
Out[4]: False
```

åęĆædIJæČşèől'xiaomingèČ¡èćńèřČçŤÍ xiaoming(), éIJĂèęĄéĞ■åĘŹStudentçśżçŽĎ__call__æ

```
In [1]: class Student():
   ...:     def __init__(self,id,name):
   ...:         self.id = id
   ...:         self.name = name
   ...:     def __repr__(self):
   ...:         return 'id = '+self.id +', name = '+self.name
   ...:     def __call__(self):
   ...:         print('I can be called')
   ...:         print(f'my name is {self.name}')
   ...:
   ...:

In [2]: t = Student('001','xiaoming')

In [3]: t()
I can be called
my name is xiaoming
```

3.12 12 å■Ąè¡ňASCII

æ§ěçIJŃå■Ąè£ŻåĹűæŢťæŢřåŕźåžŤçŽĎASCIIå■Ůçňę

```
In [1]: chr(65)
Out[1]: 'A'
```

3.13 13 ASCIIè¡ňå■Ą

æ§ěçIJŃæ§ŘäÿłASCIIå■ŮçňęářźåžŤçŽĎå■Ąè£ŻåĹűæŢř

```
In [1]: ord('A')
Out[1]: 65
```

3.14 14 éİŹæĂĄæŰźæşŢ

classmethod èčĚéěřåŹĺářźåžŤçŽĎåĞ¡æŢřäÿ■éIJĂèęĄåőđä¿ŃåŇŰïijŇäÿ■éIJĂèęĄ selfåŔĆæŢřïijŇä¡Eçňňäÿ˘ĂäÿłåŔĆæŢřéIJĂèęĄæŸŕèąĺçđ'žèĞłèžńçśžçŽĎ cls åŔĆæŢřïijŇåŕäžěæİěěřČçŤĺçśžçŽĎåśđæĂğïijŇçśžçŽĎæŰźæşŢïijŇåőđä¿ŃåŇŰářźèśąç■Ľ˘ãĂĆ

```
In [1]: class Student():
    ...:     def __init__(self,id,name):
    ...:         self.id = id
    ...:         self.name = name
    ...:     def __repr__(self):
    ...:         return 'id = '+self.id +', name = '+self.name
    ...:     @classmethod
    ...:     def f(cls):
    ...:         print(cls)
```

3.15 15 æĽ'ğèąŇå■Ůçňęäÿšèąĺçđ'žçŽĎäžčçăĄ

åřĘå■Ůçňęäÿšçij Űèŕśæ ĹŘpythonèČ¡èŕĘåĹńæĹŰåŔŕæĽ'ğèąŇçŽĎäžčçăĄïijŇäź§åŔŕäžěåřĘæŰĞå■Ůèŕź

```
In [1]: s  = "print('helloworld')"

In [2]: r = compile(s,"<string>", "exec")

In [3]: r
Out[3]: <code object <module> at 0x0000000005DE75D0, file "<string>
 ↪", line 1>

In [4]: exec(r)
helloworld
```

3.16 16 åĹŻåžžåđ■æŢř

åĹŻåžžäÿĂäÿłåđ■æŢř

```
In [1]: complex(1,2)
Out[1]: (1+2j)
```

3.17 17 åŁĺæĂĄåĹăéŹďåśđæĂğ

åĹăéŹďåŕźèśąçŽĎåśđæĂğ

3.18 18 è¡ñäÿžå■ŮåĚÿ

åĹŻåžžæŢřæ■őå■ŮåĚÿ

```
In [1]: dict()
Out[1]: {}

In [2]: dict(a='a',b='b')
Out[2]: {'a': 'a', 'b': 'b'}

In [3]: dict(zip(['a','b'],[1,2]))
Out[3]: {'a': 1, 'b': 2}

In [4]: dict([('a',1),('b',2)])
Out[4]: {'a': 1, 'b': 2}
```

3.19 19 äÿĂéŤőæ§ěçIJŃåŕźèśąæĽĂæIJĽæŰźæşŢ

äÿ■åÿęåŔĆæŢřæŮűè£ŤåŻđå¡ŞåL'■èŇČåżt'åĘĚçŽĎåŔŸéĞŔãĂĄæŰźæşŢåŠŇåőŽäźĽçŽĎçśżåđŃå

```
In [96]: dir(xiaoming)
Out[96]:
['__class__',
 '__delattr__',
 '__dict__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
```

(continues on next page)

(continued from previous page)

```
 '__le__',
 '__lt__',
 '__module__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__setattr__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 '__weakref__',

 'name']
```

3.20 20 åŔŰåŢĘåŠŇä¡ŹæŢř

åĹĘåĹńåŔŰåŢĘåŠŇä¡ŹæŢř

```
In [1]: divmod(10,3)
Out[1]: (3, 1)
```

3.21 21 æđŽäÿ¿åŕźèśą

è£ŤåŻđäÿĂäÿłåŔŕäżěæđŽäÿ¿çŽĎåŕźèśąïijŇèŕěåŕźèśąçŽĎnext()æŰźæşŢåřĘè£ŤåŻđäÿĂäÿłåĚČçżĎãĂĆ

```
In [1]: s = ["a","b","c"]
   ...: for i ,v in enumerate(s,1):
   ...:     print(i,v)
   ...:
1 a
2 b
3 c
```

3.22 22 èőąçőŮèąĺè¿¿åijŔ

åřĘå■Ůçňęäÿšstr å¡ŞæĹŘæIJL'æŢĹçŽĎèąĺè¿¿åijŔæİěæśĆåĂijåźűè£ŤåŻđèőąçőŮçżŞæđIJåŔŰåĞžå■Ůçň

```
In [1]: s = "1 + 3 +5"
   ...: eval(s)
```

(continues on next page)

(continued from previous page)

```
    ...:
Out[1]: 9
```

3.23 23 æ§ěçIJŃåŔŸéĞŕæĽĂå■ăå■ŮèŁĆæŢř

```
In [1]: import sys

In [2]: a = {'a':1,'b':2.0}

In [3]: sys.getsizeof(a) # å■ăçŤÍ240äÿłå■ŮèŁĆ
Out[3]: 240
```

3.24 24 è£Ğæżď'åŹÍ

åIJÍåĞ¡æŢřäÿ■èő¿åőŽè£Ğæżď'æİąäżűïijŇè£■äžčåĚČçť'äïijŇä£İçŢŹè£ŤåŻđåĂijäÿžTrueçŽĎåĚČçť'äïijŽ

```
In [1]: fil = filter(lambda x: x>10,[1,11,2,45,7,6,13])

In [2]: list(fil)
Out[2]: [11, 45, 13]
```

3.25 25 è¡ňäÿžæţőçĆźçśżåđŃ

åřĘäÿĂäÿłæŢť'æŢřæĹŰæŢřåĂijåđŃå■Ůçňęäÿšè¡ňæ■ćäÿžæţőçĆźæŢř

```
In [1]: float(3)
Out[1]: 3.0
```

åęĆæđIJäÿ■èČ¡è¡ňåŇŰäÿžæţőçĆźæŢřïijŇåĹŹäijŽæŁěValueError:

```
In [2]: float('a')
ValueError                                Traceback (most recent␣
↪call last)
<ipython-input-11-99859da4e72c> in <module>()
----> 1 float('a')

ValueError: could not convert string to float: 'a'
```

3.26 26 å■ŮçňęäÿšæäijåijŕåŇŰ

æäijåijŕåŇŰè¿ŞåĞžå■ŮçňęäÿšïijŇformat(value, format*spec)åőđèt'ĺäÿŁæŸŕèřČçŤĺäžĘvalueçŽĎ***form

```
In [104]: print("i am {0},age{1}".format("tom",18))
i am tom,age18
```

3.1415926	{:.2f}	3.14	ä£İçŢŹåřŕæŢřçĆźåŘŐäÿďä¡■
3.1415926	{:+.2f}	+3.14	åÿęçňęåŕůä£İçŢŹåřŕæŢřçĆźåŘŐäÿďä¡■
-1	{:+.2f}	-1.00	åÿęçňęåŕůä£İçŢŹåřŕæŢřçĆźåŘŐäÿďä¡■
2.71828	{:.0f}	3	äÿ■åÿęåřŕæŢř
5	{:0>2d}	05	æŢřå■ŮèąěéŻű (åąńåĚĚåůęè¿ź, åő¡åžęäÿž2)
5	{:x<4d}	5xxx	æŢřå■Ůèąěx (åąńåĚĚåŕşè¿ź, åő¡åžęäÿž4)
10	{:x<4d}	10xx	æŢřå■Ůèąěx (åąńåĚĚåŕşè¿ź, åő¡åžęäÿž4)
1000000	{:,}	1,000,000	äżěéĂŮåŕůåĹĘéŽŤçŽĎæŢřå■Ůæäijåijŕ
0.25	{:.2%}	25.00%	çŹ¿åĹĘæŕŤæäijåijŕ
1000000000	{:.2e}	1.00e+09	æŇĞæŢřèőřæşŢ
18	{:>10d}	âĂŸ 18âĂŹ	åŕşåŕźé¡Ř (éżŸèőď, åő¡åžęäÿž10)
18	{:<10d}	âĂŸ18 âĂŸ	åůęåŕźé¡Ř (åő¡åžęäÿž10)
18	{:^10d}	âĂŸ 18 âĂŸ	äÿ■éŮťåŕźé¡Ř (åő¡åžęäÿž10)

3.27 27 åĘżçżŞéŻĘåŘĹ

åĹŻåżžäÿĂäÿłäÿ■åŕŕä£őæŤźçŽĎéŻĘåŘĹãĂĆ

```
In [1]: frozenset([1,1,3,2,3])
Out[1]: frozenset({1, 2, 3})
```

åŽääÿžäÿ■åŕŕä£őæŤźïijŇæĽĂäżěæšąæIJĽåČŕsetéĆčæăůçŽĎaddåŠŇpopæŰźæşŢ

3.28 28 åŁĺæĂĄèŐůåŘŰåŕźèśąåśđæĂğ

èŐůåŘŰåŕźèśąçŽĎåśđæĂğ

```
In [1]: class Student():
   ...:     def __init__(self,id,name):
   ...:         self.id = id
   ...:         self.name = name
   ...:     def __repr__(self):
   ...:         return 'id = '+self.id +', name = '+self.name

In [2]: xiaoming = Student(id='001',name='xiaoming')
```

(continues on next page)

(continued from previous page)

```
In [3]: getattr(xiaoming,'name') #␣
↪èŐůåŔŰxiaomingè£Źäÿłåőđä¿ŃçŽĎnameåśđæĂğåĂij
Out[3]: 'xiaoming'
```

3.29 29 åŕźèśąæŸŕŕåŘęæIJL'è£ŹäÿłåśđæĂğ

```
In [1]: class Student():
   ...:     def __init__(self,id,name):
   ...:         self.id = id
   ...:         self.name = name
   ...:     def __repr__(self):
   ...:         return 'id = '+self.id +', name = '+self.name

In [2]: xiaoming = Student(id='001',name='xiaoming')
In [3]: hasattr(xiaoming,'name')
Out[3]: True

In [4]: hasattr(xiaoming,'address')
Out[4]: False
```

3.30 30 è£ŤåŻđåŕźèśąçŽĎåŞĹåÿŇåĂij

è£ŤåŻđåŕźèśąçŽĎåŞĹåÿŇåĂijïijŇåĂijå¿ŮæşĺæĎŔçŽĎæŸŕèĞłåőŽäźL'çŽĎåőđä¿ŃéČ¡æŸŕåŔŕåŞĹåÿŇç
dict, setç■L'åŔŕåŔŸåŕźèśąéČ¡æŸŕäÿ■åŔŕåŞĹåÿŇçŽĎ(unhashable)

```
In [1]: hash(xiaoming)
Out[1]: 6139638

In [2]: hash([1,2,3])
TypeError                                 Traceback (most recent␣
↪call last)
<ipython-input-32-fb5b1b1d9906> in <module>()
----> 1 hash([1,2,3])

TypeError: unhashable type: 'list'
```

3.31 31 äÿĂéŤőåÿőåŁľ

è£ŤåŻđåŕźèśąçŽĎåÿőåŁľæŰĞæąč

```
In [1]: help(xiaoming)
Help on Student in module __main__ object:

class Student(builtins.object)
 |  Methods defined here:
 |
 |  __init__(self, id, name)
 |
 |  __repr__(self)
 |
 |  Data descriptors defined here:
 |
 |  __dict__
 |      dictionary for instance variables (if defined)
 |
 |  __weakref__
 |      list of weak references to the object (if defined)
```

3.32 32 åŕźèśąéÙÍçĽŇåŕů

è£ŤåŻđåŕźèśąçŽĎåĘĚå■ŸåIJřåİĂ

```
In [1]: id(xiaoming)
Out[1]: 98234208
```

3.33 33 èŐůåŔŰçŤÍæĹůè¿ŞåĚě

èŐůåŔŰçŤÍæĹůè¿ŞåĚěåĘĚåőź

```
In [1]: input()
aa
Out[1]: 'aa'
```

3.34 34 è¡ňäÿžæŢťåđŃ

int(x, base =10) , xåŕŕèČ¡äÿžå■ŮçňęäÿšæĹŰæŢřåĂijïijŇåřĘx è¡ňæ■ćäÿžäÿĂäÿłæŹőéĂŽæŢťæŢřãĂĆåęĆæđIJåŔĆæŢřæŸŕå■ŮçňęäÿšïijŇéĆčäźĹåőČåŔŕèČ¡åŇĚåŘńçňęåŔ

```
In [1]: int('12',16)
Out[1]: 18
```

3.35 35 isinstance

åĹďæŰ■*object*æŸŕåŘęäÿžçśż*classinfo*çŽĎåőđäżŃïijŇæŸŕè£ŤåŻđtrue

```
In [1]: class Student():
   ...:     def __init__(self,id,name):
   ...:         self.id = id
   ...:         self.name = name
   ...:     def __repr__(self):
   ...:         return 'id = '+self.id +', name = '+self.name

In [2]: xiaoming = Student(id='001',name='xiaoming')

In [3]: isinstance(xiaoming,Student)
Out[3]: True
```

3.36 36 çĹűå■ŘåĚşçşżéĽťåőŽ

```
In [1]: class undergraduate(Student):
    ...:     def studyClass(self):
    ...:         pass
    ...:     def attendActivity(self):
    ...:         pass

In [2]: issubclass(undergraduate,Student)
Out[2]: True

In [3]: issubclass(object,Student)
Out[3]: False

In [4]: issubclass(Student,object)
Out[4]: True
```

åęĆæđIJclassæŸŕclassinfoåĚČçžĎäÿ■æ§ŘäÿłåĚčçť ăçŽĎå■ŘçśżïijŇäź§äijŽè£ŤåŻđTrue

```
In [1]: issubclass(int,(int,float))
Out[1]: True
```

3.37 37 åĹŻåżžè£■äżčåŹĺçśżåđŃ

ä¡£çŤĺiter(obj, sentinel), è£ŤåŻđäÿĂäÿłåŕŕè£■äżčåŕźèšą, sentinelåŘŕçIJAçŢě(äÿĂæŮęè£■äżčåĹřæ■ď åĚČçť ăïijŇçńŇå■şçżĹæ■ć)

```
In [1]: lst = [1,3,5]

In [2]: for i in iter(lst):
    ...:     print(i)
    ...:
1
3
5
```

```
In [1]: class TestIter(object):
    ...:     def __init__(self):
    ...:         self.l=[1,3,2,3,4,5]
    ...:         self.i=iter(self.l)
    ...:     def __call__(self):  #åőŽäźL'äžE¸__call__
 ↪æŰźæşŢçŽĎçśżçŽĎåőđä¿ŃæŸŕåŕŕèřČçŤÍçŽĎ
    ...:         item = next(self.i)
    ...:         print ("__call__ is called,fowhich would return",
 ↪item)
    ...:         return item
    ...:     def __iter__(self): #æŤŕæŇĄè£■äżčå■Ŕèőő(å■şåőŽäźL'æIJL'_
 ↪_iter__()åĞ¡æŢř)
    ...:         print ("__iter__ is called!!")
    ...:         return iter(self.l)
In [2]: t = TestIter()
In [3]: t() # åŻăäÿžåőđçŐřäžE¸__call__ïijŇæL'Ăäżětåőđä¿ŃèČ¡èćńèřČçŤÍ
__call__ is called,which would return 1
Out[3]: 1

In [4]: for e in TestIter(): # åŻăäÿžåőđçŐřäžE¸__iter__
 ↪æŰźæşŢïijŇæL'Ăäżětè Č¡èćńè£■äżč
    ...:     print(e)
    ...:
__iter__ is called!!
1
3
2
3
4
5
```

3.38 38 æĽĂæIJL'åŕźèśąäźŃæăź

object æŸŕæĽĂæIJL'çśżçŽĎå§žçśż

```
In [1]: o = object()

In [2]: type(o)
```

(continues on next page)

(continued from previous page)

```
Out[2]: object
```

3.39 45 æĽŞåijĂæŰĞäżű

è£ŤåŻđæŰĞäżűåŕźèśą

```
In [1]: fo = open('D:/a.txt',mode='r', encoding='utf-8')

In [2]: fo.read()
Out[2]: '\ufefflife is not so long,\nI use Python to play.'
```

modeåŔŰåĂijèąĺïijŽ

å■Ůçňę	æĎŔäźĽ
'r'	èŕżåŔŰïijĹéżŸèőď'ïijĽ
'w'	åĘŹåĚěïijŇåźűåĚĹæĹłæŰ■æŰĞäżű
'x'	æŐŠåőČæĂğåĹŻåżžïijŇåęĆæđIJæŰĞäżűåůšå■ŸåIJĺåĹŹåď'śèť'ě
'a'	åĘŹåĚěïijŇåęĆæđIJæŰĞäżűå■ŸåIJĺåĹŹåIJĺæIJńář¿è£¡åŁă
'b'	äžŇè£ŻåĹűæląåijŔ
't'	æŰĞæIJňæĺąåijŔïijĹéżŸèőď'ïijĽ
'+'	æĽŞåijĂçŤĺäžŐæŻť'æŰřïijĹèŕżåŔŰäÿŐåĘŹåĚěïijĽ

3.40 40 æňąåźĆ

baseäÿžåžŢçŽĎexpæňąåźĆïijŇåęĆæđIJmodçżŹåĞžïijŇåŔŰä¡Ź

```
In [1]: pow(3, 2, 4)
Out[1]: 1
```

3.41 41 æĽŞå■ř

```
In [5]: lst = [1,3,5]

In [6]: print(lst)
[1, 3, 5]

In [7]: print(f'lst: {lst}')
lst: [1, 3, 5]
```

(continues on next page)

(continued from previous page)

```
In [8]: print('lst:{}'.format(lst))
lst:[1, 3, 5]

In [9]: print('lst:',lst)
lst: [1, 3, 5]
```

3.42 42 åĹŻåžžåśđæĂğçŽĎäÿďçğ■æŰźåijŕ

è£ŤåŻđ property åśđæĂğïijŇåĚÿåđŃçŽĎçŤĺæşŢïijŽ

```
class C:
    def __init__(self):
        self._x = None

    def getx(self):
        return self._x

    def setx(self, value):
        self._x = value

    def delx(self):
        del self._x
    # ä¡£çŤĺpropertyçśżåĹŻåžž property åśđæĂğ
    x = property(getx, setx, delx, "I'm the 'x' property.")
```

ä¡£çŤĺpythonèčĚéěřåŹĺïijŇåőđçŐřäÿŐäÿŁåőŇåĚĺäÿĂæăůçŽĎæŢĹæđIJäžčçăĄïijŽ

```
class C:
    def __init__(self):
        self._x = None

    @property
    def x(self):
        return self._x

    @x.setter
    def x(self, value):
        self._x = value

    @x.deleter
    def x(self):
        del self._x
```

3.43 43 åĹŻåžžrangeåžŕåĹů

1) range(stop)
2) range(start, stop[,step])

çŤ§æĹŘäÿĂäÿłäÿ■åŕŕåŔŸåžŕåĹŮïijŽ

```
In [1]: range(11)
Out[1]: range(0, 11)

In [2]: range(0,11,1)
Out[2]: range(0, 11)
```

3.44 50 åŕ■åŘŚè£■äżčåŹÍ

```
In [1]: rev = reversed([1,4,2,3,1])

In [2]: for i in rev:
   ...:     print(i)
   ...:
1
3
2
4
1
```

3.45 45 åŻŻèĹ■äžŤåĚě

åŻŻèĹ■äžŤåĚěïijŇndigitsäżčèąĺåřŕæŢřçĆźåŘŐä£ÍçŢŹåĞää¡■ïijŽ

```
In [11]: round(10.0222222, 3)
Out[11]: 10.022

In [12]: round(10.05,1)
Out[12]: 10.1
```

3.46 46 è¡ňäÿžéŻĘåŘĹçśżåđŃ

è£ŤåŻđäÿĂäÿłsetåŕźèśąïijŇéŻĘåŘĹåĘĚäÿ■åĚĄèőÿæIJL'éĞ■åd'■åĚČçť'ăïijŽ

```
In [159]: a = [1,4,2,3,1]

In [160]: set(a)
Out[160]: {1, 2, 3, 4}
```

3.47 47 è¡ňäÿžåĹĞçĽĞåŕźèšą

class slice(*start*, *stop*[, *step*])

è£ŤåŻđäÿĂäÿłèąĺçđ'žçŤś range(start, stop, step) æĽĂæŇĞåőŽçť'ćåijŢéŻĘçŽĎ sliceåŕźèšąïijŇåőČèől'äžčçăĄåŔŕèŕżæĂğãĂĄåŔŕçżť'æŁđ'æĂğåŔŸåě¡ãĂĆ

```
In [1]: a = [1,4,2,3,1]

In [2]: my_slice_meaning = slice(0,5,2)

In [3]: a[my_slice_meaning]
Out[3]: [1, 2, 1]
```

3.48 48 æŃ£æİěåřśçŤĺçŽĎæŐŠåžŔåĞ¡æŢř

æŐŠåžŔïijŽ

```
In [1]: a = [1,4,2,3,1]

In [2]: sorted(a,reverse=True)
Out[2]: [4, 3, 2, 1, 1]

In [3]: a = [{'name':'xiaoming','age':18,'gender':'male'},{'name':'
   ...: xiaohong','age':20,'gender':'female'}]
In [4]: sorted(a,key=lambda x: x['age'],reverse=False)
Out[4]:
[{'name': 'xiaoming', 'age': 18, 'gender': 'male'},
 {'name': 'xiaohong', 'age': 20, 'gender': 'female'}]
```

3.49 49 æśĆåŠŇåĞ¡æŢř

æśĆåŠŇïijŽ

```
In [181]: a = [1,4,2,3,1]

In [182]: sum(a)
Out[182]: 11
```

(continues on next page)

(continued from previous page)

```
In [185]: sum(a,10) #æśĆåŠŇçŽĎåĹİåğŃåĂijäÿž10
Out[185]: 21
```

3.50 50 è¡ňåĚČçżĎ

tuple() åřĘåŕźèśąè¡ňäÿžäÿĂäÿłäÿ■åŔŕåŔŸçŽĎåžŔåĹŮçśżåđŃ

```
In [16]: i_am_list = [1,3,5]
In [17]: i_am_tuple = tuple(i_am_list)
In [18]: i_am_tuple
Out[18]: (1, 3, 5)
```

3.51 51 æ§ěçIJŃåŕźèśąçśżåđŃ

class type(*name*, *bases*, *dict*)

äijååĚěäÿĂäÿłåŔĆæŢřæŮűïijŇè£ŤåŻđ *object* çŽĎçśżåđŃïijŽ

```
In [1]: class Student():
   ...:     def __init__(self,id,name):
   ...:         self.id = id
   ...:         self.name = name
   ...:     def __repr__(self):
   ...:         return 'id = '+self.id +', name = '+self.name
   ...:
   ...:

In [2]: xiaoming = Student(id='001',name='xiaoming')
In [3]: type(xiaoming)
Out[3]: __main__.Student

In [4]: type(tuple())
Out[4]: tuple
```

3.52 52 èĄŽåŘĹè£■äżčåŹÍ

åĹŻåżžäÿĂäÿłèĄŽåŘĹäžĘæİěèĞłæŕŔäÿłåŔŕè£■äżčåŕźèśąäÿ■çŽĎåĚČçť ăçŽĎè£■äżčåŹĺïijŽ

```
In [1]: x = [3,2,1]
In [2]: y = [4,5,6]
In [3]: list(zip(y,x))
```

(continues on next page)

(continued from previous page)

```
Out[3]: [(4, 3), (5, 2), (6, 1)]

In [4]: a = range(5)
In [5]: b = list('abcde')
In [6]: b
Out[6]: ['a', 'b', 'c', 'd', 'e']
In [7]: [str(y) + str(x) for x,y in zip(a,b)]
Out[7]: ['a0', 'b1', 'c2', 'd3', 'e4']
```

3.53 53 nonlocalçŤĺäžŐåĘĚåţŇåĞ¡æŢřäÿ■

åĚşéŤőèŕ■nonlocalåÿÿçŤĺäžŐåĞ¡æŢřåţŇåěŮäÿ■ïijŇåčřæŸŐåŔŸéĞŔiäÿžéİđåśĂéČĺåŔŸéĞŔïijŻ

åęĆæđIJäÿ■åčřæŸŐïijŇi+=1èąĺæŸŐiäÿžåĞ¡æŢřwrapperåĘĚçŽĎåśĂéČĺåŔŸéĞŔïijŇåŻääÿžåIJĺi+=1åi
variableçŽĎéŤŹèŕŕãĂĆ

```
def excepter(f):
    i = 0
    t1 = time.time()
    def wrapper():
        try:
            f()
        except Exception as e:
            nonlocal i
            i += 1
            print(f'{e.args[0]}: {i}')
            t2 = time.time()
            if i == n:
                print(f'spending time:{round(t2-t1,2)}')
    return wrapper
```

3.54 54 global åčřæŸŐåĚĺåśĂåŔŸéĞŔ

åĚĹåŻđç■ŤäÿžäżĂäźĹèęĄæIJL'globalïijŇäÿĂäÿłåŔŸéĞŔèćńåď'ŽäÿłåĞ¡æŢřåijŢçŤĺïijŇæČşèőľ'åĚĺå

```
i = 5
def f():
    print(i)

def g():
    print(i)
```

(continues on next page)

(continued from previous page)

```
    pass

f()
g()
```

fåŠŇgäÿd'äÿłåĞ¡æŢřéČ¡èČ¡åĚśäžńåŔŸéĞŔiïijŇçÍŃåžŔæšąæIJL'æŁěéŤŹïijŇæL'Ăäžěäż Ű äžňä¿İçĎűäÿ

ä¡ĘæŸŕïijŇåęĆæđIJæĹŚæČşèęĄæIJL'äÿłåĞ¡æŢřåŕźiéĂŠåćđïijŇè£Źæăůïijž

```
def h():
    i += 1

h()
```

æ■d'æŮűæL'ğèąŇçÍŃåžŔïijŇbang, åĞžéŤŹäžĘïijĄ æŁŻåĞžåijĆåÿÿïijžUnboundLocalErrorïijŇåŐ

globalåřśæŸŕäÿžèğčåĘşæ■d'éŮőécŸèĂŇèćńæŔŘåĞžïijŇåIJĺåĞ¡æŢřhåĘĚïijŇæŸ¿çd'žåIJřåŚŁèŕL'çij

3.55 55 éŞ¿åijŔærŤè¿Č

```
i = 3
print(1 < i < 3)   # False
print(1 < i <= 3)   # True
```

3.56 56 äÿ■çŤĺelseåŠŇifåőđçŐřèőąçőŮåŹÍ

```
from operator import *


def calculator(a, b, k):
    return {
        '+': add,
        '-': sub,
        '*': mul,
        '/': truediv,
        '**': pow
    }[k](a, b)
```

(continues on next page)

(continued from previous page)

```
calculator(1, 2, '+')   # 3
calculator(3, 4, '**')   # 81
```

3.57 57 éŞżåijŕæŞ■ä¡IJ

```
from operator import (add, sub)


def add_or_sub(a, b, oper):
    return (add if oper == '+' else sub)(a, b)


add_or_sub(1, 2, '-')   # -1
```

3.58 58 äžd'æ■ćäÿd'åĚČçt'ă

```
def swap(a, b):
    return b, a


print(swap(1, 0))   # (0,1)
```

3.59 59 åŐżælJĂæśĆåźşåİĞ

```
def score_mean(lst):
    lst.sort()
    lst2=lst[1:(len(lst)-1)]
    return round((sum(lst2)/len(lst2)),1)

lst=[9.1, 9.0,8.1, 9.7, 19,8.2, 8.6,9.8]
score_mean(lst) # 9.1
```

3.60 60 æL'Şå■ř99äźŸæşŢèąÍ

æL'Şå■řåĞžåęĆäÿŃæăijåijŕçŽĎäźŸæşŢèąÍ

```
1*1=1
1*2=2   2*2=4
1*3=3   2*3=6   3*3=9
1*4=4   2*4=8   3*4=12  4*4=16
1*5=5   2*5=10  3*5=15  4*5=20  5*5=25
1*6=6   2*6=12  3*6=18  4*6=24  5*6=30  6*6=36
1*7=7   2*7=14  3*7=21  4*7=28  5*7=35  6*7=42  7*7=49
1*8=8   2*8=16  3*8=24  4*8=32  5*8=40  6*8=48  7*8=56  8*8=64
1*9=9   2*9=18  3*9=27  4*9=36  5*9=45  6*9=54  7*9=63  8*9=72
↪9*9=81
```

äÿĂåĚśæIJL'10 èąŇïijŇçňňièąŇçŽĎçňňjåĹŮç■L'äžŐïijŽj*iïijŇ

åĚűäÿ■,

iåŔŰåĂijèŇČåŻt'ïijŽ1<=i<=9

jåŔŰåĂijèŇČåŻt'ïijŽ1<=j<=i

æăźæ■őä¿Ńå■ŘåĹĘæđŘçŽĎèŕ■èÍĂæŔŔè£řïijŇè¡ňåŇŰäÿžåęĆäÿŃäżčçăĄïijŽ

```
for i in range(1,10):
    ...:     for j in range(1,i+1):
    ...:         print('%d*%d=%d'%(j,i,j*i),end="\t")
    ...:     print()
```

3.61 61 åĚĺåśŢåijĂ

åŕźäžŐåęĆäÿŃæŢřçżĎïijŽ

```
[[[1,2,3],[4,5]]]
```

åęĆä¡Ţåő ŇåĚĺåśŢåijĂæĹŘäÿĂçżt'çŽĎãĂĆè£ŹäÿłåŕŔä¿Ńå■ŘåőđçŐřçŽĎflattenæŸŕéĂŠå¡ŠçĽĹïijŇ

```
from collections.abc import *

def flatten(lst, out_lst=None):
    if out_lst is None:
        out_lst = []
    for i in lst:
        if isinstance(i, Iterable): # åĹd'æŰ■iæŸŕåŘęåŔŕè£■äžč
            flatten(i, out_lst)  # åř¿æŢřéĂŠå¡Š
        else:
            out_lst.append(i)    # äžğçŤŞçżŞædIJ
    return out_lst
```

èřČçŤÍflatten:

```
print(flatten([[1,2,3],[4,5]]))
print(flatten([[1,2,3],[4,5]], [6,7]))
print(flatten([[[1,2,3],[4,5,6]]]))
# çżŞæđIJïijž
[1, 2, 3, 4, 5]
[6, 7, 1, 2, 3, 4, 5]
[1, 2, 3, 4, 5, 6]
```

numpyéĞŇçŽĎflatenäÿŐäÿŁéİćçŽĎåĞ¡æŢřåőđçŐřæIJL'äžŻå¿őåęŹçŽĎäÿ■åŘŇïijž

```
import numpy
b = numpy.array([[1,2,3],[4,5]])
b.flatten()
array([list([1, 2, 3]), list([4, 5])], dtype=object)
```

3.62 62 åĹŮèąĺç■L'åĹĘ

```
from math import ceil

def divide(lst, size):
    if size <= 0:
        return [lst]
    return [lst[i * size:(i+1)*size] for i in range(0,␣
↪ceil(len(lst) / size))]


r = divide([1, 3, 5, 7, 9], 2)
print(r)  # [[1, 3], [5, 7], [9]]

r = divide([1, 3, 5, 7, 9], 0)
print(r)  # [[1, 3, 5, 7, 9]]

r = divide([1, 3, 5, 7, 9], -3)
print(r)  # [[1, 3, 5, 7, 9]]
```

3.63 63 åĹŮèąĺåŐŃçijľ

```
def filter_false(lst):
    return list(filter(bool, lst))


r = filter_false([None, 0, False, '', [], 'ok', [1, 2]])
print(r)  # ['ok', [1, 2]]
```

3.64 64 æŻťéŢ£åĹŮèąĺ

```
def max_length(*lst):
    return max(*lst, key=lambda v: len(v))


r = max_length([1, 2, 3], [4, 5, 6, 7], [8])
print(f'æŻt’éŢ£çŽĎåĹŮèąĺæŸŕ{r}')  # [4, 5, 6, 7]

r = max_length([1, 2, 3], [4, 5, 6, 7], [8, 9])
print(f'æŻt’éŢ£çŽĎåĹŮèąĺæŸŕ{r}')  # [4, 5, 6, 7]
```

3.65 65 æśĆäijŮæŢř

```
def top1(lst):
    return max(lst, default='åĹŮèąĺäÿžçl’ž', key=lambda v: lst.
↪count(v))

lst = [1, 3, 3, 2, 1, 1, 2]
r = top1(lst)
print(f'{lst}äÿ■åĞžçŐřæňąæŢřæIJĂåd’ŽçŽĎåĚČçt’ääÿž:{r}')  # [1, 3, 3,␣
↪2, 1, 1, 2]äÿ■åĞžçŐřæňąæŢřæIJĂåd’ŽçŽĎåĚČçt’ääÿž:1
```

3.66 66 åď Žèąĺäź Ńæ IJĂ

```
def max_lists(*lst):
    return max(max(*lst, key=lambda v: max(v)))


r = max_lists([1, 2, 3], [6, 7, 8], [4, 5])
print(r)  # 8
```

3.67 67 åĹŮèąĺæ§ěéĞ■

```
def has_duplicates(lst):
    return len(lst) == len(set(lst))


x = [1, 1, 2, 2, 3, 2, 3, 4, 5, 6]
y = [1, 2, 3, 4, 5]
has_duplicates(x)  # False
has_duplicates(y)  # True
```

3.68 68 åĹŮèąĺåŔ■è¡ň

```
def reverse(lst):
    return lst[::-1]


r = reverse([1, -2, 3, 4, 1, 2])
print(r)  # [2, 1, 4, 3, -2, 1]
```

3.69 69 æţőçĆźæŢřç■Ľåů ő æŢřåĹŮ

```
def rang(start, stop, n):
    start,stop,n = float('%.2f' % start), float('%.2f' % stop),int('
↪%.d' % n)
    step = (stop-start)/n
    lst = [start]
    while n > 0:
        start,n = start+step,n-1
        lst.append(round((start), 2))
    return lst

rang(1, 8, 10) # [1.0, 1.7, 2.4, 3.1, 3.8, 4.5, 5.2, 5.9, 6.6, 7.3,␣
↪8.0]
```

3.70 70 æŇĽæİąäżűåĹĘçżĎ

```
def bif_by(lst, f):
    return [ [x for x in lst if f(x)],[x for x in lst if not f(x)]]

records = [25,89,31,34]
bif_by(records, lambda x: x<80) # [[25, 31, 34], [89]]
```

3.71 71 mapåődçŐřåŘŠéĞŔè£ŘçőŮ

```
#åd'ŽåžŔåĹŮè£ŘçőŮåĞ¡æŢřâĂĬmap(function,iterabel,iterable2)
lst1=[1,2,3,4,5,6]
lst2=[3,4,5,6,3,2]
list(map(lambda x,y:x*y+1,lst1,lst2))
### [4, 9, 16, 25, 16, 13]
```

3.72 72 åĂijælJĂåď'ğçŽĎå■ŮåĚÿ

```
def max_pairs(dic):
    if len(dic) == 0:
        return dic
    max_val = max(map(lambda v: v[1], dic.items()))
    return [item for item in dic.items() if item[1] == max_val]


r = max_pairs({'a': -10, 'b': 5, 'c': 3, 'd': 5})
print(r)  # [('b', 5), ('d', 5)]
```

3.73 73 åŘĹåźűäÿď'äÿłå■ŮåĚÿ

```
def merge_dict(dic1, dic2):
    return {**dic1, **dic2}  # python3.
↪5åŘŐæŤŕæŇĄçŽĎäÿĂèąŇäżčçăĄåőđçŐřåŘĹåźűå■ŮåĚÿ

merge_dict({'a': 1, 'b': 2}, {'c': 3})  # {'a': 1, 'b': 2, 'c': 3}
```

3.74 74 topnå■ŮåĚÿ

```
from heapq import nlargest

# è£ŤåŻđå■ŮåĚÿdåL'■näÿłæIJĂåd'ğåĂijåŕźåžŤçŽĎéŤő

def topn_dict(d, n):
    return nlargest(n, d, key=lambda k: d[k])

topn_dict({'a': 10, 'b': 8, 'c': 9, 'd': 10}, 3)  # ['a', 'd', 'c']
```

3.75 75 åijĆä¡■èŕ■

```
from collections import Counter

# æčĂæŞěäÿd'äÿłå■ŮçňęäÿšæŸŕåŘę␣
↪çŻÿåŘŇå■Ůæŕ■åijĆåžŔèŕ■ïijŇçőĂçğřïijŽäžŠäÿžåŔŸä¡■èŕ■

def anagram(str1, str2):
    return Counter(str1) == Counter(str2)
```

(continues on next page)

(continued from previous page)

```
anagram('eleven+two', 'twelve+one')  # True␣
↪è£ŹæŸŕäÿĂåŕźçěđåźÍçŽĎåŔŸä¡■èŕ■
anagram('eleven', 'twelve')  # False
```

3.76 76 éĂżè¿ŚäÿŁåŘĹåźűå■ŮåĚÿ

(1) äÿd'çğ■åŘĹåźűå■ŮåĚÿæŰźæş Ţ
è£ŹæŸŕäÿĂèĹňçŽĎå■ŮåĚÿåŘĹåźűåĘŹæş Ţ

```
dic1 = {'x': 1, 'y': 2 }
dic2 = {'y': 3, 'z': 4 }
merged1 = {**dic1, **dic2} # {'x': 1, 'y': 3, 'z': 4}
```

ä£őæŤźmerged[âĂŸxâĂŹ]=10ïijŇdic1äÿ■çŽĎxåĂijäÿ■åŔŸïijŇmergedæŸŕéĞ■æŰřçŤ§æĹŘçŽĎäÿĂ

ä¡ĘæŸŕïijŇChainMapå■ť'äÿ■åŘŇïijŇåőČåIJĺåĘĚéČĺåĹŻåžžäžĘäÿĂäÿłåőźçžşè£ŹäžŻå■ŮåĚÿçŽĎåĹŮ

```
from collections import ChainMap
merged2 = ChainMap(dic1,dic2)
print(merged2) # ChainMap({'x': 1, 'y': 2}, {'y': 3, 'z': 4})
```

3.77 77 åŚ¡åŘ■åĚČçżĎæŔŘénŸåŔŕèŕżæĂğ

```
from collections import namedtuple
Point = namedtuple('Point', ['x', 'y', 'z'])  #␣
↪åőŽäźL'åŘ■å■ŮäÿžPointçŽĎåĚČçěŰïijŇå■ŮæőţåśđæĂğæIJL'x,y,z
lst = [Point(1.5, 2, 3.0), Point(-0.3, -1.0, 2.1), Point(1.3, 2.8, -
↪2.5)]
print(lst[0].y - lst[1].y)
```

ä¡£çŤĺåŚ¡åŘ■åĚČçżĎåĘŹåĞžæĺěçŽĎäžčçăĄåŔŕèŕżæĂğæŻť'åě¡ïijŇårď'åĚűåď'ĎçŘĘäÿŁçŹ¿äÿŁå■Čäÿł

3.78 78 æăůæIJňæŁ¡æăů

ä¡£çŤĺsampleæŁ¡æăůïijŇåęĆäÿŃä¿Ńå■ŘäžŐ100äÿłæăůæIJňäÿ■éŽŔæIJžæŁ¡æăů10äÿłãĂĆ

```
from random import randint,sample
lst = [randint(0,50) for _ in range(100)]
print(lst[:5])# [38, 19, 11, 3, 6]
lst_sample = sample(lst,10)
print(lst_sample) # [33, 40, 35, 49, 24, 15, 48, 29, 37, 24]
```

3.79 79 éĞ■æťŮæŢřæ■őéŻĘ

ä¡£çŤĺshuffleçŤĺæİěéĞ■æťŮæŢřæ■őéŻĘïijŇåĂijå¿ŮæşĺæĎŔ‘‘shuffle‘‘æŸŕåŕźlståřśåIJř(in place)æťŮçL’ŇïijŇèŁĆçIJĄå■ŸåĆĺçľžéŮt’

```
from random import shuffle
lst = [randint(0,50) for _ in range(100)]
shuffle(lst)
print(lst[:5]) # [50, 3, 48, 1, 26]
```

3.80 80 10äÿłåİĞåŇĂåĹĘåÿČçŽĎåİŘæăĞçĆź

randomæĺąåİŮäÿ■çŽĎuniform(a,b)çŤ§æĹŘ[a,b)åĘĚçŽĎäÿĂäÿłéŽŔæIJžæŢřïijŇåęĆäÿŃçŤ§æĹŘ1

