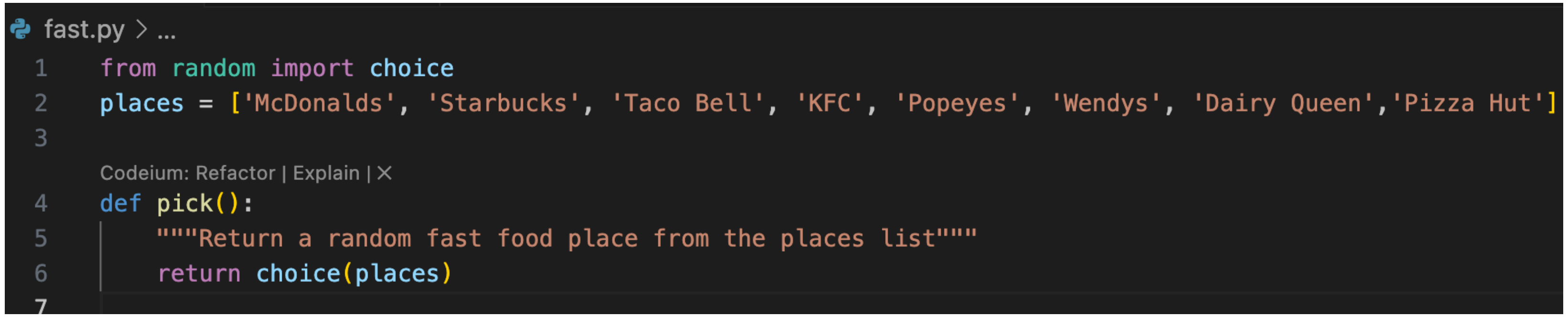
Python
Files, Exception Handling, and Modules

馬誠佑

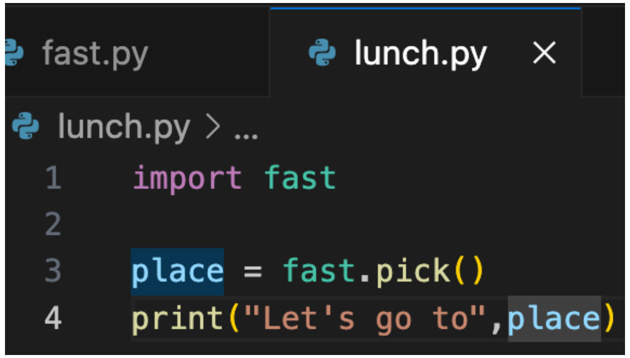
2025/04/25

Module & Package

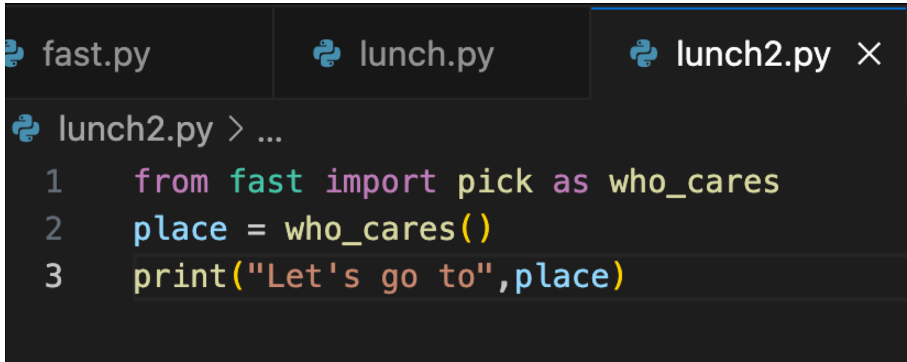
- Module: 存在有任意python程式碼的檔案，任何python程式碼都可以當成模組被其他程式碼使用。
- 匯入模組：import module，其中的module為其他Python檔檔名，不包含副檔名.py



```python
from random import choice
places = ['McDonalds', 'Starbucks', 'Taco Bell', 'KFC', 'Popeyes', 'Wendys', 'Dairy Queen','Pizza Hut']

def pick():
    """Return a random fast food place from the places list"""
    return choice(places)

```

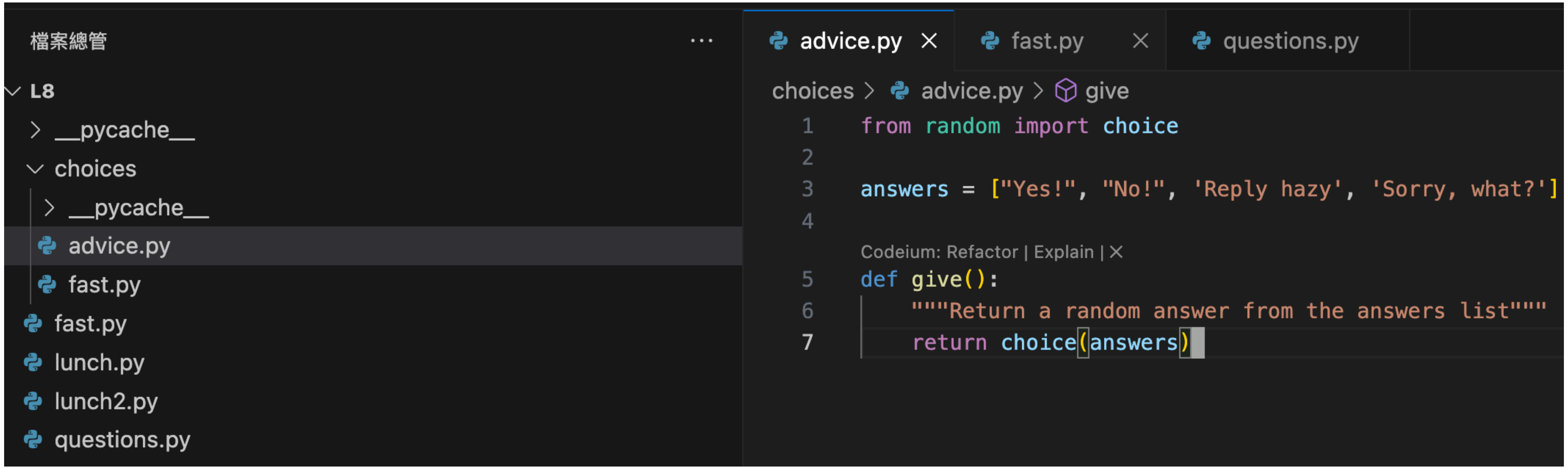


```python
import fast

place = fast.pick()
print("Let's go to",place)
```



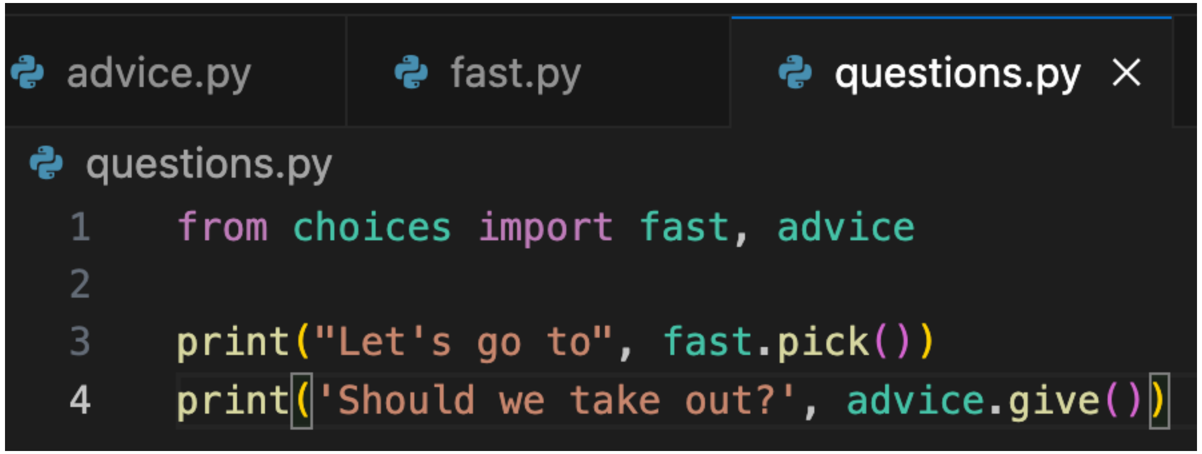
```python
from fast import pick as who_cares
place = who_cares()
print("Let's go to",place)
```

Package



```
L8
  __pycache__
  choices
    __pycache__
    advice.py
    fast.py
  fast.py
  lunch.py
  lunch2.py
  questions.py
```

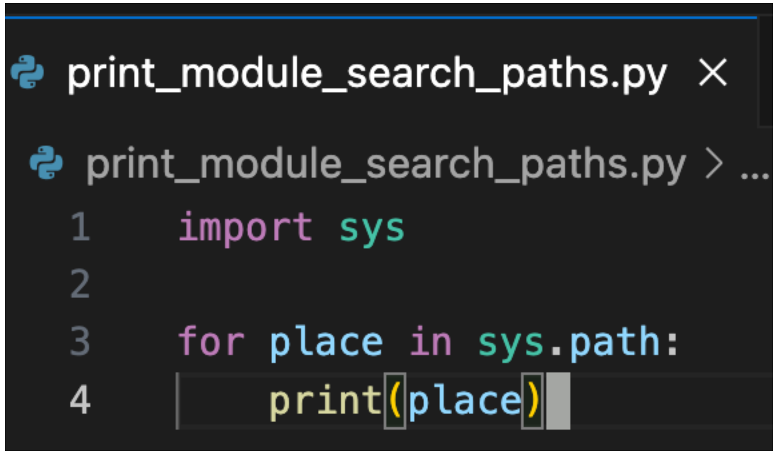
```python
from random import choice

answers = ["Yes!", "No!", 'Reply hazy', 'Sorry, what?']

def give():
    """Return a random answer from the answers list"""
    return choice(answers)
```



```python
from choices import fast, advice

print("Let's go to", fast.pick())
print('Should we take out?', advice.give())
```

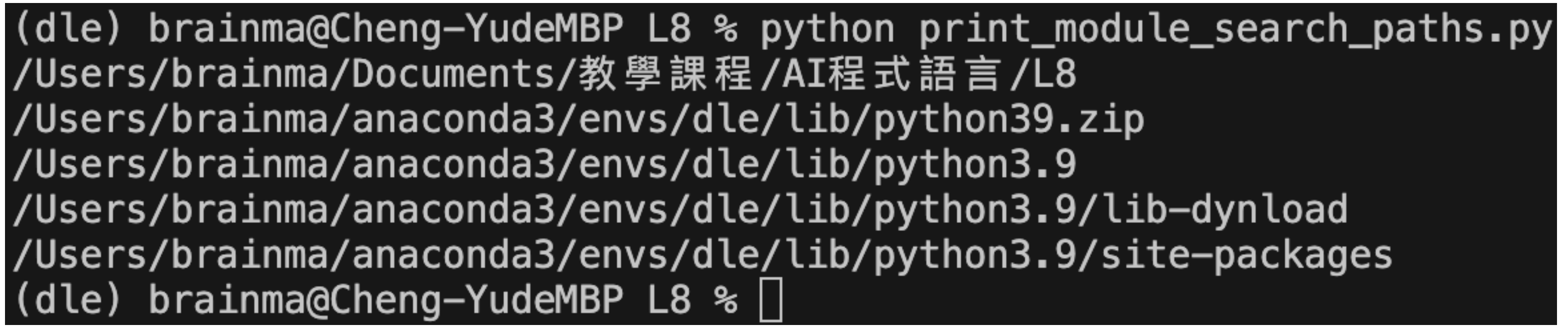
Module Searching Path

- sys.path: 所有直譯器會尋找模組的路徑list，可讀取或修改該串列。



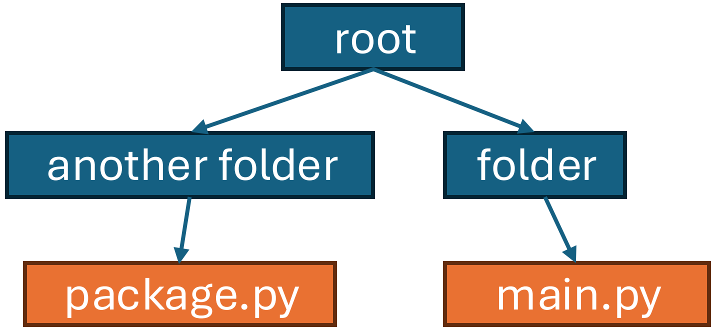
print_module_search_paths.py ×
print_module_search_paths.py > ...

```python
import sys

for place in sys.path:
    print(place)
```

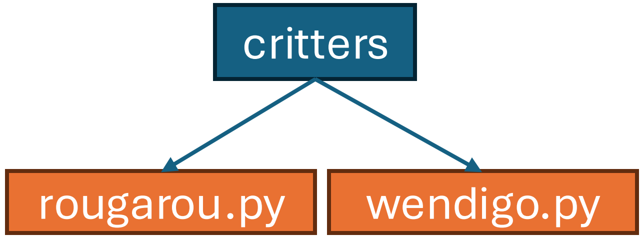


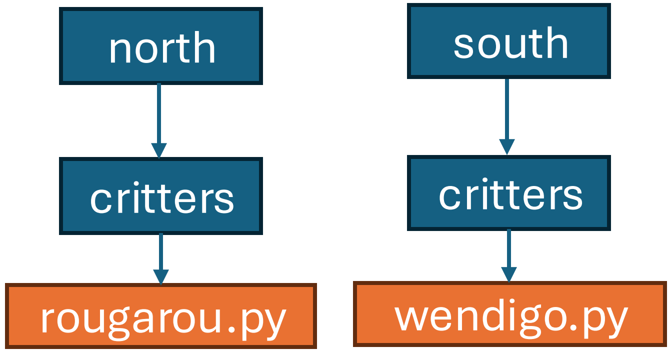
```
(dle) brainma@Cheng-YudeMBP L8 % python print_module_search_paths.py
/Users/brainma/Documents/教學課程/AI程式語言/L8
/Users/brainma/anaconda3/envs/dle/lib/python39.zip
/Users/brainma/anaconda3/envs/dle/lib/python3.9
/Users/brainma/anaconda3/envs/dle/lib/python3.9/lib-dynload
/Users/brainma/anaconda3/envs/dle/lib/python3.9/site-packages
(dle) brainma@Cheng-YudeMBP L8 % 
```

相對與絕對匯入



- 若package.py 與呼叫的程式在同一個目錄，可以用from . import package來匯入
- 若是在上一層目錄可以用from .. import package來匯入
- 若是在名為 anotherFolder的同層資料夾，可以用 from ..anotherFolder import package