```
from random import uniform
In [1]: [(uniform(0,10),uniform(0,10)) for _ in range(10)]
Out[1]:
[(9.244361194237328, 7.684326645514235),
 (8.129267671737324, 9.988395854203773),
 (9.505278771040661, 2.8650440524834107),
 (3.84320100484284, 1.7687190176304601),
 (6.095385729409376, 2.377133802224657),
 (8.522913365698605, 3.2395995841267844),
 (8.827829601859406, 3.9298809217233766),
 (1.4749644859469302, 8.038753079253127),
 (9.005430657826324, 7.58011186920019),
 (8.700789540392917, 1.2217577293254112)]
```

3.81 81 10äÿłéńŸæŰŕåĹĘåÿČçŽĎåİŘæăĞçĆź

randomæĺąåİŮäÿ■çŽĎgauss(u,sigma)çŤ§æĹŘåİĞåĂijäÿžu,æăĞåĞĘåůőäÿžsigmaçŽĎæżąèűşéńŸæŰŕåĹĘåÿČçŽĎåĂijïijŇåęĆäÿŃçŤ§æĹŘ10äÿłäžŇçżt’åİŘæăĞçĆźïijŇæ2*x-1)æżąèűşåİĞåĂijäÿž0ïijŇæăĞåĞĘåůőäÿž1çŽĎéńŸæŰŕåĹĘåÿČïijŽ

```
from random import gauss
x = range(10)
y = [2*xi+1+gauss(0,1) for xi in x]
points = list(zip(x,y))
### 10äÿłäžŇçżt’çĆźïijŽ
[(0, -0.86789025305992),
 (1, 4.738439437453464),
 (2, 5.190278040856102),
 (3, 8.05270893133576),
 (4, 9.979481700775292),
```

(continues on next page)

(continued from previous page)

```
 (5, 11.960781766216384),
 (6, 13.025427054303737),
 (7, 14.02384035204836),
 (8, 15.33755823101161),
 (9, 17.565074449028497)]
```

3.82 82 chainénŸæŢĹäÿšèĄŤåď'Žäÿłåőźå'Źĺåŕźèšą

chainåĞ¡æŢřäÿšèĄŤaåŠŇbïijŇåĚijéą¿åĘĚå■ŸæŢĹçŐĞåŘŇæŮűåĘŹæşŢæŻť'åŁääijŸéŻĚãĂĆ

```
from itertools import chain
a = [1,3,5,0]
b = (2,4,6)

for i in chain(a,b):
  print(i)
### çżŞædIJ
1
3
5
0
2
4
6
```

3.83 83 æŞ■ä¡IJåĞ¡æŢřåŕźèšą

```
In [31]: def f():
    ...:     print('i\'m f')
    ...:

In [32]: def g():
    ...:     print('i\'m g')
    ...:

In [33]: [f,g][1]()
i'm g
```

åĹŻåżžåĞ¡æŢřåŕźèšąçŽĎlistïijŇæăźæ■őæČşèęĄèřČçŤĺçŽĎindexïijŇæŰźä¿£çż§äÿĂèřČçŤĺãĂĆ

3.84 84 çŤ§æĹŘéĂĘåžŕåžŕåĹů

```
list(range(10,-1,-1)) # [10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
```

çňňäÿĽäÿłåŕĆæŢřäÿžèť§æŮűïijŇèąĺçď žäżŐçňňäÿĂäÿłåŕĆæŢřåijĂåğŃéĂŠåĞŕïijŇçżĹæ■ćåĹřçňňäžŇ

3.85 85 åĞ¡æŢřçŽĎäžŤçśżåŕĆæŢřä¡£çŤĺä¿Ńå■Ř

pythonäžŤçśżåŕĆæŢřïijŽä¡■ç¡őåŕĆæŢřïijŇåĚşéŤőå■ŮåŕĆæŢřïijŇéżŸèőď åŕĆæŢřïijŇåŕŕåŔŸä¡■ç¡őa

```
def f(a,*b,c=10,**d):
  print(f'a:{a},b:{b},c:{c},d:{d}')
```

éżŸèőď åŕĆæŢř‘‘c‘‘äÿ■èČ¡ä¡■äžŐåŕŕåŔŸåĚşéŤőå■ŮåŕĆæŢř‘‘d‘‘åŘŐ.

èřČçŤĺf:

```
In [10]: f(1,2,5,width=10,height=20)
a:1,b:(2, 5),c:10,d:{'width': 10, 'height': 20}
```

åŕŕåŔŸä¡■ç¡őåŕĆæŢřbåőđåŕĆåŘŐèćńèğčæđŘäÿžåĚČçżĎ(2, 5);èĂŇcåŔŰå¿ŮéżŸèőď åĂij10; dèćńèğčæđŘäÿžå■ŮåĚÿ.

åĘ■æňąèřČçŤĺf:

```
In [11]: f(a=1,c=12)
a:1,b:(),c:12,d:{}
```

a=1äijäåĚěæŮűaåřśæŸŕåĚşéŤőå■ŮåŕĆæŢřïijŇb,déČ¡æIJłäijäåĂijïijŇcèćńäijäåĚě12ïijŇèĂŇéIđéżŸèőd

æşĺæĎŔèğĆåŕ§åŕĆæŢřa, æŮćåŔŕäžěf(1),äź§åŔŕäžěf(a=1) åĚűåŔŕèŕżæĂğæŕŤçňňäÿĂçğ■æŻť åě¡ïijŇåżžèőőä¡£çŤĺf(a=1)ãĂĆåęĆæđIJèęĄåijžåĹűä¡£çŤĺf(a=1) ïijŇéIJ

```
def f(*,a,*b):
  print(f'a:{a},b:{b}')
```

æ■ď æŮűf(1)èřČçŤĺïijŇářĘäijŽæŁěéŤŹïijŽTypeError: f() takes 0 positional arguments but 1 was given

åŔłèČ¡f(a=1) æĽ■èČ¡OK.

èŕť æŸŐåĽ■éIćçŽĎ*åŔŚæŇëä¡IJçŤĺïijŇåőČåŔŸäÿžåŔłèČ¡äijäåĚěåĚşéŤőå■ŮåŕĆæŢřïijŇéĆčäźĹåęĆä

```
In [22]: for name,val in signature(f).parameters.items():
    ...:     print(name,val.kind)
    ...:
a KEYWORD_ONLY
b VAR_KEYWORD
```

åŔŕçIJŃåĹřåŔĆæŢřaçŽĎçśżåđŃäÿžKEYWORD_ONLYïijŇäź§åřśæŸŕäżĚäżĚäÿžåĚşéŤőå■ŮåŔĆæŢřãĂĆ

ä¡ĘæŸŕïijŇåęĆæđIJfåőŽäźĽäÿžïijŽ

```
def f(a,*b):
  print(f'a:{a},b:{b}')
```

æ§ěçIJŃåŔĆæŢřçśżåđŃïijŽ

```
In [24]: for name,val in signature(f).parameters.items():
    ...:     print(name,val.kind)
    ...:
a POSITIONAL_OR_KEYWORD
b VAR_POSITIONAL
```

åŔŕäżěçIJŃåĹřåŔĆæŢřaæŮćåŔŕäżěæŸŕä¡■ç¡őåŔĆæŢřäź§åŔŕæŸŕåĚşéŤőå■ŮåŔĆæŢřãĂĆ

3.86 86 ä¡£çŤĺsliceåŕźèšą

çŤ§æĹŘåĚşäžŐèŻŃçşŢçŽĎåžŔåĹŮcake1ïijŽ

```
In [1]: cake1 = list(range(5,0,-1))

In [2]: b = cake1[1:10:2]

In [3]: b
Out[3]: [4, 2]

In [4]: cake1
Out[4]: [5, 4, 3, 2, 1]
```

åE■çŤ§æĹŘäÿĂäÿłåžŔåĹŮïijŽ

```
In [5]: from random import randint
   ...: cake2 = [randint(1,100) for _ in range(100)]
   ...: # åŘŇæăůäżěéŮt'éŽŤäÿž2åĽĞåL'■10äÿłåĚČçt'äïijŇå¿ŮåĹřåĽĞçL'Ğd
   ...: d = cake2[1:10:2]
In [6]: d
Out[6]: [75, 33, 63, 93, 15]
```

ä¡ăçIJŃïijŇæĹŚäžňä¡£çŤĺåŘŇäÿĂçğ■åĽĞæşŢïijŇåĹĘåĹńåĽĞåijĂäÿđ'äÿłèŻŃçşŢcake1,cake2.
åŘŐæÌěåŘŚçŐřè£Źçğ■åĽĞæşŢæđĄäÿžçżŔåĚÿïijŇåŔĹæŃ£åőČåŐżåĽĞæŽt'åđ'ŽçŽĎåőźåŹĺåŕźèšąãĂĆ

éĆčäźĹïijŇäÿžäżĂäźĹäÿ■æŁŁè£Źçğ■åĽĞæşŢåŕĄèčĚäÿžäÿĂäÿłåŕźèšąåŚćïij§äžŐæŸŕåřśæIJL'äžĘsliceå

åőŽäźL'sliceåŕźèšąæđĄäÿžçőĂå■ŢïijŇåęĆæŁŁäÿŁéÍćçŽĎåĽĞæşŢåőŽäźL'æĹŘsliceåŕźèšąïijŽ

```
perfect_cake_slice_way = slice(1,10,2)
#åŐżåĹĞcake1
cake1_slice = cake1[perfect_cake_slice_way]
```

(continues on next page)

(continued from previous page)

```
cake2_slice = cake2[perfect_cake_slice_way]

In [11]: cake1_slice
Out[11]: [4, 2]

In [12]: cake2_slice
Out[12]: [75, 33, 63, 93, 15]
```

äÿŐäÿŁéİćçŽĎçżŞæđIJäÿĂèĞťãĂĆ

åŕźäžŐéĂĘåŘŚåžŔåĹŮåĹĞçĽĞïijŇsliceåŕźèśąäÿĂæăůåŔŕèąŇïijŽ

```
a = [1,3,5,7,9,0,3,5,7]
a_ = a[5:1:-1]

named_slice = slice(5,1,-1)
a_slice = a[named_slice]

In [14]: a_
Out[14]: [0, 9, 7, 5]

In [15]: a_slice
Out[15]: [0, 9, 7, 5]
```

éćŚçźĄä¡£çŤĺåŘŇäÿĂåĹĞçĽĞçŽĎæŞ■ä¡IJåŔŕä¡£çŤĺsliceåŕźèśąæŁ¡åĞžæİěïijŇåď■çŤĺçŽĎåŘŇæŮűè£

CHAPTER 4

äžÑãĂĄPythonå■Ůçňęäÿšå ŠŇæ■čåĹŹ

å■Ůçňęäÿšæ Ůăæ L’Ăäÿ■åIJĺïijŇå■Ůçňęäÿšç ŽĎåď ĎçŘĘäź§æ ŸŕæIJĂåÿÿèğĄçŽĎæŞ■ä¡IJãĂĆæIJňçńăèŁ

4.1 1 åŔ■è¡ňå■Ůçňęäÿš

```
st="python"
#æŰźæşŢ1
''.join(reversed(st))
#æŰźæşŢ2
st[::-1]
```

4.2 2 å■Ůçňęäÿšå ĹĞçL’ĞæŞ■ä¡IJ

```
å■Ůçňęäÿšå ĹĞçL’ĞæŞ■ä¡IJâĂŤâĂŤæ§ěæL’¿æŻ£æ■ć3æĹŰ5çŽĎåĂ■æŢř
In [1]:[str("java"[i%3*4:]+"python"[i%5*6:] or i) for i in range(1,
↪15)]
OUT[1]:['1',
 '2',
 'java',
 '4',
 'python',
 'java',
 '7',
 '8',
 'java',
 'python',
```

(continues on next page)

(continued from previous page)

```
 '11',
 'java',
 '13',
 '14']
```

4.3 3 joinäÿšèĄŤå■Ůçňęäÿš

```
In [4]: mystr = ['1',
   ...:  '2',
   ...:  'java',
   ...:  '4',
   ...:  'python',
   ...:  'java',
   ...:  '7',
   ...:  '8',
   ...:  'java',
   ...:  'python',
   ...:  '11',
   ...:  'java',
   ...:  '13',
   ...:  '14']

In [5]: ','.join(mystr) #çŤĺéĂŮåŕůè£đæŐěå■Ůçňęäÿš
Out[5]: '1,2,java,4,python,java,7,8,java,python,11,java,13,14'
```

4.4 4 å■Ůçňęäÿšç˛Ďå■ÙèŁĆéŢ£åžę

```
def str_byte_len(mystr):
    return (len(mystr.encode('utf-8')))


str_byte_len('i love python')  # 13(äÿłå■ÙèŁĆ)
str_byte_len('å■Ůçňę')  # 6(äÿłå■ÙèŁĆ)
```

äžěäÿŃæŸŕræ■čåĹŹéČĺåĹĘ

```
import re
```

4.5 5 æ§ěæĽ¿çňňäÿĂäÿłåŇźéĚ■äÿš

```
s = 'i love python very much'
pat = 'python'
r = re.search(pat,s)
print(r.span()) #(7,13)
```

4.6 6 æ§ěæĽ¿æĽĂæIJL'1çŽĎçť́ćåijŢ

```
s = 'åśśäÿIJçIJAą¡■åİŁåÿĆéİŠåůđçňń1äÿ■å■ęéńŸäÿL'1çŔ■'
pat = '1'
r = re.finditer(pat,s)
for i in r:
    print(i)

# <re.Match object; span=(9, 10), match='1'>
# <re.Match object; span=(14, 15), match='1'>
```

4.7 7 \d åŇźéĚ■æŢřå■Ů[0-9]

findallæĽ¿åĞžåĚĺéČĺä¡■ç¡őçŽĎæĽĂæIJL'åŇźéĚ■

```
s = 'äÿĂåĚś20èąŇäżčçăĄè£ŘèąŇæŮűéŮt'13.59s'
pat = r'\d+' #␣
↪+èąĺçd'žåŇźéĚ■æŢřå■Ů(\dèąĺçd'žæŢřå■ŮçŽĎéĂŽçŤĺå■Ůçňę)1æňąæĹŰåd'Žæňą
r = re.findall(pat,s)
print(r)
# ['20', '13', '59']
```

4.8 8 åŇźéĚ■æţőçĆźæŢřåŠŇæŢť́æŢř

?èąĺçď'žåĽ■äÿĂäÿłå■ŮçňęåŇźéĚ■0æĹŰ1æňą

```
s = 'äÿĂåĚś20èąŇäżčçăĄè£ŘèąŇæŮűéŮt'13.59s'
pat = r'\d+\.?\d+' # ?èąĺçd'žåŇźéĚ■åřŕæŢřçĆź(\.
↪)0æňąæĹŰ1æňąïijŇè£Źçĝ■åĘŹæşŢæIJL'äÿłåřŕbugïijŇäÿ■èČ¡åŇźéĚ■åĹřäÿłä¡■æŢřçŽĎæŢt'æŢ
r = re.findall(pat,s)
print(r)
# ['20', '13.59']

# æŻt'åě¡çŽĎåĘŹæşŢïijž
pat = r'\d+\.\d+|\d+' # A|BïijŇåŇźéĚ■Aåd'śèt'ěæL'■åŇźéĚ■B
```

4.9 9 ^åŇźéĚ■å■Ůçňęäÿšç ŽĎåijĂåď'ť

```
s = 'This module provides regular expression matching operations␣
↪similar to those found in Perl'
pat = r'^[emrt]' # æ$ěæL'¿äżěå■Ůçňęe,m,ræĹŰtåijĂåğŃçŽĎå■Ůçňęäÿš
r = re.findall(pat,s)
print(r)
# [],
↪åŻääÿžå■Ůçňęäÿšç ŽĎåijĂåd't'æŸŕå■Ůçňę`T`ïijŇäÿ■åIJĺemrtåŇźéĚ■èŇČåŻt'åĘĚïijŇæL'Ăä
IN [11]: s2 = 'email for me is guozhennianhua@163.com'
re.findall('^[emrt].*',s2)# åŇźéĚ■äżěe,m,r,
↪tåijĂåğŃçŽĎå■ŮçňęäÿšïijŇåŘŐéĺćæŸŕåd'Žäÿłäżżæ ĎŔå■Ůçňę
Out[11]: ['email for me is guozhennianhua@163.com']
```

4.10 10 re.I å£¡çŢěåď ğářŔåĘŹ

```
s = 'That'
pat = r't'
r = re.findall(pat,s,re.I)
In [22]: r
Out[22]: ['T', 't']
```

4.11 11 çŘĘèğčcompileçŽĎä¡IJçŤÍ

åęĆæđIJèęĄåĄŽå¿Ĺåď'ŽæňąåŇźéĚ■ïijŇåŔŕäżěåĚĹçijŰèŕŚåŇźéĚ■äÿšïijŽ

```
import re
pat = re.compile('\W+') # \W åŇźéĚ■äÿ■æŸŕæŢřå■ŮåŠŇå■Ůæŕ■çŽĎå■Ůçňę
has_special_chars = pat.search('ed#2@edc')
if has_special_chars:
    print(f'str contains special characters:{has_special_chars.
↪group(0)}')

###è¿ŞåĞžçżŞædIJ:
 # str contains special characters:#

### åĘ■æňąä¡£çŤĺpatæ■čåĹŹçijŰèŕŚåŕźèśą åĄŽåŇźéĚ■
again_pattern = pat.findall('guozhennianhua@163.com')
if '@' in again_pattern:
    print('possibly it is an email')
```

4.12 12 ä¡£çŤÍ()æ■ŢèŐůå■Ţèŕ■ïijŇäÿ■æČşåÿęçľžæăij

ä¡£çŤÍ () æ■ŢèŐů

```
s = 'This module provides regular expression matching operations␣
↪similar to those found in Perl'
pat = r'\s([a-zA-Z]+)'
r = re.findall(pat,s)
print(r) #['module', 'provides', 'regular', 'expression', 'matching
↪', 'operations', 'similar', 'to', 'those', 'found', 'in', 'Perl']
```

çIJŃåĹřæŔŘåŘŰå■Ţèŕ■äÿ■æIJłåŇĚæŃńçňňäÿĂäÿłå■Ţèŕ■ïijŇä¡£çŤÍ?
èąĺçď žåĽ■éÍćå■Ůçňęå Ğžç Őř0æňąæĹŰ1æňąïijŇä¡ĘæŸŕæ■ď å■Ůçňęè£ŸæIJĽèąĺçď žèť łå£ČæĹŰéÍđèť łå£Čå

```
s = 'This module provides regular expression matching operations␣
↪similar to those found in Perl'
pat = r'\s?([a-zA-Z]+)'
r = re.findall(pat,s)
print(r) #['This', 'module', 'provides', 'regular', 'expression',
↪'matching', 'operations', 'similar', 'to', 'those', 'found', 'in',
↪ 'Perl']
```

4.13 13 splitåĹĘåĽšå■Ţèŕ■

ä¡£çŤÍäžěäÿŁæŰźæşŢåĹĘåĽšå■Ţèŕ■äÿ■æŸŕçőĂæť AçŽĎïijŇäžĚäžĚæŸŕäÿžäžĘæijŢçď žãĂĆåĹĘåĽšå

```
s = 'This module provides regular expression matching operations␣
↪similar to those found in Perl'
pat = r'\s+'
r = re.split(pat,s)
print(r) # ['This', 'module', 'provides', 'regular', 'expression',
↪'matching', 'operations', 'similar', 'to', 'those', 'found', 'in',
↪ 'Perl']

### äÿŁéÍćè£ŹåŔěèŕÍäź$åŔŕçŻt'æŐěä¡£çŤÍstrèĞłåÿęçŽĎsplitåĞ¡æŢřïijŽ
s.split(' ') #ä¡£çŤÍçl'žæăijåĹĘéŽŤ

###␣
↪ä¡ĘæŸŕïijŇåŕźäžŐéčŐæăijçňęæŻt'åŁååd'■æÍĆçŽĎæČĚåĘţïijŇsplitæŮäèČ¡äÿžåŁŻïijŇåŔłè

s = 'This,,,   module ; \t   provides|| regular ; '
words = re.split('[,\s;|]+',s)
↪#è£ŹæăůåĹĘéŽŤåĞžæÍěïijŇæIJĂåŘŐäijŽæIJL'äÿĂäÿłçl'žå■Ůçňęäÿš
words = [i for i in words if len(i)>0]
```

4.14 14 matchäżŐå■Ůçňęäÿšåij̈ÅåğŃä¡■ç¡őåŇźéĚ■

æşĺæĎŕmatch,searchç■L'çŽĎäÿ■åŘŇïijŽ
1) matchåĞ¡æŢř

```
import re
### match
mystr = 'This'
pat = re.compile('hi')
pat.match(mystr) # None
pat.match(mystr,1) # äżŐä¡■ç¡ő1åd'ĎåijÄåğŃåŇźéĚ■
Out[90]: <re.Match object; span=(1, 3), match='hi'>
```

2) searchåĞ¡æŢř
searchæŸŕäżŐå■Ůçňęäÿšç̈ŽĎäżżæĎŕä¡■ç¡őåijÄåğŃåŇźéĚ■

```
In [91]: mystr = 'This'
    ...: pat = re.compile('hi')
    ...: pat.search(mystr)
Out[91]: <re.Match object; span=(1, 3), match='hi'>
```

4.15 15 æŻ£æ■ćåŇźéĚ■çŽĎå■Řäÿš

subåĞ¡æŢřåőđçŐřåŕźåŇźéĚ■å■Řäÿšç̈ŽĎæŻ£æ■ć

```
content="hello 12345, hello 456321"
pat=re.compile(r'\d+') #èęĄæŻ£æ■ćçŽĎéČĺåĹĘ
m=pat.sub("666",content)
print(m) # hello 666, hello 666
```

4.16 16 èť̈łå£Čæ■ŢèŐů

(.*)èąĺçd'žæ■ŢèŐůäżżæĎŕåd'Žäÿłå■Ůçňęïij̈Ňår¡åŕŕèČ¡åd'ŽçŽĎåŇźéĚ■å■Ůçňę

```
content='<h>ddedadsad</h><div>graph</div>bb<div>math</div>cc'
pat=re.compile(r"<div>(.*)</div>")  #èt'łål'łæĺąåijŕ
m=pat.findall(content)
print(m) #åŇźéĚ■çżŞædIJäÿžïijŽ ['graph</div>bb<div>math']
```

4.17 17 éİđèťłå£Čæ■ŢèŐů

äżĚæůżåŁăäÿĂäÿłéŮőåŘů(?)ïijŇå¿ŮåĹřçżŞæđIJåőŇåĚĺäÿ■åŘŇïijŇè£ŹæŸŕéİđèťłå£ČåŇźéĚ■ïijŇéĂŽ

```
content='<h>ddedadsad</h><div>graph</div>bb<div>math</div>cc'
pat=re.compile(r"<div>(.*?)</div>")
m=pat.findall(content)
print(m) # ['graph', 'math']
```

éİđèťłå£Čæ■ŢèŐůïijŇèğĄåě¡åřśæŤűãĂĆ

4.18 18 åÿÿçŤĺåĚČå■Ůçňęæ Äżçżş

```
. åŇźéĚ■äżżæĎŔå■Ůçňę
^ åŇźéĚ■å■ŮçňęäÿšåijĂåğŃä¡■ç¡ő
$ åŇźéĚ■å■Ůçňęäÿšäÿ■çżŞæİŞçŽĎä¡■ç¡ő
* åL'■éİćçŽĎåŐ§å■ŘéĞ■åd'■0æňąãĂĄ1æňąãĂĄåd'Žæňą
? åL'■éİćçŽĎåŐ§å■ŘéĞ■åd'■0æňąæĹŰèĂĚ1æňą
+ åL'■éİćçŽĎåŐ§å■ŘéĞ■åd'■1æňąæĹŰåd'Žæňą
{n} åL'■éİćçŽĎåŐ§å■ŘåĞžçŐřäžE n æňą
{n,} åL'■éİćçŽĎåŐ§å■ŘèĞşářŚåĞžçŐř n æňą
{n,m} åL'■éİćçŽĎåŐ§å■ŘåĞžçŐřæňąæŢřäżŃäžŐ n-m äźŃéŮt'
( ) åĹEçżĎ,éIJĂèęĄè¿ŞåĞžçŽĎéČĺåĹE
```

4.19 19 åÿÿçŤĺéĂŽçŤĺå■Ůçňęæ Äżçżş

```
\s  åŇźéĚ■çl'žçŹ¡å■Ůçňę
\w  åŇźéĚ■äżżæĎŔå■Ůæŕ■/æŢřå■Ů/äÿŃåĹŠçž£
\W  åŠŇářŔåĘŹ w çŻÿåŔ■ïijŇåŇźéĚ■äżżæĎŔå■Ůæŕ■/æŢřå■Ů/
 ↪äÿŃåĹŠçž£äžěåd'ŰçŽĎå■Ůçňę
\d  åŇźéĚ■å■Ąè£ŻåĹűæŢřå■Ů
\D  åŇźéĚ■éŹd'äžEå■Ąè£ŻåĹűæŢřäžěåd'ŰçŽĎåĂij
[0-9]  åŇźéĚ■äÿĂäÿł0-9äźŃéŮt'çŽĎæŢřå■Ů
[a-z]  åŇźéĚ■åřŔåĘŹèŃśæŰĞå■Ůæŕ■
[A-Z]  åŇźéĚ■åd'ğåĘŹèŃśæŰĞå■Ůæŕ■
```

4.20 20 çĹňåŘŰçŹ¿åžęéęŰéąţæăĞécŸ

```
import re
from urllib import request

#çĹňèŹńçĹňåŘŰçŹ¿åžęéęŰéąţåĘĚåőź
```

(continues on next page)

(continued from previous page)

```
data=request.urlopen("http://www.baidu.com/").read().decode()

#åĹĘædŘç¡Śéąţ,çąőåőŽæ■čåĹŽèąÍè¿¿åijŕ
pat=r'<title>(.*?)</title>'

result=re.search(pat,data)
print(result) <re.Match object; span=(1358, 1382), match='<title>
 ↪çŹ¿åžęäÿĂäÿŃïijÑä¡ăåřśç§ěéĄŞ</title>'>

result.group() # çŹ¿åžęäÿĂäÿŃïijÑä¡ăåřśç§ěéĄŞ
```

4.21 21 æĽźéĞŕRè¡ňåŇŰäÿžéľijåşřæăijåijŕ(Camel)

æŢřæ■őåžŞå■Ůæőţ̊åŘ■æĽźéĞŕRè¡ňåŇŰäÿžéľijåşřæăijåijŕ

åĹĘædŘè£ĞçÍŃ

```
# çŤÍåĹřçŽĎæ■čåĹŽäÿšèőšèğč
# \s æŇĞåŇźéĚ■ïijž [ \t\n\r\f\v]
# A|Bïijžèąĺçd'žåŇźéĚ■Aäÿšæĺ̊Bäÿš
# re.sub(pattern, newchar, string):
#␣
 ↪substitueäžčæŽ£ïijŇçŤÍnewcharå■ŮçňęæŽ£äžčäÿŐpatternåŇźéĚ■çŽĎå■ŮçňęæĽ'ĂæIJL'.
 ↪
```

```
# title(): è¡ňåŇŰäÿžåd'ğåĘŹïijÑä¿Ńå■Řïijž
# 'Hello world'.title() # 'Hello World'
```

```
# print(re.sub(r"\s|_|", "", "He llo_worl\td"))
s = re.sub(r"(\s|_|-)+", " ",
           'some_database_field_name').title().replace(" ", "")
#çżŞædIJïijž SomeDatabaseFieldName
```

```
#␣
 ↪åŕŕäżěçIJŃåĹřæ■d'æŰűçŽĎçňňäÿĂäÿłå■Ůçňęäÿžåd'ğåĘŹïijÑéIJĂèęAè¡ňåŇŰäÿžårŕåĘŹ
s = s[0].lower()+s[1:]  # æIJĂçżĹçżŞædIJ
```

æŢť'çŘĘäžěäÿŁåĹĘædŘå¿ŮåĹřåęĆäÿŃäżčçăĄïijž

```
import re
def camel(s):
    s = re.sub(r"(\s|_|-)+", " ", s).title().replace(" ", "")
    return s[0].lower() + s[1:]

# æĽźéĞŕRè¡ňåŇŰ
```

(continues on next page)

(continued from previous page)

```
def batch_camel(slist):
    return [camel(s) for s in slist]
```

æţŃèŕŢçżŞæđIJïijŽ

```
s = batch_camel(['student_id', 'student\tname', 'student-add'])
print(s)
# çżŞæđIJ
['studentId', 'studentName', 'studentAdd']
```

4.22 22 árĘçăĄåőL'åĚÍæčĂæ§ě

árĘçăĄåőL'åĚÍèęĄæśĆïijŽ1)èęĄæśĆárĘçăĄäÿž6åĹř20ä¡■;
2)árĘçăĄåŔłåŇĚåŘńèŃśæŰĞå■Ůæŕ■åŠŇæŢřå■Ů

```
pat = re.compile(r'\w{6,20}') #␣
↪è£ŹæŸŕéŤŹèŕŕçŽĎïijŇåżääÿž\wéĂŽéĚ■çňęåŇźéĚ■çŽĎæŸŕå■Ůæŕ■ïijŇæŢřå■ŮåŠŇäÿŃåĹŠçž£ïij
# ä¡£çŤÍæIJĂçÍşçŽĎæŰźæşŢïijŽ\da-zA-
↪Zæżąèűş`árĘçăĄåŔłåŇĚåŘńèŃśæŰĞå■Ůæŕ■åŠŇæŢřå■Ů`
pat = re.compile(r'[\da-zA-Z]{6,20}')
```

éĂĽçŤÍæIJĂä£ÍéŹľçŽĎfullmatchæŰźæşŢïijŇæ§ěçIJŃæŸŕåŘęæŢť'äÿłå■ŮçňęäÿšéČ¡åŇźéĚ■ïijŽ

```
pat.fullmatch('qaz12') # è£ŤåžđđNone, éŢ£åžęåřŔäžŐ6
pat.fullmatch('qaz12wsxedcrfvtgb67890942234343434') # None␣
↪éŢ£åžęåd'ğäžŐ22
pat.fullmatch('qaz_231') # None åŘńæIJL'äÿŃåĹŠçž£
pat.fullmatch('n0passw0Rd')
Out[4]: <re.Match object; span=(0, 10), match='n0passw0Rd'>
```

CHAPTER 5

äÿĽãĂĄPythonæŰĞäżűåŠŇæŮěæIJ§

PythonæŰĞäżűIOæŞ■ä¡IJæűĽåŔŁæŰĞäżűèŕżåĘŹæŞ■ä¡IJïijŇèŐůåŔŰæŰĞäżűåŘŐçijĂåŘ■ïijŇä£őæ

PythonæŮěæIJ§çńäèŁĆïijŇçŤśèąĺçď'žåď'ğæŮěæIJ§çŽĎcalendar, dateæĺąåİŮïijŇéĂŘæÿŘè£ĞæÿąåĹřèąĺçď'žæŮűéŮť'åĹżåžęæŻť'åřŔçŽĎæĺąåİŮïijŽdatetime, timeæĺąåİŮïijŇæŇĽ'çĚğæ■ď'éĂżè¿ŚåśŢåijĂãĂĆ

äÿĂåĚśæĂżçżŞæIJĂåÿÿçŤĺçŽĎ24äÿłåĚşäžŐæŰĞäżűåŠŇæŮűéŮť'åď'ĎçŘĘæĺąåİŮçŽĎä¿Ńå■ŘãĂĆ

5.1 1 èŐůåŔŰåŘŐçijĂåŘ■

```
import os
file_ext = os.path.splitext('./data/py/test.py')
front,ext = file_ext
In [5]: front
Out[5]: './data/py/test'

In [6]: ext
Out[6]: '.py'
```

5.2 2 æŰĞäżűèŕżæŞ■ä¡IJ

```
import os
# åĹŻåżžæŰĞäżűåď'ź

def mkdir(path):
```

(continues on next page)

(continued from previous page)

```
    isexists = os.path.exists(path)
    if not isexists:
        os.mkdir(path)
# èŕżåŔŰæŰĞäżűä£ąæĄŕ

def openfile(filename):
    f = open(filename)
    fllist = f.read()
    f.close()
    return fllist  # è£ŤåŽđèŕżåŔŰåĘĚåőź
```

5.3 3 æŰĞäżűåĘŹæŞ■ä¡IJ

```
# åĘŹåĚěæŰĞäżűä£ąæĄŕ
# example1
#␣
 ↪wåĘŹåĚëïijŇåęĆædIJæŰĞäżűå■ŸåIJĺïijŇåĹŹæÿĚçl'žåĘĚåőźåŘŐåĘŹåĚëïijŇäÿ■å■ŸåIJĺåĹŹå
f = open(r"./data/test.txt", "w", encoding="utf-8")
print(f.write("æţŃèŕŢæŰĞäżűåĘŹåĚě"))
f.close

# example2
#␣
 ↪aåĘŹåĚëïijŇæŰĞäżűå■ŸåIJĺïijŇåĹŽåIJĺæŰĞäżűåĘĚåőźåŘŐè£¡åŁäåĘŹåĚëïijŇäÿ■å■ŸåIJĺåĹ
f = open(r"./data/test.txt", "a", encoding="utf-8")
print(f.write("æţŃèŕŢæŰĞäżűåĘŹåĚě"))
f.close

# example3
# withåĚşéŤőå■Ůçşżçż$äijŽèĞłåŁĺåĚşéŮ■æŰĞäżűåŠŇåd'ĎçŘĘåijĆåÿÿ
with open(r"./data/test.txt", "w") as f:
    f.write("hello world!")
```

5.4 4 èůŕå¿Ďäÿ■çŽĎæŰĞäżűåŘ■

```
In [11]: import os
    ...: file_ext = os.path.split('./data/py/test.py')
    ...: ipath,ifile = file_ext
    ...:

In [12]: ipath
Out[12]: './data/py'
```

(continues on next page)

(continued from previous page)

```
In [13]: ifile
Out[13]: 'test.py'
```

5.5 5 æĽźéĞŔä£őæŤźæŰĞäżűåŘŐçijĂ

æĽźéĞŔä£őæŤźæŰĞäżűåŘŐçijĂ

æIJňä¿Ńå■Řä¡£çŤÍPythonçŽĎosæÍąåİŮåŠŇ argparseæÍąåİŮïijŇářĘåůëä¡IJçŻőå¡Ţwork_diräÿŃæ

éĂŽè£ĞæIJňä¿Ńå■ŘïijŇåď ğåőűářĘäijŽåď ğæęĆæÿĚæěŽargparseæÍąåİŮçŽĎäÿżèęAçŤÍæşŢãĂĆ

åŕijåĚěæÍąåİŮ

```
import argparse
import os
```

åőŽäźĽèĎŽæIJňåŔĆæŢř

```
def get_parser():
    parser = argparse.ArgumentParser(
        description='åůëä¡IJçŻőå¡Ţäÿ■æŰĞäżűåŘŐçijĂåŘ■ä£őæŤź')
    parser.add_argument('work_dir', metavar='WORK_DIR', type=str,␣
↪nargs=1,
                        help='ä£őæŤźåŘŐçijĂåŘ■çŽĎæŰĞäżűçŻőå¡Ţ')
    parser.add_argument('old_ext', metavar='OLD_EXT',
                        type=str, nargs=1, help='åŐ$æİěçŽĎåŘŐçijĂ')
    parser.add_argument('new_ext', metavar='NEW_EXT',
                        type=str, nargs=1, help='æŰřçŽĎåŘŐçijĂ')
    return parser
```

åŘŐçijĂåŘ■æĽźéĞŔä£őæŤź

```
def batch_rename(work_dir, old_ext, new_ext):
    """
␣
↪äijäéĂŠå¡$åL'■çŻőå¡ŢïijŇåŐ$æİěåŘŐçijĂåŘ■ïijŇæŰřçŽĎåŘŐçijĂåŘ■åŘŐïijŇæL'źéĞŔéĞ■å$
    """
    for filename in os.listdir(work_dir):
        # èŐůåŔŰå¿ŮåĹřæŰĞäżűåŘŐçijĂ
        split_file = os.path.splitext(filename)
        file_ext = split_file[1]
        # åőŽä¡■åŘŐçijĂåŘ■äÿžold_ext çŽĎæŰĞäżű
        if old_ext == file_ext:
            # ä£őæŤźåŘŐæŰĞäżűçŽĎåőŇæŢt'åŘ■çğř
            newfile = split_file[0] + new_ext
            # åőđçŐřéĞ■åŚ¡åŘ■æŞ■ä¡IJ
            os.rename(
                os.path.join(work_dir, filename),
```

(continues on next page)

(continued from previous page)

```
                os.path.join(work_dir, newfile)
            )
    print("åőŇæĹŘéĞ■åŚ¡åŘ■")
    print(os.listdir(work_dir))
```

åőđçŐřMain

```
def main():
    """
    mainåĞ¡æŢř
    """
    # åŚ¡äżd'èąŇåŔĆæŢř
    parser = get_parser()
    args = vars(parser.parse_args())
    # äżŐåŚ¡äżd'èąŇåŔĆæŢřäÿ■ä¿ÌæňąèğčæđŘåĞžåŔĆæŢř
    work_dir = args['work_dir'][0]
    old_ext = args['old_ext'][0]
    if old_ext[0] != '.':
        old_ext = '.' + old_ext
    new_ext = args['new_ext'][0]
    if new_ext[0] != '.':
        new_ext = '.' + new_ext

    batch_rename(work_dir, old_ext, new_ext)
```

5.6 6 xlsæĽźéĞŔè¡ňæ■ćæĹŘxlsx

```
import os


def xls_to_xlsx(work_dir):
    """
 ↪äijäéĂŠå¡ŞåL'■çŻőå¡ŢïijŇåŐ$æÌěåŘŐçijĂåŘ■ïijŇæŰřçŽĎåŘŐçijĂåŘ■åŘŐïijŇæL'źéĞŔéĞ■åŚ
    """
    old_ext, new_ext = '.xls', '.xlsx'
    for filename in os.listdir(work_dir):
        # èŐůåŘŰå¿ŮåĹřæŰĞäżűåŘŐçijĂ
        split_file = os.path.splitext(filename)
        file_ext = split_file[1]
        # åőŽä¡■åŘŐçijĂåŘ■äÿžold_ext çŽĎæŰĞäżű
        if old_ext == file_ext:
            # ä£őæŤźåŘŐæŰĞäżűçŽĎåőŇæŢt'åŘ■çğř
            newfile = split_file[0] + new_ext
            # åőđçŐŕéĞ■åŚ¡åŘ■æŞ■ä¡IJ
            os.rename(
                os.path.join(work_dir, filename),
```

(continues on next page)

(continued from previous page)

```
                os.path.join(work_dir, newfile)
            )
    print("åőŇæĹŘéĞ■åŚ¡åŘ■")
    print(os.listdir(work_dir))


xls_to_xlsx('./data')

# è¿ŞåĞžçżŞædIJïijŽ
# ['cut_words.csv', 'email_list.xlsx', 'email_test.docx', 'email_
↪test.jpg', 'email_test.xlsx', 'geo_data.png', 'geo_data.xlsx',
'iotest.txt', 'pyside2.md', 'PySimpleGUI-4.7.1-py3-none-any.whl',
↪'test.txt', 'test_excel.xlsx', 'ziptest', 'ziptest.zip']
```

5.7 7 åőŽåĹűæŰĞäżűäÿ■åŘŇèąŇ

æŕŤè¿Čäÿd'äÿłæŰĞäżűåIJĺåŞłäžŻèąŇåĘĚåőźäÿ■åŘŇïijŇè£ŤåŻđè£ŹäžŻèąŇçŽĎçijŰåŔůïijŇèąŇåŔůçij

åőŽäźĽçż§èőąæŰĞäżűèąŇæŢřçŽĎåĞ¡æŢř

```
# çż§èőąæŰĞäżűäÿłæŢř
    def statLineCnt(statfile):
        print('æŰĞäżűåŘ■ïijŽ'+statfile)
        cnt = 0
        with open(statfile, encoding='utf-8') as f:
            while f.readline():
                cnt += 1
            return cnt
```

çż§èőąæŰĞäżűäÿ■åŘŇäźŇåd'ĎçŽĎå■ŘåĞ¡æŢřijŽ

```
# moreèąĺçd'žåŘńæIJL'æŻt'åd'ŽèąŇæŢřçŽĎæŰĞäżű
        def diff(more, cnt, less):
            difflist = []
            with open(less, encoding='utf-8') as l:
                with open(more, encoding='utf-8') as m:
                    lines = l.readlines()
                    for i, line in enumerate(lines):
                        if line.strip() != m.readline().strip():
                            difflist.append(i)
            if cnt - i > 1:
                difflist.extend(range(i + 1, cnt))
            return [no+1 for no in difflist]
```

äÿżåĞ¡æŢřijŽ

```
# è£ŤåŻđçŽĎçżŞæđIJèąŇåŕůäżŐ1åijĂåğŃ
# listèąĺçd'žfileAåŠŇfileBäÿ■åŘŇçŽĎèąŇçŽĎçijŰåŕů

def file_diff_line_nos(fileA, fileB):
    try:
        cntA = statLineCnt(fileA)
        cntB = statLineCnt(fileB)
        if cntA > cntB:
            return diff(fileA, cntA, fileB)
        return diff(fileB, cntB, fileA)

    except Exception as e:
        print(e)
```

æŕŤè¿Čäÿd'äÿłæŰĞäžűAåŠŇBïijŇæŃ£çŻÿåŕźè¿Čç§■çŽĎæŰĞäžűåŐżæŕŤè¿ČïijŇè£Ğæżd'èąŇåŘŐçŽĹ
æŽĆæIJłèĂČèŹŚæ§Řäÿ łæŰĞäžűæIJĂåŘŐåŕŕèČ¡æIJL'çŽĎåd'Žèą Ňçl'žèą Ňç■L'çL'źæőŁæČĚåĘţ
ä¡£çŤĺfile_diff_line_nos åĞ¡æŢřïijŽ

```
if __name__ == '__main__':
    import os
    print(os.getcwd())

    '''
    ä¿Ńå■ŘïijŽ
    fileA = "'hello world!!!!!'\
            'nice to meet you'\
            'yes'\
            'no1'\
            'jack'"
    fileB = "'hello world!!!!!'\
            'nice to meet you'\
            'yes' "
    '''
    diff = file_diff_line_nos('./testdir/a.txt', './testdir/b.txt')
    print(diff)  # [4, 5]
```

åĚşäžŐæŰĞäžűæŕŤè¿ČçŽĎïijŇåőđéŹĚäÿŁïijŇåIJĺPythonäÿ■æIJL'åŕźåžŤæĺąåİŮdifflib
, æŔŘä¿ŻæŻt'åd'ŽåĚűäžŰæăijåijŔçŽĎæŰĞäžűæŻt'èŕęçżEçŽĎæŕŤè¿ČïijŇåd'ğåőűåŔŕåŔĆèĂČïijŽ

https://docs.python.org/3/library/difflib.html?highlight=difflib#module-difflib

5.8 8 èŐůåŔŰæŇĞåőŽåŘŐçijĂåŘ■çŽĎæŰĞäžű

```
import os

def find_file(work_dir,extension='jpg'):
    lst = []
```

(continues on next page)

(continued from previous page)

```
    for filename in os.listdir(work_dir):
        print(filename)
        splits = os.path.splitext(filename)
        ext = splits[1] # æŃ£åĹřæL'l'åśŢåŘ■
        if ext == '.'+extension:
            lst.append(filename)
    return lst

r = find_file('.','md')
print(r) # è£ŤåŽđæL'ĂæIJL'çŻőå¡ŢäÿŃçŽĎmdæŰĞäżű
```

5.9 12 åźťçŽĎæŮěåŐĘåŻ¿

```
import calendar
from datetime import date
mydate = date.today()
year_calendar_str = calendar.calendar(2019)
print(f"{mydate.year}åźt'çŽĎæŮěåŐĘåŻ¿ïijŽ{year_calendar_str}\n")
```

æL'Şå■řçżŞæđIJïijŽ

```
2019

      January                   February                  March
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr␣
↪Sa Su
    1  2  3  4  5  6                   1  2  3                  1 ␣
↪2  3
 7  8  9 10 11 12 13       4  5  6  7  8  9 10       4  5  6  7  8 ␣
↪9 10
14 15 16 17 18 19 20      11 12 13 14 15 16 17      11 12 13 14 15␣
↪16 17
21 22 23 24 25 26 27      18 19 20 21 22 23 24      18 19 20 21 22␣
↪23 24
28 29 30 31               25 26 27 28               25 26 27 28 29␣
↪30 31

       April                     May                       June
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr␣
↪Sa Su
 1  2  3  4  5  6  7             1  2  3  4  5                  ␣
↪1  2
 8  9 10 11 12 13 14       6  7  8  9 10 11 12       3  4  5  6  7 ␣
↪8  9
15 16 17 18 19 20 21      13 14 15 16 17 18 19      10 11 12 13 14␣
↪15 16
22 23 24 25 26 27 28      20 21 22 23 24 25 26      17 18 19 20 21␣
↪22 23
```

(continues on next page)

(continued from previous page)

```
29 30                    27 28 29 30 31              24 25 26 27 28␣
↪29 30

        July                    August                  September
Mo Tu We Th Fr Sa Su     Mo Tu We Th Fr Sa Su     Mo Tu We Th Fr␣
↪Sa Su
 1  2  3  4  5  6  7                1  2  3  4                    ␣
↪   1
 8  9 10 11 12 13 14      5  6  7  8  9 10 11      2  3  4  5  6 ␣
↪7  8
15 16 17 18 19 20 21     12 13 14 15 16 17 18      9 10 11 12 13␣
↪14 15
22 23 24 25 26 27 28     19 20 21 22 23 24 25     16 17 18 19 20␣
↪21 22
29 30 31                 26 27 28 29 30 31        23 24 25 26 27␣
↪28 29
                                                  30

      October                  November                 December
Mo Tu We Th Fr Sa Su     Mo Tu We Th Fr Sa Su     Mo Tu We Th Fr␣
↪Sa Su
    1  2  3  4  5  6                1  2  3                    ␣
↪   1
 7  8  9 10 11 12 13      4  5  6  7  8  9 10      2  3  4  5  6 ␣
↪7  8
14 15 16 17 18 19 20     11 12 13 14 15 16 17      9 10 11 12 13␣
↪14 15
21 22 23 24 25 26 27     18 19 20 21 22 23 24     16 17 18 19 20␣
↪21 22
28 29 30 31              25 26 27 28 29 30        23 24 25 26 27␣
↪28 29
                                                  30 31
```

5.10 13 åĹďæŰ■æŸŕåŘęäÿžéŮřáźť

```
import calendar
from datetime import date

mydate = date.today()
is_leap = calendar.isleap(mydate.year)
print_leap_str = "%såźt'æŸŕéŮřáźt'" if is_leap else "
↪%såźt'äÿ■æŸŕéŮřáźt'\n"
print(print_leap_str % mydate.year)
```

æĽŞå■řçżŞæđIJïijŽ

```
2019åźt'äÿ■æŸŕéŮřåźt'
```

5.11 14 æIJĹçŽĎæŮěåŐĘåŻż

```
import calendar
from datetime import date

mydate = date.today()
month_calendar_str = calendar.month(mydate.year, mydate.month)

print(f"{mydate.year}åźt'-{mydate.month}æIJĹçŽĎæŮěåŐĘåŻż¡ïijŽ{month_
↪calendar_str}\n")
```

æL'Şå■řçżŞæđIJïijŽ

```
December 2019
Mo Tu We Th Fr Sa Su
                   1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31
```

5.12 15 æIJĹæIJL'åĞăåď'ľ

```
import calendar
from datetime import date

mydate = date.today()
weekday, days = calendar.monthrange(mydate.year, mydate.month)
print(f'{mydate.year}åźt'-{mydate.month}
↪æIJĹçŽĎçňňäÿĂåd'l'æŸŕéĆčäÿĂåŚĺçŽĎçňň{weekday}åd'l'\n')
print(f'{mydate.year}åźt'-{mydate.month}æIJĹåĚśæIJL'{days}åd'l'\n')
```

æL'Şå■řçżŞæđIJïijŽ

```
2019åźt'-12æIJĹçŽĎçňňäÿĂåd'l'æŸŕéĆčäÿĂåŚĺçŽĎçňň6åd'l'

2019åźt'-12æIJĹåĚśæIJL'31åd'l'
```

5.13 16 ælJĹçňňäÿĂåď'

```
from datetime import date
mydate = date.today()
month_first_day = date(mydate.year, mydate.month, 1)
print(f"å¡ŞæIJĹçňňäÿĂåd'l':{month_first_day}\n")
```

æĽ'Şå■řçżŞæđIJïijŽ

```
å¡ŞæIJĹçňňäÿĂåd'l':2019-12-01
```

5.14 17 ælJĹæIJĂåŘŐäÿĂåď'

```
from datetime import date
import calendar
mydate = date.today()
_, days = calendar.monthrange(mydate.year, mydate.month)
month_last_day = date(mydate.year, mydate.month, days)
print(f"å¡ŞæIJĹæIJĂåŘŐäÿĂåd'l':{month_last_day}\n")
```

æĽ'Şå■řçżŞæđIJïijŽ

```
å¡ŞæIJĹæIJĂåŘŐäÿĂåd'l':2019-12-31
```

5.15 18 èŐůåŔŰå¡ŞåĽ'■æŮűéŮť

```
from datetime import date, datetime
from time import localtime

today_date = date.today()
print(today_date)   # 2019-12-22

today_time = datetime.today()
print(today_time)   # 2019-12-22 18:02:33.398894

local_time = localtime()
print(strftime("%Y-%m-%d %H:%M:%S", local_time))   #␣
↪è¡ňåŇŰäÿžåőŽåĹűçŽĎæäijåijĺ 2019-12-22 18:13:41
```

5.16 19 å■Ůçňęæ ŮűéŮťè¡ňæ ŮűéŮť

```
from time import strptime

# parse str time to struct time
struct_time = strptime('2019-12-22 10:10:08', "%Y-%m-%d %H:%M:%S")
print(struct_time) # struct_timeçśżåđŃåřśæŸŕtimeäÿ■çŽĎäÿĂäÿłçśż

# time.struct_time(tm_year=2019, tm_mon=12, tm_mday=22, tm_hour=10,␣
↪tm_min=10, tm_sec=8, tm_wday=6, tm_yday=356, tm_isdst=-1)
```

5.17 20 æŮűéŮťè¡ňå■ŮçňęæŮűéŮť

```
from time import strftime, strptime, localtime

In [2]: print(localtime()) #è£ŹæŸŕè¿ŞåĚěçŽĎæŮűéŮt'
Out[2]: time.struct_time(tm_year=2019, tm_mon=12, tm_mday=22, tm_
↪hour=18, tm_min=24, tm_sec=56, tm_wday=6, tm_yday=356, tm_isdst=0)

print(strftime("%m-%d-%Y %H:%M:%S", localtime()))  #␣
↪è¡ňåŇŰäÿžåőžåĹűçŽĎæäijåijĺŔ
# è£ŹæŸŕå■Ůçňęäÿšèąĺçd'žçŽĎæŮűéŮt'ïijž   12-22-2019 18:26:21
```

CHAPTER 6

åŻŻãĂĄPythonåď Žçž£çÍŃ

çŤÍåřŔä¿Ńå■ŘèőšPythonåď Žçž£çÍŃïijŇéĂŽä£ŮæŸŞæĞĆãĂĆ

6.1 1 éżŸèőď åŘŕåŁÍäÿżçž£çÍŃ

äÿĂèĹňçŽĎïijŇçÍŃåžŔéżŸèőď æĽğèąŇåŔłåIJÍäÿĂäÿłçž£çÍŃïijŇè£Źäÿłçž£çÍŃçğřäÿžäÿżçž£çÍŃïijŇä¿Ń

åŕijåĚěçž£çÍŃçŻÿåĚşçŽĎæÍąåİŮ `threading`:

```
import threading
```

threadingçŽĎçśżæŰźæşŢ `current_thread()` è£ŤåŻđå¡ŞåĽ■çž£çÍŃïijŽ

```
t = threading.current_thread()
print(t) # <_MainThread(MainThread, started 139908235814720)>
```

æĽĂäżěïijŇéłŇèŕĄäžĘçÍŃåžŔéżŸèőď æŸŕåIJÍ`MainThead`äÿ■æĽğèąŇãĂĆ

`t.getName()` èŐůå¿Ůè£Źäÿłçž£çÍŃçŽĎåŘ■å■Ůïij ŇåĚűäżŰåÿÿçŤÍæŰźæşŢïijŇ`getName()` èŐůå¿Ů

```
print(t.getName()) # MainThread
print(t.ident) # 139908235814720
print(t.isAlive()) # True
```

äżěäÿŁè£ŹäžŻäžĚæŸŕäżŃçż■åď Žçž£çÍŃçŽĎèČŇæŹŕçŞěèŕĘïijŇåŻääÿžåĹřçŽőåĽ■äÿžæ■ćïijŇæĹŚäż

6.2 2 åĹŻåžžçž£çÍŃ

åĹŻåžžäÿĂäÿłçž£çÍŃïijŽ

```
my_thread = threading.Thread()
```

åĹŻåžžäÿĂäÿłåŘ■çğřäÿžmy_threadçŽĎçž£çÍŃïijŽ

```
my_thread = threading.Thread(name='my_thread')
```

åĹŻåžžçž£çÍŃçŽĎçŻőçŽĎæŸŕåŚŁèŕĽåőČåÿőåŁľæĹŚäżňåĄŽäžŻäżĂäźĹïijŇåĄŽäžŻäżĂäźĹéĂŽè£Ğå

```
def print_i(i):
    print('æL'Şå■ři:%d'%(i,))
my_thread = threading.Thread(target=print_i,args=(1,))
```

my_threadçž£çÍŃåůšçżŔåĚĺĽŕæ■ęèčËïijŇä¡ĘæŸŕæĹŚäżňå¿ŮæŇL'äÿŃåŔŚåřĎæŇL'éŠőïijŇåŘŕåŁĺs

```
my_thread().start()
```

æL'Şå■řçżŞæđIJåęĆäÿŃïijŇåĚűäÿ■argsæŇĞåőŽåĞ¡æŢřprint_iéIJĂèęĄçŽĎåŔĆæŢřïïijŇçśżåđŃäÿ

```
æL'Şå■ři:1
```

èĞşæ■d'ïijŇåd'Žçž£çÍŃçŻÿåĚşçŽĎæäÿå£Čç§ěèŕĘçĆźïijŇåůšçżŔæĂżçŻŞåőŇærŢãĂĆä¡ĘæŸŕïijŇäźĚäż

æŐěäÿŃæİěïijŇèĄŁèĄŁåžŤçŤĺåd'Žçž£çÍŃçijŰçÍŃïijŇæIJĂæIJñèt'ÍçŽĎäÿĂäžŻäÿIJèě£ãĂĆ

6.3 3 äžd'æŻ£èŐůå¿ŮCPUæŮűéŮt'çL'Ğ

äÿžäžĘæŻt'åě¡èğčéĞŁïijŇåĄĞåőŽèőąçőŮæIJžæŸŕå■Ţæäÿç ŽĎïijŇår¡çőąåŕźäžŐcpythonïijŇè£ŹäÿłåA

åijĂè¿§3äÿłçž£çÍŃïijŇèčĚåĹřthreadsäÿ■:

```
import time
from datetime import datetime
import threading


def print_time():
    for _ in range(5): # åIJÍæŕŔäÿłçž£çÍŃäÿ■æL'Şå■ř5æňą
        time.sleep(0.1) # æÍąæŃ$æL'Şå■řåL'■çŽĎçŻÿåĚşåd'ĎçŘĘéĂżè¿Ś
        print('å¡ŞåL'■çž£çÍŃ%s,æL'Şå■řçżŞæİ§æŮűéŮt'äÿž:%s'
↪%(threading.current_thread().getName(),datetime.today()))


threads = [threading.Thread(name='t%d'%(i,),target=print_time) for␣
↪i in range(3)]
```

åŘŕåŁĺ3äÿłçž£çĺŃïijŽ

```
[t.start() for t in threads]
```

æĽŞå■řçżŞæđIJåęĆäÿŃïijŇt0,t1,t2äÿĽäÿłçž£çĺŃïijŇæăźæ■őæŞ■ä¡IJçşżçż§çŽĎèřČåžęçőŮæşŢïijŇè

```
å¡ŞåĽ■çž£çĺŃt0,æĽŞå■řçżŞæİ$æŮűéŮt'äÿž:2020-01-12 02:27:15.705235
å¡ŞåĽ■çž£çĺŃt1,æĽŞå■řçżŞæİ$æŮűéŮt'äÿž:2020-01-12 02:27:15.705402
å¡ŞåĽ■çž£çĺŃt2,æĽŞå■řçżŞæİ$æŮűéŮt'äÿž:2020-01-12 02:27:15.705687
å¡ŞåĽ■çž£çĺŃt0,æĽŞå■řçżŞæİ$æŮűéŮt'äÿž:2020-01-12 02:27:15.805767
å¡ŞåĽ■çž£çĺŃt1,æĽŞå■řçżŞæİ$æŮűéŮt'äÿž:2020-01-12 02:27:15.805886
å¡ŞåĽ■çž£çĺŃt2,æĽŞå■řçżŞæİ$æŮűéŮt'äÿž:2020-01-12 02:27:15.806044
å¡ŞåĽ■çž£çĺŃt0,æĽŞå■řçżŞæİ$æŮűéŮt'äÿž:2020-01-12 02:27:15.906200
å¡ŞåĽ■çž£çĺŃt2,æĽŞå■řçżŞæİ$æŮűéŮt'äÿž:2020-01-12 02:27:15.906320
å¡ŞåĽ■çž£çĺŃt1,æĽŞå■řçżŞæİ$æŮűéŮt'äÿž:2020-01-12 02:27:15.906433
å¡ŞåĽ■çž£çĺŃt0,æĽŞå■řçżŞæİ$æŮűéŮt'äÿž:2020-01-12 02:27:16.006581
å¡ŞåĽ■çž£çĺŃt1,æĽŞå■řçżŞæİ$æŮűéŮt'äÿž:2020-01-12 02:27:16.006766
å¡ŞåĽ■çž£çĺŃt2,æĽŞå■řçżŞæİ$æŮűéŮt'äÿž:2020-01-12 02:27:16.007006
å¡ŞåĽ■çž£çĺŃt2,æĽŞå■řçżŞæİ$æŮűéŮt'äÿž:2020-01-12 02:27:16.107564
å¡ŞåĽ■çž£çĺŃt0,æĽŞå■řçżŞæİ$æŮűéŮt'äÿž:2020-01-12 02:27:16.107290
å¡ŞåĽ■çž£çĺŃt1,æĽŞå■řçżŞæİ$æŮűéŮt'äÿž:2020-01-12 02:27:16.107741
```

6.4 4 åď̌Žçž£çĺŃæŁćåď̌žåŘŇäÿĂäÿłåŔŸéĞŔ

åď̌Žçž£çĺŃçijŰçĺŃïijŇå■ŸåIJĺæŁćåď̌žåŘŇäÿĂäÿłåŔŸéĞŔçŽĎéŮőécŸãĂĆ

æŕŤåęĆäÿŃéİćä¿Ńå■ŘïijŇåĹŻåžžçŽĎ10äÿłçž£çĺŃåŘŇæŮűçńđäžĽåĚĺåśĂåŔŸéĞŔa:

```
import threading


a = 0
def add1():
    global a
    a += 1
    print('%s  adds a to 1: %d'%(threading.current_thread().
↪getName(),a))

threads = [threading.Thread(name='t%d'%(i,),target=add1) for i in␣
↪range(10)]
[t.start() for t in threads]
```

æĽğèąŇçżŞæđIJïijŽ

```
t0  adds a to 1: 1
t1  adds a to 1: 2
t2  adds a to 1: 3
t3  adds a to 1: 4
t4  adds a to 1: 5
t5  adds a to 1: 6
t6  adds a to 1: 7
t7  adds a to 1: 8
t8  adds a to 1: 9
t9  adds a to 1: 10
```

çžŞæđIJäÿĂåĹĞæ■čåÿÿïijŇæŕŔäÿłçž£çĺŇæĽğèąŇäÿĂæňąïijŇæŁŁaçŽĎåĂijåŁă1ïijŇæIJĂåŘŐa åŔŸäÿž10ïijŇäÿĂåĹĞæ■čåÿÿãĂĆ

è£ŘèąŇäÿŁéİćäžčçăĄå■ĄåĞăéĄ■ïijŇäÿĂåĹĞäź§éČ¡æ■čåÿÿãĂĆ

æĽĂäžëïijŇæĹŚäžňèČ¡äÿŃçžŞèőžïijŽè£Źæőţäžčçă ĄæŸŕçž£çĺŇåőĽåĚĺçŽĎåŘŮïij§

NOïijĄ

åď Žçž£çĺŇäÿ■ïijŇåŔłèęĄå■ŸåIJĺåŘŇæŮűèŕžåŔŰåŠŇä£őæŤźäÿĂäÿłåĚĺåśĂåŔŸéĞŔçŽĎæČĚåĘţïijŇå

åř¡çőąïijŇæIJL'æŮűïijŇæ§ŘäžŻæČĚåĘţçŽĎèţĎæžŘçńđäžL'ïijŇæŽt'éIJšåĞžéŮőécŸçŽĎæęĆçŐĞæđĄä

æIJňä¿Ńäÿ■ïijŇåęĆæđIJçž£çĺŃ0 åIJĺä£őæŤźaåŘŐïijŇåĚűäżŰæ§ŘäžŻçž£çĺŃè£ŸæŸŕgetåĹřçŽĎæŸŕæš

ä¡ĘæŸŕåIJĺæIJňä¿Ńäÿ■ïijŇa = a + 1è£Źçğ■ä£őæŤźæŞ■ä¡IJïijŇèŁśèt'źçŽĎæŮűéŮt'åď łç§■äžĘïijŇç

6.5 5 äžčçăĄçĺ■ä¡IJæŤźåŁĺïijŇåŔńéŮőécŸæŽt'éIJšåĞžæİě

åŘłèęĄåijĎæŸŐçŹ¡éŮőécŸæŽt'éIJšçŽĎåŐ§åŻäïijŇåŔńéŮőécŸåĞžçŐřè£ŸæŸŕäÿ■åŻřéŽ¿çŽĎãĂĆ

æČşèśąæŢřæ■őåžŞçŽĎåĘŹåĚěæŞ■ä¡IJïijŇäÿĂèĹňéIJĂèęĄèĂŮèt'źæĹŚäžňåŔŕäžěæĎ§ç§ěçŽĎæŮűéŮt'

äÿžäžĘæĺąæŃ§è£ŹäÿłåĘŹåĚěåŁĺä¡IJïijŇçőĂåŇŰæIJ§éŮt'ïijŇæĹŚäžňåŘłéIJĂèęĄåžűéŢ£ä£őæŤźåŔŸéĆ

```
import threading
import time


a = 0
def add1():
    global a
    tmp = a + 1
    time.sleep(0.2) # åžűæŮű0.2çğŠïijŇæĺąæŃ§åĘŹåĚěæL'ĂéIJĂæŮűéŮt'
    a = tmp
    print('%s  adds a to 1: %d'%(threading.current_thread().
↪getName(),a))

threads = [threading.Thread(name='t%d'%(i,),target=add1) for i in␣
↪range(10)]
[t.start() for t in threads]
```

éĞ■æŰřè£ŘèąŇäžčçăĄïijŇåŕłéIJĂäÿĂæňąïijŇéŮőécŸçńŃél'ňåőŇåĚĺæŽt'éIJšïijŇçżŞæđIJåęĆäÿŃïijŽ

```
t0  adds a to 1: 1
t1  adds a to 1: 1
t2  adds a to 1: 1
t3  adds a to 1: 1
t4  adds a to 1: 1
t5  adds a to 1: 1
t7  adds a to 1: 1
t6  adds a to 1: 1
t8  adds a to 1: 1
t9  adds a to 1: 1
```

çIJŃåĹřïijŇ10äÿłçž£çĺŃåĚĺéČĺè£ŘèąŇåŘŐïijŇaçŽĎåĂijåŔłçŻÿå¡ŞäžŐäÿĂäÿłçž£çĺŃæL'ğèąŇçŽĎçżŞ

äÿŃéİćåĹĘæđŘïijŇäÿžäżĂäźĹäijŽåĞžçŐřäÿŁéİćçŽĎçżŞæđIJïijŽ

è£ŹæŸŕäÿĂäÿłå¿ĹæIJL'èŕť'æIJ■åŁŻçŽĎä¿Ńå■ŘïijŇåŻääÿžåIJĺä£őæŤźaåL'■ïijŇæIJL'0.2çğŞçŽĎäijŚçIJ

äžěäÿŁæIJĂæäÿå£ČçŽĎäÿL'èąŇäžčçăĄïijŽ

```
tmp = a + 1
time.sleep(0.2) # åżűæŮű0.2çğŠïijŇæĺąæŃ$åĘźåĚěæL'ĂéIJĂæŮűéÛt'
a = tmp
```

6.6 6 åŁääÿŁäÿĂæŁŁéŤĄïijŇéĄ£åĚ■äžěäÿŁæČĚåĘţåĞžçŐř

ç§ěéĄŞéŮőécŸåĞžçŐřçŽĎåŐ§åŻäåŘŐïijŇèęĄæČşä£őåd'■éŮőécŸïijŇäź§æšąéĆčäźĹåd'■æİĆãĂĆ

éĂŽè£Ğpythonäÿ■æŘŘä¿ŻçŽĎéŤĄæIJžåĹűïijŇæ§Řæőţäžčçă ĄåŔłèČ¡å■Ţçž£çĺŃæL'ğèąŇæŮűïijŇäÿŁé

åĹŻåżžäÿĂæŁŁéŤĄlocka:

```
import threading
import time


locka = threading.Lock()
```

éĂŽè£Ğ locka.acquire() èŐůå¿ŮéŤĄïijŇéĂŽè£Ğlocka.release()éĞŁæŤ¿éŤĄïijŇåőČäżñäźŃéŮť'çŽĎè£ŹäžŻäżčçăĄïijŇåŔłèČ¡å■Ţçž£çĺŃæL'ğèąŇãĂĆ

```
a = 0
def add1():
    global a
    try:
        locka.acquire() # èŐůå¿ŮéŤĄ
        tmp = a + 1
        time.sleep(0.2) # åżűæŮű0.
↪2çğŠïijŇæĺąæŃ$åĘźåĚěæL'ĂéIJĂæŮűéÛt'
```

(continues on next page)

(continued from previous page)

```
        a = tmp
    finally:
        locka.release() # éĞŁæŤżéŤĄ
    print('%s  adds a to 1: %d'%(threading.current_thread().
↪getName(),a))

threads = [threading.Thread(name='t%d'%(i,),target=add1) for i in␣
↪range(10)]
[t.start() for t in threads]
```

æĽğèąŇçżŞæđIJåęĆäÿŃïijŽ

```
t0  adds a to 1: 1
t1  adds a to 1: 2
t2  adds a to 1: 3
t3  adds a to 1: 4
t4  adds a to 1: 5
t5  adds a to 1: 6
t6  adds a to 1: 7
t7  adds a to 1: 8
t8  adds a to 1: 9
t9  adds a to 1: 10
```

äÿĂèţůæ■čåÿÿïijŇåĚűåőđè£ŹåůšçżŕæŸŕå■Ţçž£çÍŃéąžåžŕæĽğèąŇäžĘïijŇåřśæIJňä¿Ńå■ŘèĂŇèÍĂïijŇ

çÍŃåžŕäÿ■åŕłæIJĽäÿĂæŁŁéŤĄïijŇéĂŽè£Ğ `try...finally`è£ŸèČ¡çąőä£İäÿ■åŔŞçŤ§æ■żéŤĄãĆ

æşÍæĎŕä¡£çŤĺåIJžåŘĹïijŇéĄ£åĚ■æ■żéŤĄïijŇæŸŕæĹŚäžňåIJĺä¡£çŤĺåď Žçž£çÍŃåijĂåŔŚæŮűéIJĂèęĄæ

CHAPTER 7

äžŤãĂĄPythonäÿĽåďğåĹľåŹÍ

Pythonäÿ■çŽĎäÿL’åď ğåĹ’åŹĺåŇĚæŃňïijŽè£■äżčåŹÍïijŇçŤ$æĹŘåŹÍïijŇèčĚéěřåŹÍïijŇåĹ’çŤÍåè
itertoolsæŔŘä¿Żè£■äžčåŹÍçŻÿåĚşçŽĎæŞ■ä¡IJãĂĆæ■ď’éČĺåĹĘæŤűå¡ŢæIJL’æĎŔæĂİçŽĎä¿Ńå■ŘåĚśè

7.1 1 áŕżæĽ’¿çňňnæňąåĞžçŐřä¡■ç¡ő

```
def search_n(s, c, n):
    size = 0
    for i, x in enumerate(s):
        if x == c:
            size += 1
        if size == n:
            return i
    return -1



print(search_n("fdasadfadf", "a", 3))# çżŞæđIJäÿž7ïijŇæ■čçąő
print(search_n("fdasadfadf", "a", 30))# çżŞæđIJäÿž-1ïijŇæ■čçąő
```

7.2 2 æŰŘæşćéĆčåěŚæŢřåĹŮåĽ■néąź

```
def fibonacci(n):
    a, b = 1, 1
    for _ in range(n):
        yield a
```

(continues on next page)

(continued from previous page)

```
        a, b = b, a + b


list(fibonacci(5))  # [1, 1, 2, 3, 5]
```

7.3 3 æĽżåĞžæĽĂælJĽéĞ■åď■åĚČçťă

```
from collections import Counter


def find_all_duplicates(lst):
    c = Counter(lst)
    return list(filter(lambda k: c[k] > 1, c))


find_all_duplicates([1, 2, 2, 3, 3, 3])  # [2,3]
```

7.4 4 èĄŤåŘĹçż§èőąæňąæŢř

Counterårźèšąéůt'åŕŕäžěåĄŽæŢřå■ęè£ŘçőŮ

```
from collections import Counter
a = ['apple', 'orange', 'computer', 'orange']
b = ['computer', 'orange']

ca = Counter(a)
cb = Counter(b)
#Counterårźèšąéůt'åŕŕäżěåĄŽæŢřå■ęè£ŘçőŮ
ca + cb  # Counter({'orange': 3, 'computer': 2, 'apple': 1})


# è£ŻäÿĂæ■ěæŁ¡èšąïijŇåőđçŐřåd'ŽäÿłåĹŮèąĺåĘĚåĚČçt'ăçŽĎäÿłæŢřçż§èőą


def sumc(*c):
    if (len(c) < 1):
        return
    mapc = map(Counter, c)
    s = Counter([])
    for ic in mapc: # ic æŸŕäÿĂäÿłCounterårźèšą
        s += ic
    return s
```

(continues on next page)

(continued from previous page)

```
#Counter({'orange': 3, 'computer': 3, 'apple': 1, 'abc': 1, 'face': 1})
sumc(a, b, ['abc'], ['face', 'computer'])
```

7.5 5 groupbyå■Ţå■ŮæőţǻLĘçżĎ

åď'l'ærřŤèőřå¡Ţïij Ž

```
a = [{'date': '2019-12-15', 'weather': 'cloud'},
 {'date': '2019-12-13', 'weather': 'sunny'},
 {'date': '2019-12-14', 'weather': 'cloud'}]
```

æŇL'çĚğåď'l'ærřŤå■Ůæőţweather ǻLĘçżĎæśĞæĂžïij Ž

```
from itertools import groupby
for k, items in  groupby(a,key=lambda x:x['weather']):
     print(k)
```

è¿ŞåĞžçżŞæđIJçIJŃåĞžïijŇåĹĘçżĎåď śèt'ěïijĄåŐ§åŻäïijŽåĹĘçżĎåL'■å£Ěéążæ ŇL'çĚğåĹĘçżĎå■Ůæőţ

```
cloud
sunny
cloud
```

ä£őæŤźäžčçăĄïij Ž

```
a.sort(key=lambda x: x['weather'])
for k, items in  groupby(a,key=lambda x:x['weather']):
     print(k)
     for i in items:
         print(i)
```

è¿ŞåĞžçżŞæđIJïij Ž

```
cloud
{'date': '2019-12-15', 'weather': 'cloud'}
{'date': '2019-12-14', 'weather': 'cloud'}
sunny
{'date': '2019-12-13', 'weather': 'sunny'}
```

7.6 6 itemgetteråŠŇkeyåĞ¡æŢř

æşĺæĎŔåĹřsortåŠŇgroupbyæL'ĂçŤĺçŽĎkeyåĞ¡æŢřïijŇéŹd'äžĘlambdaåĘŹæşŢåď Űïij Ńè£ŸæIJI

```
a = [{'date': '2019-12-15', 'weather': 'cloud'},
 {'date': '2019-12-13', 'weather': 'sunny'},
 {'date': '2019-12-14', 'weather': 'cloud'}]
from operator import itemgetter
from itertools import groupby

a.sort(key=itemgetter('weather'))
for k, items in groupby(a, key=itemgetter('weather')):
     print(k)
     for i in items:
         print(i)
```

çżŞæđIJïijŽ

```
cloud
{'date': '2019-12-15', 'weather': 'cloud'}
{'date': '2019-12-14', 'weather': 'cloud'}
sunny
{'date': '2019-12-13', 'weather': 'sunny'}
```

7.7 7 groupbyåď̌Žå■Ůæőţ̌åĹĘçżĎ

itemgetteræŸŕäÿĂäÿłçśżïijŇitemgetter('weather')è£ŤåŻđäÿĂäÿłåŔŕèřČçŤĺçŽĎåŕźèśąïijŇ

```
from operator import itemgetter
from itertools import groupby

a.sort(key=itemgetter('weather', 'date'))
for k, items in groupby(a, key=itemgetter('weather')):
     print(k)
     for i in items:
         print(i)
```

çżŞæđIJåęĆäÿŃïijŇä¡£çŤĺweatheråŠŇdateäÿď̌äÿłå■ŮæőţæŐŠåžŔaïijŇ

```
cloud
{'date': '2019-12-14', 'weather': 'cloud'}
{'date': '2019-12-15', 'weather': 'cloud'}
sunny
{'date': '2019-12-13', 'weather': 'sunny'}
```

æşĺæĎŔè£ŹäÿłçżŞæđIJäÿŐäÿŁéİćçżŞæđIJæIJL̀äžŻå¿őåęŹäÿ■åŘŇïijŇè£ŹäÿłæŻt̀åď̌ŽæŸŕæĹŚäžňæČş

7.8 8 sumåĞ¡æŢřèőąçőŮåŠŇèĄŽåŘĹåŘŇæŮűåĄŽ

Pythonäÿ■çŽĎèĄŽåŘĹçśżåĞ¡æŢřsum,min,maxçňňäÿĂäÿłåŔĆæŢřæŸŕiterableçśżåđŇïijŇäÿĂèĹň

```
a = [4,2,5,1]
sum([i+1 for i in a]) # 16
```

ä¡£çŤĺåĹŮèąĺçŤ§æĹŘåijŕ[i+1 for i in a]åĹŻåžžäÿĂäÿłéŢ£åžęäÿŐaäÿĂèąŇçŽĎäÿt'æŮűåĹŮè

èŕŢæČşåęĆæđIJä¡ăçŽĎæŢřçżĎaéŢ£åžęå■ĄçŹ¿äÿĞçžğïijŇåĘ■åĹŻåžžäÿĂäÿłè£ŹæăůçŽĎäÿt'æŮűåĹŮè

```
a = [4,2,5,1]
sum(i+1 for i in a) # 16
```

æ■d'æŮűi+1 for i in aæŸŕ(i+1 for i in a)çŽĎçőĂåĘŹïijŇå¿ŮåĹřäÿĂäÿłçŤ§æĹŘåŹĺ(g

```
In [8]:(i+1 for i in a)
OUT [8]:<generator object <genexpr> at 0x000002AC7FFA8CF0>
```

çŤ§æĹŘåŹĺæŕŔè£■äžčäÿĂæ■ěåŘŘåĞž(yield)äÿĂäÿłåĚČçt'ăăźűèőąçőŮåŠŇèĄŽåŘĹåŘŐïijŇè£ŻåĚě

7.9 9 liståĹĘçżĎ(çŤ§æĹŘåŹĺçL'Ĺ)

```
from math import ceil

def divide_iter(lst, n):
    if n <= 0:
        yield lst
        return
    i, div = 0, ceil(len(lst) / n)
    while i < n:
        yield lst[i * div: (i + 1) * div]
        i += 1

list(divide_iter([1, 2, 3, 4, 5], 0))  # [[1, 2, 3, 4, 5]]
list(divide_iter([1, 2, 3, 4, 5], 2))  # [[1, 2, 3], [4, 5]]
```

7.10 10 åĹŮèąĺåĚĺåśŢåijĂïijĹçŤ§æĹŘåŹĺçL'ĹïijL'

```
#åd'ŽåśĆåĹŮèąĺåśŢåijĂæĹŘå■ŢåśĆåĹŮèąĺ
a=[1,2,[3,4,[5,6],7],8,["python",6],9]
def function(lst):
    for i in lst:
        if type(i)==list:
            yield from function(i)
        else:
            yield i
print(list(function(a))) # [1, 2, 3, 4, 5, 6, 7, 8, 'python', 6, 9]
```

7.11 11 æţŃèŕŢåĞ¡æŢřè£ŘèąŇæŮűéŮťçŽĎèčĚéěřåŹÍ

```
#æţŃèŕŢåĞ¡æŢřæL'ğèąŇæŮűéŮt'çŽĎèčĚéěřåŹÍçd'žä¿Ń
import time
def timing_func(fn):
    def wrapper():
        start=time.time()
        fn()    #æL'ğèąŇäijääĚěçŽĎfnåŔĆæŢř
        stop=time.time()
        return (stop-start)
    return wrapper
@timing_func
def test_list_append():
    lst=[]
    for i in range(0,100000):
        lst.append(i)
@timing_func
def test_list_compre():
    [i for i in range(0,100000)]  #åĹŮèąÍçŤ$æĹŘåijŔ
a=test_list_append()
c=test_list_compre()
print("test list append time:",a)
print("test list comprehension time:",c)
print("append/compre:",round(a/c,3))

test list append time: 0.0219423770904541
test list comprehension time: 0.007980823516845703
append/compre: 2.749
```

7.12 12 çż§èőąåijĆåÿÿåĞžçŐřæňąæŢřåŠŇæŮűéŮťçŽĎèčĚéěřåŹ

åĘŹäÿĂäÿłèčĚéěřåŹÍïijŇçż§èőąæ§ŘäÿłåijĆåÿÿéĞ■åď■åĞžçŐřæŇĞåőŽæňąæŢřæŮűïijŇçżŔåŐĘçŽĎæ

```
import time
import math


def excepter(f):
    i = 0
    t1 = time.time()
    def wrapper():
        try:
            f()
        except Exception as e:
            nonlocal i
            i += 1
            print(f'{e.args[0]}: {i}')
```

(continues on next page)

(continued from previous page)

```
            t2 = time.time()
            if i == n:
                print(f'spending time:{round(t2-t1,2)}')
    return wrapper
```

åĚşéŤőèŕ■nonlocalåÿÿçŤĺäžŐåĞ¡æŢřåţŇåěŮäÿ■ïijŇåčřæŸŐåŔŸéĞŔiäÿžéİđåśĂéČĺåŔŸéĞŔïijŻ

åęĆæđIJäÿ■åčřæŸŐïijŇi+=1èąĺæŸŐiäÿžåĞ¡æŢřwrapperåĘĚçŽĎåśĂéČĺåŔŸéĞŔïijŇåŻääÿžåIJĺi+variableçŽĎéŤŹèŕŕãĂĆ

ä¡£çŤĺåĹŻåžžçŽĎèčĚéěřåĞ¡æŢřexcepter, næŸŕåijĆåÿÿåĞžçŐřçŽĎæňąæŢřãĂĆ

åĚśæţŃèŕŢäžĘäÿd'çśżåÿÿèğĄçŽĎåijĆåÿÿïijŽèćńéŻűéŹd'åŠŇæŢřçżĎèűŁçŢŇãĂĆ

```
n = 10 # except count

@excepter
def divide_zero_except():
    time.sleep(0.1)
    j = 1/(40-20*2)

# test zero divived except
for _ in range(n):
    divide_zero_except()


@excepter
def outof_range_except():
    a = [1,3,5]
    time.sleep(0.1)
    print(a[3])
# test out of range except
for _ in range(n):
    outof_range_except()
```

æĽŞå■řåĞžæİěçŽĎçżŞæđIJåęĆäÿŃïijŽ

```
division by zero: 1
division by zero: 2
division by zero: 3
division by zero: 4
division by zero: 5
division by zero: 6
division by zero: 7
division by zero: 8
division by zero: 9
division by zero: 10
spending time:1.01
list index out of range: 1
list index out of range: 2
```

(continues on next page)

(continued from previous page)

```
list index out of range: 3
list index out of range: 4
list index out of range: 5
list index out of range: 6
list index out of range: 7
list index out of range: 8
list index out of range: 9
list index out of range: 10
spending time:1.01
```

7.13 13 æţŃèŕŢè£ŘèąŇæŮűéŢ£çŽĎèčĚéěřåŹÍ

```
#æţŃèŕŢåĞ¡æŢřæL'ğèąŇæŮűéŮt'çŽĎèčĚéěřåŹÍçd'žä¿Ń
import time
def timing(fn):
    def wrapper():
        start=time.time()
        fn()    #æL'ğèąÑäijääĚěçŽĎfnåŔĆæŢř
        stop=time.time()
        return (stop-start)
    return wrapper

@timing
def test_list_append():
    lst=[]
    for i in range(0,100000):
        lst.append(i)

@timing
def test_list_compre():
    [i for i in range(0,100000)]  #åĹŮèąÍçŤ$æĹŘåijŔ

a=test_list_append()
c=test_list_compre()
print("test list append time:",a)
print("test list comprehension time:",c)
print("append/compre:",round(a/c,3))