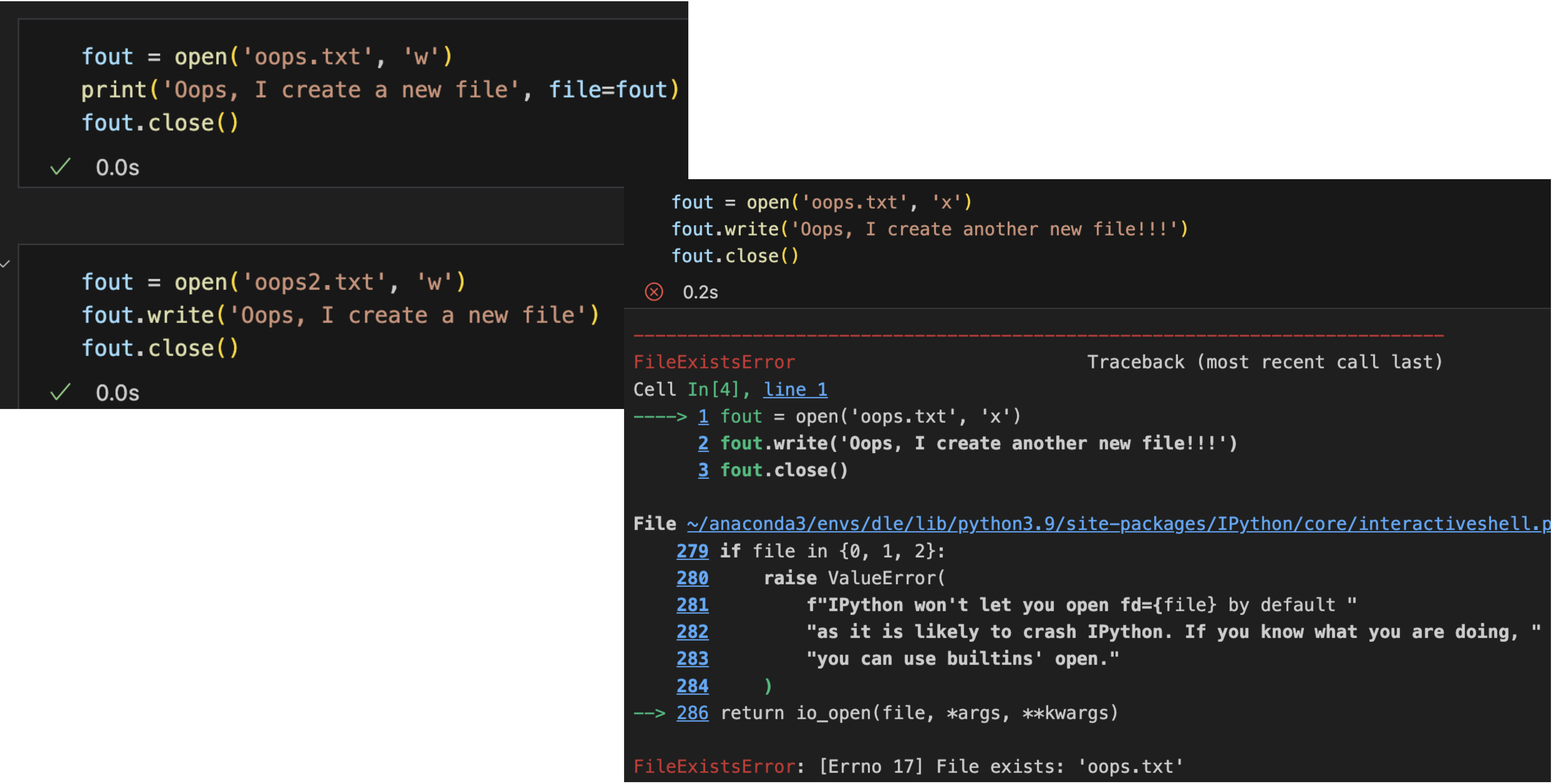


from critters import rougarou, wendigo

from critters import rougarou, wendigo => 正確 只要 north 跟south資料夾都有在sys.path中

File input & output

- 用open()建立或開啟ex: fileObj = open(filename, mode)
- mode可選擇
  - ‘r’ 讀取
  - ‘w’ 寫入，如果檔案不存在，python會建立它。如果檔案存在，python會覆寫它。
  - ‘x’寫入，但只有在檔案還不存在時。
  - ‘a’ 附加，在檔案原本結尾處寫入（如果檔案存在的話。）
  - ‘t’ or 無，寫成文字檔。（只能放在第二個字元）
  - ‘b’ 寫成二進制檔。（只能放在第二個字元）
- 最後寫入或讀取完後需要呼叫 fileObj.close()來確保完成任何寫入動作與釋放記憶體。



```
fout = open('oops.txt', 'w')
print('Oops, I create a new file', file=fout)
fout.close()
```

✓ 0.0s

```
fout = open('oops2.txt', 'w')
fout.write('Oops, I create a new file')
fout.close()
```

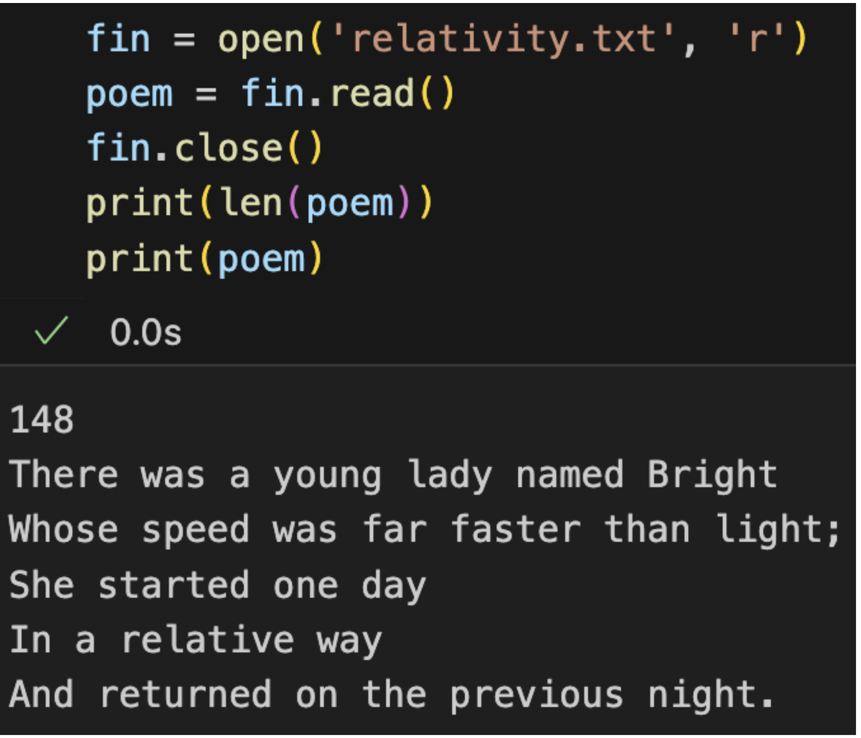
✓ 0.0s

```
fout = open('oops.txt', 'x')
fout.write('Oops, I create another new file!!!')
fout.close()
```

⊗ 0.2s

```
---------------------------------------------------------------------------
FileExistsError                           Traceback (most recent call last)
Cell In[4], line 1
----> 1 fout = open('oops.txt', 'x')
      2 fout.write('Oops, I create another new file!!!')
      3 fout.close()

File ~/anaconda3/envs/dle/lib/python3.9/site-packages/IPython/core/interactiveshell.p
    279 if file in {0, 1, 2}:
    280     raise ValueError(
    281         f"IPython won't let you open fd={file} by default "
    282         "as it is likely to crash IPython. If you know what you are doing, "
    283         "you can use builtins' open."
    284     )
--> 286 return io_open(file, *args, **kwargs)

FileExistsError: [Errno 17] File exists: 'oops.txt'
```

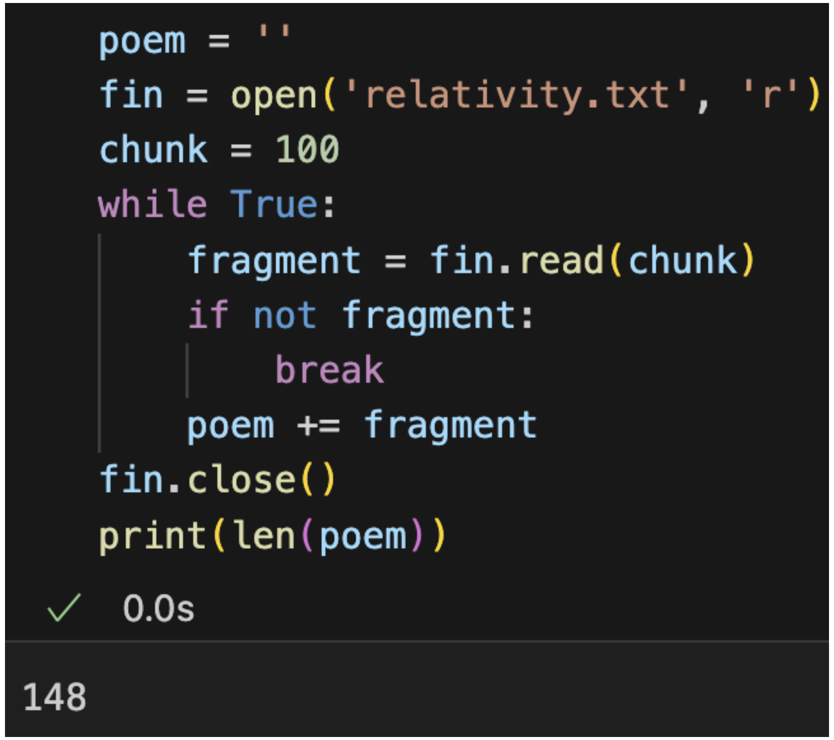
read()、readline()、readlines()



```python
fin = open('relativity.txt', 'r')
poem = fin.read()
fin.close()
print(len(poem))
print(poem)
```

✓ 0.0s

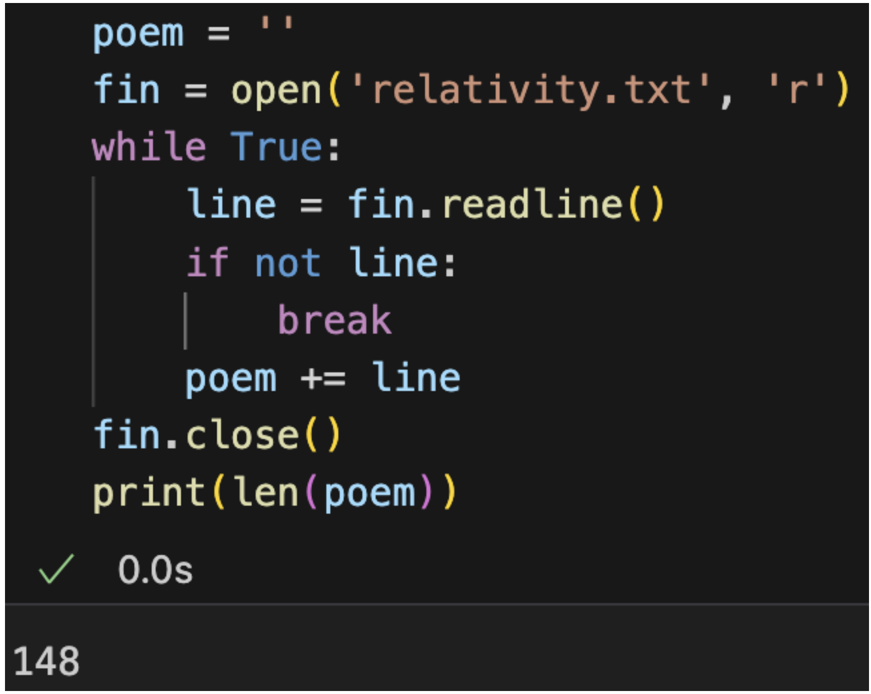
```
148
There was a young lady named Bright
Whose speed was far faster than light;
She started one day
In a relative way
And returned on the previous night.
```



```python
poem = ''
fin = open('relativity.txt', 'r')
chunk = 100
while True:
    fragment = fin.read(chunk)
    if not fragment:
        break
    poem += fragment
fin.close()
print(len(poem))
```

✓ 0.0s

```
148
```

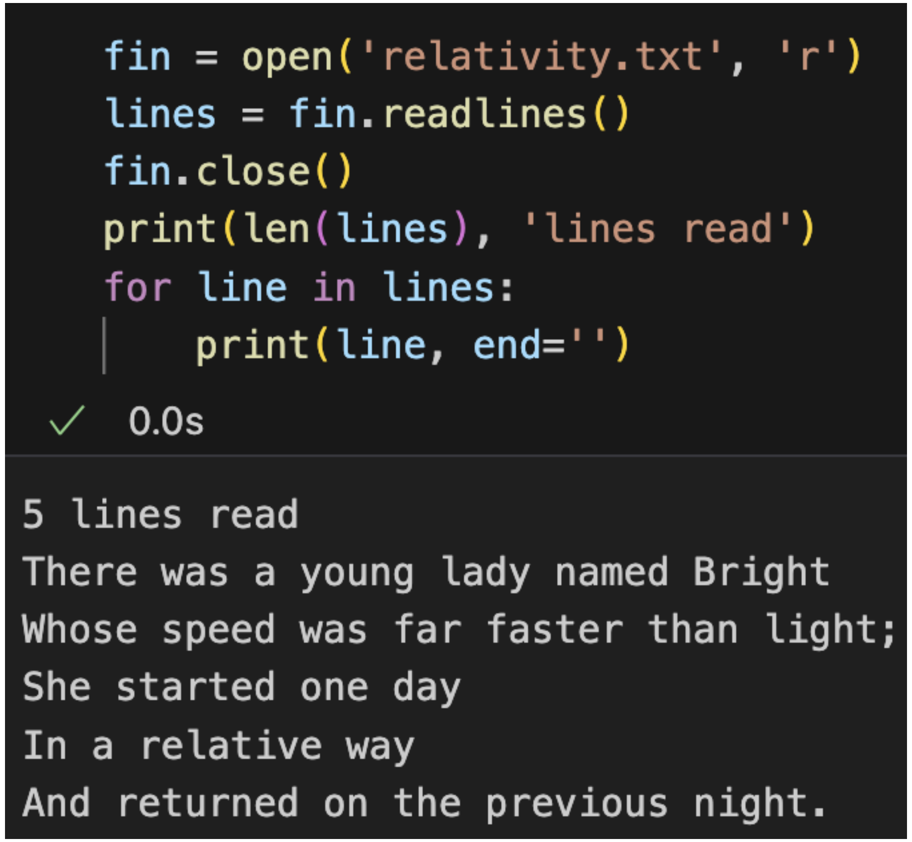


```python
poem = ''
fin = open('relativity.txt', 'r')
while True:
    line = fin.readline()
    if not line:
        break
    poem += line
fin.close()
print(len(poem))
```