# test list append time: 0.0219
# test list comprehension time: 0.00798
# append/compre: 2.749
```

7.14 14 èčĚéěřåŹÍéĂŽä£ŮçŘĘèğč

åĘ■çIJŃäÿĂäÿłèčĚéěřåŹÍïijŽ

```
def call_print(f):
  def g():
    print('you\'re calling %s function'%(f.__name__,))
  return g
```

ä¡£çŤĺcall_printèčĚéěřåŹĺïijŽ

```
@call_print
def myfun():
  pass

@call_print
def myfun2():
  pass
```

myfun()åŘŐè£ŤåŻđïijŽ

```
In [27]: myfun()
you're calling myfun function

In [28]: myfun2()
you're calling myfun2 function
```

ä¡£çŤĺcall_print

ä¡ăçIJŃïijŇ@call_printæŤ¿ç¡őåIJĺäżżä¡ŢäÿĂäÿłæŰřåőŽäźL'çŽĎåĞ¡æŢřäÿŁéİćïijŇéČ¡äijŽézŸèőd'

è£ŹæŸŕäÿžäżĂäźĹåŚćïij§æşĺæĎŔèğĆåŕ§æŰřåőŽäźL'çŽĎcall_printåĞ¡æŢř(åŁăäÿŁ@åŘŐä¿£æŸŕ

```
def call_print(f):
  def g():
    print('you\'re calling %s function'%(f.__name__,))
  return g
```

åőČå£ĚéążæŐěåŔŮäÿĂäÿłåĞ¡æŢřfïijŇçĎűåŘŐè£ŤåŻđåŔęåd'ŰäÿĂäÿłåĞ¡æŢřg.

èčĚéěřåŹĺæIJňèt'Í

æIJňèt'ĺäÿŁïijŇåőČäÿŐäÿŃéİćçŽĎèřČçŤĺæŰźåijŔæŢĹædIJæŸŕç■L'æŢĹçŽĎïijŽ

```
def myfun():
  pass

def myfun2():
  pass

def call_print(f):
  def g():
    print('you\'re calling %s function'%(f.__name__,))
  return g
```

äÿŃéİćæŸŕæIJĂéĞ■èęĄçŽĎäżčçăĄïijŽ

```
myfun = call_print(myfun)
myfun2 = call_print(myfun2)
```

åď ğåőűçIJŃæŸŐçŹ¡åŘŮïij§äź§åřśæŸŕcall_print(myfun)åŘŐäÿ■æŸŕè£ŤåŻđäÿĂäÿłåĞ¡æŢřåŘŮïijŇçĎú

åĘ■æňąèřČçŤÍmyfun, myfun2æŮűïijŇæŢĹæđIJæŸŕè£ŹæăůçŽĎïijŽ

```
In [32]: myfun()
you're calling myfun function

In [33]: myfun2()
you're calling myfun2 function
```

ä¡ăçIJŃïijŇè£ŹäÿŐèčĚéěřåŹÍçŽĎåőđçŐřæŢĹæđIJæŸŕäÿĂæĺąäÿĂæăůçŽĎãĂĆèčĚéěřåŹÍçŽĎåĘŹæşŢål
= call_print(myfun)ïijŇmyfun2 = call_print(myfun2)ïijŇä¡ĘæŸŕèčĚéěřåŹÍçŽĎè£ŹçğåřA

7.15 15 åőŽåĹűéĂŠåĞŔè£■äžčåŹÍ

```
↪#çijŰåĘŹäÿĂäÿłè£■äżčåŹÍïijŇéĂŽè£Ğå¿łçŐŕèŕ■åŔĕïijŇåőđçŐřårźæ§Řäÿłæ■čæŢt'æŢřçŽĎä¿
↪
class Descend(Iterator):
    def __init__(self,N):
        self.N=N
        self.a=0
    def __iter__(self):
        return self
    def __next__(self):
        while self.a<self.N:
            self.N-=1
            return self.N
        raise StopIteration

descend_iter=Descend(10)
print(list(descend_iter))
[9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
```

æăÿå£ČèęAçĆźïijŽ

1 __nex__åŘ■å■Ůäÿ■èČ¡åŔŸïijŇåőđçŐřåőŽåĹűçŽĎè£■äžčéĂżè¿Ś

2 raise StopIterationïijŽéĂŽè£Ğ raise äÿ■æŰ■çÍŃåžŔïijŇå£Ěéążè£ŹæăůåĘŹ

CHAPTER 8

åĚ■ãĂĄPythonçżŸåŻ¿

Pythonåÿÿç˜ĺçŽĎçżŸåŻ¿åůěåĚůåŇĚæŃňïijŽmatplotlib, seaborn, plotlyç■L'ïijŇäžěåŔŁäÿĂäžŽåĚűäżŰäÿŞçŤĺäžŐçżŸåĹűæ§ŘçśžåŻ¿åęĆèŕ■äžŠåŻ¿ç■L'çŽĎåŇĚïijŇæŔŘçż

8.1 1 turtleçżŸåĹűåěěè£ŘäžŤçŐŕåŻ¿

turtleçżŸåŻ¿çŽĎåĞ¡æŢřéİđåÿÿåě¡çŤĺïijŇå§žæIJňçIJŃåĹřåĞ¡æŢřåŘ■å■ŮïijŇåřśèČ¡ç§ěéĄŞåőČçŽĎåŘ

1 åŕijåĚěåžŞ

```
import turtle as p
```

2 åőŽäźL'çŤżåIJĘåĞ¡æŢř

```
def drawCircle(x,y,c='red'):
    p.pu()# æŁňèţůçŤżçňŤ
    p.goto(x,y) # çżŸåĹűåIJĘçŽĎèţůåğŃä¡■ç¡ő
    p.pd()# æŤ¿äÿŃçŤżçňŤ
    p.color(c)# çżŸåĹűcèL'šåIJĘçŐŕ
    p.circle(30,360) #çżŸåĹűåIJĘïijžå■Łå¿ĎïijŇèğŠåžę
```

3 çŤżçňŤå§žæIJňèő¿ç¡ő

```
p = turtle
p.pensize(3) # çŤżçňŤåřžåŕÿèő¿ç¡ő3
```

4 çżŸåĹűäžŤçŐŕåŻ¿

èřČçŤĺçŤżåIJĘåĞ¡æŢř

```
drawCircle(0,0,'blue')
drawCircle(60,0,'black')
drawCircle(120,0,'red')
drawCircle(90,-30,'green')
drawCircle(30,-30,'yellow')

p.done()
```

çżŞæđIJïijŽ

8.2 2 turtleçżŸåĹűæijńåď'éŻłèŁś

åŕijåĚěæĺąåİŮ

åŕijåĚě `turtle`åžŞåŠŇ pythonçŽĎ `random`

```
import turtle as p
import random
```

çżŸåĹűéŻłèŁś

```
def snow(snow_count):
    p.hideturtle()
    p.speed(500)
    p.pensize(2)
    for i in range(snow_count):
        r = random.random()
        g = random.random()
        b = random.random()
        p.pencolor(r, g, b)
        p.pu()
        p.goto(random.randint(-350, 350), random.randint(1, 270))
        p.pd()
        dens = random.randint(8, 12)
        snowsize = random.randint(10, 14)
        for _ in range(dens):
            p.forward(snowsize)   #␣
↪åŘŚå¡ŞåL'■çĬżçňĬæŰźåŘŚçğżåŁĺsnowsizeåČŔçt'äéŢ£åžę
            p.backward(snowsize)   #␣
↪åŘŚå¡ŞåL'■çĬżçňĬçŻÿåŔ■æŰźåŘŚçğżåŁĺsnowsizeåČŔçt'äéŢ£åžę
            p.right(360 / dens)   # éąžæŮűéŠĹçğżåŁĺ360 / densåžę
```

çżŸåĹűåIJřéİć

```
def ground(ground_line_count):
    p.hideturtle()
    p.speed(500)
    for i in range(ground_line_count):
        p.pensize(random.randint(5, 10))
        x = random.randint(-400, 350)
        y = random.randint(-280, -1)
        r = -y / 280
        g = -y / 280
        b = -y / 280
        p.pencolor(r, g, b)
        p.penup()  # æŁňètůçŤżçňŤ
        p.goto(x, y)  # èől'çŤżçňŤçğżåŁĺåĹřæ■d'ä¡■ç¡ő
        p.pendown()  # æŤ¿äÿŃçŤżçňŤ
        p.forward(random.randint(40, 100))  #␣
↪çIJijå¡ŞåL'■çŤżçňŤæŰźåŘŚåŘŚåL'■çğżåŁĺ40~100èůİçęż
```

äÿżåĞ¡æŢř

```
def main():
    p.setup(800, 600, 0, 0)
    # p.tracer(False)
    p.bgcolor("black")
    snow(30)
    ground(30)
    # p.tracer(True)
    p.mainloop()

main()
```

åŁĺæĂĄåŻ¿çżŞæđIJåśŢçd'žïijŽ

8.3 3 wordcloudèŕ■äžŚåŻ¿

```
import hashlib
import pandas as pd
from wordcloud import WordCloud
geo_data=pd.read_excel(r"../data/geo_data.xlsx")
print(geo_data)
# 0     æůśåIJş
# 1     æůśåIJş
# 2     æůśåIJş
# 3     æůśåIJş
# 4     æůśåIJş
# 5     æůśåIJş
```

(continues on next page)

(continued from previous page)

```
# 6      æůśåIJş
# 7      åź£åůđ
# 8      åź£åůđ
# 9      åź£åůđ

words = ','.join(x for x in geo_data['city'] if x != [])
↪#ç■ŻéĂL'åĞžéÍđçl'žåĹŮèąĺåĂij
wc = WordCloud(
    background_color="green", #èČŇæŹŕéćIJèL'š"green"çż£èL'š
    max_words=100, #æŸ¿çd'žæIJĂåd'ğèŕ■æŢř
    font_path='./fonts/simhei.ttf', #æŸ¿çd'žäÿ■æŰĞ
    min_font_size=5,
    max_font_size=100,
    width=500  #åŻ¿åźĚåő¡åžę
    )
x = wc.generate(words)
x.to_file('../data/geo_data.png')
```

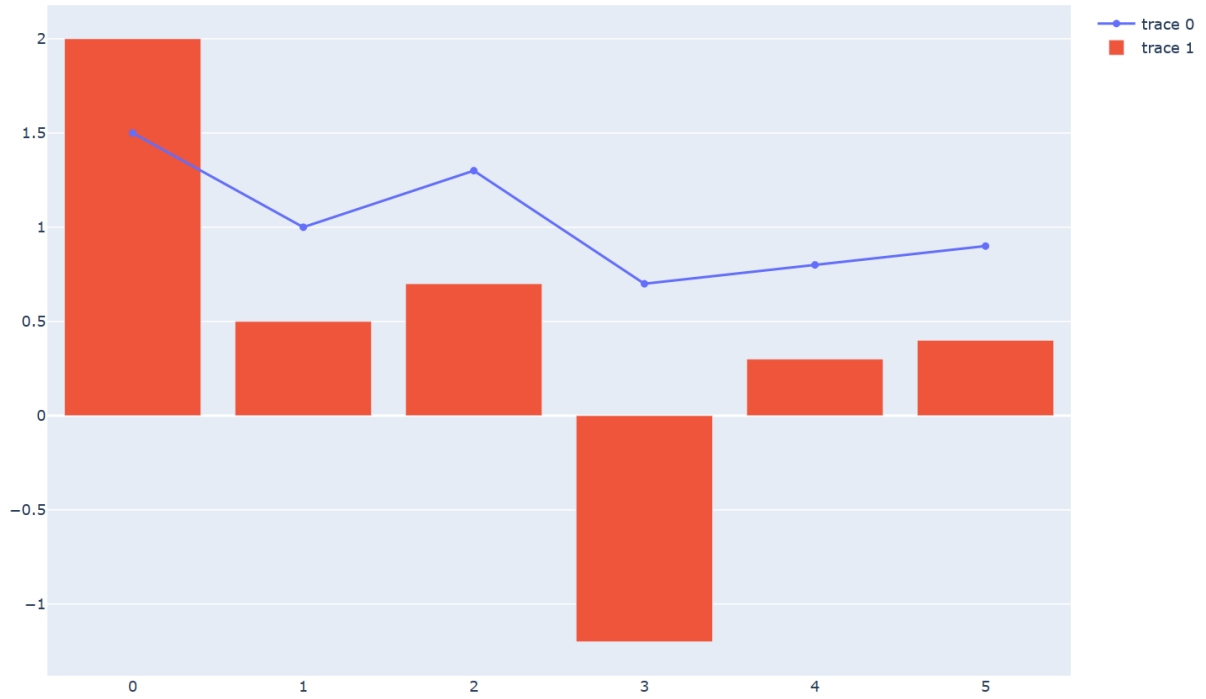
8.4 4 plotlyçŤżæ§śçŁűåŻ¿åŠŇæŁŸçž£åŻ¿

```
#æŞśçŁűåŻ¿+æŁŸçž£åŻ¿
import plotly.graph_objects as go
fig = go.Figure()
fig.add_trace(
    go.Scatter(
        x=[0, 1, 2, 3, 4, 5],
        y=[1.5, 1, 1.3, 0.7, 0.8, 0.9]
    ))
fig.add_trace(
    go.Bar(
        x=[0, 1, 2, 3, 4, 5],
        y=[2, 0.5, 0.7, -1.2, 0.3, 0.4]
```

(continues on next page)

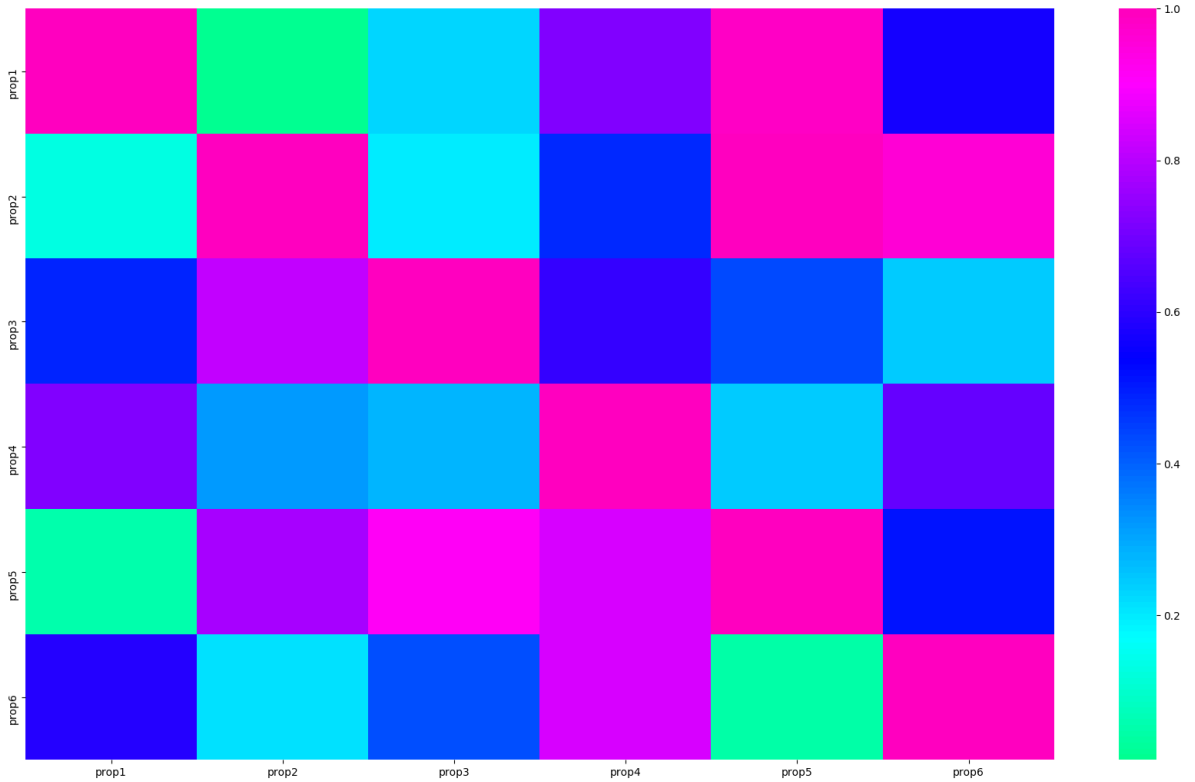
(continued from previous page)

```
    ))
fig.show()
```



8.5 5 seabornçČ■åŁŻåŻ¿

```
# åŕijåĚěåžŞ
import seaborn as sns
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# çŤŞæĹŘæŢřæ■őéŻĘ
data = np.random.random((6,6))
np.fill_diagonal(data,np.ones(6))
features = ["prop1","prop2","prop3","prop4","prop5", "prop6"]
data = pd.DataFrame(data, index = features, columns=features)
print(data)
# çżŸåĹűçČ■åŁŻåŻ¿
heatmap_plot = sns.heatmap(data, center=0, cmap='gist_rainbow')
plt.show()
```



8.6 6 matplotlibæŁŸçž£åŻ¿

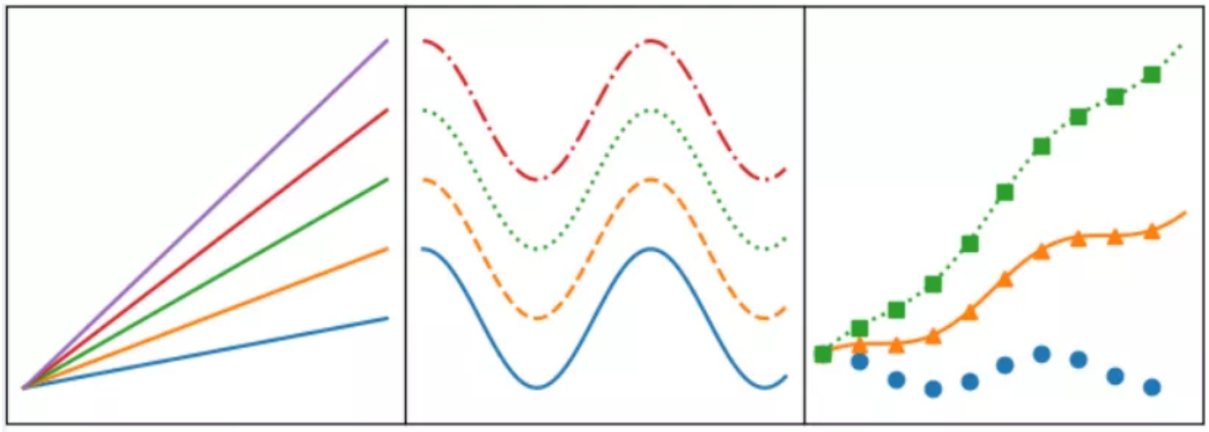
æÍąåİÙåŘ■çğřïijŽexample_utils.pyïijŇéĞŇéİćåŇĚæŃňäÿL'äÿłåĞ¡æŢřïijŇåŘĎèĞłåŁ§èČ¡åęĆäÿŃïijŽ

```
import matplotlib.pyplot as plt

# åĹŽåżžçĬżåŻ¿figåŠŇaxes
def setup_axes():
    fig, axes = plt.subplots(ncols=3, figsize=(6.5,3))
    for ax in fig.axes:
        ax.set(xticks=[], yticks=[])
    fig.subplots_adjust(wspace=0, left=0, right=0.93)
    return fig, axes
# åŻ¿çL'ĞæăĞéćŸ
def title(fig, text, y=0.9):
    fig.suptitle(text, size=14, y=y, weight='semibold', x=0.98, ha=
↪'right',
                 bbox=dict(boxstyle='round', fc='floralwhite', ec='
↪#8B7E66',
                           lw=2))
# äÿžæŢřæ■őæůżåŁăæŰĞæIJňæşĺéĞŁ
def label(ax, text, y=0):
    ax.annotate(text, xy=(0.5, 0.00), xycoords='axes fraction', ha=
↪'center',
                style='italic',
                bbox=dict(boxstyle='round', facecolor='floralwhite',
```

(continues on next page)

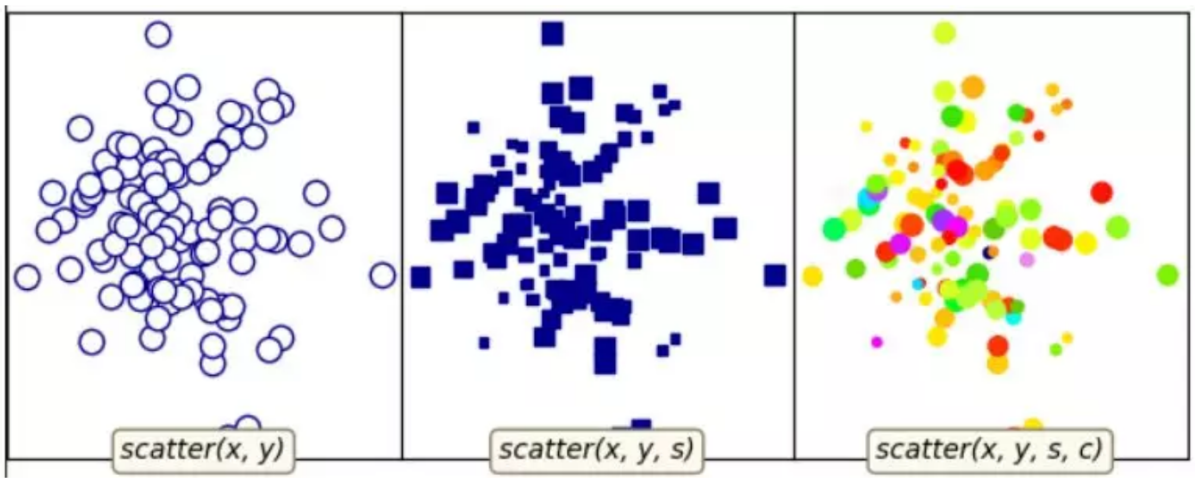
(continued from previous page)

```
ec='#8B7E66'))
```



```
import numpy as np
import matplotlib.pyplot as plt

import example_utils

x = np.linspace(0, 10, 100)

fig, axes = example_utils.setup_axes()
for ax in axes:
    ax.margins(y=0.10)

# å■ŘåŻ¿1 éżŸèőd'plotåd'ŽæÌąçž£ïijŇéćIJèL'šçşżçżŞåĹĘéĚ■
for i in range(1, 6):
    axes[0].plot(x, i * x)

# å■ŘåŻ¿2 åśŢçd'žçž£çŽĎäÿ■åŘŇlinestyle
for i, ls in enumerate(['-', '--', ':', '-.']):
    axes[1].plot(x, np.cos(x) + i, linestyle=ls)

# å■ŘåŻ¿3 åśŢçd'žçž£çŽĎäÿ■åŘŇlinestyleåŠŇmarker
for i, (ls, mk) in enumerate(zip(['', '-', ':'], ['o', '^', 's'])):
    axes[2].plot(x, np.cos(x) + i * x, linestyle=ls, marker=mk,␣
↪markevery=10)

# èő¿ç¡őæăĞéćŸ
# example_utils.title(fig, '"ax.plot(x, y, ...)": Lines and/or␣
↪markers', y=0.95)
# ä£Ìå■ŸåŻ¿çL'Ğ
fig.savefig('plot_example.png', facecolor='none')
# åśŢçd'žåŻ¿çL'Ğ
plt.show()
```

8.7 7 matplotlibæŢčçĆźåŻ¿



åŕźåžŤäžčçăĄïijŽ

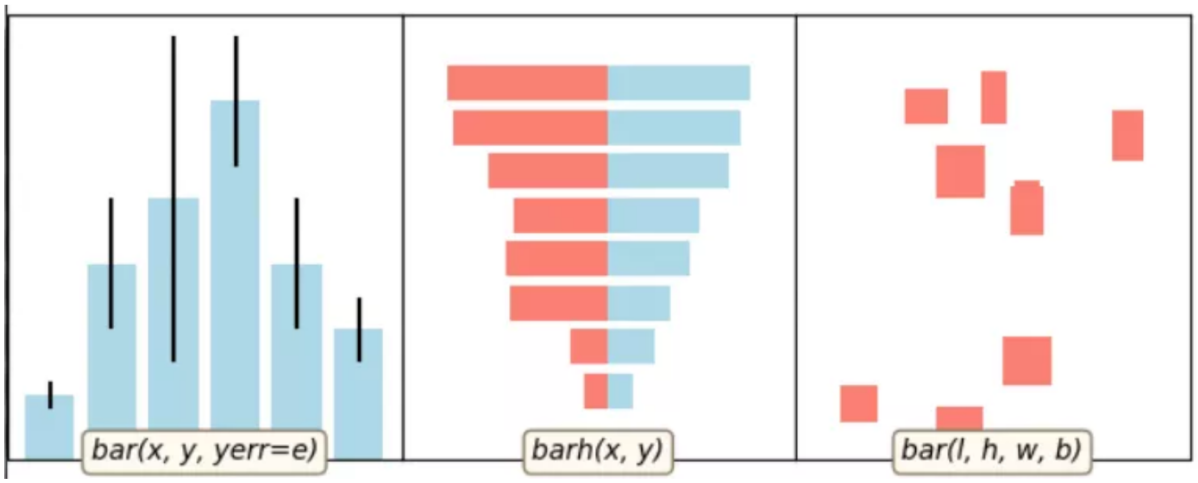
```
"""
æŢčçĆźåŻ¿çŽĎå$žæIJňçŤĺæşŢ
"""
import numpy as np
import matplotlib.pyplot as plt

import example_utils

# éŽŘæIJžçŤ$æĹŘæŢřæ■ő
np.random.seed(1874)
x, y, z = np.random.normal(0, 1, (3, 100))
t = np.arctan2(y, x)
size = 50 * np.cos(2 * t)**2 + 10

fig, axes = example_utils.setup_axes()

# å■ŘåŻ¿1
axes[0].scatter(x, y, marker='o',  color='darkblue', facecolor=
↪'white', s=80)
example_utils.label(axes[0], 'scatter(x, y)')

# å■ŘåŻ¿2
axes[1].scatter(x, y, marker='s', color='darkblue', s=size)
example_utils.label(axes[1], 'scatter(x, y, s)')

# å■ŘåŻ¿3
axes[2].scatter(x, y, s=size, c=z,  cmap='gist_ncar')
example_utils.label(axes[2], 'scatter(x, y, s, c)')

# example_utils.title(fig, '"ax.scatter(...)": Colored/scaled␣
↪markers',
```

(continues on next page)

(continued from previous page)

```
#                         y=0.95)
fig.savefig('scatter_example.png', facecolor='none')

plt.show()
```

8.8 8 matplotlibæ§śçŁűåŻ¿



åŕźåžŤäżčçăĄïijŽ

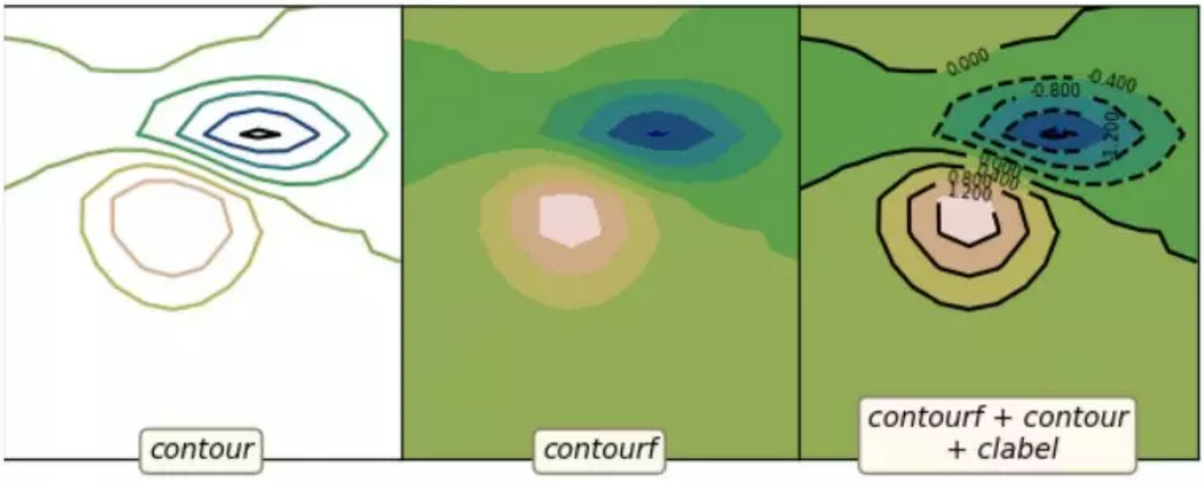
```
import numpy as np
import matplotlib.pyplot as plt

import example_utils


def main():
    fig, axes = example_utils.setup_axes()

    basic_bar(axes[0])
    tornado(axes[1])
    general(axes[2])

    # example_utils.title(fig, '"ax.bar(...)": Plot rectangles')
    fig.savefig('bar_example.png', facecolor='none')
    plt.show()

# å■ŘåŻ¿1
def basic_bar(ax):
    y = [1, 3, 4, 5.5, 3, 2]
    err = [0.2, 1, 2.5, 1, 1, 0.5]
    x = np.arange(len(y))
    ax.bar(x, y, yerr=err, color='lightblue', ecolor='black')
```

(continues on next page)

(continued from previous page)

```
    ax.margins(0.05)
    ax.set_ylim(bottom=0)
    example_utils.label(ax, 'bar(x, y, yerr=e)')

# å■ŘåŻ¿2
def tornado(ax):
    y = np.arange(8)
    x1 = y + np.random.random(8) + 1
    x2 = y + 3 * np.random.random(8) + 1
    ax.barh(y, x1, color='lightblue')
    ax.barh(y, -x2, color='salmon')
    ax.margins(0.15)
    example_utils.label(ax, 'barh(x, y)')

# å■ŘåŻ¿3
def general(ax):
    num = 10
    left = np.random.randint(0, 10, num)
    bottom = np.random.randint(0, 10, num)
    width = np.random.random(num) + 0.5
    height = np.random.random(num) + 0.5
    ax.bar(left, height, width, bottom, color='salmon')
    ax.margins(0.15)
    example_utils.label(ax, 'bar(l, h, w, b)')


main()
```

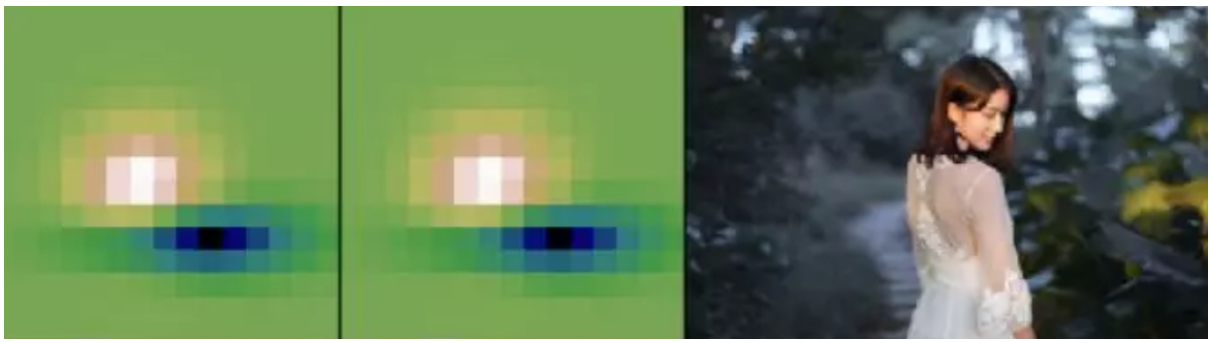
8.9 9 matplotlibç■L'éńŸçž£åŻ¿



åŕźåžŤäžčçăĄïijŽ

```
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.cbook import get_sample_data

import example_utils

z = np.load(get_sample_data('bivariate_normal.npy'))

fig, axes = example_utils.setup_axes()

axes[0].contour(z, cmap='gist_earth')
example_utils.label(axes[0], 'contour')

axes[1].contourf(z, cmap='gist_earth')
example_utils.label(axes[1], 'contourf')

axes[2].contourf(z, cmap='gist_earth')
cont = axes[2].contour(z, colors='black')
axes[2].clabel(cont, fontsize=6)
example_utils.label(axes[2], 'contourf + contour\n + clabel')

# example_utils.title(fig, '"contour, contourf, clabel": Contour/
↪label 2D data',
#                     y=0.96)
fig.savefig('contour_example.png', facecolor='none')

plt.show()
```

8.10 10 imshowåŻ¿



åŕźåžŤäžčçăĄïijŽ

```
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.cbook import get_sample_data
from mpl_toolkits import axes_grid1
```

(continues on next page)

(continued from previous page)

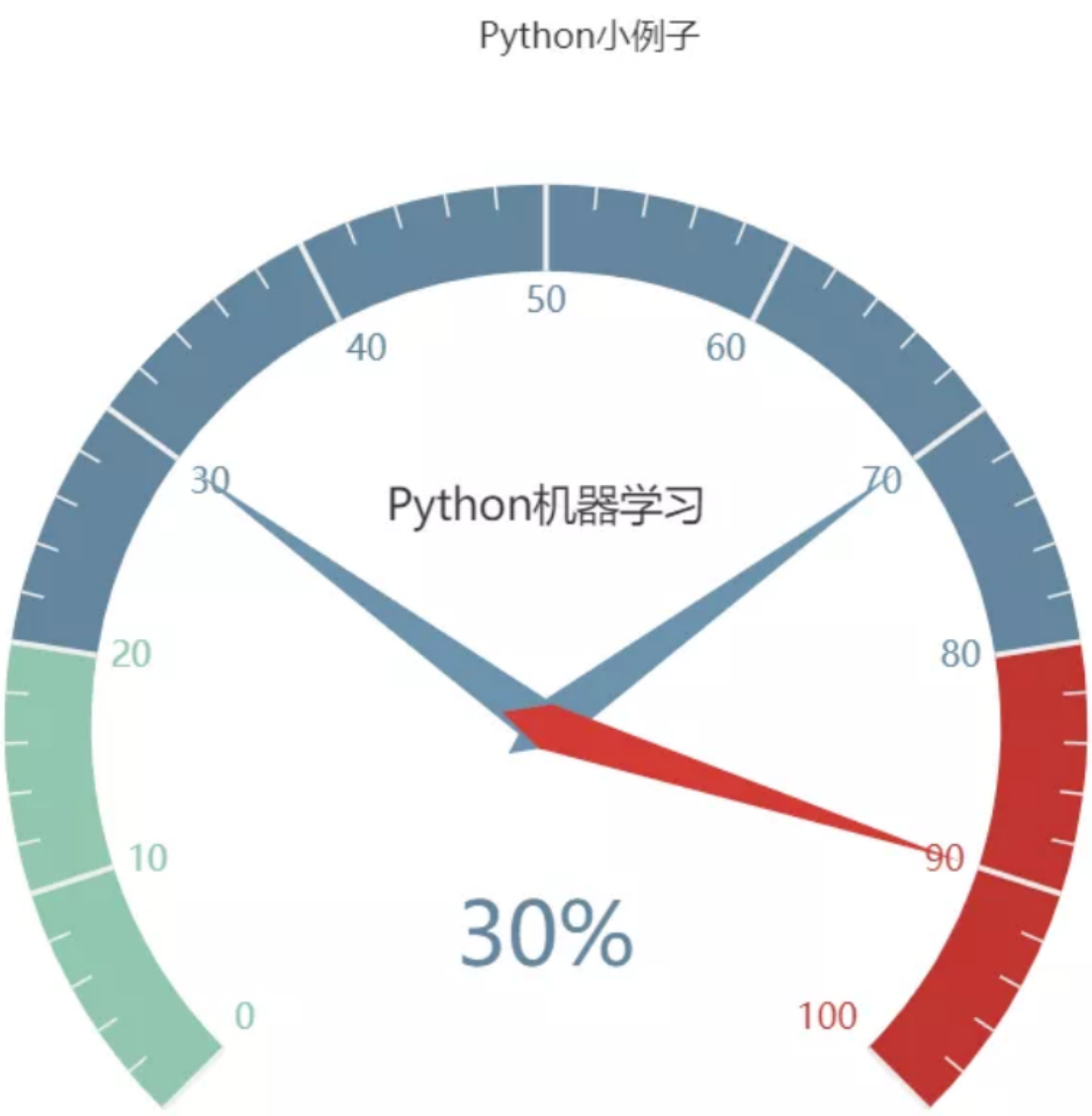
```
import example_utils


def main():
    fig, axes = setup_axes()
    plot(axes, *load_data())
    # example_utils.title(fig, '"ax.imshow(data, ...)": Colormapped␣
↪or RGB arrays')
    fig.savefig('imshow_example.png', facecolor='none')
    plt.show()


def plot(axes, img_data, scalar_data, ny):

    # éżŸèőd'çž£æĂğæŔŠåĂij
    axes[0].imshow(scalar_data, cmap='gist_earth', extent=[0, ny,␣
↪ny, 0])

    # æIJĂè£ŚéĆżæŔŠåĂij
    axes[1].imshow(scalar_data, cmap='gist_earth', interpolation=
↪'nearest',
                   extent=[0, ny, ny, 0])

    # åśŢçd'žRGB/RGBAæŢřæ■ő
    axes[2].imshow(img_data)


def load_data():
    img_data = plt.imread(get_sample_data('5.png'))
    ny, nx, nbands = img_data.shape
    scalar_data = np.load(get_sample_data('bivariate_normal.npy'))
    return img_data, scalar_data, ny


def setup_axes():
    fig = plt.figure(figsize=(6, 3))
    axes = axes_grid1.ImageGrid(fig, [0, 0, .93, 1], (1, 3), axes_
↪pad=0)

    for ax in axes:
        ax.set(xticks=[], yticks=[])
    return fig, axes


main()
```

8.11 11 pyechartsçżŸåĹűäżłèąĺçŻŸ

ä¡£çŤĺpip install pyecharts åőL'èčĚïijŇçL'ĹæIJňäÿž v1.6ïijŇpyechartsçżŸåĹűäżłèąĺçŻŸïijŇåŕłéIJĂèęĄå

```
from pyecharts import charts

# äżłèąĺçŻŸ
gauge = charts.Gauge()
gauge.add('PythonåřŕäżŃå■Ř', [('PythonæIJžåźĺå■ęäźă', 30), (
↪'Pythonå§žçąĂ', 70.),
                          ('Pythonæ■čåĹŹ', 90)])
gauge.render(path="./data/äżłèąĺçŻŸ.html")
print('ok')
```

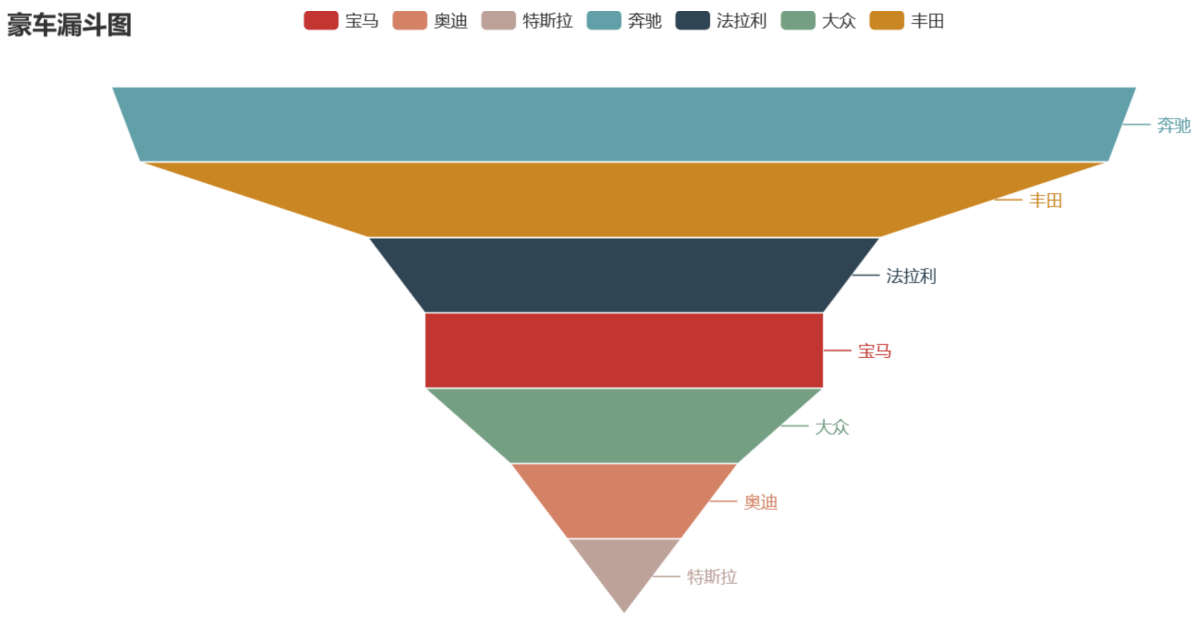
äżłèąĺçŻŸäÿ■åĚśåśŢçd'žäÿL'éąźïijŇæŕŕéąźçŽĎæŕŤä¿ŃäÿžŠ30%,70%,90%ïijŇåęĆäÿŃåŻ¿éżŸèőd'åŘ■ç



8.12 12 pyechartsæijŕræŰŮåŻż

```
from pyecharts import options as opts
from pyecharts.charts import Funnel, Page
from random import randint

def funnel_base() -> Funnel:
  c = (
    Funnel()
    .add("èśłè¡ę", [list(z) for z in zip(['åőİél'ň', 'æşŢæŃL'åĹl'',
↪'åěŤél'ř', 'åěěè£ł', 'åd'ğäijŮ', 'äÿřçŤř', 'çL'źæŰŕæŃL''],
                [randint(1, 20) for _ in range(7)])])
    .set_global_opts(title_opts=opts.TitleOpts(title=
↪"èśłè¡ęæijŕræŰŮåŻż"))
  )
  return c
funnel_base().render('./img/car_fnnel.html')
```

äžě7çğ■è¡ęåđŃåŕŁæ§Řäÿłåśđæ˘ĂğåĂijçżŸåĹűçŽĎæijŕræŰŮåŻ¿ïijŇåśđæĂğåĂijåď ğèűŁéĺàè£ŚæijŕræŰ



8.13 13 pyechartsæŮěåŐĘåŻż

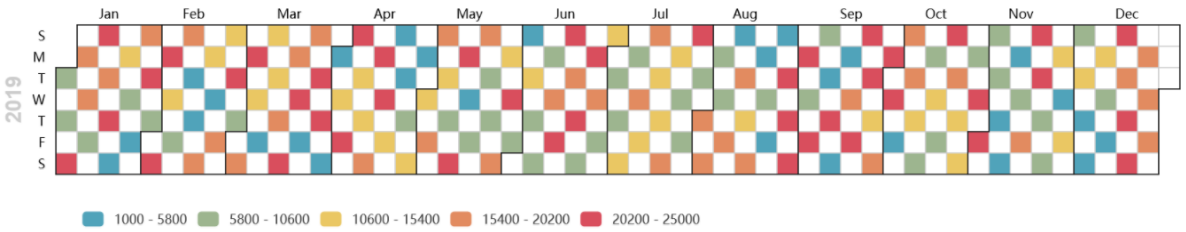
```
import datetime
import random
from pyecharts import options as opts
from pyecharts.charts import Calendar

def calendar_interval_1() -> Calendar:
    begin = datetime.date(2019, 1, 1)
```

(continues on next page)

(continued from previous page)

```
    end = datetime.date(2019, 12, 27)
    data = [
        [str(begin + datetime.timedelta(days=i)), random.
↪randint(1000, 25000)]
        for i in range(0, (end - begin).days + 1, 2)  #␣
↪éŽŤåd'l'çż§èőą
    ]
    calendar = (
      Calendar(init_opts=opts.InitOpts(width="1200px")).add(
            "", data, calendar_opts=opts.CalendarOpts(range_="2019
↪"))
        .set_global_opts(
            title_opts=opts.TitleOpts(title="Calendar-
↪2019åźt'æ■ěæŢřçż§èőą"),
            visualmap_opts=opts.VisualMapOpts(
                max_=25000,
                min_=1000,
                orient="horizontal",
                is_piecewise=True,
                pos_top="230px",
                pos_left="100px",
            ),
        )
    )
    return calendar

calendar_interval_1().render('./img/calendar.html')
```

çżŸåĹű2019åźť1æIJĹ1æŮěåĹř12æIJĹ27æŮěçŽĎæ■ěèąŇæŢřïijŇåőŸæŰźçżŹåĞžçŽĎåŻ¿å¡ćåő¡åžę90
InitOpts(width="1200px")åĄŽåĞžå¿őèřČïijŇåźűäÿŤvisualmapæŸ¿çd'žæĽĂæIJĽæ■ěæŢřïijŇæ



8.14 14 pyechartsçżŸåĹűgraphåŻ¿

```
import json
import os
from pyecharts import options as opts
from pyecharts.charts import Graph, Page

def graph_base() -> Graph:
```

(continues on next page)

(continued from previous page)

```
    nodes = [
        {"name": "cus1", "symbolSize": 10},
        {"name": "cus2", "symbolSize": 30},
        {"name": "cus3", "symbolSize": 20}
    ]
    links = []
    for i in nodes:
        if i.get('name') == 'cus1':
            continue
        for j in nodes:
            if j.get('name') == 'cus1':
                continue
            links.append({"source": i.get("name"), "target": j.get(
↪"name")})
    c = (
        Graph()
        .add("", nodes, links, repulsion=8000)
        .set_global_opts(title_opts=opts.TitleOpts(title="customer-
↪influence"))
    )
    return c
```

æđĎåżžåŻ¿ïijŇåĚűäÿ■åőćæĹůçĆź1äÿŐåĚűäżŰäÿd'äÿłåőćæĹůéČ¡æšąæIJL'åĚşçşż(link)ïijŇäź§åřśæŸ

8.15 15 pyechartsæřťçŘČåŻ¿

```
from pyecharts import options as opts
from pyecharts.charts import Liquid, Page
from pyecharts.globals import SymbolType

def liquid() -> Liquid:
```

(continues on next page)

(continued from previous page)

```
    c = (
        Liquid()
        .add("lq", [0.67, 0.30, 0.15])
        .set_global_opts(title_opts=opts.TitleOpts(title="Liquid"))
    )
    return c

liquid().render('../../img/liquid.html')
```

æřť çŘČåŻ¿çŽĎåŘŰåĂij[0.67, 0.30, 0.15]èąĺçd'žäÿŃåŻ¿äÿ■çŽĎäÿL'äÿłæşćæţłçž£ïijŇäÿ

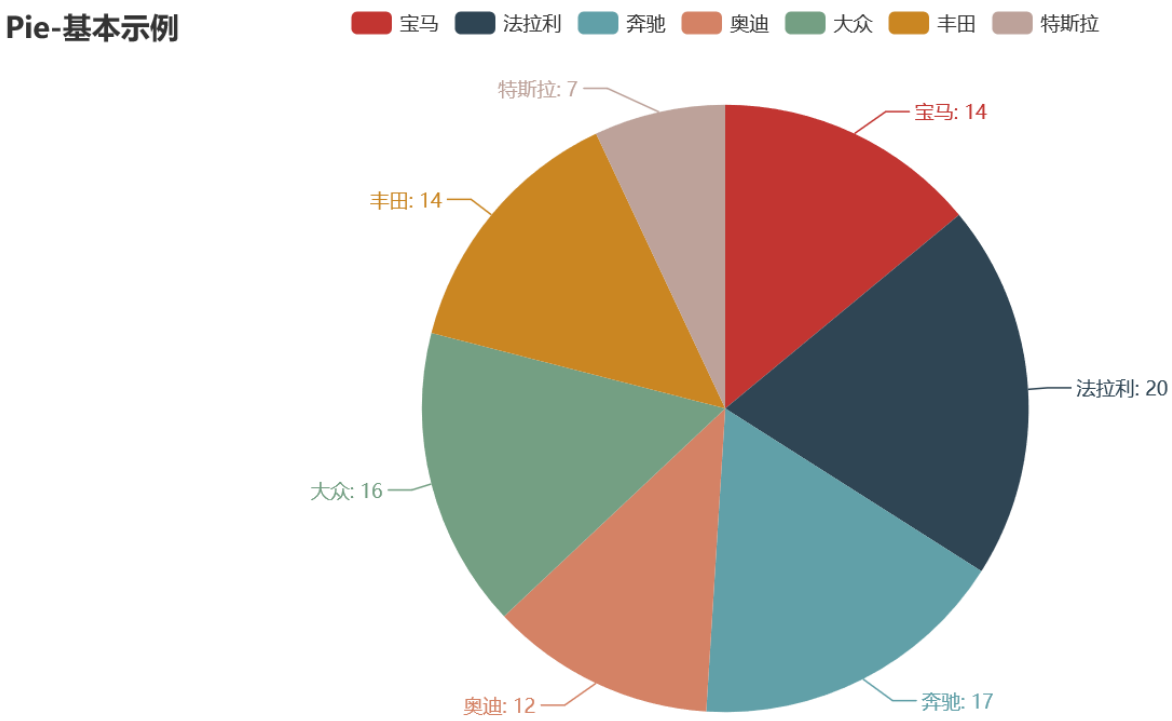
8.16 16 pyechartséěijåŻ¿

```
from pyecharts import options as opts
from pyecharts.charts import Pie
from random import randint

def pie_base() -> Pie:
    c = (
        Pie()
        .add("", [list(z) for z in zip(['åőİél'ň', 'æşŢæŃL'åĹl'',
↪'åěŤél'ř', 'åěëè£ł', 'åd'ğäijŮ', 'äÿřçŤř', 'çL'źæŰŕæŃL''],
                                       [randint(1, 20) for _ in␣
↪range(7)])])
        .set_global_opts(title_opts=opts.TitleOpts(title="Pie-
↪å§žæIJňçd'žä¿Ń"))
        .set_series_opts(label_opts=opts.LabelOpts(formatter="{b}:
↪{c}"))
    )
    return c

pie_base().render('../../img/pie_pyecharts.html')
```

8.17 17 pyechartsædĄåİŘæăĞåŻ¿

```
import random
from pyecharts import options as opts
from pyecharts.charts import Page, Polar

def polar_scatter0() -> Polar:
    data = [(alpha, random.randint(1, 100)) for alpha in␣
↪range(101)] # r = random.randint(1, 100)
    print(data)
```

(continues on next page)



(continued from previous page)

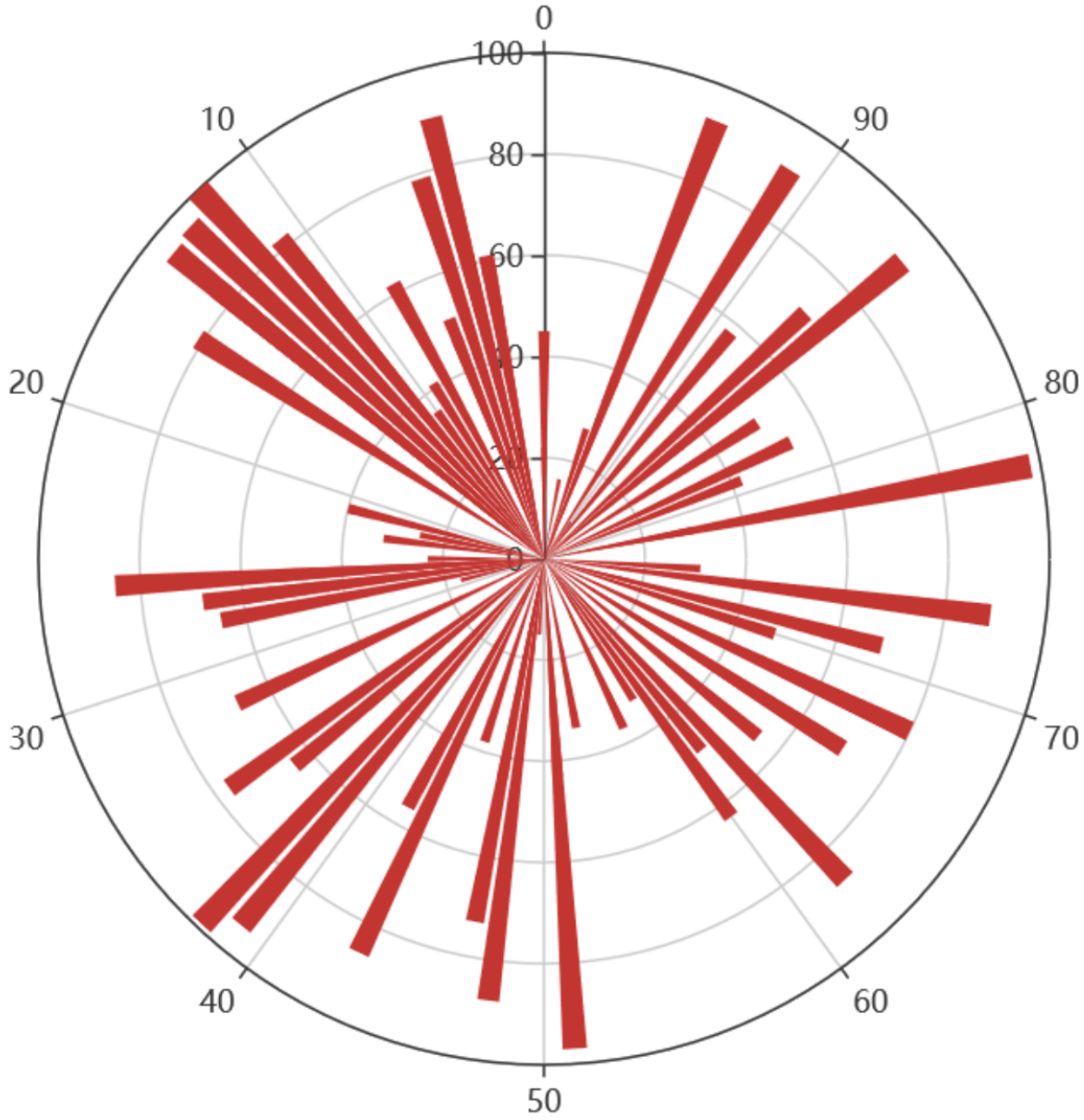
```
    c = (
        Polar()
        .add("", data, type_="bar", label_opts=opts.LabelOpts(is_
↪show=False))
        .set_global_opts(title_opts=opts.TitleOpts(title="Polar"))
    )
    return c

polar_scatter0().render('./img/polar.html')
```

æđĄåİŘæăĞèąÍçđ'žäÿž(åđ'źèğŠ,å■Łå¿Ď)ïijŇåęĆ(6,94)èąÍçđ'žåđ'źèğŠäÿž6ïijŇå■Łå¿Ď94çŽĎçĆźïi

8.18 18 pyechartsèŕ■äžŚåŻ¿

```
from pyecharts import options as opts
from pyecharts.charts import Page, WordCloud
from pyecharts.globals import SymbolType

words = [
    ("Python", 100),
    ("C++", 80),
    ("Java", 95),
    ("R", 50),
    ("JavaScript", 79),
    ("C", 65)
```

(continues on next page)


0
100
10
90
80
60
20
80
30
70
40
60
50

(continued from previous page)

```
]

def wordcloud() -> WordCloud:
    c = (
        WordCloud()
        # word_size_range: å■Ţèŕ■å■Ůä¡Şåd'ğåřŔèŇČåŻt'
        .add("", words, word_size_range=[20, 100], shape='cardioid')
        .set_global_opts(title_opts=opts.TitleOpts(title="WordCloud
↪"))
    )
    return c

wordcloud().render('./img/wordcloud.html')
```

("C", 65)èąĺçď'žåIJĺæIJňæňąçż§èőąäÿ■Cèŕ■èĺĂåĞžçŐř65æňą

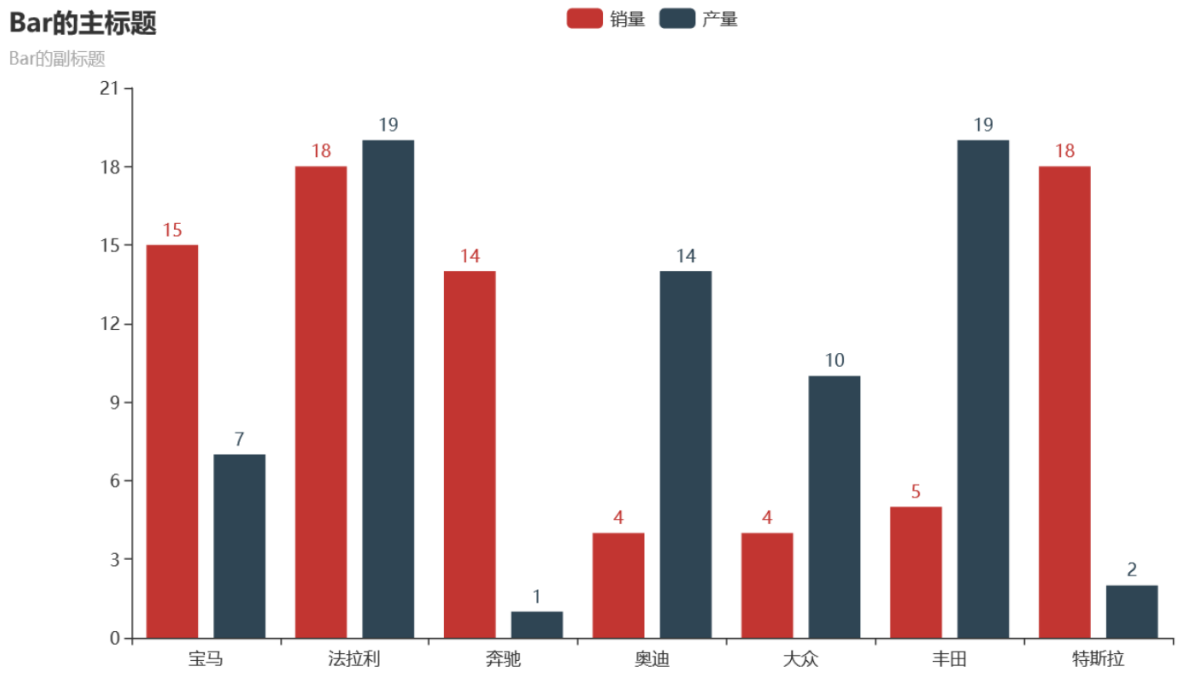
8.19 19 pyechartsçşżåĹŮæ§śçŁűåŻ¿

```
from pyecharts import options as opts
from pyecharts.charts import Bar
from random import randint

def bar_series() -> Bar:
    c = (
        Bar()
        .add_xaxis(['åőİél'ň', 'æşŢæŃL'åĹl'', 'åěŤél'ř', 'åěèè£ł',
↪'åd'ğäijŮ', 'äÿřçŤř', 'çL'źæŰŕæŃL'', ''])
        .add_yaxis("éŤĂéĞŔ", [randint(1, 20) for _ in range(7)])
        .add_yaxis("äžğéĞŔ", [randint(1, 20) for _ in range(7)])
        .set_global_opts(title_opts=opts.TitleOpts(title=
↪"BarçŽĎäÿżæäĞéćŸ", subtitle="BarçŽĎåL'ŕæäĞéćŸ"))
```

(continues on next page)

(continued from previous page)

```
    )
    return c

bar_series().render('./img/bar_series.html')
```



8.20 20 pyechartsçČ■åŁŻåŻ¿

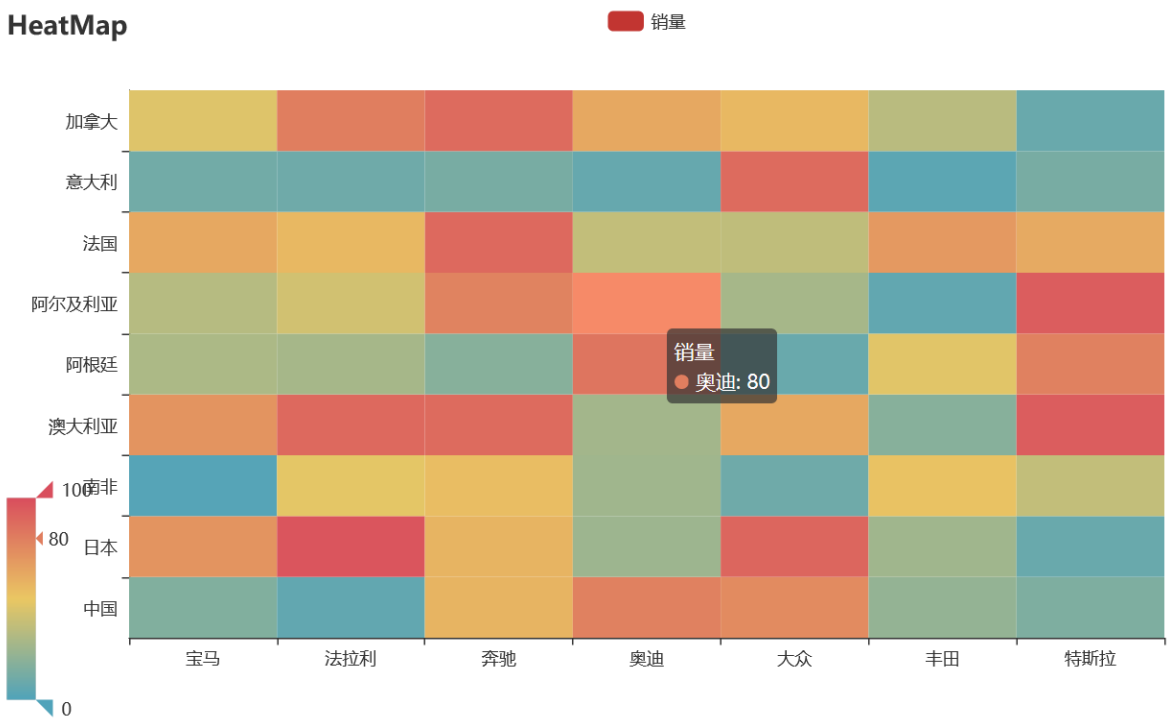
```
import random
from pyecharts import options as opts
from pyecharts.charts import HeatMap

def heatmap_car() -> HeatMap:
    x = ['åőİél’ň', 'æşŢæŃL’åĹl’', 'åěŤél’ř', 'åěëè£ł', 'åd’ğäijŮ',
↪'äÿřçŤř', 'çL’źæŰŕæŃL’']
    y = ['äÿ■åŻ¡','æŮěæIJň','å■Ůéİđ','æ¿şåd’ğåĹl’äžŽ','éŸ£æăźåżů',
↪'éŸ£åřŤåŔŁåĹl’äžŽ','æşŢåŻ¡','æĎŔåd’ğåĹl’','åŁăæŃ£åd’ğ']
    value = [[i, j, random.randint(0, 100)]
             for i in range(len(x)) for j in range(len(y))]
    c = (
        HeatMap()
        .add_xaxis(x)
        .add_yaxis("éŤĂéĞŔ", y, value)
        .set_global_opts(
            title_opts=opts.TitleOpts(title="HeatMap"),
            visualmap_opts=opts.VisualMapOpts(),
        )
```

(continues on next page)

(continued from previous page)

```
    )
    return c

heatmap_car().render('./img/heatmap_pyecharts.html')
```

çČ■åŁŻåŻ¿æŔŕè£řçŽĎåőđéŹĚæŸŕäÿL'çžt'åĚşçşżïijŇxè¡t'èąĺçd'žè¡ęåđŃïijŇyè¡t'èąĺçd'žåŻ¡åőűïijŇæŕŔ



8.21 21 matplotlibçżŸåĹűåŁĺçŤż

matplotlibæŸŕpythonäÿ■æIJĂçżŔåĚÿçŽĎçżŸåŻ¿åŇĚïijŇéĞŇéİćanimationæĺąåİÙèČ¡çżŸåĹűå

éęŰåĚĹåŕijåĚěåřŔä¿Ńå■Řä¡£çŤĺçŽĎæĺąåİÙïijŽ

```
from matplotlib import pyplot as plt
from matplotlib import animation
from random import randint, random
```

çŤ§æĹŘæŢřæ■őïijŇframes_countæŸŕåÿğçŽĎäÿłæŢřïijŇdata_countæŕŔäÿłåÿğçŽĎæ§śå■Řäÿłæ

```
class Data:
    data_count = 32
    frames_count = 2

    def __init__(self, value):
        self.value = value
        self.color = (0.5, random(), random()) #rgb
```

(continues on next page)

(continued from previous page)

```
    # éĂăæŢřæ■ő
    @classmethod
    def create(cls):
        return [[Data(randint(1, cls.data_count)) for _ in␣
↪range(cls.data_count)]
                for frame_i in range(cls.frames_count)]
```

çżŸåĹűåŁĺçŤżïijŽanimation.FuncAnimationåĞ¡æŢřçŽĎåŻđèřČåĞ¡æŢřçŽĎåŔĆæŢřfièąĺçď'žçcla()æÿĚéŹď'äÿŁäÿĂåÿğãĂĆ

```
def draw_chart():
    fig = plt.figure(1, figsize=(16, 9))
    axs = fig.add_subplot(111)
    axs.set_xticks([])
    axs.set_yticks([])

    # çŤ§æĹŘæŢřæ■ő
    frames = Data.create()

    def animate(fi):
        axs.cla()  # clear last frame
        axs.set_xticks([])
        axs.set_yticks([])
        return axs.bar(list(range(Data.data_count)),        # X
                       [d.value for d in frames[fi]],       # Y
                       1,                                   # width
                       color=[d.color for d in frames[fi]]  # color
                       )
    # åŁĺçŤżåśŢçd'ž
    anim = animation.FuncAnimation(fig, animate, frames=len(frames))
    plt.show()


draw_chart()
```

CHAPTER 9

äÿČãĂĄ PythonäźŃåİŚ

æĂżçżŞPythonæIJĂåőźæŸŞåĞžéŤŹçŽĎç§ěèŕĘçĆźïijŇéĂŽè£ĞåřŔä¿Ńå■ŘçŽĎå¡ćåijŔïijŇäÿĂäÿĂåŚĹ

9.1 1 åŘńå■Ţäÿłå ĚČçť äçŽĎåĚČçżĎ

Pythonäÿ■æIJL’äžŻåĞ¡æŢřçŽĎåŔĆæŢřçśżåđŃäÿžåĚČçżĎïijŇåĚűåĘĚæIJL’1äÿłåĚČçť ăïijŇè£ŹæăůåĹŻ

```
c = (5) # NO!
```

åőČåőđéŹĚåĹŻåżžäÿĂäÿłæŢť åđŃåĚČçť ă5ïijŇå£Ěéążèę ĄåIJĺåĚČçť ăåŘŐåŁääÿĂäÿłéĂŮåŔů:

```
c = (5,) # YES!
```

9.2 2 éžŸèőďåŔĆæŢřèő¿äÿžçľž

åŘńæIJL’éżŸèőď åŔĆæŢřçŽĎåĞ¡æŢřïijŇåęĆæđIJçśżåđŃäÿžåőźåŹĺïijŇäÿŤèő¿ç¡őäÿžçľ žïijŽ

```
def f(a,b=[]):  # NO!
    print(b)
    return b

ret = f(1)
ret.append(1)
ret.append(2)
# å¡ŞåĘ■èřČçŤĺf(1)æŮűïijŇécĎèőąæL’Şå■řäÿž []
```

(continues on next page)

(continued from previous page)

```
f(1)
# ä¡ĘæŸŕå■t'äÿž [1,2]
```

è£ŹæŸŕåŔŕåŔŸçśžåđŃçŽĎéżŸèőď'åŔĆæŢřäźŃåİŠïijŇèŕůåŁąå£Ěèő¿ç¡őæ■ď'çśžéżŸèőď'åŔĆæŢřäÿžNo

```
def f(a,b=None): # YES!
    pass
```

9.3 3 åĚšäžńåŔŸéĞŔæIJłçżŠåőŽäźŃåİŚ

æIJL'æŮűæČşèęĄåď'Žäÿ łåĞ¡æŢřåĚšäžńäÿĂäÿłåĚĺåśĂåŔŸéĞŔïijŇä¡Ęå■t'åIJÍæ§ŘäÿłåĞ¡æŢřåĘĚèŕŢåŻ

```
i = 1
def f():
    i+=1 #NO!

def g():
    print(i)
```

åžŤèŕěåIJÍfåĞ¡æŢřåĘĚæŸ¿çď'žåčřæŸŐiäÿžglobalåŔŸéĞŔïijŽ

```
i = 1
def f():
    global i # YES!
    i+=1
```

9.4 4 lambdaèĞłçŤśåŔĆæŢřäźŃåİŚ

æŐŠåžŔåŠŇåĹĘçżĎçŽĎkeyåĞ¡æŢřåÿÿä¡£çŤĺlambdaïijŇèąĺè¿¿æŻt'åŁăçőĂæt'ĄïijŇä¡ĘæŸŕæIJL'äÿłåİŚ

```
a = [lambda x: x+i for i in range(3)] # NO!
for f in a:
    print(f(1))
# ä¡ăåŔŕèČ¡æIJŞæIJŻè¿ŞåĞžïijž1,2,3
```

ä¡ĘæŸŕåőđéŹĚå■t'è¿ŞåĞž: 3,3,3. åőŽäźL'lambdaä¡£çŤĺçŽĎiècńçğřäÿžèĞłçŤśåŔĆæŢřïijŇåőČåŔłåIJĺè

```
a = 0
a = 1
a = 2
def f(a):
    print(a)
```

æ■čçąőåĄŽæşŢæŸŕè¡ňåŇŰèĞłçŤśåŔĆæŢřäÿžlambdaåĞ¡æŢřçŽĎéżŸèőd'åŔĆæŢřïijŽ

```
a = [lambda x,i=i: x+i for i in range(3)] # YES!
```

9.5 5 åŘĎçğ■åŕĆæŢřä¡£çŤĺäźŃåİŚ

Pythonåijžåď ğåď ŽåŔŸïijŇåŐ§åŻääźŃäÿĂåIJĺäžŐåĞ¡æŢřåŔĆæŢřçśżåđŃçŽĎåď ŽæäůåŇŰãĂĆæŰźä¿‡

(1) SyntaxError: positional argument follows keyword argument

(2) TypeError: f() missing 1 required keyword-only argument: âĂŸbâĂŹ

(3) SyntaxError: keyword argument repeated

(4) TypeError: f() missing 1 required positional argument: âĂŸbâĂŹ

(5) TypeError: f() got an unexpected keyword argument âĂŸaâĂŹ

(6) TypeError: f() takes 0 positional arguments but 1 was given

æĂżçżŞäÿżèęAçŽĎåŔĆæŢřä¡£çŤĺèğĎåĹŹ

ä¡■ç¡őåŔĆæŢř

ä¡■ç¡őåŔĆæŢřçŽĎåőŽäźL'ïijŽåĞ¡æŢřèřČçŤĺæŮűæäźæ■őåĞ¡æŢřåőŽäźL'çŽĎåŔĆæŢřä¡■ïijĹå¡ćå

```
def f(a):
  return a

f(1) # ä¡■ç¡őåŔĆæŢř
```

ä¡■ç¡őåŔĆæŢřäÿ■èČ¡çijžåřŚïijŽ

```
def f(a,b):
  pass

f(1) # TypeError: f() missing 1 required positional argument: 'b'
```

èğĎåĹŹ1ïijŽä¡■ç¡őåŔĆæŢřå£Ěéążäÿ ĂäÿĂåŕźåžŤïijŇçijžäÿĂäÿ■åŔŕ

åĚşéŤőå■ŮåŔĆæŢř

åIJĺåĞ¡æŢřèřČçŤĺæŮűïijŇéĂŽè£ĞâĂŸéŤő–åĂijâĂŹæŰźåijŔäÿžåĞ¡æŢřå¡ćåŔĆäijäåĂijïijŇäÿ■çŤĺæŇL

```
def f(a):
  print(f'a:{a}')
```

è£ŹäźĹèřČçŤĺïijŇaåřśæŸŕåĚşéŤőå■ŮåŔĆæŢřïijŽ

```
f(a=1)
```

ä¡ĘæŸŕäÿŃéİćèřČçŤĺåřśäÿ■OK:

```
f(a=1,20.) # SyntaxError: positional argument follows keyword␣
↪argument
```

èğĎåĹŹ2ïijŽåĚşéŤőå■ŮåŔĆæŢřå£Ěéążå IJĺä¡■ç¡őåŔĆæŢřåŕşè¿ź

äÿŃéİćèřČçŤĺäź§äÿ■OK:

```
f(1,width=20.,width=30.) #SyntaxError: keyword argument repeated
```

èğĎåĹŹ3ïijŽåŕźåŘŇäÿĂäÿłå¡ćåŔĆäÿ■èČ¡éĞ■åd'■äijăåĂij

éżŸèőď'åŔĆæŢř

åIJĺåőŽäźL'åĞ¡æŢřæŮűïijŇåŕäžěäÿžå¡ćåŔĆæŔŘä¿ŻéżŸèőď'åĂijãĂĆåŕźäžŐæIJL'éżŸèőď'åĂijçŽĎå¡ć

```
def f(a,b=1):
  print(f'a:{a}, b:{b}')
```

èğĎåĹŹ4ïijŽæŮäèőžæŸŕåĞ¡æŢřçŽĎåőŽäźL'è£ŸæŸŕèřČçŤĺïijŇéżŸèőď'åŔĆæŢřçŽĎåőŽäźL'åžŤ

åŘłåIJĺåőŽäźL'æŮűęţŇåĂijäÿĂæňąïijŻéżŸèőď'åŔĆæŢřéĂŽåÿÿåžŤèŕěåőŽäźL'æĹŘäÿ■åŕŕåŘŸçśžåđŃ

åŕŕåŘŸä¡■ç¡őåŔĆæŢř

åęĆäÿŃåőŽäźL'çŽĎåŔĆæŢřaäÿžåŕŕåŘŸä¡■ç¡őåŔĆæŢřïijŽ

```
def f(*a):
  print(a)
```

èřČçŤĺæŰźæşŢïijŽ

```
f(1) #æL'Şå■řçżŞæđIJäÿžåĚČçżĎïijŽ (1,)
f(1,2,3) #æL'Şå■řçżŞæđIJïijŽ(1, 2, 3)
```

ä¡ĘæŸŕïijŇäÿ■èČ¡è£ŹäźĹèřČçŤĺïijŽ

```
f(a=1) # TypeError: f() got an unexpected keyword argument 'a'
```

åŕŕåŘŸåĚşéŤőå■ŮåŔĆæŢř

åęĆäÿŃaæŸŕåŕŕåŘŸåĚşéŤőå■ŮåŔĆæŢřïijŽ

```
def f(**a):
  print(a)
```

èřČçŤĺæŰźæşŢïijŽ

```
f(a=1) #æL'Şå■řçżŞæđIJäÿžå■ŮåĚÿïijŽ{'a': 1}
f(a=1,b=2,width=3) #æL'Şå■řçżŞæđIJïijŽ{'a': 1, 'b': 2, 'width': 3}
```

ä¡ĘæŸŕïijŇäÿ■èČ¡è£ŹäźĹèřČçŤĺïijŽ

```
f(1) TypeError: f() takes 0 positional arguments but 1 was given
```

æŐěäÿŃæİěïijŇå■ŢçŃňæŐÍéĂĄåĹĘæđŘäÿĂäÿłåřŕäżŃå■ŘïijŇçżijåŘĹäžěäÿŁåŘĎçğ■åŕĆæŢřçśżåđŃç

9.6 6 åĹŮèąĺåĹăéŹď'äźŃåİŠ

åĹăéŹď'äÿĂäÿłåĹŮèąĺäÿ■çŽĎåĚČçt'ăïijŇæ■d'åĚČçt'ăåŕŕèČ¡åIJĺåĹŮèąĺäÿ■éĞ■åd'■åd'Žæňąïij Ž

```
def del_item(lst,e):
    return [lst.remove(i) for i in e if i==e] # NO!
```

èĂČèŹŠåĹăéŹď'è£ŹäÿłåžŕåĹŮ[1,3,3,3,5]äÿ■çŽĎåĚČçt'ă3ïijŇçżŞæđIJåŕŠçŐřåŕłåĹăéŹď'åĚűäÿ■äÿd'ä

```
del_item([1,3,3,3,5],3) # çżŞæđIJïijž[1,3,5]
```

æ■čçąőåĄŽæşŢïijŽ

```
def del_item(lst,e):
    d = dict(zip(range(len(lst)),lst)) # YES! æđĎéĂăå■ŮåĚÿ
    return [v for k,v in d.items() if v!=e]
```

9.7 7 åĹŮèąĺå£ńéĂ§åd'■åĹűäźŃåİŠ

åIJĺpythonäÿ■⋆äÿŐåĹŮèąĺæŞ■ä¡IJïijŇåőđçŐřå£ńéĂ§åĚČçt'ăåd'■åĹűïijŽ

```
a = [1,3,5] * 3 # [1,3,5,1,3,5,1,3,5]
a[0] = 10 # [10, 2, 3, 1, 2, 3, 1, 2, 3]
```

åęĆæđIJåĹŮèąĺåĚČçt'ăäÿžåĹŮèąĺæĹŰå■ŮåĚÿç■L'åd'■åŘĹçśżåđŃïijŽ

```
a = [[1,3,5],[2,4]] * 3 # [[1, 3, 5], [2, 4], [1, 3, 5], [2, 4], [1,
↪ 3, 5], [2, 4]]

a[0][0] = 10 #
```

çżŞæđIJåŕŕèČ¡åĞžäźŐä¡ăçŽĎæĎŕæŰŹïijŇåĚűäżŰa[1[0]ç■L'äź§èćńä£őæŤźäÿž10

```
[[10, 3, 5], [2, 4], [10, 3, 5], [2, 4], [10, 3, 5], [2, 4]]
```

è£ŹæŸŕåŽăäÿž*åd'■åĹűçŽĎåd'■åŘĹåŕźèśąéČ¡æŸŕæţĚåijŢçŤĺïijŇäź§åřśæŸŕèŕť'id(a[0])äÿŐid(a[2])éŮĺ

```
a = [[] for _ in range(3)]
```

9.8 8 å■Ůçňęäÿšéľžç Ţ Ź

```
In [1]: a = 'something'
    ...: b = 'some'+'thing'
    ...: id(a)==id(b)
Out[1]: True
```

åęĆæđIJäÿŁéİćä¿Ńå■Řè£ŤåŻđTrueïijŇä¡ĘæŸŕäÿŃéİćä¿Ńå■ŘäÿžäżĂäź ĹæŸŕFalse:

```
In [1]: a = '@zglg.com'

In [2]: b = '@zglg'+'.com'

In [3]: id(a)==id(b)
Out[3]: False
```

è£ŹäÿŐCpython çijŰèŕŚäijŸåŇŰçŻÿåĚşïijŇèąŇäÿžçğřäÿžå■Ůçňęäÿšél'żçŢŹïijŇä¡Ęél'żçŢŹçŽĎå■

9.9 9 çŻÿåŘŇåĂijçŽĎäÿ■åŔŕåŔŸåŕźèšą

```
In [5]: d = {}
    ...: d[1] = 'java'
    ...: d[1.0] = 'python'

In [6]: d
Out[6]: {1: 'python'}

### key=1,value=javaçŽĎéŤőåĂijåŕźçěđåŹĺæűĹåd'šäžĘ
In [7]: d[1]
Out[7]: 'python'
In [8]: d[1.0]
Out[8]: 'python'
```

è£ŹæŸŕåŻääÿžåĚůæIJL'çŻÿåŘŇåĂijçŽĎäÿ■åŔŕåŔŸåŕźèšąåIJÍPythonäÿ■ågŃçźĹåĚůæIJL'çŻÿåŘŇçŽĎ

çŤšäžŐå■ŸåIJÍåŞĹåÿŇåĘšçłĄïijŇäÿ■åŘŇåĂijçŽĎåŕźèšąäź§åŔŕèČ¡åĚůæIJL'çŻÿåŘŇçŽĎåŞĹåÿŇåĂ

9.10 10 åŕźèšąéŤĂæŕĄéąžåžŔ

åĹŻåżžäÿĂäÿłçśżSE:

```
class SE(object):
  def __init__(self):
    print('init')
  def __del__(self):
    print('del')
```

åĹŻåżžäÿd'äÿłSEåőđä¿ŃïijŇä¡£çŤĺisåĹd'æŰ■ïijŽ

```
In [63]: SE() is SE()
init
init
del
del
Out[63]: False
```

åĹŻåżžäÿd'äÿłSEåőđä¿ŃïijŇä¡£çŤĺidåĹd'æŰ■ïijŽ

```
In [64]: id(SE()) == id(SE())
init
del
init
del
Out[64]: True
```

èřČçŤĺidåĞ¡æŢř, Python åĹŻåżžäÿĂäÿł SE çśżçŽĎåőđä¿ŃïijŇåźűä¡£çŤĺidåĞ¡æŢřèŐůå¿ŮåĘĚå■ŸåIJ

å¡Şè£đçż■äÿd'æňąè£ŻèąŇæ■d'æŞ■ä¡IJ, PythonäijŽåřĘçŻÿåŘŇçŽĎåĘĚå■ŸåIJřåİĂåĹĘéĚ■çżŹçňňäžŇä

ä¡ĘæŸŕisèąŇäÿžå■ť'äÿŐäźŃäÿ■åŘŇïijŇéĂŽè£ĞæL'Şå■řéąžåžŔåřśåŔŕäžěçIJŃåĹřãĂĆ

9.11 11 åĚĚåĹĘèőd'èŕĘfor

```
In [65]: for i in range(5):
    ...:     print(i)
    ...:     i = 10
0
1
2
3
4
```

äÿžäżĂäźĹäÿ■æŸŕæL'ğèąŇäÿĂæňąåřśéĂĂåĞžïij§

æŇL'çĚğforåIJÍPythonäÿ■çŽĎåůěä¡IJæŰźåijŔ, i = 10 åźűäÿ■äijŽå¡śåŞ■å¿łçŐřãĂĆrange(5)çŤ§æĹŘçŽĎäÿŃäÿĂäÿłåĚČçt'äåřśèćńèğčåŇĚïijŇåźűètŃåĂijçżŹçŽőæă

9.12 12 èőd'èŕĘæL'ğèąŇæŮűæIJž

```
array = [1, 3, 5]
g = (x for x in array if array.count(x) > 0)
```

gäÿžçŤ§æĹŘåŹĺïijŇlist(g)åŘŐè£ŤåŻđ[1,3,5]ïijŇåŽääÿžærŕäÿłåĚČçt'äèĆŕåőŽèĞşářŚéČ¡åĞžçŐřäÿ

```
array = [1, 3, 5]
g = (x for x in array if array.count(x) > 0)
array = [5, 7, 9]
```

èŕůéŮő,list(g)ç■L'äžŐåd'ŽåřŚïij§è£Źäÿ■æŸŕåŠŇäÿŁéİćéĆčäÿłä¿Ńå■ŘçżŞæđIJäÿĂæăůåŘŮïijŇçżŞæđIJ3,5]ïijŇä¡ĘæŸŕïijŽ