✓ 0.0s

```
148
```

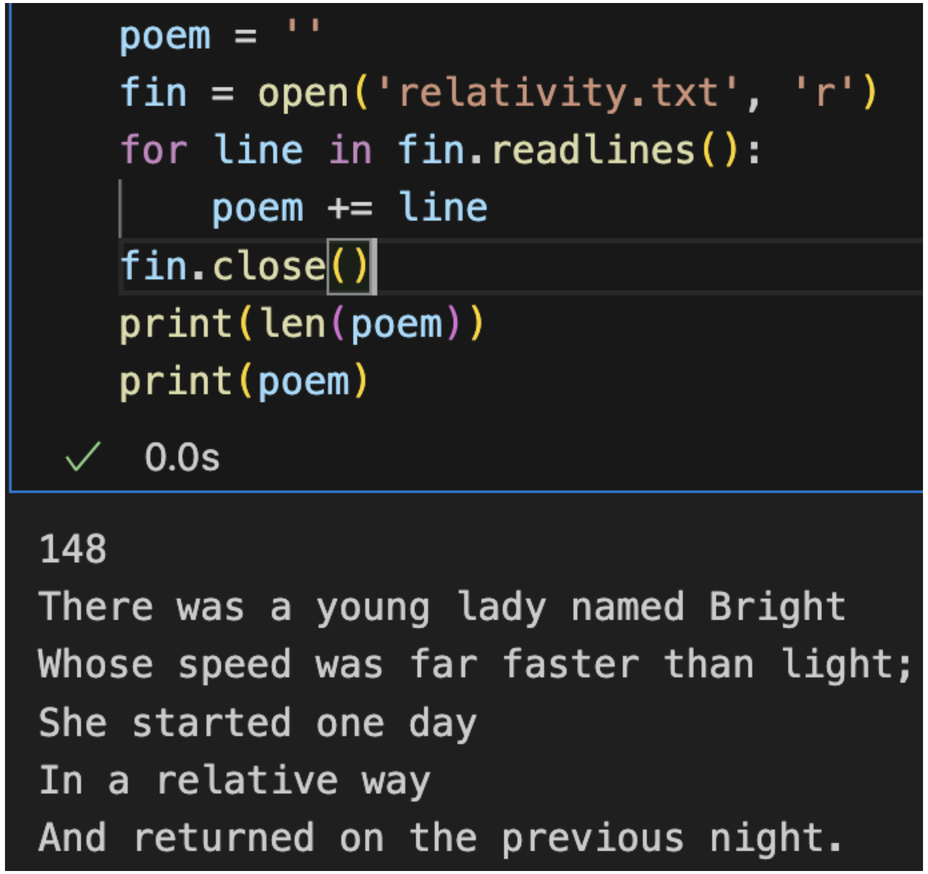
read()、readline()、readlines()-2



```python
fin = open('relativity.txt', 'r')
lines = fin.readlines()
fin.close()
print(len(lines), 'lines read')
for line in lines:
    print(line, end='')
```

```
✓  0.0s
```

```
5 lines read
There was a young lady named Bright
Whose speed was far faster than light;
She started one day
In a relative way
And returned on the previous night.
```

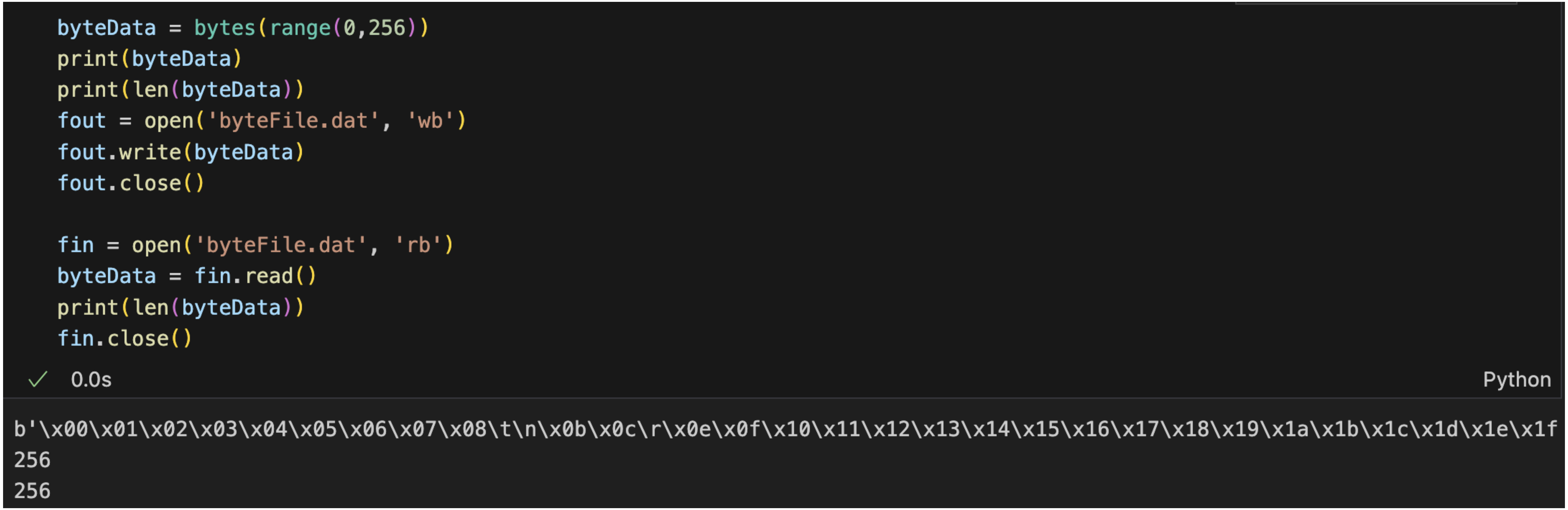


```python
poem = ''
fin = open('relativity.txt', 'r')
for line in fin.readlines():
    poem += line
fin.close()
print(len(poem))
print(poem)
```

```
✓  0.0s
```

```
148
There was a young lady named Bright
Whose speed was far faster than light;
She started one day
In a relative way
And returned on the previous night.
```

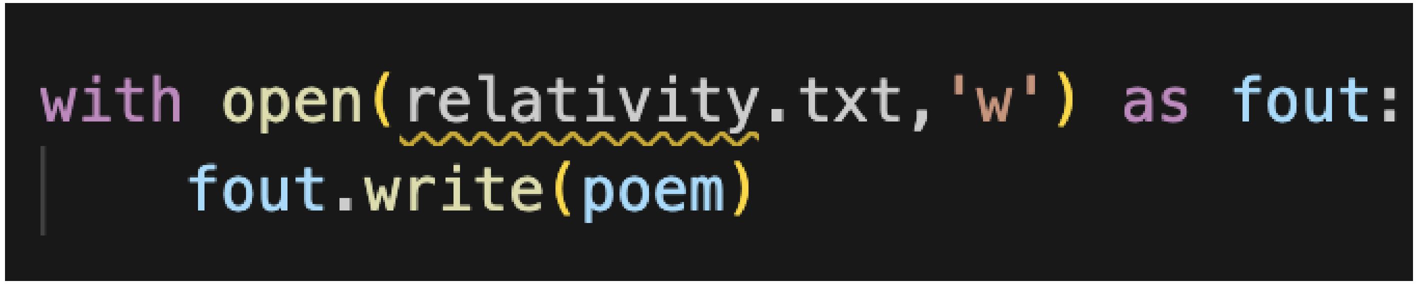
write()、read() 二進制檔案



```python
byteData = bytes(range(0,256))
print(byteData)
print(len(byteData))
fout = open('byteFile.dat', 'wb')
fout.write(byteData)
fout.close()

fin = open('byteFile.dat', 'rb')
byteData = fin.read()
print(len(byteData))
fin.close()
```

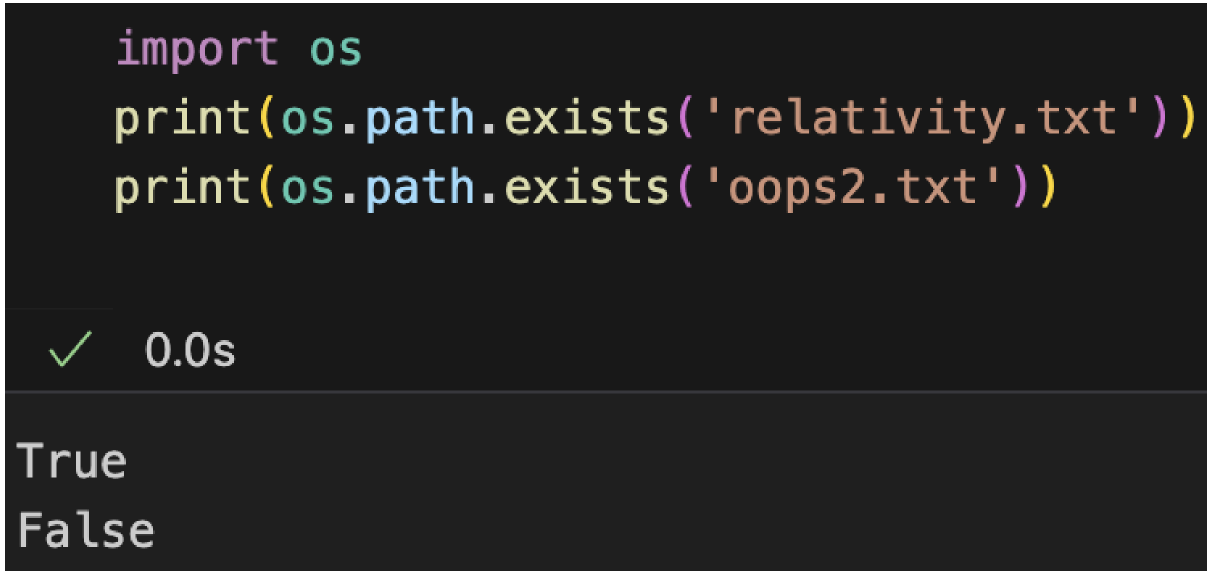
```
b'\x00\x01\x02\x03\x04\x05\x06\x07\x08\t\n\x0b\x0c\r\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f
256
256
```

用with來自動關閉檔案



```
with open(relativity.txt,'w') as fout:
    fout.write(poem)
```

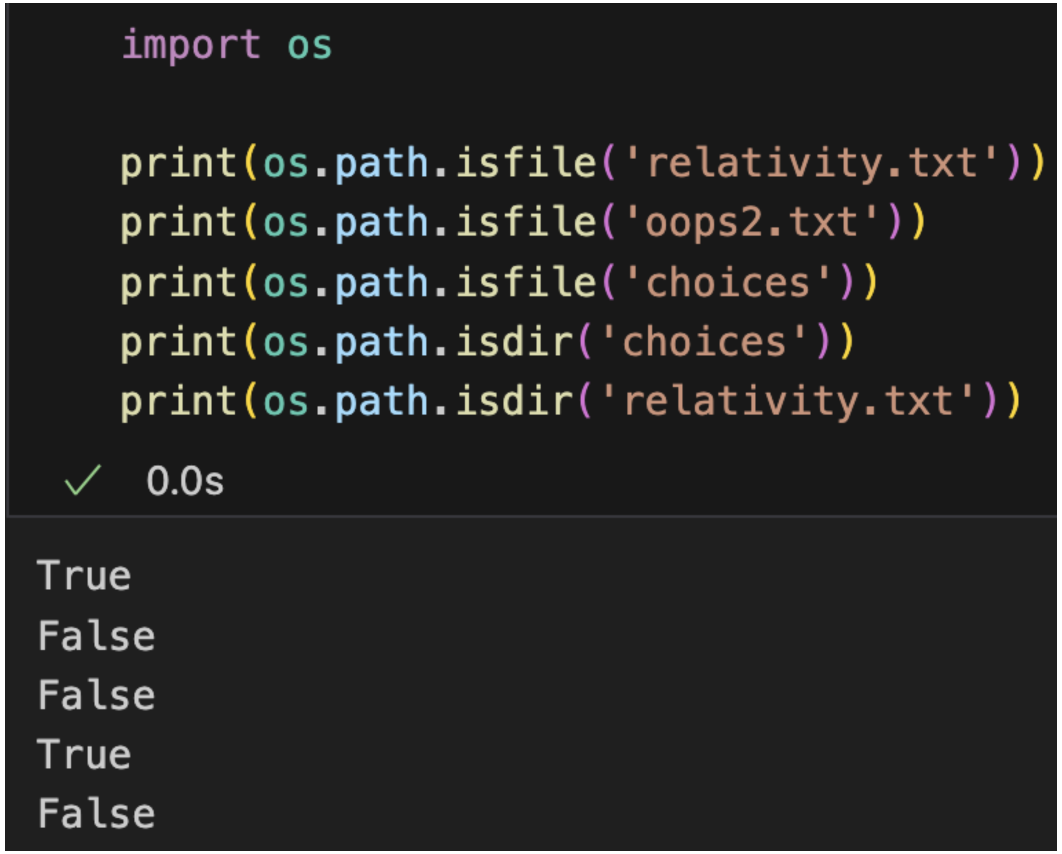
用exists()來檢測檔案是否存在



```python
import os
print(os.path.exists('relativity.txt'))
print(os.path.exists('oops2.txt'))
```

```
True
False
```

用isfile()、isdir()來確認型態

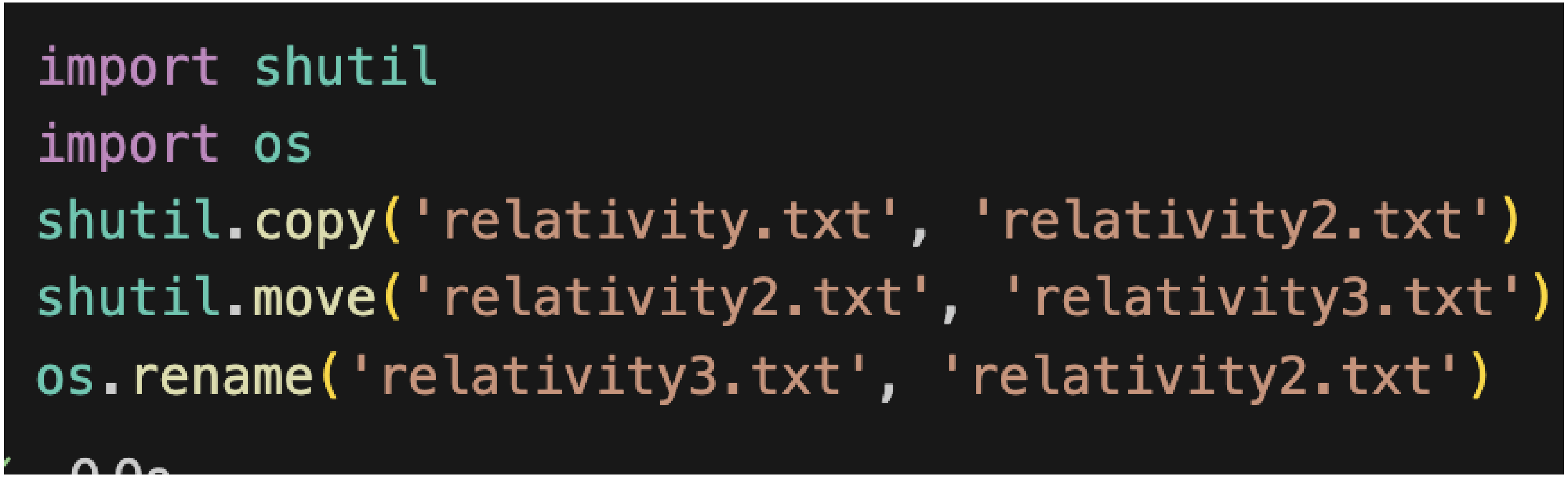


```
import os

print(os.path.isfile('relativity.txt'))
print(os.path.isfile('oops2.txt'))
print(os.path.isfile('choices'))
print(os.path.isdir('choices'))
print(os.path.isdir('relativity.txt'))
```