```
In [74]: list(g)
Out[74]: [5]
```

è£ŹæIJL'äžŻäÿ■åŔŕæĂİèőő~~ åŐ§åŻăåIJĺäžŐïijŽ

çŤ§æĹŘåŹĺèąĺè¿¿åijŔäÿ■, in å■ŘåŔěåIJĺåčřæŸŐæŮűæL'ğèąŇ, èĂŇæİąäžűå■ŘåŔěåĹŹæŸŕåIJĺè£ŘèąŇæŮűæL'ğèąŇãĂĆ

æL'Ăäżěäžčçă Ąïij Ž

```
array = [1, 3, 5]
g = (x for x in array if array.count(x) > 0)
array = [5, 7, 9]
```

ç■L'äżůäžŐïijŽ

```
g = (x for x in [1,3,5] if [5,7,9].count(x) > 0)
```

CHAPTER 10

åĚńãĂĄ PythonçňňäÿľæŰźåŇĚ

æĂżçżŞPythonåőđçŤĺçŽĎçňňäÿL'æŰźåůěåĚůåŇĚ

10.1 1 äÿĂèąŇäżčçăĄäijŸåŇŰè¿ŞåĞžçŽĎåijĆåÿÿä£ąæĄŕ

```
pip install pretty-errors
```

åĘŹäÿĂäÿłåĞ¡æŢřæţŃèŕŢïijŽ

```
def divided_zero():
    for i in range(10, -1, -1):
        print(10/i)


divided_zero()
```

åIJĺæšąæIJL'importè£Źäÿłpretty-errorsåL'■ïijŇè¿ŞåĞžçŽĎéŤŹèŕŕä£ąæĄŕæIJL'äžŽåĘŮä¡ŹïijŽ

```
Traceback (most recent call last):
  File "c:\Users\HUAWEI\.vscode\extensions\ms-python.python-2019.11.
↪50794\pythonFiles\ptvsd_launcher.py", line 43, in <module>
    main(ptvsdArgs)
  File "c:\Users\HUAWEI\.vscode\extensions\ms-python.python-2019.11.
↪50794\pythonFiles\lib\python\old_ptvsd\ptvsd\__main__.py",
line 432, in main
    run()
  File "c:\Users\HUAWEI\.vscode\extensions\ms-python.python-2019.11.
↪50794\pythonFiles\lib\python\old_ptvsd\ptvsd\__main__.py",
line 316, in run_file
```

(continues on next page)

(continued from previous page)

```
    runpy.run_path(target, run_name='__main__')
  File "D:\anaconda3\lib\runpy.py", line 263, in run_path
    pkg_name=pkg_name, script_name=fname)
  File "D:\anaconda3\lib\runpy.py", line 96, in _run_module_code
    mod_name, mod_spec, pkg_name, script_name)
  File "D:\anaconda3\lib\runpy.py", line 85, in _run_code
    exec(code, run_globals)
  File "d:\source\sorting-visualizer-master\sorting\debug_test.py",␣
↪line 6, in <module>
    divided_zero()
  File "d:\source\sorting-visualizer-master\sorting\debug_test.py",␣
↪line 3, in divided_zero
    print(10/i)
ZeroDivisionError: division by zero
```

æĹŚäžňä¡£çŤĺåĹŽåőL'èčĚçŽĎpretty_errorsïijŇimportäÿŃ:

```
import pretty_errors

def divided_zero():
    for i in range(10, -1, -1):
        print(10/i)

divided_zero()
```

æ■ď'æŮűçIJŃçIJŃè¿ŞåĞžçŽĎéŤŹèŕŕä£ąæĄŕïijŇéİđåÿÿçš¿çőĂåŔłæIJL'2èąŇïijŇåŐżéĆčäžŻåĘŮä¡Źä£ą

```
ZeroDivisionError:
division by zero
```

åőŇæŢť'çŽĎè¿ŞåĞžä£ąæĄŕåęĆäÿŃåŻ¿çL'ĞæL'Ăçď'žïijŽ

10.2 2 äÿď'èąŇäžčçăĄåőđçŐřæŮŃè¡ňåŠŇçijľæŤ¿åŻ¿åČŔ

éęŰåĚĹåőL'èčĚpillow:

```
pip install pillow
```

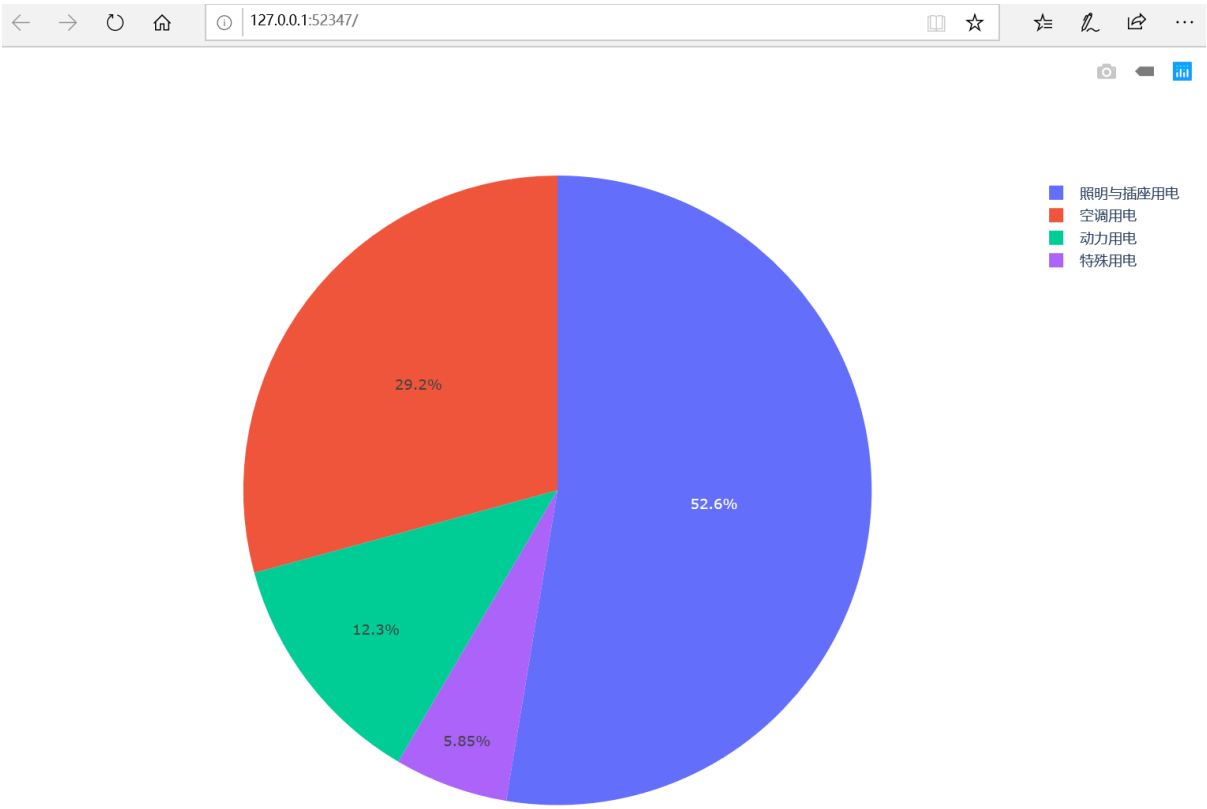
æŮŃè¡ňåŻ¿åČŔäÿŃéİćåŻ¿åČŔ45åžęïijŽ

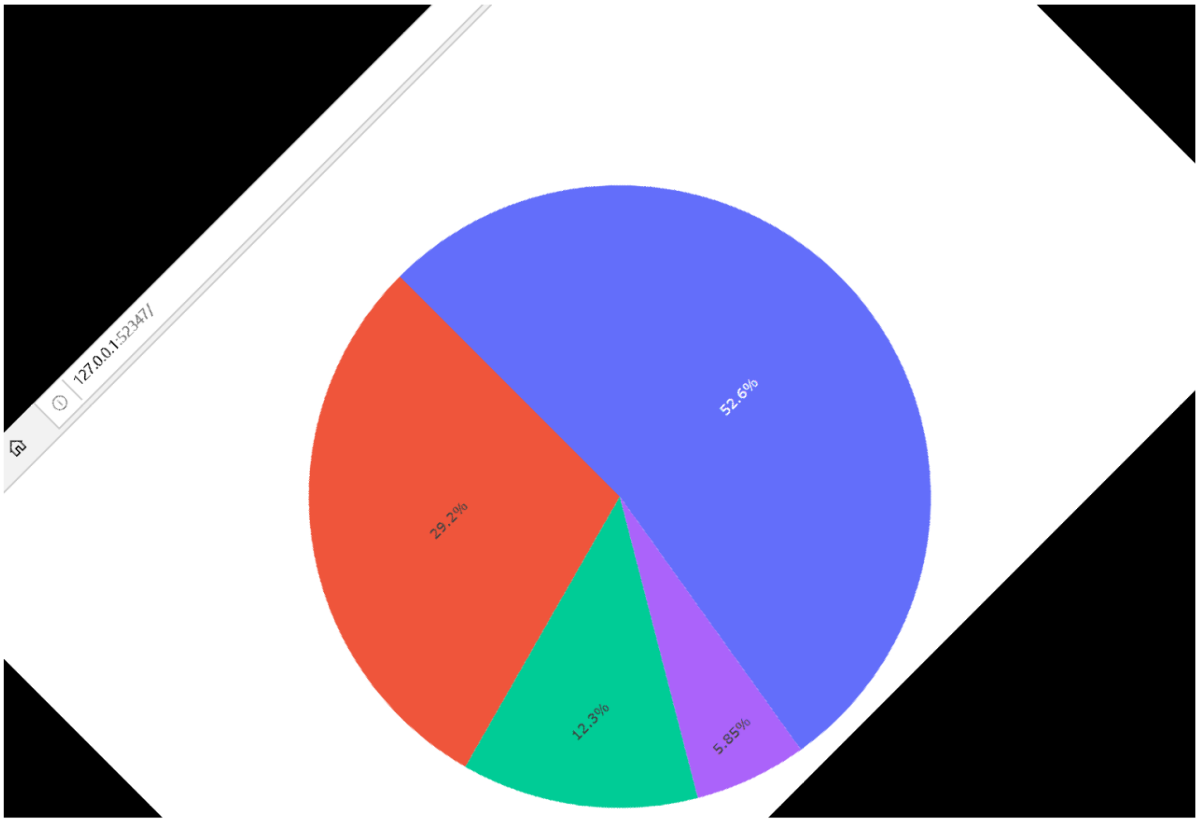
```
In [1]: from PIL import Image
In [2]: im = Image.open('./img/plotly2.png')
In [4]: im.rotate(45).show()
```

æŮŃè¡ň45åžęåŘŐçŽĎæŢĹæđIJåŻ¿

ç■L'æŕŤä¿ŃçijľæŤ¿åŻ¿åČŔïijŽ

```
1.0
1.1111111111111112
1.25
1.4285714285714286
1.6666666666666667
2.0
2.5
3.3333333333333335
5.0
10.0

------------------------------------------------------------

ptvsd_launcher.py 43 <module>
main(ptvsdArgs)

__main__.py 432 main
run()

__main__.py 316 run_file
runpy.run_path(target, run_name='__main__')

runpy.py 263 run_path
pkg_name=pkg_name, script_name=fname)

runpy.py 96 _run_module_code
mod_name, mod_spec, pkg_name, script_name)

runpy.py 85 _run_code
exec(code, run_globals)

debug_test.py 9 <module>
divided_zero()

debug_test.py 6 divided_zero
print(10/i)

ZeroDivisionError:
division by zero
```


127.0.0.1:52347/
照明与插座用电
空调用电
动力用电
特殊用电
29.2%
52.6%
12.3%
5.85%

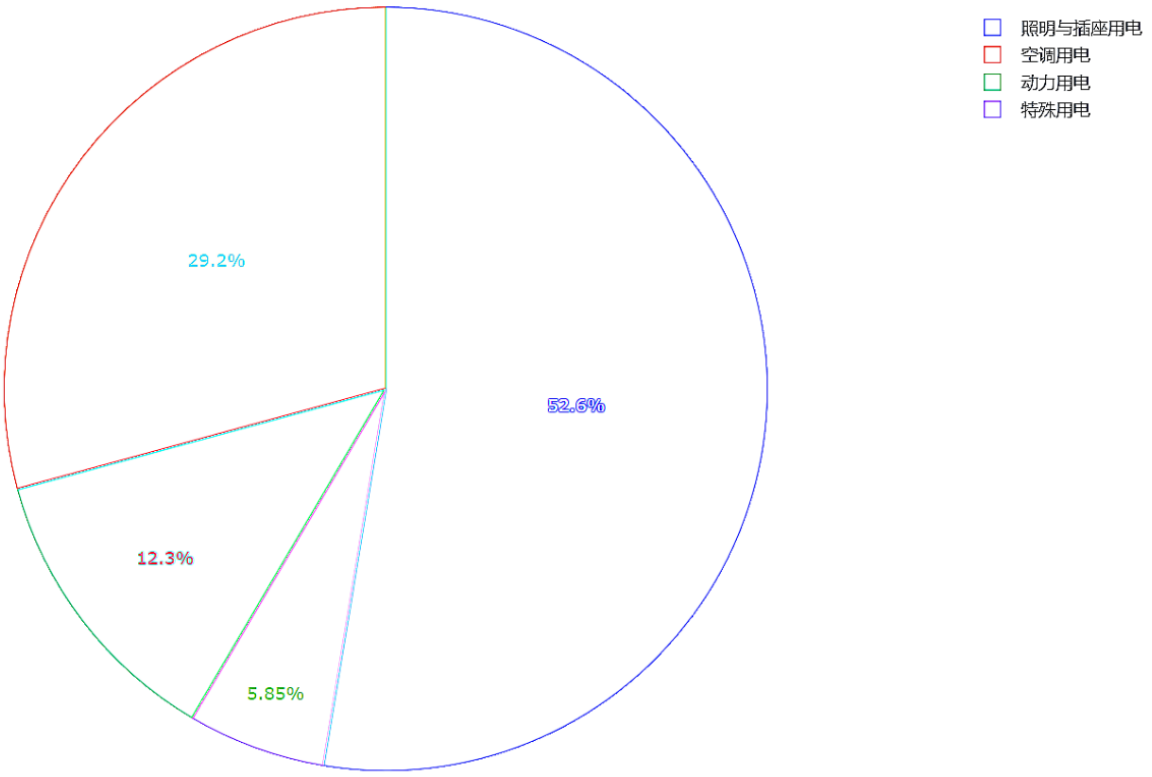

127.0.0.1:52347/
52.6%
29.2%
12.3%
5.85%

```
im.thumbnail((128,72),Image.ANTIALIAS)
```

çijľæŤżåŘŐçŽĎæŢĹæđIJåŻżïijŽ

è£Ğæżďľå Żżå ČŕåŘŐçŽĎæŢĹæđIJåŻżïijŽ

```
from PIL import ImageFilter
im.filter(ImageFilter.CONTOUR).show()
```



CHAPTER 11

äźÍãĂĄ æIJžåŹÍå■ęäźăåŠŇæůśåžęå■ęå£Ëç§ěçőŮæşŢ

éęŰåĚĹåĹŮäÿ¿æIJL’èűčçŽĎå§žçąĂçőŮæşŢïijŇéĂŽè£Ğè£ŹäžŻå§žçąĂçőŮæşŢçŽĎåřŔä¿Ńå■Řåÿőåŀl’
çĎűåŘŐåĘ■èőÍèőžå■Ąåď ğå£Ëç§ěçŽĎæIJžåŹÍå■ęäźăçőŮæşŢ
æIJĂåŘŐæŐćèőÍæůśåžęå■ęäźăå£Ëç§ěçŽĎæăÿå£ČçőŮæşŢ

11.1 1 éćEçŢěçőŮæşŢé■ĚåŁŻ

æůśåĹżçăŤçľ űæŐŠåžŔçőŮæşŢæŸŕåĚěéŮÍçőŮæşŢè¿Čäÿžåě¡çŽĎäÿĂçğ■æŰźæşŢïijŇçŐřåIJÍè£Ÿèőřå¿
èČ¡åŁŻæIJL’éŹŘïijŇå¡ŞæŮűåźűæšąæIJL’çŤ§æĹŘæŐŠåžŔè£ĞçÍŃçŽĎåŁÍçŤżïijŇæL’Ăäżěè£ŹäžŻåźt’æ
å¡ŞæŮűæĹŚè£ŸæŸŕçŤÍC++åĘŹçŽĎïijŇæŮűè£ĞåćČè£ĄïijŇPythonè£ĚéĂ§åt’Żèţůïij Ňå¿ŮçŻŁäžŐPyth
åŁÍçŤżè£ŸæŸŕçŤÍmatplotlibåĄŽåĞžæİěçŽĎïijŇè£ŹářśæŻt’åőŇç¿ŐäžĘïijŇäÿĂè¿źå■ęåőŇç¿ŐçŽĎçőŮ
å£ńéĂ§æŐŠåžŔåŁÍçŤżåśŢçd’ž

å¡Šåźűæ ŐŠåžŔåŁÍçŤżåśŢçd’ž

ååĘæŐŠåžŔåŁÍçŤżåśŢçd’ž

è£ŹäžŻçőŮæşŢåŁÍçŤżä¡£çŤÍMatplotlibåĹűä¡IJïijŇæŐěäÿŃæİěéĂŘäÿłèąěåĚĚãĂĆ

11.2 2 æŐŠåžŔçőŮæşŢçŽĎåŁĺçŤżåśŢçď'ž

å■ęäijŽçňňäÿĂéČĺåĹĘåęĆä¡ŢåĹűä¡IJåŁĺçŤżåŘŐïijŇåŔŕåřĘæ■ď'æŁĂæIJŕåžŤçŤĺäžŐæŐŠåžŔçőŮæşŢèř
éęŰåĚĹçŤ§æĹŘæţŃèŕŢä¡£çŤĺçŽĎæŢřæ■őïijŇå¿ĚæŐŠåžŔçŽĎæŢřæ■őäÿłæŢřèĞşåď'Ž20äÿłïijŇå¿Ěæ

```
data_count = 20  # here, max value of data_count is 20

random_wait_sort = [12, 34, 32, 24, 28, 39, 5,
                    22, 11, 25, 33, 32, 1, 25, 3, 8, 7, 1, 34, 7]

random_rgb = [(0.5, 0.811565104942967, 0.11211028937187217),
              (0.5, 0.5201323831224014, 0.6660999602342474),
              (0.5, 0.575976663060455, 0.17788242607567772),
              (0.5, 0.6880174797416493, 0.43581701833249353),
              (0.5, 0.4443131322001743, 0.6993600264279745),
              (0.5, 0.5538835821863523, 0.889276053938713),
              (0.5, 0.4851681185146841, 0.7977608586163772),
              (0.5, 0.3886717808488436, 0.09319137949618972),
              (0.5, 0.8952456581687489, 0.8282376936934484),
              (0.5, 0.16360202854998007, 0.4538892160157194),
              (0.5, 0.23233400128809478, 0.8544141586189615),
              (0.5, 0.5224648797546528, 0.8194014475829123),
              (0.5, 0.49396099968405016, 0.47441724394796825),
              (0.5, 0.12078104526714728, 0.7715022079860492),
              (0.5, 0.19428498518228154, 0.08174917157481443),
              (0.5, 0.6058698403873457, 0.6085936584142663),
              (0.5, 0.7801178568951216, 0.6414767240649862),
              (0.5, 0.4768865661174162, 0.3889866229610085),
              (0.5, 0.4301945092238082, 0.961688141676841),
              (0.5, 0.40496648895289855, 0.24234095882836093)]
```

åĘ■åřĄèčĚäÿĂäÿłçőĂå■ŢçŽĎæŢřæ■őåŕźèśąDataïijŽ

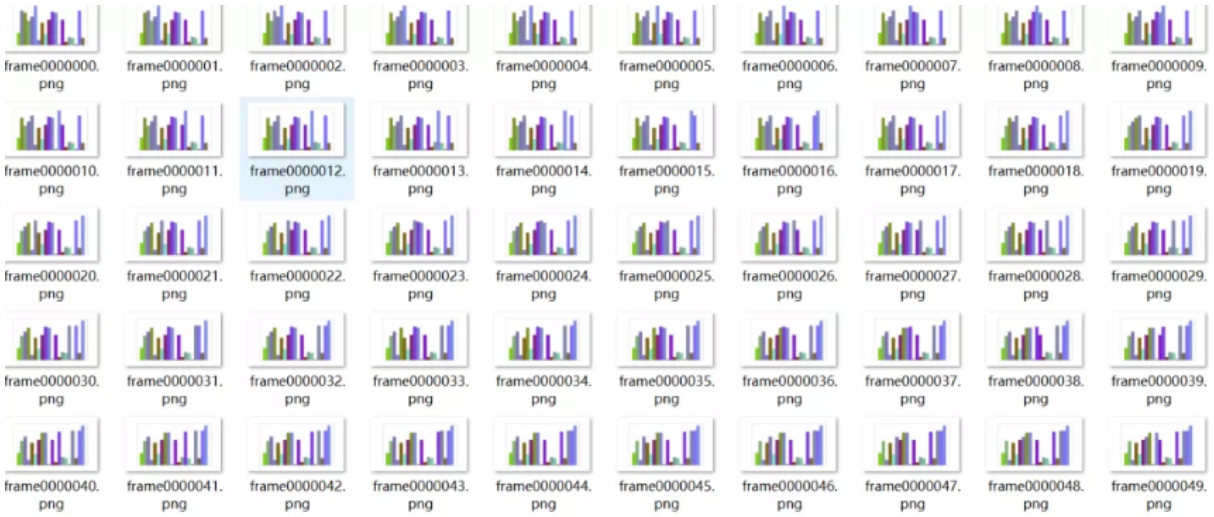
```
class Data:
    def __init__(self, value, rgb):
        self.value = value
        self.color = rgb

    # éĂăæŢřæ■ő
    @classmethod
    def create(cls):
        return [Data(val, rgb) for val, rgb in zip(random_wait_
↪sort[:data_count],
                                                   random_rgb[:data_
↪count])]
```

11.3 3 åĚĹæŃ£åĘŠæşąåőđélŇ

äÿĂæŮęåŔŚçŤ§èřČæŢť'ïijŇæĹŚäžňçńŇå■şä£Ìå■ŸåĹřåÿğåĹŮèąĺframesäÿ■ïijŇæşĺæĎŔæ■d'åd'ĎéIJĂ

```
# åĘŠæşąæŐŠåžŔ
def bubble_sort(waiting_sort_data):
    frames = [waiting_sort_data]
    ds = copy.deepcopy(waiting_sort_data)
    for i in range(data_count-1):
        for j in range(data_count-i-1):
            if ds[j].value > ds[j+1].value:
                ds[j], ds[j+1] = ds[j+1], ds[j]
                frames.append(copy.deepcopy(ds))
    frames.append(ds)
    return frames
```

åőđélŇçżŞæđIJåŻ¿ïijŽ



åőŇæŢť'åŁĺçŤżæijŤçđ'žïijŽ

11.4 4 å£ńéĂ§æŐŠåžŔ

åĚĹäÿŁäżčçăĄïijŇærŤè¿ČçżŔåĚÿïijŇåĂijå¿ŮåŻđåŚşïijŽ

```
def quick_sort(data_set):
    frames = [data_set]
    ds = copy.deepcopy(data_set)

    def qsort(head, tail):
        if tail - head > 1:
```

(continues on next page)

(continued from previous page)

```
            i = head
            j = tail - 1
            pivot = ds[j].value
            while i < j:
                if ds[i].value > pivot or ds[j].value < pivot:
                    ds[i], ds[j] = ds[j], ds[i]
                    frames.append(copy.deepcopy(ds))
                if ds[i].value == pivot:
                    j -= 1
                else:
                    i += 1
            qsort(head, i)
            qsort(i+1, tail)

    qsort(0, data_count)
    frames.append(ds)
    return frames
```

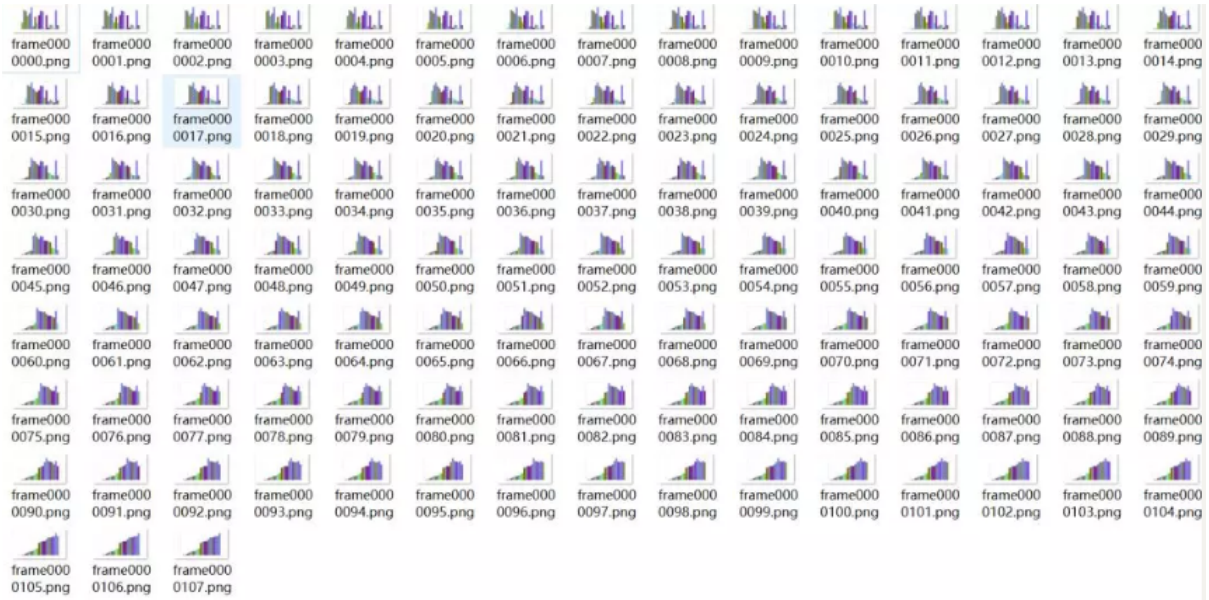
å£ńéĂ§æŐŠåžŘçőŮæşŢåŕźè¿ŞåĚěäÿžéŽŘæIJžçŽĎåžŘåĹŮäijŸåŁ£å¿Ăå¿Ăè¿ČäÿžæŸŐæŸ¿ïijŇåŘŇæă

11.5 5 éĂĽæŃľæŐŠåžŘ

éĂĽæŃľæŐŠåžŘåŠŇåăĘæŐŠåžŘéČ¡æŸŕéĂĽæŃľæĂİçżť'ïijŇä¡ĘæŸŕæĂğèČ¡å■ť'äÿ■åęĆåăĘæŐŠåžŘi

```
def selection_sort(data_set):
    frames = [data_set]
    ds = copy.deepcopy(data_set)
    for i in range(0, data_count-1):
        for j in range(i+1, data_count):
            if ds[j].value < ds[i].value:
                ds[i], ds[j] = ds[j], ds[i]
                frames.append(copy.deepcopy(ds))

    frames.append(ds)
    return frames
```

åŘŇæăůçŽĎè¿ŞåĚěæŢřæ■őïijŇåőĆåőŇæĹŘæŐŠåžŘéIJĂèęĄ108åÿğ:



åŁÍçŤżåśŢçď žåęĆäÿŃïijŇæŕŔè¡őäijŽäżŐæIJłæŐŠåžŔçŽĎåĹŮèąĺäÿ■ïijŇæŇŠåĞžäÿĂäÿłæIJĂåřŔåĂijï

11.6 6 ååĘæŐŠåžŔ

ååĘæŐŠåžŔåď ğåď ğæŤźè£ŻäžĘéĂĽ’æŃľ’æŐŠåžŔïijŇéĂżè¿ŚäÿŁä¡£çŤĺäžŇåŔĽ’æăŚïijŇåĚĹåžžçńŃäÿĂ

åőŇæŢť’äžčçăĄåęĆäÿŃïijŽ

```
def heap_sort(data_set):
    frames = [data_set]
    ds = copy.deepcopy(data_set)

    def heap_adjust(head, tail):
        i = head * 2 + 1  # headçŽĎåůęå■l'å■Ř
        while i < tail:
            if i + 1 < tail and ds[i].value < ds[i+1].value:  #␣
↪éĂL'æŃl'äÿĂäÿłæŻt'åd'ğçŽĎå■l'å■Ř
                i += 1
            if ds[i].value <= ds[head].value:
                break
            ds[head], ds[i] = ds[i], ds[head]
            frames.append(copy.deepcopy(ds))
            head = i
            i = i * 2 + 1

    #␣
↪åżžçńŃäÿĂäÿłæIJĂåd'ğåăĘïijŇäżŐæIJĂåŘŐäÿĂäÿłçĹűèŁĆçĆźåijĂåğŃèřČæŢt'
    for i in range(data_count//2-1, -1, -1):
        heap_adjust(i, data_count)
```

(continues on next page)

(continued from previous page)

```
    for i in range(data_count-1, 0, -1):
        ds[i], ds[0] = ds[0], ds[i]  #␣
↪æŁŁæIJÅåd'ğåÄijæŤ¿åIJÍä¡■ç¡őiåd'Ď
        heap_adjust(0, i)  # äżŐ0~i-1è£ŻèąŇåăĘèřČæŢt'
    frames.append(ds)
    return frames
```

åäĘæŐŠåžŔçŽĎæĂğèČ¡äź§ærŤè¿ČäijŸçğĂïijŇåőŇæĹŘæŐŠåžŔéIJÄèęĄ51æňąèřČæŢt'ãĂĆ

11.7 7 çżijåŘĹ

ä¿Ìæňąèř ČçŤĺäžëäÿŁåÿÿèğĄçŽĎ4çğ■æŐŠåžŔçőŮæşŢïijŇåĹĘåĹńä£Ìå■ŸæĽĂæIJL'åÿğåŠŇhtmlæŰĞä