✓ 0.0s

```
True
False
False
True
False
```

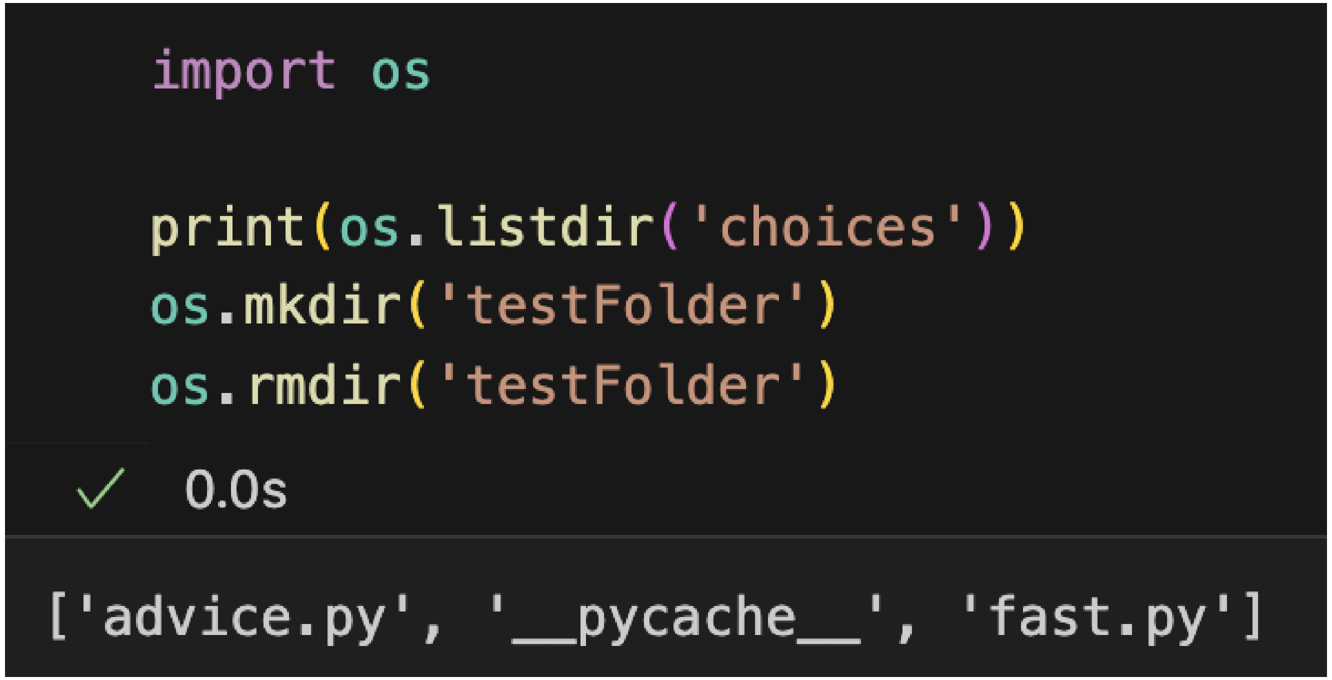
copy()、move()、rename()



```python
import shutil
import os
shutil.copy('relativity.txt', 'relativity2.txt')
shutil.move('relativity2.txt', 'relativity3.txt')
os.rename('relativity3.txt', 'relativity2.txt')
```

```
✓ 0.0s
```

listdir()、mkdir()、rmdir()

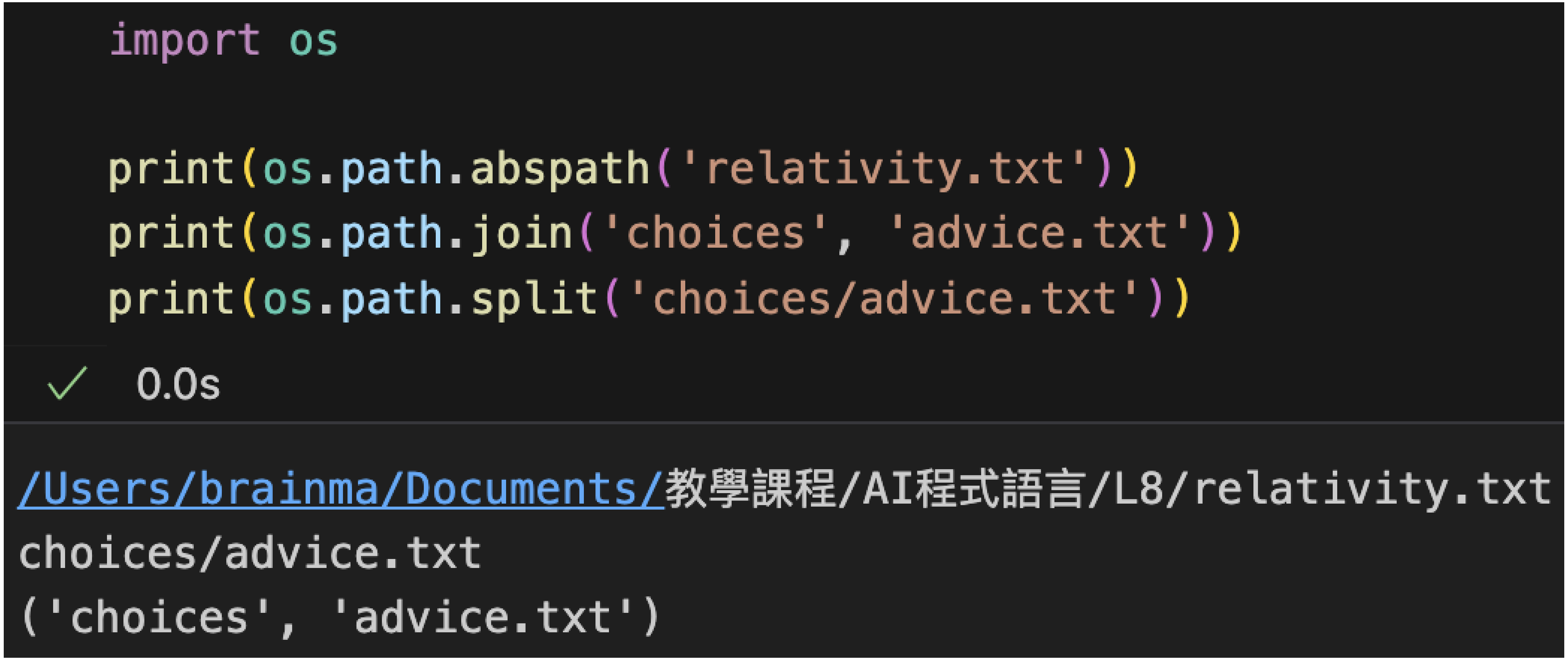


```python
import os

print(os.listdir('choices'))
os.mkdir('testFolder')
os.rmdir('testFolder')
```

✓ 0.0s

```
['advice.py', '__pycache__', 'fast.py']
```

abspath()、os.path.join()、os.path.split()

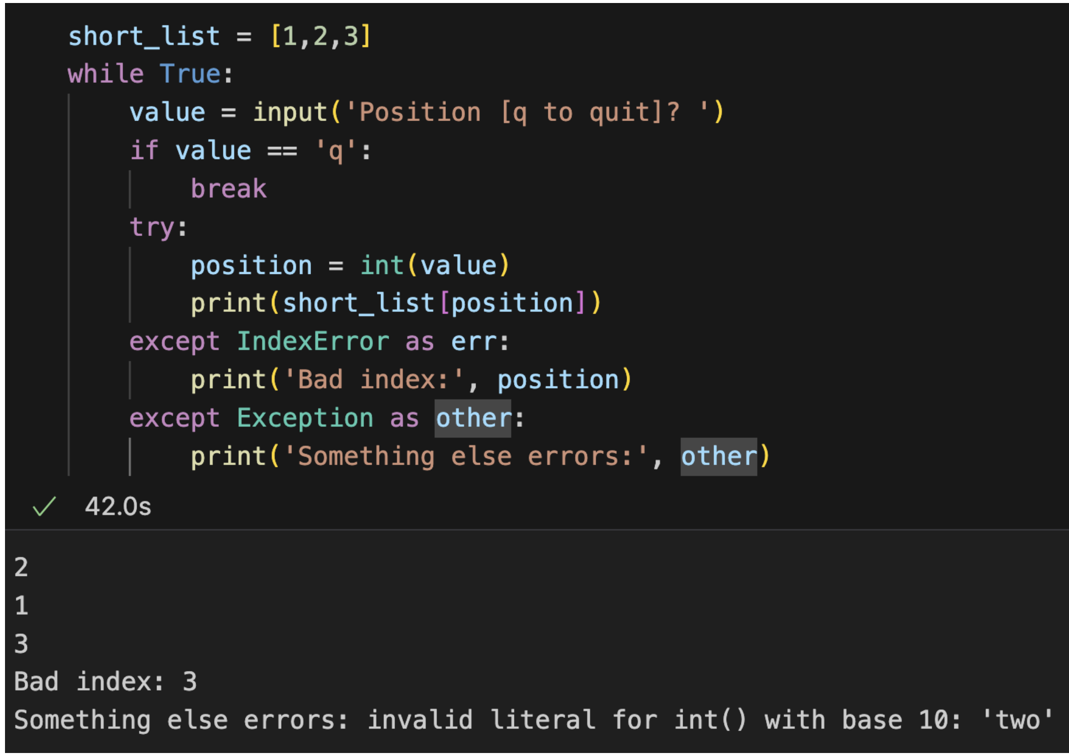


```python
import os

print(os.path.abspath('relativity.txt'))
print(os.path.join('choices', 'advice.txt'))
print(os.path.split('choices/advice.txt'))
```

```
/Users/brainma/Documents/教學課程/AI程式語言/L8/relativity.txt
choices/advice.txt
('choices', 'advice.txt')
```