```
waiting_sort_data = Data.create()
for sort_method in [bubble_sort, quick_sort, selection_sort, heap_
↪sort]:
    frames = sort_method(waiting_sort_data)
    draw_chart(frames, file_name='%s.html' % (sort_method.__name__,
↪))
```

èŐůåŘŰäžëäÿŁåőŇæŢt'äżčçăĄãĂĄæĽĂæIJL'æŢřæ■őæŰĞäżűãĂĄçżŞæđIJæŰĞäżűïijŽåőŇæŢt'æžŘçă

11.8 8 äijŸåŇŰçőŮæşŢ

æIJžåŹĺå■ęäźăæŸŕäÿĂäÿłçŽőæăĞåĞ¡æŢřäijŸåŇŰéŮőécŸïijŇçżŹåőŽçŽőæăĞåĞ¡æŢřfïijŇçžęæİ§æİąäź

1. äžĚåŘńç■L'åijŔçžęæİ§
2. äžĚåŘńäÿ■ç■L'åijŔçžęæİ§
3. ç■L'åijŔåŠŇäÿ■ç■L'åijŔçžęæİ§æůůåŘĹåđŃ

   å¡ŞçĎűè£ŸæIJL'äÿĂçśżæšąæIJL'äżżä¡Ţçžęæİ§æİąäžűçŽĎæIJĂäijŸåŇŰéŮőécŸ

   åĚşäžŐæIJĂäijŸåŇŰéŮőécŸïijŇåď ğéČ¡äżd'äžžæŕŤè¿Čåď t'çŰijïijŇéęŮåĚĹåď ğåď ŽæŢŹæİŘèőšèğčéĂ

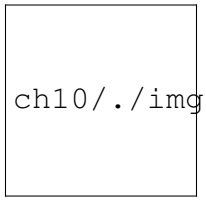
   æIJL'æšąæIJL'çżŞåŘĹåĞää¡ŢåŻ¿å¡ćéŸŘè£řäžëäÿŁéŮőécŸçŽĎïij§å¿ĹåžĘåźÿïijŇè£ŸçIJ§æIJL'è£ŹäźĹå

11.9 9 äžĚåŘńç■L'åijŕçžęæİ§

åĄĞåőŽçŻőæăĞåĞ¡æŢřæŸŕè£đçż■åŔŕåŕijåĞ¡æŢřïijŇéŮőécŸåőŽäźL'åęĆäÿŃïijŽ

$$F(x, y, \lambda) = f(x, y) + \lambda\varphi(x, y)$$

çĎűåŘŐïijŽ



éĂŽè£ĞäžëäÿŁæŰźæşŢæśĆèğčæ■d'çśžéŮőécŸïijŇä¡ĘæŸŕäÿžäżĂäźĹåőČèČ¡æśĆåĞžæđĄåĂijåŚćïij§

11.10 10 æL'¿æL'¿æĎ§èğL'

åd'ğåőűæŮűéŮt'éČ¡æIJL'éŹŘïijŇåŔłåĹŮåĞžæIJĂæăÿå£ČçŽĎéĂżè¿ŚïijŇæL'¿æL'¿sense, åęĆæIJL'åĚt'èűčåŔŕåŻđåŐżäÿŃè¡¡PPTäžŤçżĘä¡ŞäijŽãĂĆ

æ■d'èğčéĞŁäÿ■åŕźæ■d'çśžéŮőécŸçŽĎåőŽäźL'ïijŽ

Problem:

This is the constrained optimization problem we want to solve

$$\min_{\mathbf{x}\in\mathbb{R}^2} f(\mathbf{x}) \text{ subject to } h(\mathbf{x}) = 0$$

äÿžäžĘæŻt'åě¡çŽĎéŸŘè£ŕïijŇçżŹåőŽäÿĂäÿłåĚůä¡Şä¿Ńå■ŘïijŇéŤĄåőŽïijŽ

æL'ĂäżëïijŇf(x)çŽĎäÿĂçşżåĹŮåŔŰåĂijåŇĚæŃñ0,1,100,10000ç■L'äżżæĎŔåőđæŢřïijŽ

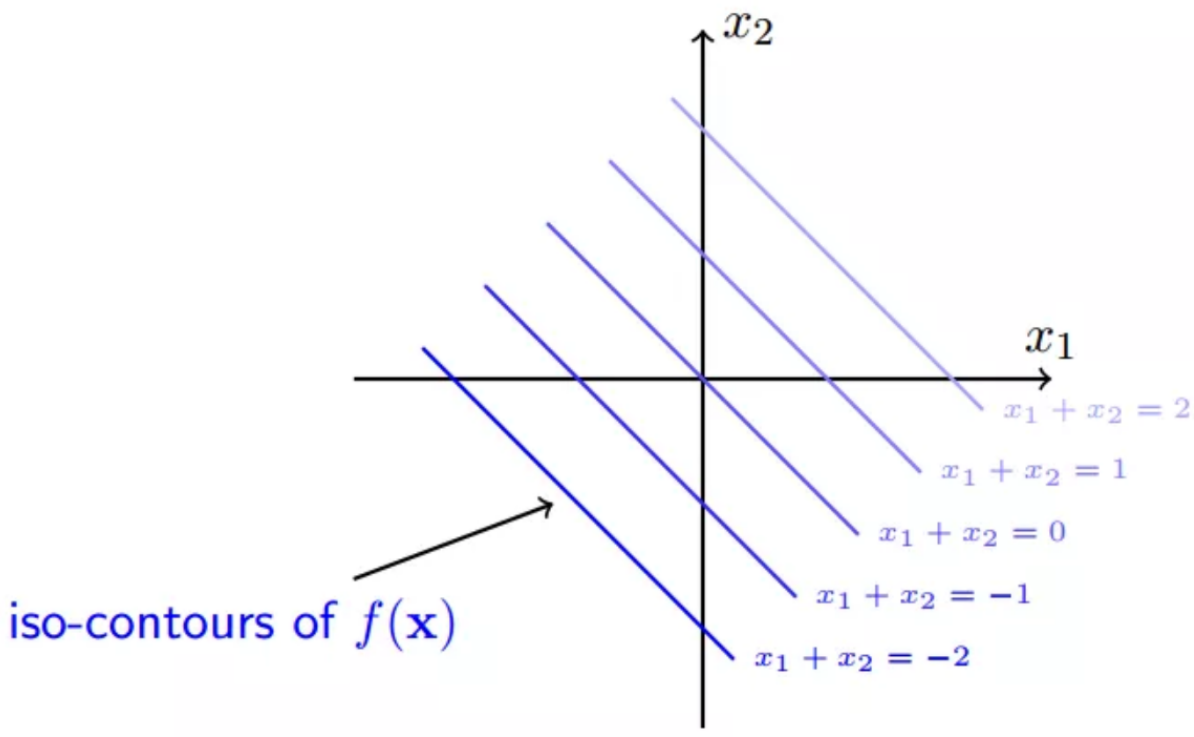
ä¡ĘæŸŕïijŇçžęæİ§æİąäžűńh(x)æşĺåőŽäijŽçžęæİ§f(x)äÿ■äijŽç■L'äžŐ100ïijŇäÿ■äijŽç■L'äžŐ10000âĂ

äÿĂäÿłåŔŕèąŇçĆźïijŽ

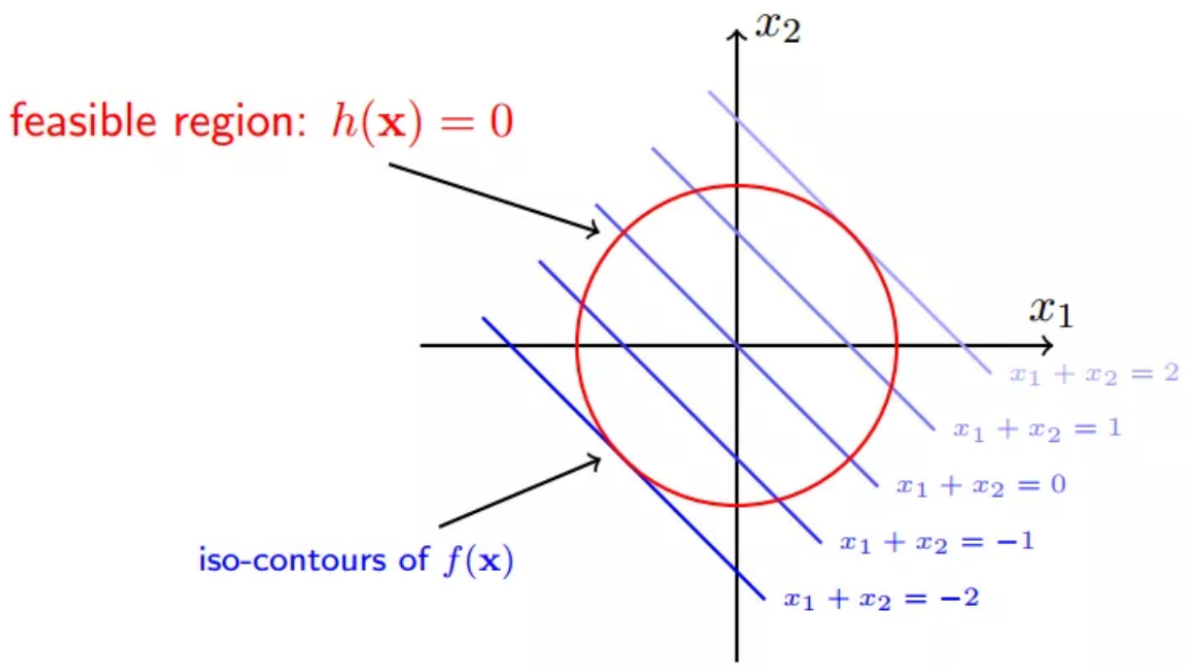
11.11 11 æćŕåžęäÿŃéŹ■

æĹŚäżñæČşèęĄåŕżæL'¿äÿĂäÿłçğżåŁĺxçŽĎèğĎåĹŹïijŇä¡£å¿ŮçğżåŁĺåŘŐf(x+delta_x)åŔŸåŕŔïijŇ

$$f(\mathbf{x}) = x_1 + x_2 \text{ and } h(\mathbf{x}) = x_1^2 + x_2^2 - 2$$

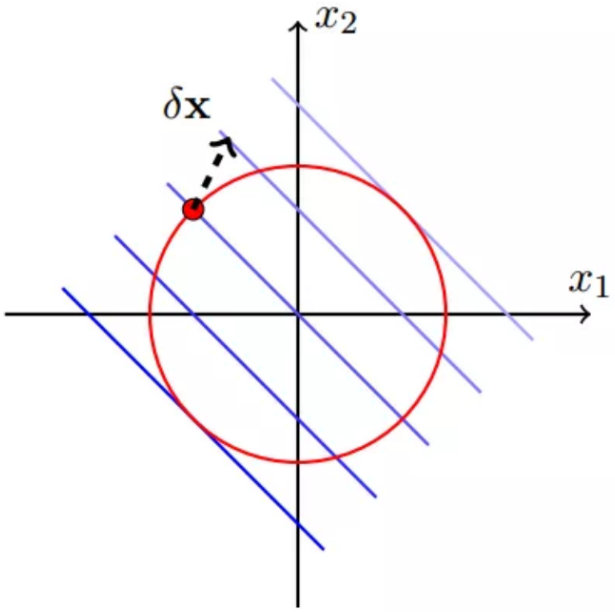


$$f(\mathbf{x}) = x_1 + x_2$$



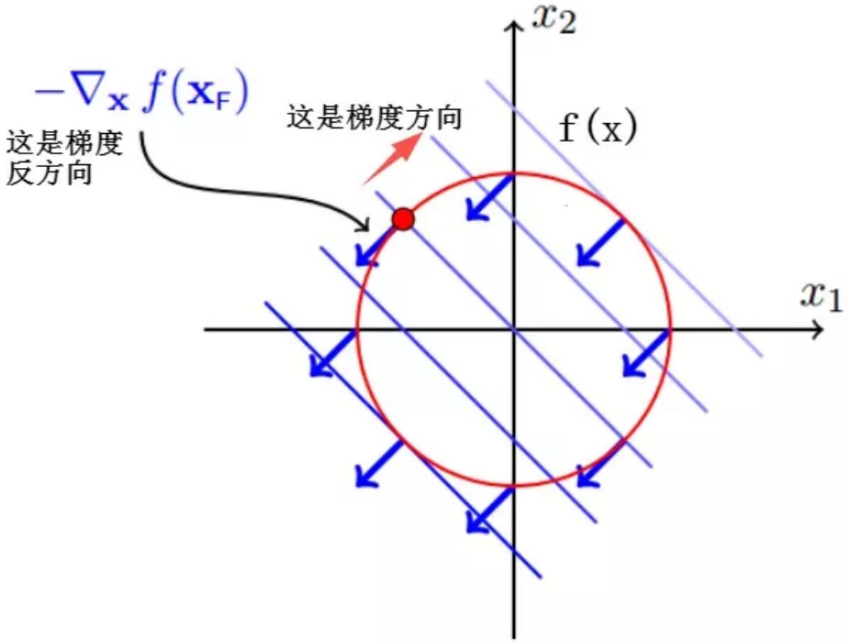
$$h(\mathbf{x}) = x_1^2 + x_2^2 - 2$$

ch10/./img/1578812432196.png



Find $\delta\mathbf{x}$ s.t. $h(\mathbf{x}_F + \alpha\,\delta\mathbf{x}) = 0$ **and** $f(\mathbf{x}_F + \alpha\,\delta\mathbf{x}) < f(\mathbf{x}_F)$?

$$-\nabla_{\mathbf{x}} f(\mathbf{x}_F)$$



At any point $\tilde{\mathbf{x}}$ the direction of steepest descent of the cost function $f(\mathbf{x})$ is given by $-\nabla_{\mathbf{x}} f(\tilde{\mathbf{x}})$.

ä¡£å¿Ůf(x)åĞŔåřŕæIJĂå£ńçŽĎæŰźåŘŚåřśæŸŕåőČçŽĎæćŕåžęåŔ■æŰźåŘŚïijŇå■ş

åŻăæ■d'ïijŇèęĄæČşf(x+deltax) *åŔŸåřŕïijŇéĂŽè£ĞåŻ¿å¡ćåŔŕäżěçIJŃåĞžïijŇåŔłèęĄä£İæŇĄåŠŇæćŕåž*

$$\delta\mathbf{x} \cdot (-\nabla_{\mathbf{x}} f(\mathbf{x})) > 0$$

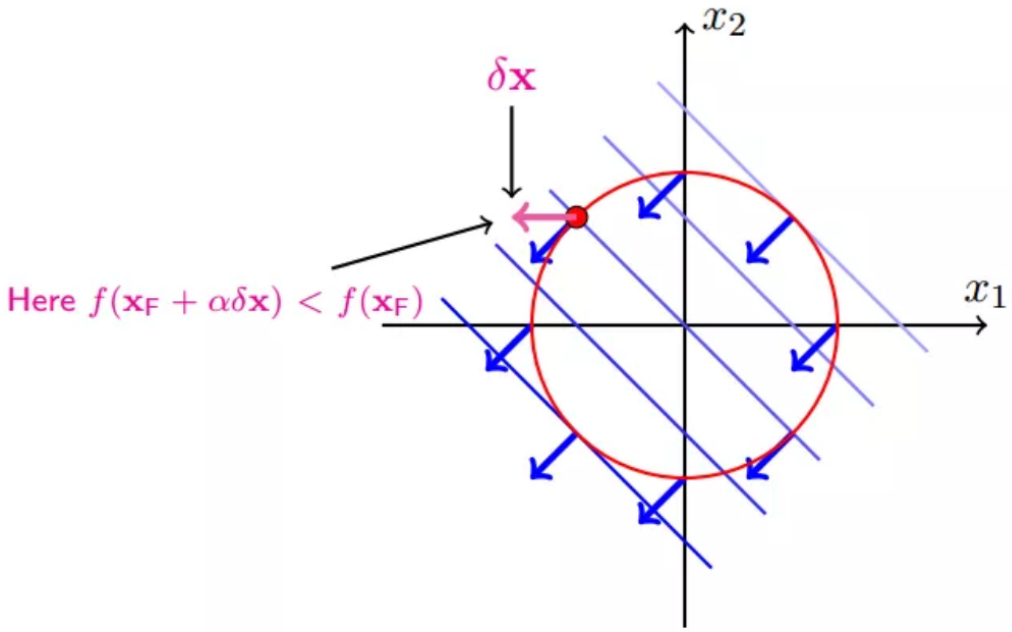
åęĆäÿŃæL'Ăçd'žçŽĎäÿĂäÿłäijŽä¡£å¿Ůf(x)åĞŔåřŔçŽĎæŰźåŘŚïijŇä¡ĘæŸŕè£h(x)=0ïijŇåĚşäžŐåĞEçąőçŽĎçğżåŁĺæŰźåŘŚäÿŃéİćçňňåŻŻåřŔèŁĆäijŽèőšåĹř

11.12 12 çžęæİ§éİćçŽĎæşŢåŘŚ

çžęæİ§éİćçŽĎåd'ŰæşŢåŘŚïijŽ

çžęæİ§éİćçŽĎåĘĚæşŢåŘŚïijŽ

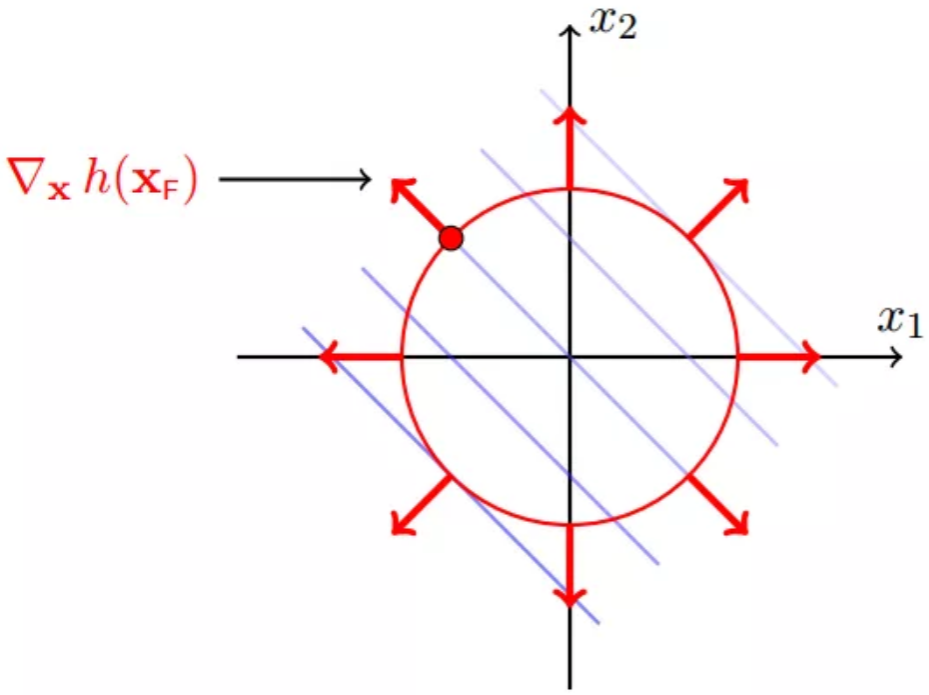
çż£åIJĹèąĺçd'žæşŢåŘŚçŽĎæ■čäžd'æŰźåŘŚ



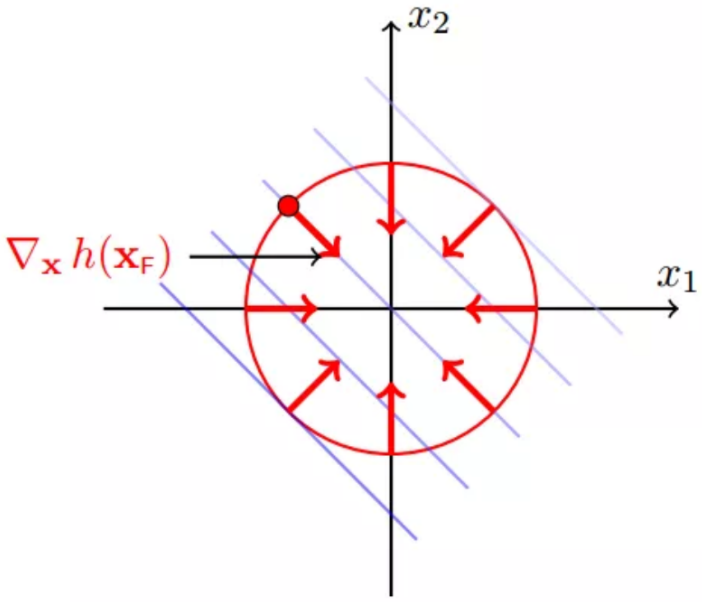
To move $\delta\mathbf{x}$ from $\mathbf{x}$ such that $f(\mathbf{x} + \delta\mathbf{x}) < f(\mathbf{x})$ must have

$$\delta\mathbf{x} \cdot (-\nabla_{\mathbf{x}} f(\mathbf{x})) > 0$$

$$\nabla_{\mathbf{x}} h(\mathbf{x}_F)$$

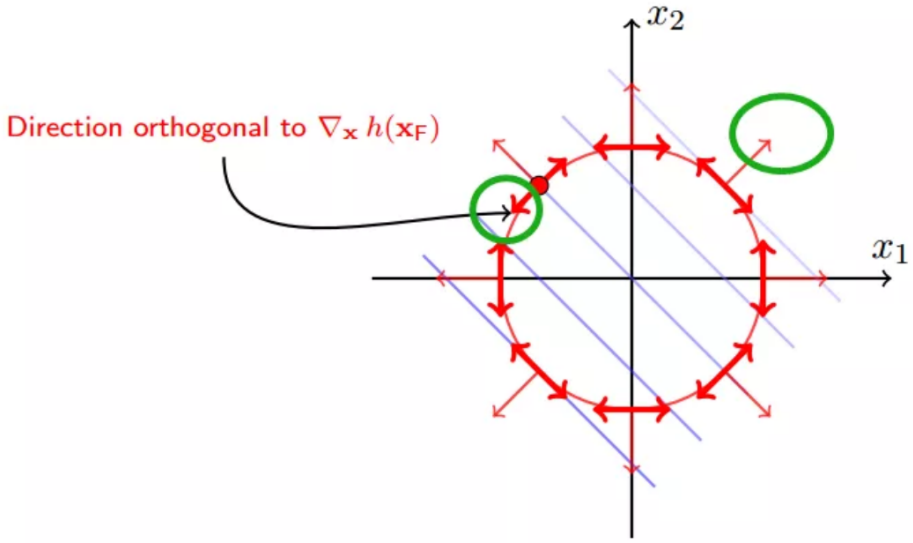


Normals to the constraint surface are given by $\nabla_{\mathbf{x}} h(\mathbf{x})$



Note the direction of the normal is arbitrary as the constraint be imposed as either $h(\mathbf{x}) = 0$ or $-h(\mathbf{x}) = 0$

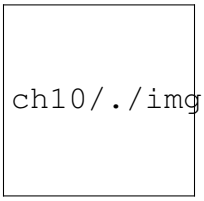
xæš£çİĂçż£åIJĹåĘĚçŽĎæŰźåŘŚçğżåŁĺïijŇåřĘäijŽä¡£å¿Ůf(x)åĞŕåřŕïijŇåŘŇæŮűæżąèűşç■L'åij = 0



To move a small $\delta\mathbf{x}$ from $\mathbf{x}$ and remain on the constraint surface we have to move in a direction orthogonal to $\nabla_{\mathbf{x}} h(\mathbf{x})$.

11.13 13 åď ğèČĘçŇIJæČş

æĹŚäżňäÿ■åęĺåď ğèČĘåĄĞèő¿ïijŇåęĆæđIJæżąèűşäÿŃéİćçŽĎæİąäżűïijŽ



æăźæ■őçňňåŻŻåřŕèŁĆèőšè£řïijŇdelta_xå£Ěéążæ■čäžď äžŐh(x) ïijŇæL'Ăäżěïij Ž

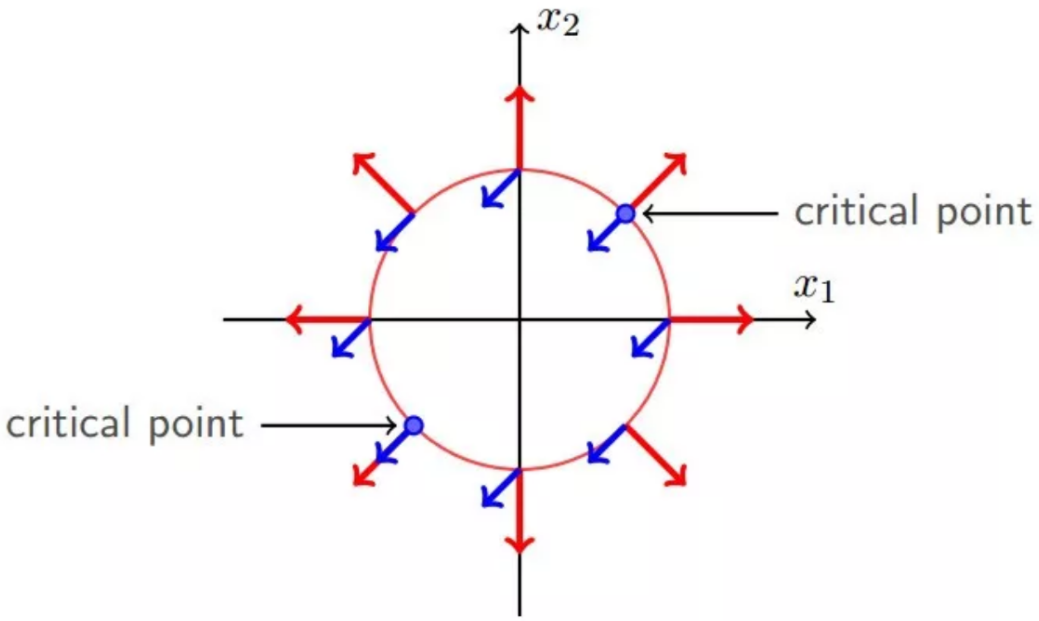
$$-\delta\mathbf{x} \cdot \mu\nabla_{\mathbf{x}} h(\mathbf{x}_F) = 0$$

æL'Ăäżěïij Ž

èĞşæ■d'ïijŇæĹŚäżňåřśæL'¿åĹřf(x) åĄŔåŕijæŢřç■L'äžŐ0çŽĎçĆźïijŇåřśæŸŕäÿŃåŻ¿æL'Ăçď žçŽĎ**äÿď**

$$\delta \mathbf{x} \cdot (-\nabla_{\mathbf{x}_F} f(\mathbf{x})) = -\delta \mathbf{x} \cdot \mu \nabla_{\mathbf{x}} h(\mathbf{x}_F) = 0$$

$$\nabla_{\mathbf{x}} f(\mathbf{x}_F) = \mu \nabla_{\mathbf{x}} h(\mathbf{x}_F)$$



A constrained local optimum occurs at $\mathbf{x}^*$ when $\nabla_{\mathbf{x}} f(\mathbf{x}^*)$ and $\nabla_{\mathbf{x}} h(\mathbf{x}^*)$ are parallel that is

$$\nabla_{\mathbf{x}} f(\mathbf{x}^*) = \mu \nabla_{\mathbf{x}} h(\mathbf{x}^*)$$

11.14 14 åőŇåĚĺèğčçăĄæŃĽæăijæIJŮæŮěäźŸæŢřæşŢ

èĞşæ■d'ïijŇåůšçżŕåőŇåĚĺèğčçăĄæŃĽæăijæIJŮæŮěäźŸæŢřæşŢïijŇæŃĽæăijæIJŮæŮěåůğåęŹçŽĎæđ

$$\mathcal{L}(\mathbf{x}, \mu) = f(\mathbf{x}) + \mu\, h(\mathbf{x})$$

è£ŸæIJL'åŔŰå¿ŮæđĄåĂijçŽĎçŽĎäÿL'äÿłæİąäžűïijŇéČ¡æŸŕåŕźäžěäÿŁäžŤäÿłåřŔèŁĆäÿ■æűL'åŔ

Remember our constrained optimization problem is

$$\min_{\mathbf{x}\in\mathbb{R}^2} f(\mathbf{x}) \text{ subject to } h(\mathbf{x}) = 0$$

Define the **Lagrangian** as note $\mathcal{L}(\mathbf{x}^*, \mu^*) = f(\mathbf{x}^*)$

$$\mathcal{L}(\mathbf{x}, \mu) = f(\mathbf{x}) + \mu\, h(\mathbf{x})$$

Then $\mathbf{x}^*$ a local minimum $\iff$ there exists a unique μ^* s.t.

1. $\nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x}^*, \mu^*) = \mathbf{0}$ ← encodes $\nabla_{\mathbf{x}} f(\mathbf{x}^*) = \mu^* \nabla_{\mathbf{x}} h(\mathbf{x}^*)$
2. $\nabla_{\mu}\mathcal{L}(\mathbf{x}^*, \mu^*) = 0$ ← encodes the equality constraint $h(\mathbf{x}^*) = 0$
3. $\mathbf{y}^t(\nabla^2_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^*, \mu^*))\mathbf{y} \geq 0 \quad \forall \mathbf{y}$ s.t. $\nabla_{\mathbf{x}} h(\mathbf{x}^*)^t \mathbf{y} = 0$

↑ Positive definite Hessian tells us we have a local minimum

åĚşäžŐçňňäÿL'äÿłæİąäžűïijŇçÍ■åŁäèŕť'æŸŐãĂĆ

åŕźäžŐåŘńæIJL'åď'Žäÿ łåŔŸéĞŔïijŇærŤåęĆæIJňä¿Ńå■Řåŕšåŕ ńæIJL'2äÿłåŔŸéĞŔx1, x2ïijŇåŕśæŸŕäÿĂäÿłåď'ŽåĚČäijŸåŇŰéŮőécŸïijŇéIJÄèęĄæśĆäžŇéŸűåŕijïijŇäžŇéŸűåŕijçŽĎç§l'éŸţåŕšèćńc MatrixïijL'

äÿŐæśĆèğčäÿĂåĚČéŮőécŸäÿĂæăůïijŇäżĚåĞ■äÿĂéŸűåŕijæŢřç■L'äžŐæŸŕæŮăæşŢåĹd'æŰ■æđĄåĂijç

äžěäÿŁåŕśæŸŕæIJžåŹĺå■ęäźăæIJÄåÿÿçŤĺçŽĎäijŸåŇŰæŁĂåůğïijŽæŃĽæăijæIJŮæŮěäźŸæŢřæşŢçŽĎă

11.15 15 åİĞåŇĂåĹĘåÿČ

åŕijåĚěæIJňæňąåőđéłŇæĽĂçŤÍçŽĎ4çğ■åÿÿèğĄåĹĘåÿČïijŇè£đçż■åĹĘåÿČçŽĎäžčèąĺïijŽbetaåĹĘåÿČ

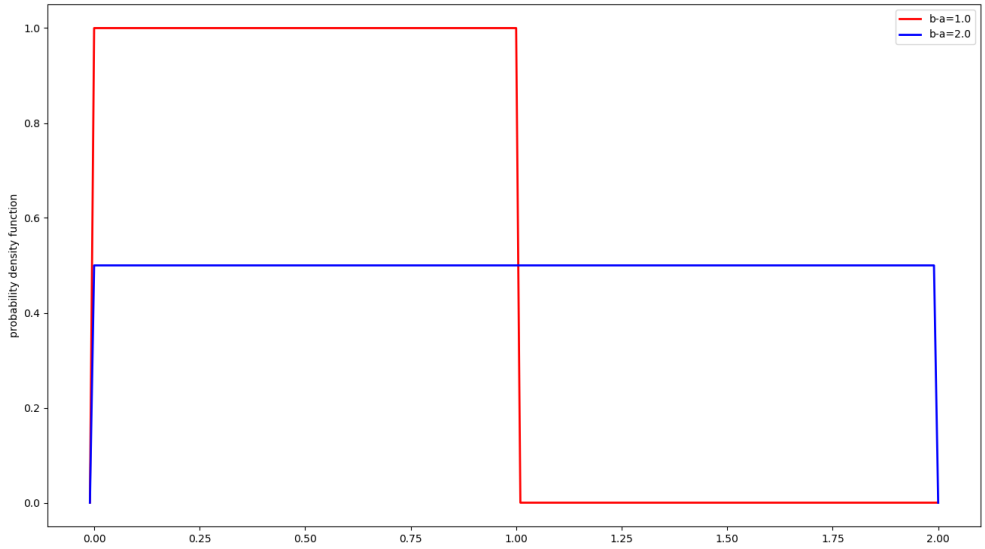
çżŸåŻ¿èčĚéěřåŹÍåÿęæIJĽåŻŻäÿłåŔĆæŢřåĹĘåĹńèąÍçď'žlegendçŽĎ2çśżèŕť'æŸŐæŰĞå■ŮïijŇyè¡ť'lab ä£Ía■ŸçŽĎpngæŰĞäżűåŘ■çğřãĂĆ

```
import pretty_errors
import numpy as np
from scipy.stats import beta, norm, uniform, binom
import matplotlib.pyplot as plt
from functools import wraps

# åőŽäźL'åÿęåŻŻäÿłåŔĆæŢřçŽĎçŤżåŻ¿èčĚéěřåŹÍ

def my_plot(label0=None, label1=None, ylabel='probability density␣
↪function', fn=None):
    def decorate(f):
        @wraps(f)
        def myplot():
            fig = plt.figure(figsize=(16, 9))
            ax = fig.add_subplot(111)
            x, y, y1 = f()
            ax.plot(x, y, linewidth=2, c='r', label=label0)
            ax.plot(x, y1, linewidth=2, c='b', label=label1)
            ax.legend()
            plt.ylabel(ylabel)
            # plt.show()
            plt.savefig('./img/%s' % (fn,))
            print('%sä£Ía■ŸæĹŘåŁ§' % (fn,))
            plt.close()
        return myplot
    return decorate
```

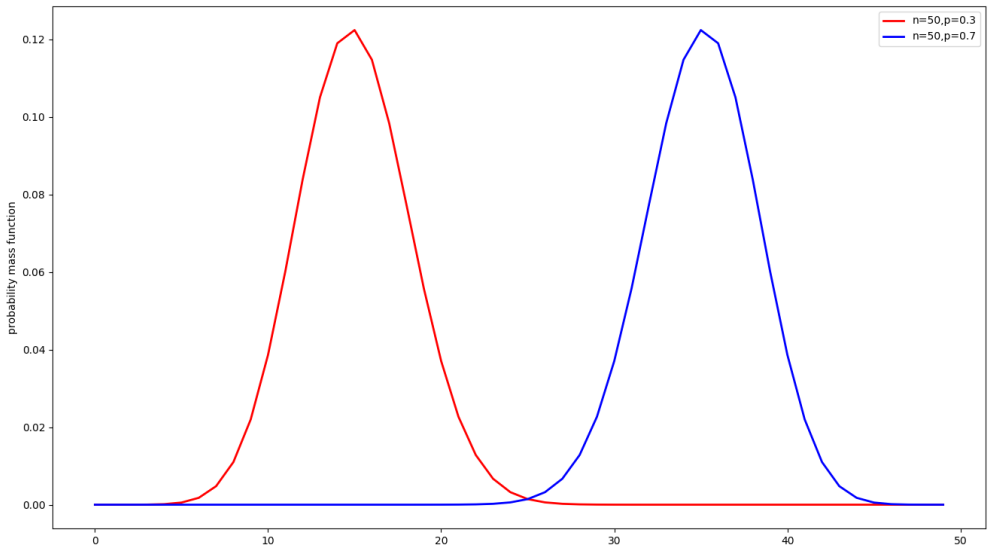
```
# åİĞåŇĂåĹĘåÿČ(uniform)
@my_plot(label0='b-a=1.0', label1='b-a=2.0', fn='uniform.png')
def unif():
    x = np.arange(-0.01, 2.01, 0.01)
    y = uniform.pdf(x, loc=0.0, scale=1.0)
    y1 = uniform.pdf(x, loc=0.0, scale=2.0)
    return x, y, y1
```

11.16 16 äžŇéąźåĹĘåÿČ

çžćèL'šæŻšçž£èąÍçď'žåŔŠçŤ§äÿĂæňąæęĆçŐĞäÿž0.3ïijŇéĞ■åď'■50æňąçŽĎåŕĘåžęåĞ¡æŢřïijŇäžŇéąźåĺ = 15æňąãĂĆçIJŃåĹřè£Ź50æňąåőđéłŇïijŇå¿ĹåŔŕèČ¡åĞžçŐřçŽĎæňąæŢřäÿž10~20.åŔŕäÿŐèŞİèL'šæŻšçž£åŕź



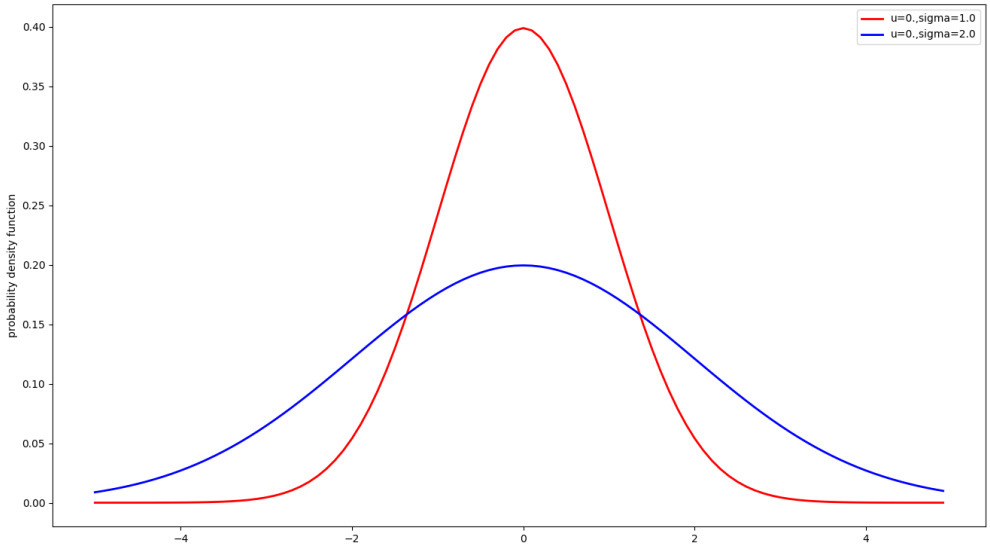

```
# äžŇéąźåĹĘåÿČ
@my_plot(label0='n=50,p=0.3', label1='n=50,p=0.7', fn='binom.png',␣
↪ylabel='probability mass function')
def bino():
    x = np.arange(50)
    n, p, p1 = 50, 0.3, 0.7
    y = binom.pmf(x, n=n, p=p)
    y1 = binom.pmf(x, n=n, p=p1)
    return x, y, y1
```



11.17 17 éńŸæŰŕåĹĘåÿČ

çžćèL'šæŻšçž£èąĺçď'žåİĞåĂijäÿž0ïijŇæăĞåĞĘåůőäÿž1.0çŽĎæęĆçŐĞåŕĘåžęåĞ¡æŢřïijŇèŞİèL'šæŻšçž£

```
# éńŸæŰŕ åĹĘåÿČ
@my_plot(label0='u=0.,sigma=1.0', label1='u=0.,sigma=2.0', fn=
↪'guass.png')
def guass():
    x = np.arange(-5, 5, 0.1)
    y = norm.pdf(x, loc=0.0, scale=1.0)
    y1 = norm.pdf(x, loc=0., scale=2.0)
    return x, y, y1
```



11.18 18 betaåĹĘåÿČ

betaåĹĘåÿČçŽĎæIJ§æIJŻåĂijåęĆäÿŃïijŇåŕŕäżŐäÿŃéİćçŽĎäÿď'æİąæŻšçž£äÿ■åŁääžěéłŇèŕĄïijŽ

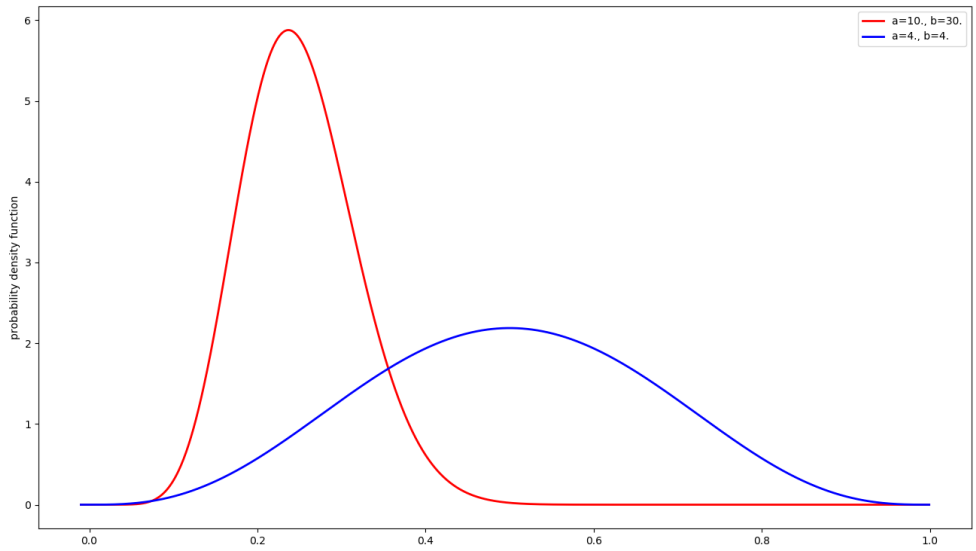
ch10/./imgimage-20200105205845965.png

```
# beta åĹĘåÿČ
@my_plot(label0='a=10., b=30.', label1='a=4., b=4.', fn='beta.png')
def bet():
```

(continues on next page)

(continued from previous page)

```
    x = np.arange(-0.01, 1, 0.001)
    y = beta.pdf(x, a=10., b=30.)
    y1 = beta.pdf(x, a=4., b=4.)
    return x, y, y1
```



CHAPTER 12

å■ĄãĂĄPandasæŢřæ■őåĹĘæđŘ

å§žäžŐKaggleçŤţå¡šå¡śèŕĎæŢřæ■őéŻĘïijŇä¡£çŤĺPandasåĄŽæŢřæ■őåĹĘæđŘïijŇår¡åŔŕèČ¡æŃĘåĹĘäÿ

12.1 1 äžĘèğčæŢřæ■ő

æŢřæ■őæİěèĞłkaggleïijŇåĚšåŇĚæŃňäÿĽäÿłæŰĞäžűïijŽ

1. movies.dat
2. ratings.dat
3. users.dat

`movies.dat`åŇĚæŃňäÿĽäÿłå■Ůæőţïij Ž[âĂŸMovie IDâĂŹ, âĂŸMovie TitleâĂŹ, âĂŸGenreâĂŹ]

ä¡£çŤĺpandasåŕijåĚěæ■d'æŰĞäžűïijŽ

```
import pandas as pd

movies = pd.read_csv('./data/movietweetings/movies.dat', delimiter=
↪'::', engine='python', header=None, names = ['Movie ID', 'Movie␣
↪Title', 'Genre'])
```

åŕijåĚěåŘŐïijŇæŸ¿çd'žåĽ■5èąŇïijŽ

```
   Movie ID                                        Movie Title  \
0         8      Edison Kinetoscopic Record of a Sneeze (1894)
1        10                 La sortie des usines LumiçŇńre (1895)
2        12                       The Arrival of a Train (1896)
3        25  The Oxford and Cambridge University Boat Race ...
```

(continues on next page)

(continued from previous page)

```
4         91                            Le manoir du diable (1896)
5        131                              Une nuit terrible (1896)
6        417                         Le voyage dans la lune (1902)
7        439                        The Great Train Robbery (1903)
8        443          Hiawatha, the Messiah of the Ojibway (1903)
9        628                          The Adventures of Dollie (1908)
                                                 Genre
0                                  Documentary|Short
1                                  Documentary|Short
2                                  Documentary|Short
3                                                NaN
4                                      Short|Horror
5                               Short|Comedy|Horror
6  Short|Action|Adventure|Comedy|Fantasy|Sci-Fi
7                        Short|Action|Crime|Western
8                                                NaN
9                                     Action|Short
```

æňąåŕijåĚěåĚűäžŰäÿď'äÿłæŢřæ■őæŰĞäžű

users.dat:

```
users = pd.read_csv('./data/movietweetings/users.dat', delimiter='::
↪', engine='python', header=None, names = ['User ID', 'Twitter ID
↪'])
print(users.head())
```

çżŞædIJïijŽ

```
   User ID  Twitter ID
0        1   397291295
1        2    40501255
2        3   417333257
3        4   138805259
4        5  2452094989
5        6   391774225
6        7    47317010
7        8    84541461
8        9  2445803544
9       10   995885060
```

rating.data:

```
ratings = pd.read_csv('./data/movietweetings/ratings.dat',␣
↪delimiter='::', engine='python', header=None, names = ['User ID',
↪'Movie ID', 'Rating', 'Rating Timestamp'])
print(ratings.head())
```

çżŞædIJïijŽ

```
   User ID  Movie ID  Rating  Rating Timestamp
0        1    111161      10        1373234211
1        1    117060       7        1373415231
2        1    120755       6        1373424360
3        1    317919       6        1373495763
4        1    454876      10        1373621125
5        1    790724       8        1374641320
6        1    882977       8        1372898763
7        1   1229238       9        1373506523
8        1   1288558       5        1373154354
9        1   1300854       8        1377165712
```

12.2 2 read_csvä¡£çŤĺèŕťæŸŐ

èŕťæŸŐïijŇæIJňæňąåŕijåĚědatæŰĞäžűä¡£çŤĺpandas.read_csvåĞ¡æŢřãĂĆ

çňňäÿĂäÿłä¡■ç¡őåŘĆæŢř./data/movietweetings/ratings.dat èąĺçď'žæŰĞäžűçŽĎçŽÿåŕźèůŕå¿Ď

çňňäžŇäÿłåĚşéŤőå■ŮåŘĆæŢřïijŽdelimiter='::'ïijŇèąĺçď'žæŰĞäžűåĹĘéŽŤçňęä¡£çŤĺ::

åŘŐéIćåĞääÿłåĚşéŤőå■ŮåŘĆæŢřåĹĘåĹńäžčèąĺä¡£çŤĺçŽĎåijŢæŞŐïijŇæŰĞäžűæšąæIJL'èąĺåď't'ïijŇæL

åŕijåĚěåŘŐdataframeçŽĎåĹŮåŘ■ä¡£çŤĺnamesåĚşéŤőå■Ůèő¿ç¡őïijŇè£ŹäÿłåŘĆæŢřåď'ğåőűåŘŕäžěèőřä

KaggleçŤţå¡śæŢřæ■őéŻĘçňňäÿĂèŁĆïijŇæĹŚäžňä¡£çŤĺæŢřæ■őåď'ĎçŘĘåĹľåŹĺ pandasïijŇ åĞ¡æŢřread_csv åŕijåĚěçżŹåőŽçŽĎäÿL'äÿłæŢřæ■őæŰĞäžűãĂĆ

```
import pandas as pd

movies = pd.read_csv('./data/movietweetings/movies.dat', delimiter=
↪'::', engine='python', header=None, names = ['Movie ID', 'Movie␣
↪Title', 'Genre'])
users = pd.read_csv('./data/movietweetings/users.dat', delimiter='::
↪', engine='python', header=None, names = ['User ID', 'Twitter ID
↪'])
ratings = pd.read_csv('./data/movietweetings/ratings.dat',␣
↪delimiter='::', engine='python', header=None, names = ['User ID',
↪'Movie ID', 'Rating', 'Rating Timestamp'])
```

çŤĺåĹřçŽĎread_csvïijŇæ§ŘäžŻéĞ■èęĄçŽĎåŘĆæŢřïijŇåęĆä¡Ţä¡£çŤĺåIJĺäÿŁäÿĂèŁĆäź§æIJL'æL'Ă

æL'¿åĞžå¿ŮåĹĘåL'■10åŰIJåL'ğ(comedy)

12.3 3 åď'ĎçŘĘçżĎåŘĹåĂij

èąĺmovieså■Ůæőţ Genreèąĺçď'žçŤţå¡śçŽĎçśżåđŃïijŇåŔŕèČ¡æIJL'åď'Žäÿłå Äijïij ŇåĹĘéŽŤçňęäÿž | ïijŇ

éŠĹåŕźè£Źçśżå■ŮæőţåŔŰåĂijïijŇåŔŕä¡£çŤĺPandasäÿ■SeriesæŔŘä¿ŻçŽĎstråĄŽäÿĂæ■ěè¡ňåŇŰïijŇa

```
mask = movies.Genre.str.contains('comedy',case=False,na=False)
```

æşĺæĎŔä¡£çŤĺçŽĎäÿd'äÿłåŔĆæŢřïijŽcase, na

caseäÿž FalseïijŇèąĺçd'žåŕźåd'ğåřŔåĘŹäÿ■æŢŔæĎ§ïijŻ
na GenreåĹŮæ§Řäÿłå■ŢåĚČæăijäÿžNaNæŮűïijŇæĹŚäżňä¡£çŤĺçŽĎåĚĚåąńåĂijïijŇæ■d'åd'Ďåąńå ĚĚäÿžFalse

è£ŤåŻđçŽĎmaskæŸŕäÿĂçżt'çŽĎSeriesïijŇçżŞæđĎäÿŐ
movies.GenreçŻÿåŘŇïijŇåŔŰåĂijäÿžTrue æĹŰ False.

èğĆåŕ§çżŞæđIJïijŽ

```
0    False
1    False
2    False
3    False
4    False
5     True
6     True
7    False
8    False
9    False
Name: Genre, dtype: bool
```

12.4 4 èő£éŮőæ§ŘåĹŮ

å¿ŮåĹřæŐl'çăĄmaskåŘŐïijŇpandaséIđåÿÿæŰźä¿£åIJřèČ¡æŔŘåŔŰåĞžçŽőæăĞèőřå¡ŢïijŽ

```
comedy = movies[mask]
comdey_ids = comedy['Movie ID']
```

äžěäÿŁïijŇåIJÍpandasäÿ■èćńæIJĂéćŚçŐĞä¡£çŤĺïijŇäÿ■åĘ■èğčéĞŁãĂĆçIJŃçżŞæđIJcomedy_ids.head()ïijŽ

```
5      131
6      417
15    2354
18    3863
19    4099
20    4100
21    4101
22    4210
23    4395
```

(continues on next page)

(continued from previous page)

```
25    4518
Name: Movie ID, dtype: int64
```

1-4äžŃçż■æŢřæ■őèŕżåĚěïijŇåd'ĎçŘEçżĎåŘĹåĂijïijŇçt'ćåijŢæŢřæ■őç■Ľ, pandasäÿ■ä¡£çŤĺè¿Čåď'ŽçŽĎåĞ¡æŢřïijŇå§žäžŐKaggleçIJ§åőđçŤţå¡śå¡śèŕĎæŢřæ■őéŻĘïijŇæIJĂåŘŐå¿ŮåĹIDïijŽ

```
5      131
6      417
15    2354
18    3863
19    4099
20    4100
21    4101
22    4210
23    4395
25    4518
Name: Movie ID, dtype: int64
```

äÿŃéİćçżğçż■æŢřæ■őæŐćçť'ćäźŃæŮĚ~

12.5 5 è£đæŐěäÿď'äÿłèąĺ

æŃ£åĹřæĽ'ĂæIJĽ'åŰIJåĽ'ğçŽĎIDåŘŐïijŇèęĄæČşæĽ'¿åĞžåĚűäÿ■åźşåİĞå¿ŮåĹĘæIJĂéńŸçŽĎåĽ'■10å

åĘ■åŻđéą¿äÿŃratingsèąĺçżŞæđĎïijŽ

```
   User ID  Movie ID  Rating  Rating Timestamp
0        1    111161      10        1373234211
1        1    117060       7        1373415231
2        1    120755       6        1373424360
3        1    317919       6        1373495763
4        1    454876      10        1373621125
5        1    790724       8        1374641320
6        1    882977       8        1372898763
7        1   1229238       9        1373506523
8        1   1288558       5        1373154354
9        1   1300854       8        1377165712
```

pandas äÿ■ä¡£çŤĺjoinåĚşèĄŤäÿď'åijåèąĺïijŇè£đæŐěå■Ůæőţæ ŸŕMovie IDïijŇåęĆæđIJéąžåĚűèĞłçĎűè£ŹäźĹä¡£çŤĺjoinïijŽ

```
combine = ratings.join(comedy, on='Movie ID', rsuffix='2')
```

åůęåŔşæżŚåŁĺïijŇæ§ěçIJŃåőŇæŢť'äžčçăĄ

åď'ğåőűåŘŕéłŇèŕĄè£Źçğ■åĘŹæşŢïijŇäžŤçżĘäÿĂçIJŃïijŇäijŽåŔŚçŐřçżŞæđIJéİđåÿÿèŕąåijĆãĂĆ

çľűåĚűåŐ§åŻäïijŇè£ŹæŸŕpandas joinåĞ¡æŢřä¡£çŤĺçŽĎäÿĂäÿłçőŮæŸŕåİŚçĆźïijŇåőČåIJĺåőŸæąčäÿ■ä Movie ID.

åůęèąĺçŽĎindexäÿ■èęĄæśĆïijŇä¡ĘæŸŕèęĄåIJĺåŔĆæŢř onäÿ■çżŹåőŽãĂĆ

äżěäÿŁæŸŕèęĄæşĺæĎŔçŽĎäÿĂçĆź

ä£őæŤźäÿžïijŽ

```
combine = ratings.join(comedy.set_index('Movie ID'), on='Movie ID')
print(combine.head(10))
```

äżěäÿŁæŸŕOKçŽĎåĘŹæşŢ

èğĆåŕ§çżŞæđIJïijŽ

```
   User ID  Movie ID  Rating  Rating Timestamp Movie Title Genre
0        1    111161      10        1373234211          NaN   NaN
1        1    117060       7        1373415231          NaN   NaN
2        1    120755       6        1373424360          NaN   NaN
3        1    317919       6        1373495763          NaN   NaN
4        1    454876      10        1373621125          NaN   NaN
5        1    790724       8        1374641320          NaN   NaN
6        1    882977       8        1372898763          NaN   NaN
7        1   1229238       9        1373506523          NaN   NaN
8        1   1288558       5        1373154354          NaN   NaN
9        1   1300854       8        1377165712          NaN   NaN
```

GenreåĹŮäÿžNaNèąĺæŸŐïijŇè£Źäÿ■æŸŕåŰIJåL'ğãĂĆéIJĂèęĄç■ŻéĂL'åĞžæ■d'åĹŮäÿ■äÿžNaN çŽĎèőřå¡ŢãĂĆ

12.6 6 æŇL'åĹŮç■ŻéĂL'

pandasæIJĂæŰźä¿£çŽĎåIJřæŰźïijŇářśæŸŕåŘŚéĞŔåŇŰè£ŘçőŮïijŇář¡åŔřèČ¡åĞŔářŠäžĘforå¿łçŐŕçŽĎ

æŇL'åĹŮç■ŻéĂL'è£Źçğ■åÿÿèğĄéIJĂæśĆïijŇèĞłçĎűåŔŕäżěè¡żæİ¿åžŤårźãĂĆ

äÿžäžĘçĚğéą¿åĹİæňąæŐěèğę pandas çŽĎæIJŃåŘŃïijŇåĹĘäÿd'æ■ěåŐżåĘŹïijŽ

```
mask = pd.notnull(combine['Genre'])
```

çżŞæđIJæŸŕäÿĂåĹŮåŔłåŘńTrue æĹŰ FalseçŽĎåÄij

```
result = combine[mask]
print(result.head())
```

çżŞæđIJäÿ■ïijŇGenreå■Ůæőţäÿ■èĞşářŠåŘńæIJL'äÿĂäÿłComedyå■ŮçňęäÿšïijŇèąĺæŸŐéłŇèŕĄäžĘæĹŚ

```
    User ID  Movie ID  Rating  Rating Timestamp                Movie␣
↪Title  \
12        1   1588173       9        1372821281       Warm Bodies␣
↪(2013)
```

(continues on next page)

(continued from previous page)

```
13         1   1711425       3        1372604878        21 & Over␣
↪(2013)
14         1   2024432       8        1372703553   Identity Thief␣
↪(2013)
17         1   2101441       1        1372633473  Spring Breakers␣
↪(2012)
28         2   1431045       7        1457733508         Deadpool␣
↪(2016)

                              Genre
12           Comedy|Horror|Romance
13                          Comedy
14    Adventure|Comedy|Crime|Drama
17              Comedy|Crime|Drama
28  Action|Adventure|Comedy|Sci-Fi
```

åůěä¡IJçť ğåijăïijŇçŐřåIJÍæŕŔåď'ľåŔłèČ¡åĘŹäÿĂçĆźãĂĆæĹŚåř¡éĞŔåĘŹçŽĎèŕęçżĘçĆźïijŇäÿĂæ■ěäÿĂ

æŸŐåď'ľçżğçż■æIJİçİĂçŻőæăĞïijŇåL'■è£ŻäÿĂçĆźçĆź~

CHAPTER 13

å■ĄäÿĂãĂĄPythonåőđæĹŸ

äÿĂäžŽåőđæĹŸçŽĎåřŔä¿Ńå■ŘïijŇçŻőåL'■æŽĆæŮűæIJL'5äÿłïijŇåŘŐæIJ§äijŽäÿ■æŰ■æL'ľåĚĚãĂĆ

13.1 1 çŐŕåćČæŘ■åžž

åŇžåĹĘåĞääÿłåřŔçŹ¡åőźæŸŞæůůæůĘçŽĎæęĆå£ţïijŽpycharmïijŇpythonèğčéĞŁåŹĺïijŇcondaåőL'èčĚïi

- pycharmæŸŕpythonåijĂåŔŚçŽĎéŻĘæĹŘåijĂåŔŚçŐŕåćČ(Integrated Development Environmentïij ŇçőĂçğřIDE)ïijŇåőČæIJňèžńæŮăæşŢæL'ğèąŇPythonäžčçăĄ
- pythonèğčéĞŁåŹĺæL'■æŸŕçIJ§æ■čæL'ğèąŇäžčçăĄçŽĎåůěåĚůïijŇpycharméĞŇåŔŕèő¿ç¡őPythonè
- anacondaæŸŕpythonåÿÿçŤĺåŇĚçŽĎåŘĹéŻĘïijŇåźűæŔŘä¿ŻçżŹæĹŚäžňä¡£çŤĺcondaåŚ¡äžd'éİđåÿÿæ
- condaåőL'èčĚïijŽæĹŚäžňåőL'èčĚè£Ğanacondaè¡ŕäżűåŘŐïijŇåřśèČ¡åd'§ä¡£çŤĺcondaåŚ¡äžd'äÿŃè¡¡
- pipåőL'èčĚïijŽçśżäijijäžŐcondaåőL'èčĚçŽĎpythonåőL'èčĚåŇĚçŽĎæŰźæşŢ

ä£őæŤźéŢIJåČŔæžŘ

åIJĺä¡£çŤĺåőL'èčĚconda åőL'èčĚæ§ŘäžŻåŇĚäijŽåĞžçŐřæĚćæĹŰåőL'èčĚåd'śèt'ěéŮőécŸïijŇæIJĂæIJ

åĚĹæ§ěçIJŃåůšçżŘåőL'èčĚè£ĞçŽĎéŢIJåČŔæžŘïijŇcmdçłŮåŘčæL'ğèąŇåŚ¡äžd'ïijŽ

```
conda config --show
```

æ§ěçIJŃéĚ■ç¡őéąźchannelsïijŇąęĆæđIJæŸ¿çd'žåÿęæIJL'tsinghuaïijŇåĹŹèŕt'æŸŐåůšåőL'èčĚè£Č

```
channels:
- https://mirrors.tuna.tsinghua.edu.cn/tensorflow/linux/cpu/
- https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/msys2/
- https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/conda-forge/
```

(continues on next page)

(continued from previous page)

```
- https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/
- https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/pytorch/
```

äÿŃäÿĂæ■ěïijŇä¡£çŤÍconda config --remove channels urlåIJřåİĂåĹăéŹd'æÿĚå■ŐéŢIJåČŕïijŇåęĆäÿŃåŚ¡äžd'åĹăéŹd'çňňäÿĂäÿłãĂĆçĎűåŘŐïijŇä¿İæňąåĹăéŹ

```
conda config --remove channels https://mirrors.tuna.tsinghua.edu.cn/
↪tensorflow/linux/cpu/
```

æůżåŁăçŻőåL'■åŕŕçŤÍçŽĎäÿ■çğŚåd'ğéŢIJåČŕæžŘïijŽ

```
conda config --add channels https://mirrors.ustc.edu.cn/anaconda/
↪pkgs/free/
```

åźűèő¿ç¡őæŘIJçt'ćæŮűæŸ¿çd'žéĂŽéĄŞåIJřåİĂïijŽ

```
conda config --set show_channel_urls yes
```

çąőèőd'æŸŕåŘęåőL'èčĚéŢIJåČŕæžŘæĹŘåŁ§ïijŇæL'ğèąŇconda config --showïijŇæL'¿åĽřchannelsåĂijäÿžåęĆäÿŃïijŽ

```
channels:
  - https://mirrors.ustc.edu.cn/anaconda/pkgs/free/
  - defaults
```

Done~

13.2 2 èĞłåŁÍç¿d'åŔŚéĆőäžű

PythonèĞłåŁÍç¿d'åŔŚéĆőäžű

```
import smtplib
from email import (header)
from email.mime import (text, application, multipart)
import time

def sender_mail():
    smt_p = smtplib.SMTP()
    smt_p.connect(host='smtp.qq.com', port=25)
    sender, password = '113097485@qq.com', "**************"
    smt_p.login(sender, password)
    receiver_addresses, count_num = [
        'guozhennianhua@163.com', 'xiaoxiazi99@163.com'], 1
    for email_address in receiver_addresses:
        try:
            msg = multipart.MIMEMultipart()
            msg['From'] = "zhenguo"
```

(continues on next page)

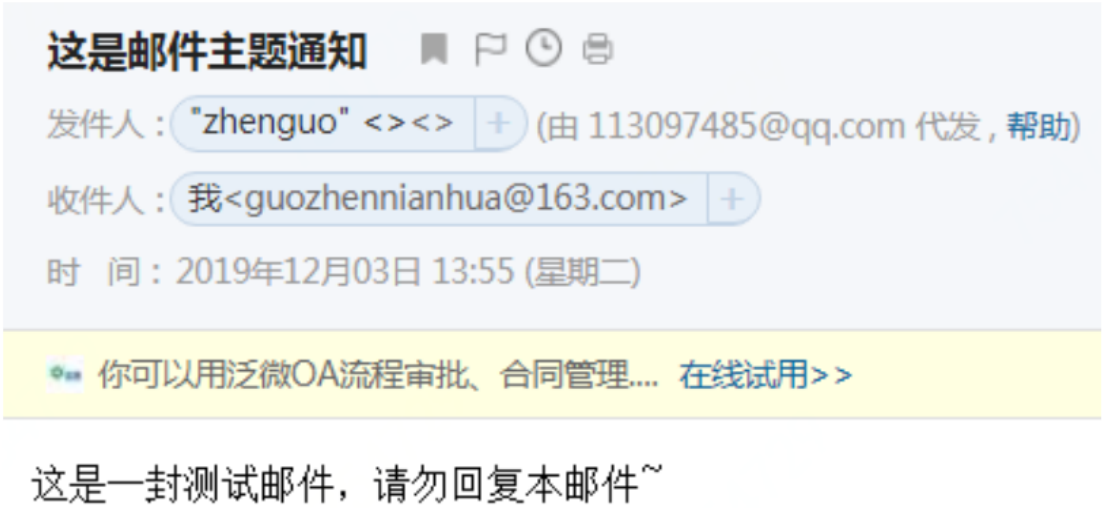
(continued from previous page)

```
            msg['To'] = email_address
            msg['subject'] = header.Header('è£ŹæŸŕéĆőäżűäÿżéćŸéĂŽç§ě
↪', 'utf-8')
            msg.attach(text.MIMEText(
                'è£ŹæŸŕäÿĂåřĄætŃèŕŢéĆőäżűïijŇèŕůåŃ£åŽđåd’■æIJňéĆőäżű~
↪', 'plain', 'utf-8'))
            smt_p.sendmail(sender, email_address, msg.as_string())
            time.sleep(10)
            print('çňň%dæňąåŔŚéĂĄçżŹ%s' % (count_num, email_
↪address))
            count_num = count_num + 1
        except Exception as e:
            print('çňň%dæňąçżŹ%såŔŚéĂĄéĆőäżűåijĆåÿÿ' % (count_num,␣
↪email_address))
            continue
    smt_p.quit()

sender_mail()
```

æşÍæĎŔïijŽ

åŔŚéĂĄéĆőçőśæŸŕqqéĆőçőśïijŇæL’ĂäżěèęĄåIJÍqqéĆőçőśäÿ■èő¿ç¡őåijĂåŘŕSMTPæIJ■åŁąïijŇèő¿ç¡őåőŇæ

åŔŚéĂĄåŘŐçŽĎæĹłåŻ¿ïijŽ



13.3 3 äžŇåĹĘæŘIJçťć

äžŇåĹĘæŘIJçťćæŸŕçÍŃåžŔåŚŸå£Ëåď’ĞçŽĎçőŮæşŢïijŇæŮăèőžäżĂäźĹåIJžåŘĹïijŇéČ¡èęĄéİđåÿÿçE§

åřŕäżŃå■Řæŕŕŕè£řïijŽ
åIJĺæIJL'åžŕæŢřçžĎarräÿ■ïijŇæŇĞåőŽåŇžéŮť[left,
right]èŇČåŻť'åĘĚïijŇæ§ěæL'¿åĚČçť'ăx
åęĆæđIJäÿ■å■ŸåIJĺïijŇè£ŤåŻđ-1

äžŇåĹĘæŘIJçť'ćbinarySearchåőđçŐřçŽĎäÿžéĂżè¿Ś

```
def binarySearch(arr, left, right, x):
    while left <= right:

        mid = int(left + (right - left) / 2); #␣
↪æL'¿åĹřäÿ■éŮt'ä¡■ç¡őãĂĆæśĆäÿ■çĆźåĘźæĹŘ(left+right)/
↪2æŻt'åőźæŸŞæžćåĞžïijŇæL'Ăäżěäÿ■åżžèőőè£ŹæăůåĘŹ

        # æčĂæŞěxæŸŕåŘęåĞžçŐřåIJĺä¡■ç¡őmid
        if arr[mid] == x:
            print('found %d åIJĺçt'ćåijŢä¡■ç¡ő%d åd'Ď' %(x,mid))
            return mid

            # åĄĞåęĆxæŻt'åd'ğïijŇåĹŹäÿ■åŕŕèČ¡åĞžçŐřåIJĺåůęå■ŁéČĺåĹĘ
        elif arr[mid] < x:
            left = mid + 1 #æŘIJçt'ćåŇžéŮt'åŔŸäÿž[mid+1,right]
            print('åŇžéŮt'çijl'åřŕäÿž[%d,%d]' %(mid+1,right))

        #␣
↪åŘŇçŘĘïijŇåĄĞåęĆxæŻt'åřŕïijŇåĹŹäÿ■åŕŕèČ¡åĞžçŐřåIJĺåŕşå■ŁéČĺåĹĘ
        elif x<arr[mid]:
            right = mid - 1 #æŘIJçt'ćåŇžéŮt'åŔŸäÿž[left,mid-1]
            print('åŇžéŮt'çijl'åřŕäÿž[%d,%d]' %(left,mid-1))

    # åĄĞåęĆæŘIJçt'ćåĹřè£ŹéĞŇïijŇèąĺæŸŐxæIJłåĞžçŐřåIJĺ[left,right]äÿ■
    return -1
```

åIJĺIpythonäžd'äžŠçŢŇéİćäÿ■ïijŇèřČçŤĺbinarySearchçŽĎåřŔDemo:

```
In [8]: binarySearch([4,5,6,7,10,20,100],0,6,5)
åŇžéŮt'çijl'åřŕäÿž[0,2]
found 5 at 1
Out[8]: 1

In [9]: binarySearch([4,5,6,7,10,20,100],0,6,4)
åŇžéŮt'çijl'åřŕäÿž[0,2]
åŇžéŮt'çijl'åřŕäÿž[0,0]
found 4 at 0
Out[9]: 0

In [10]: binarySearch([4,5,6,7,10,20,100],0,6,20)
åŇžéŮt'çijl'åřŕäÿž[4,6]
found 20 at 5
```

(continues on next page)

(continued from previous page)

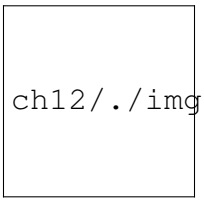
```
Out[10]: 5

In [11]: binarySearch([4,5,6,7,10,20,100],0,6,100)
åŇžéŮt'çijl'ářŔäÿž[4,6]
åŇžéŮt'çijl'ářŔäÿž[6,6]
found 100 at 6
Out[11]: 6
```

13.4 4 çĹňåŔŰåď'ærŤæŢřæ■őåźűèğčædŘæÿľåžęåĂij

çĹňåŔŰåď'ærŤæŢřæ■őåźűèğčædŘæÿľåžęåĂij

çť'ăæİŘæİěèĞłæIJŃåŔŃècĄçż■ïijŇæĎ§èřćïijĄ

çĹňåŔŰçŽĎhtml çżŞædĎ




```
import requests
from lxml import etree
import pandas as pd
import re

url = 'http://www.weather.com.cn/weather1d/101010100.shtml#input'
with requests.get(url) as res:
    content = res.content
    html = etree.HTML(content)
```

éĂŽè£ĞlxmlæĺąåİŮæŔŘåŔŰåĂij

lxmlæŕŤbeautifulsoupèğčædŘåIJĺæ§ŘäžŻåIJžåŘĹæŻťéńŸæŢĹ

```
location = html.xpath('//*[@id="around"]//a[@target="_blank"]/span/
↪text()')
temperature = html.xpath('//*[@id="around"]/div/ul/li/a/i/text()')
```

çżŞædIJïijŽ

```
['éęŹæšş', 'æű£åůđ', 'åŤŘåśś', 'æšğåůđ', 'åd'l'æt'ě', 'åżŁåİŁ',
↪'åd'łåŐ§', 'ç§şåőűåžĎ', 'æű£éź£', 'åijăåőűåŔč', 'ä£İåőŽ', 'äÿL'æšş
↪', 'åŇŮäžňå■ŤåžŹ', 'åŇŮäžňåŻ¡å■ŘçŻŚ', 'äÿ■åŻ¡åIJřèt'Ĺå■ŽçL'l'éęĘ',
↪'æIJĹåİŻåĚň
åŻ■', 'æŸŐå§ŐåćŹéĄŮåİĂåĚňåŻ■', 'åŇŮäžňåÿĆèğĎåĹŠåśŢèğĹéęĘ', 'äżĂåĹźæţů
↪', 'å■ŮéŤčéijŞåůů', 'åd'l'åİŻåĚňåŻ■', 'åŇŮæţůåĚňåŻ■', 'æŹŕåśśåĚňåŻ■
↪', 'åŇŮäžňæţůæt'NéęĘ']
```

(continues on next page)

(continued from previous page)

```
['11/-5ÂřC', '14/-5ÂřC', '12/-6ÂřC', '12/-5ÂřC', '11/-1ÂřC', '11/-
↪5ÂřC', '8/-7ÂřC', '13/-2ÂřC', '8/-6ÂřC', '5/-9ÂřC', '14/-6ÂřC',
↪'11/-4ÂřC', '13/-3ÂřC'
, '13/-3ÂřC', '12/-3ÂřC', '12/-3ÂřC', '13/-3ÂřC', '12/-2ÂřC', '12/-
↪3ÂřC', '13/-3ÂřC', '12/-2ÂřC', '12/-2ÂřC', '12/-2ÂřC', '12/-3ÂřC']
```

æđĎéĂăDataFrameåŕźèśą

```
df = pd.DataFrame({'location':location, 'temperature':temperature})
print('æÿl’åžęåĹŮ')
print(df['temperature'])
```

æ■čåĹŹèğčæđŘæÿl’åžęåĂij

```
df['high'] = df['temperature'].apply(lambda x: int(re.match('(-?[0-
↪9]*?)/-?[0-9]*?ÂřC', x).group(1) ) )
df['low'] = df['temperature'].apply(lambda x: int(re.match('-?[0-
↪9]*?/(-?[0-9]*?)ÂřC', x).group(1) ) )
print(df)
```

èŕęçżĘèŕť æŸŐå■Řå■ŮçňęåĹŻæ■ŢèŐů

éŹď’ážĘçőĂå■ŢåIJřåĹď’æŰ■æŸŕåŘęåŇźéĚ■äźŃåď’ŰïijŇæ■čåĹŹèąÍè¿¿åijŔè£ŸæIJL’æŔŘåŔŰå■Řäÿšç
8}) $åĹĘåĹńåőŽäźL’ážĘäÿd’äÿłçżĎïijŇåŔŕäžěçŻť’æŐěäžŐåŇźéĚ■çŽĎå■Ůçňęäÿšäÿ■æŔŘåŔŰåĞžåŇžåŔů

```
m = re.match(r'^(\d{3})-(\d{3,8})$', '010-12345')
print(m.group(0))
print(m.group(1))
print(m.group(2))

# 010-12345
# 010
# 12345
```

åęĆæđIJæ■čåĹŹèąÍè¿¿åijŔäÿ■åőŽäźL’ážĘçżĎïijŇåřśåŔŕäžěåIJÍMatchåŕźèśąäÿŁçŤÍgroup()æŰźæşŢ

æşÍæĎŘåĹřgroup(0)æřÿè£IJæŸŕåŐ§åğŃå■ŮçňęäÿšïijŇgroup(1)ãĂĄgroup(2)âĂęâĂęèąÍçď’žç

æIJĂçżĹçżŞæđIJ

```
Name: temperature, dtype: object
    location temperature  high  low
0         éęŹæšş     11/-5ÂřC    11   -5
1         æű£åůđ     14/-5ÂřC    14   -5
2         åŤŘåśś     12/-6ÂřC    12   -6
3         æšğåůđ     12/-5ÂřC    12   -5
4         åd’l’æt’ě     11/-1ÂřC    11   -1
5         åżŁåİŁ     11/-5ÂřC    11   -5
6         åd’łåŐ§       8/-7ÂřC     8   -7
```

(continues on next page)

(continued from previous page)

```
7          çŞşåőűåžĎ      13/-2ÂřC     13    -2
8           æű£éź£       8/-6ÂřC      8    -6
9          åijäåőűåŕč       5/-9ÂřC      5    -9
10          ä£Ìåőž      14/-6ÂřC     14    -6
11          äÿL'æšş      11/-4ÂřC     11    -4
12       åŇŮäžňå■Ťåžź      13/-3ÂřC     13    -3
13      åŇŮäžňåż¡å■ŘçŻŚ      13/-3ÂřC     13    -3
14    äÿ■åż¡åIJřèt'Íå■ŽçL'l'éęĘ      12/-3ÂřC     12    -3
15       æIJĹåİżåĚňåż■      12/-3ÂřC     12    -3
16    æŸŐå$Őåćźé ĄŮåİÅåĚňåż■      13/-3ÂřC     13    -3
17   åŇŮäžňåÿĆèğĎåĹŠåśŢèğĹéęĘ      12/-2ÂřC     12    -2
18        äżĂåĹźætů      12/-3ÂřC     12    -3
19       å■ŮéŤčéijŞåůů      13/-3ÂřC     13    -3
20       åd'l'åİżåĚňåż■      12/-2ÂřC     12    -2
21       åŇŮætůåĚňåż■      12/-2ÂřC     12    -2
22       æŹŕåśśåĚňåż■      12/-2ÂřC     12    -2
23      åŇŮäžňætůæt'ŃéęĘ      12/-3ÂřC     12    -3
```

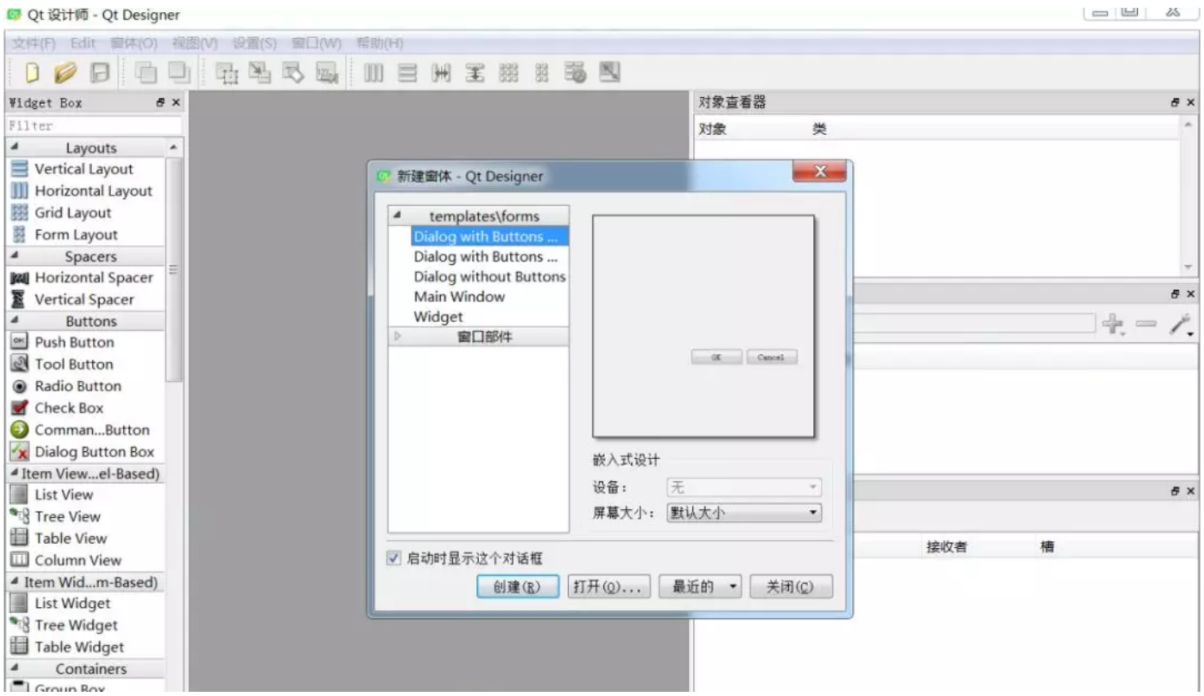
13.5 5 åĹűä¡IJåřŔèĂŇç¿ŐçŽĎèőąçőŮåŹÍ

1) uièő¿èőą

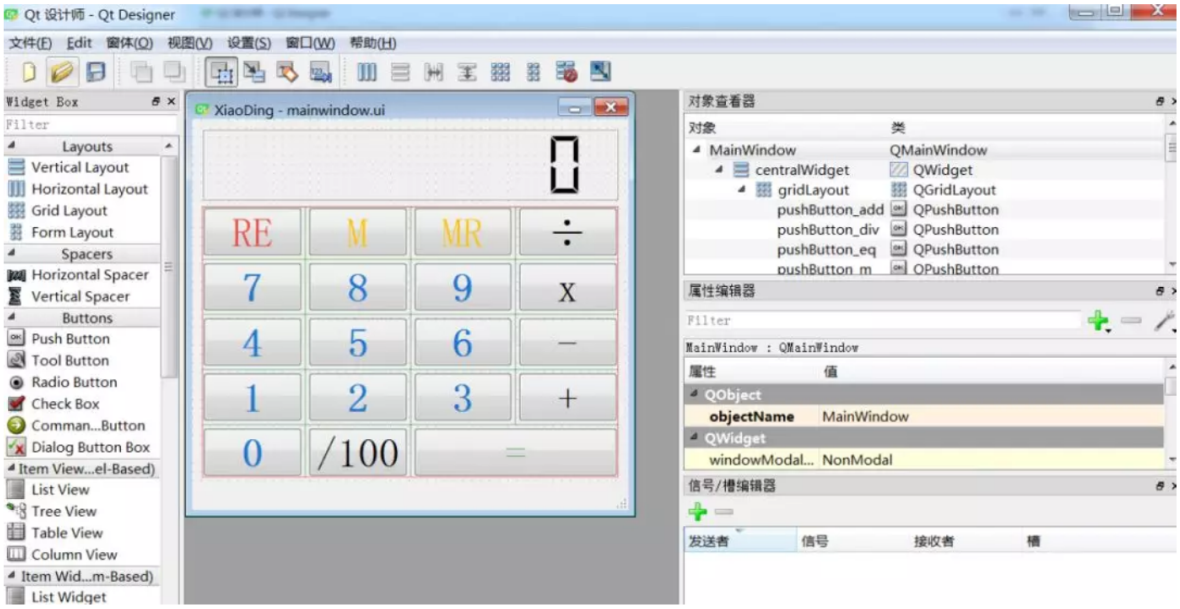
ä¡£çŤÍqt designer ïijŇæŇL'èčĚanacondaåŘŐïijŇåIJĺåęĆäÿŃèůŕå¿ĎæL'¿åĹřïijŽ

conda3.05\Library\bin

designer.exeæŰĞäžűïijŇåŔŇåĞżåŘŕåŁĺïijŽ



åĹŻåžžçłŮä¡ŞïijŇåŚ¡åŘ■äÿžXiaoDingïijŇæŢt'äÿłçŽĎçŢŇéĺćåęĆäÿŃæL'Ăçd'žïijŽ



qt èő¿èőąåŹĺæŔŘä¿ŻçŽĎåÿÿçŤĺæŐğäžűå§žæIJňéČ¡èČ¡æżąèűşåijĂåŔŚéIJĂæśĆïijŇéĂŽè£ĞæŃŰå

æIJĂçżĹèő¿èőąçŽĎèőąçőŮåŹĺXiaoDingçŢŇéİćåęĆäÿŃïijŇ

æŕŤåęĆïijŇåĚűäÿ■äÿĂäÿłçŤĺäžŐèőąçőŮåŹĺæŸ¿çd'žçŽĎåŕźèśąïijŽlcdNumberïijŇåŕźèśąçŽĎçśżåđŃäjNumberãĂĆåŔşä¿ğäÿžèőąçőŮåŹĺäÿ■çŤĺåĹřçŽĎæL'ĂæIJL'åŕźèśąãĂĆ

2) è¡ňpyæŰĞäžű

ä¡£çŤĺåęĆäÿŃåŚ¡äżd'ïijŇåřĘèő¿èőąåě¡çŽĎuiæŰĞäžűè¡ňäÿžpyæŰĞäžűïijŽ

```
pyuic5 -o ./calculator/MainWindow.py ./calculator/mainwindow.ui
```

3) èőąçŐŮåŹĺåőđçŐřéĂżè¿Ś

åŕijåĚěåžŞïijŽ

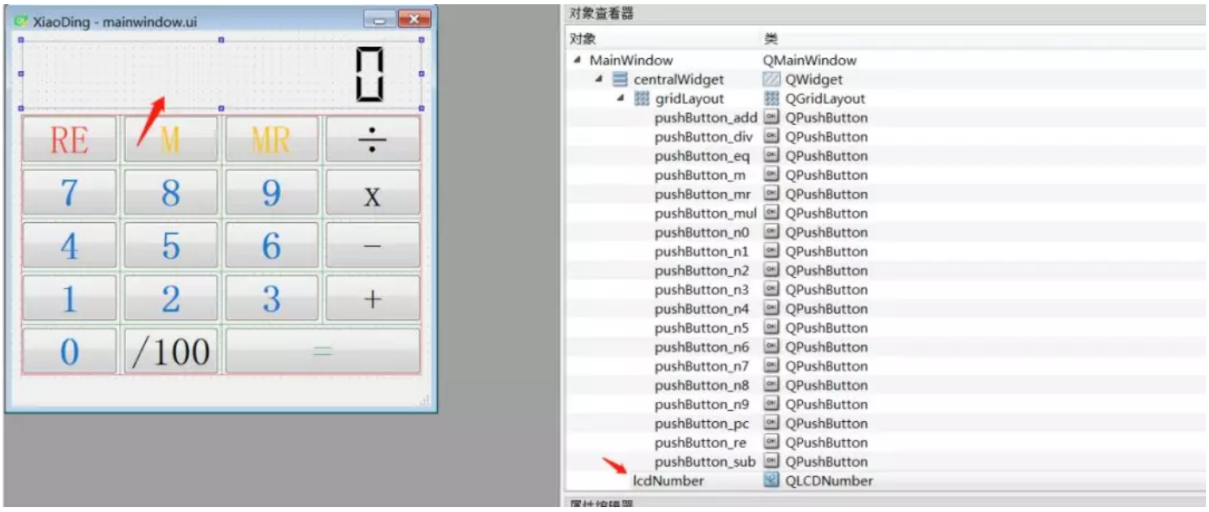
```
from PyQt5.QtGui import *
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *

import operator

from MainWindow import Ui_MainWindow
```

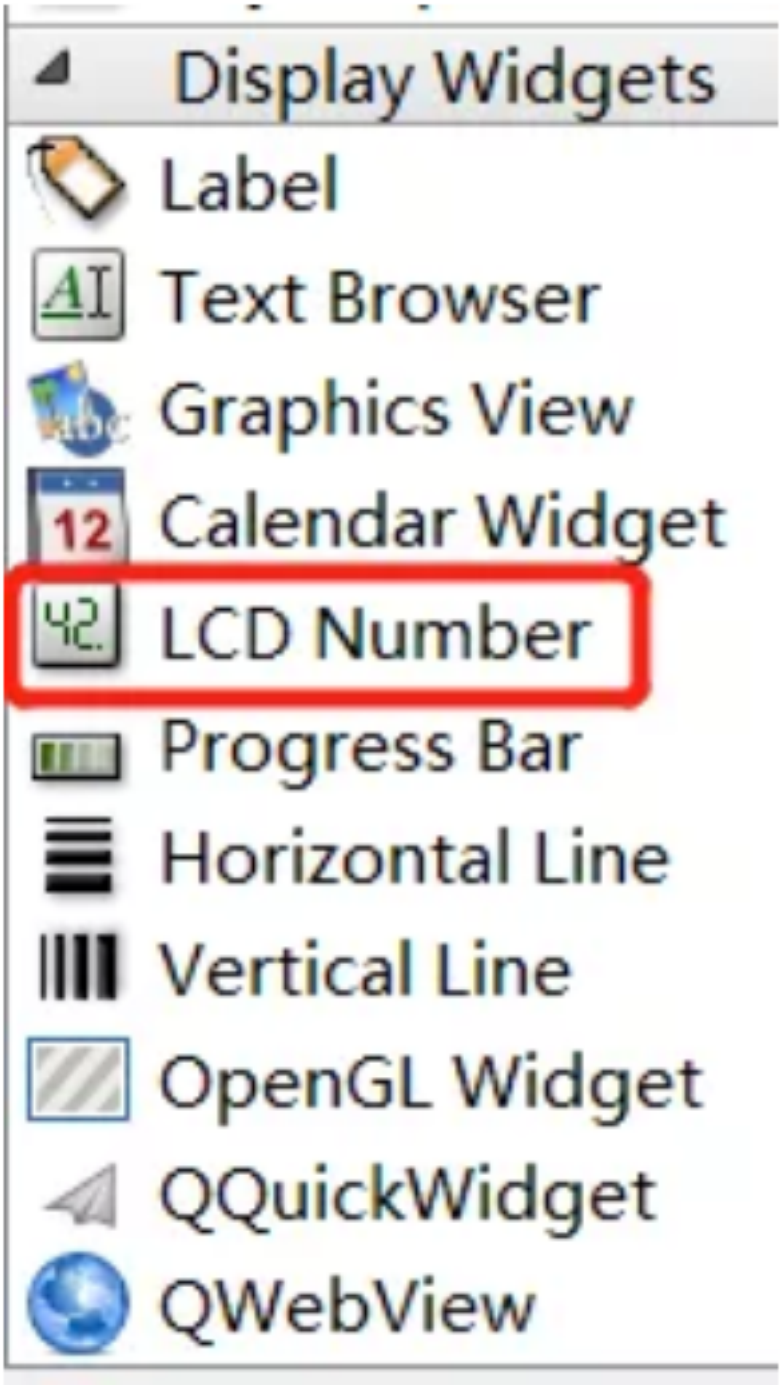
äÿžécŸäžčçăĄéĂżè¿Śå¿Ĺçš¿çőĂïijŽ

```
# Calculator state.
READY = 0
INPUT = 1


class MainWindow(QMainWindow, Ui_MainWindow):
    def __init__(self, *args, **kwargs):
```

(continues on next page)


XiaoDing - mainwindow.ui
0
RE
M
MR
÷
7
8
9
x
4
5
6
−
1
2
3
+
0
/100
=


XiaoDing - mainwindow.ui
对象查看器
对象
类
MainWindow
QMainWindow
centralWidget
QWidget
gridLayout
QGridLayout
pushButton_add
pushButton_div
pushButton_eq
pushButton_m
pushButton_mr
pushButton_mul
pushButton_n0
pushButton_n1
pushButton_n2
pushButton_n3
pushButton_n4
pushButton_n5
pushButton_n6
pushButton_n7
pushButton_n8
pushButton_n9
pushButton_pc
pushButton_re
pushButton_sub
lcdNumber
QPushButton
QLCDNumber


Display Widgets
Label
Text Browser
Graphics View
Calendar Widget
LCD Number
Progress Bar
Horizontal Line
Vertical Line
OpenGL Widget
QQuickWidget
QWebView

(continued from previous page)

```
        super(MainWindow, self).__init__(*args, **kwargs)
        self.setupUi(self)

        # Setup numbers.
        for n in range(0, 10):
            getattr(self, 'pushButton_n%s' % n).pressed.
↪connect(lambda v=n: self.input_number(v))

        # Setup operations.
        self.pushButton_add.pressed.connect(lambda: self.
↪operation(operator.add))
        self.pushButton_sub.pressed.connect(lambda: self.
↪operation(operator.sub))
        self.pushButton_mul.pressed.connect(lambda: self.
↪operation(operator.mul))
        self.pushButton_div.pressed.connect(lambda: self.
↪operation(operator.truediv))  # operator.div for Python2.7

        self.pushButton_pc.pressed.connect(self.operation_pc)
        self.pushButton_eq.pressed.connect(self.equals)

        # Setup actions
        self.actionReset.triggered.connect(self.reset)
        self.pushButton_ac.pressed.connect(self.reset)

        self.actionExit.triggered.connect(self.close)

        self.pushButton_m.pressed.connect(self.memory_store)
        self.pushButton_mr.pressed.connect(self.memory_recall)

        self.memory = 0
        self.reset()

        self.show()
```

å§žçąĂæŰźæşŢïijŽ

```
def input_number(self, v):
    if self.state == READY:
        self.state = INPUT
        self.stack[-1] = v
    else:
        self.stack[-1] = self.stack[-1] * 10 + v

    self.display()

def display(self):
    self.lcdNumber.display(self.stack[-1])
```

æŇL'éŠőRE,M, REářźåžŤçŽĎåőđçŐřéĂżè¿Śïij Ž

```
def reset(self):
    self.state = READY
    self.stack = [0]
    self.last_operation = None
    self.current_op = None
    self.display()

def memory_store(self):
    self.memory = self.lcdNumber.value()

def memory_recall(self):
    self.state = INPUT
    self.stack[-1] = self.memory
    self.display()
```

+,−,x,/,/100åŕźåžŤåőđçŐřæŰźæşŢïijŽ

```
def operation(self, op):
        if self.current_op:  # Complete the current operation
            self.equals()

        self.stack.append(0)
        self.state = INPUT
        self.current_op = op

    def operation_pc(self):
        self.state = INPUT
        self.stack[-1] *= 0.01
        self.display()
```

=åŔůåŕźåžŤçŽĎæŰźæşŢåőđçŐřïijŽ

```
def equals(self):
       if self.state == READY and self.last_operation:
           s, self.current_op = self.last_operation
           self.stack.append(s)

       if self.current_op:
           self.last_operation = self.stack[-1], self.current_op

           try:
               self.stack = [self.current_op(*self.stack)]
           except Exception:
               self.lcdNumber.display('Err')
               self.stack = [0]
           else:
               self.current_op = None
               self.state = READY
               self.display()
```

mainåĞ¡æŢřïijŽ

```
if __name__ == '__main__':
    app = QApplication([])
    app.setApplicationName("XiaoDing")

    window = MainWindow()
    app.exec_()
```

CHAPTER 14

åĚşäžŐ

Python åřŕä¿Ńå■Ř

äžžçŤ§èŃęç§■ïijŇæĹŚçŤĺPythonïijĄ

å■ęäźžååřŕä¿Ńå■Řïij ŇæŻťæŸŕåćđæůżæřŤæřŻ

çěİä¡ăå■ęäźžååřŕä¿Ńå■Řå£ńäźŘ~~

æňćè£ŐåĚşæşĺãĂŁPythonåřŕä¿Ńå■ŘãĂŃåőŸæŰźåĚňäijŮåŕůïijŇæŕŕåď'l'çňňäÿĂæŮűéŮť'èŐůåŕŰæI

åęĆæđIJè£ŹäžŻåřŕä¿Ńå■ŘïijŇåŕźä¡ăæIJL'åÿőåŁl'ïijŇèŕůæĹŚåŰİæİŕåŠŰåŢą~~
è£ŹæăůæĹŚæŻť'æIJL'åŁĺåŁŻïijŇåĘŹåĞžæŻť'åď'Žåě¡çŽĎåřŕä¿Ńå■ŘãĂĆäżżæĎŕéĞŚéćİïijŇéČ¡æŸŕéijŞåŁŚ

推荐使用微信支付

Emily (**霞)

微信支付