Error handle: try and except

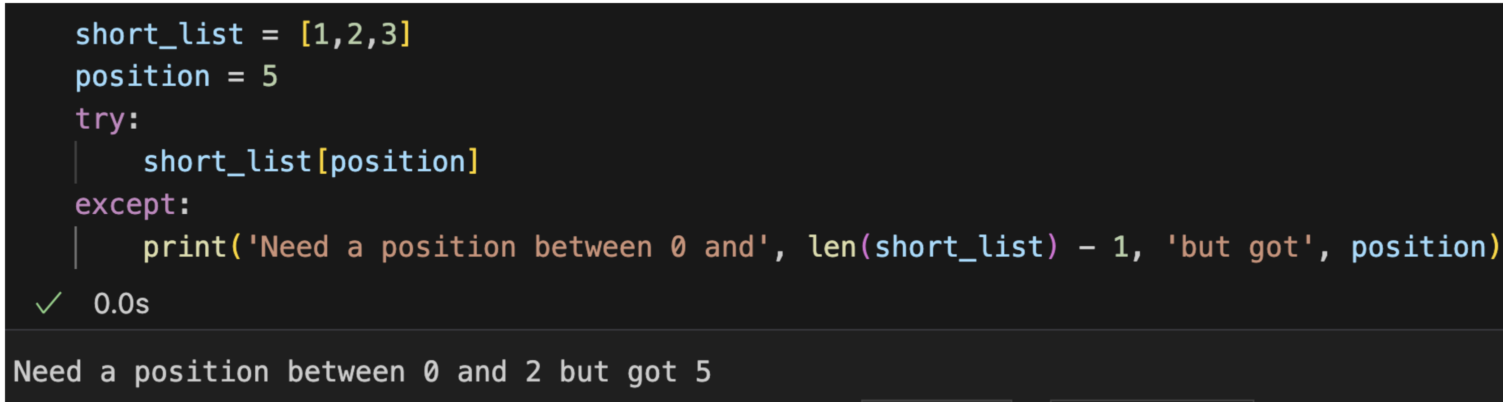
```
try:
        .....
except <ExceptionType1>:
        <handler1>
...
except <ExceptionTypeN>:
        <handlerN>
except:
        <handlerExcept> #有發生上面沒定義的error時執行
else:
        <process_else> #沒有任何error發生時執行
finally:
        <process_finally> #一定會執行
```



```python
short_list = [1,2,3]
while True:
    value = input('Position [q to quit]? ')
    if value == 'q':
        break
    try:
        position = int(value)
        print(short_list[position])
    except IndexError as err:
        print('Bad index:', position)
    except Exception as other:
        print('Something else errors:', other)
```

✓ 42.0s

```
2
1
3
Bad index: 3
Something else errors: invalid literal for int() with base 10: 'two'
```



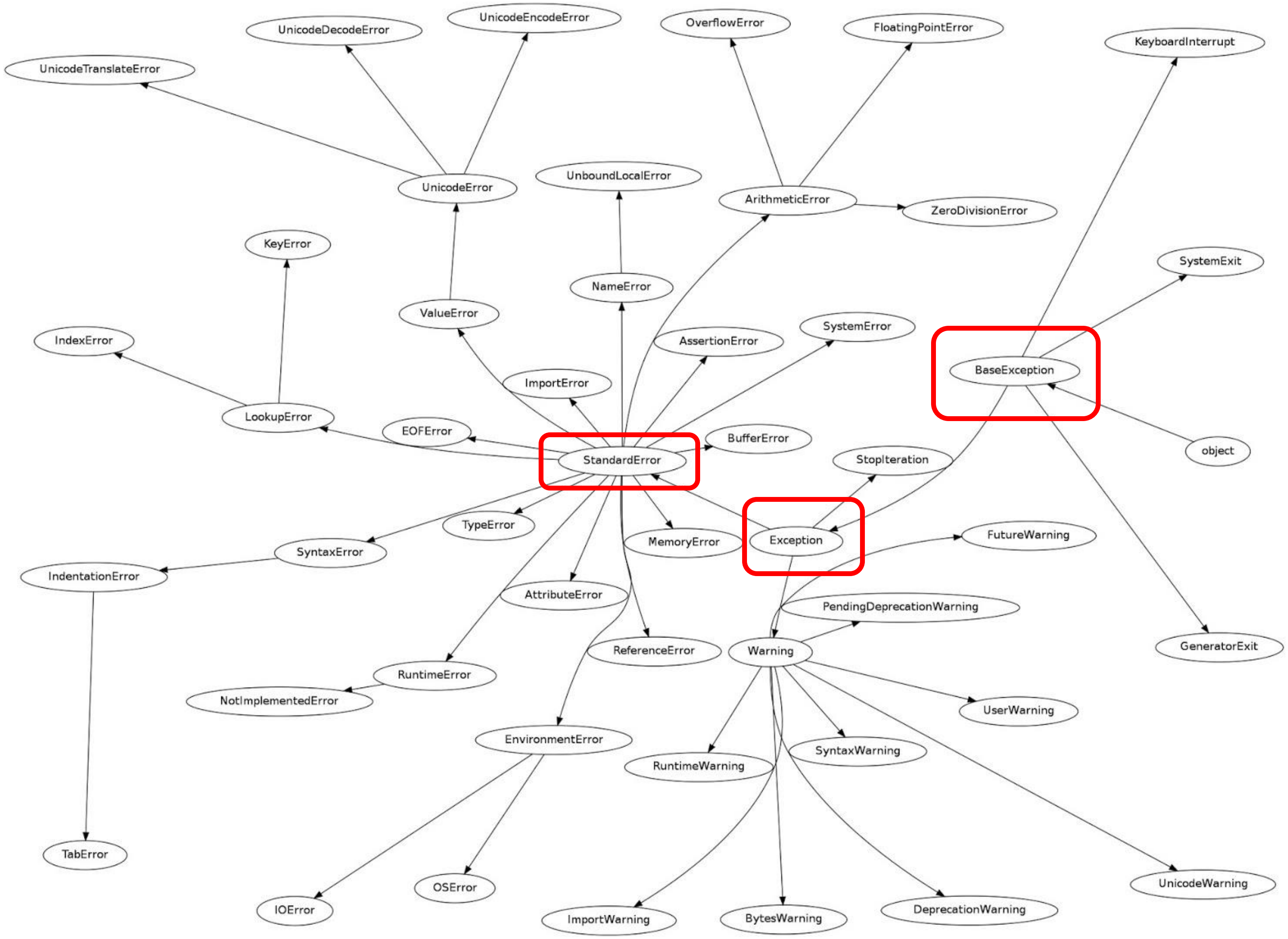
```python
short_list = [1,2,3]
position = 5
try:
    short_list[position]
except:
    print('Need a position between 0 and', len(short_list) - 1, 'but got', position)
```

✓ 0.0s

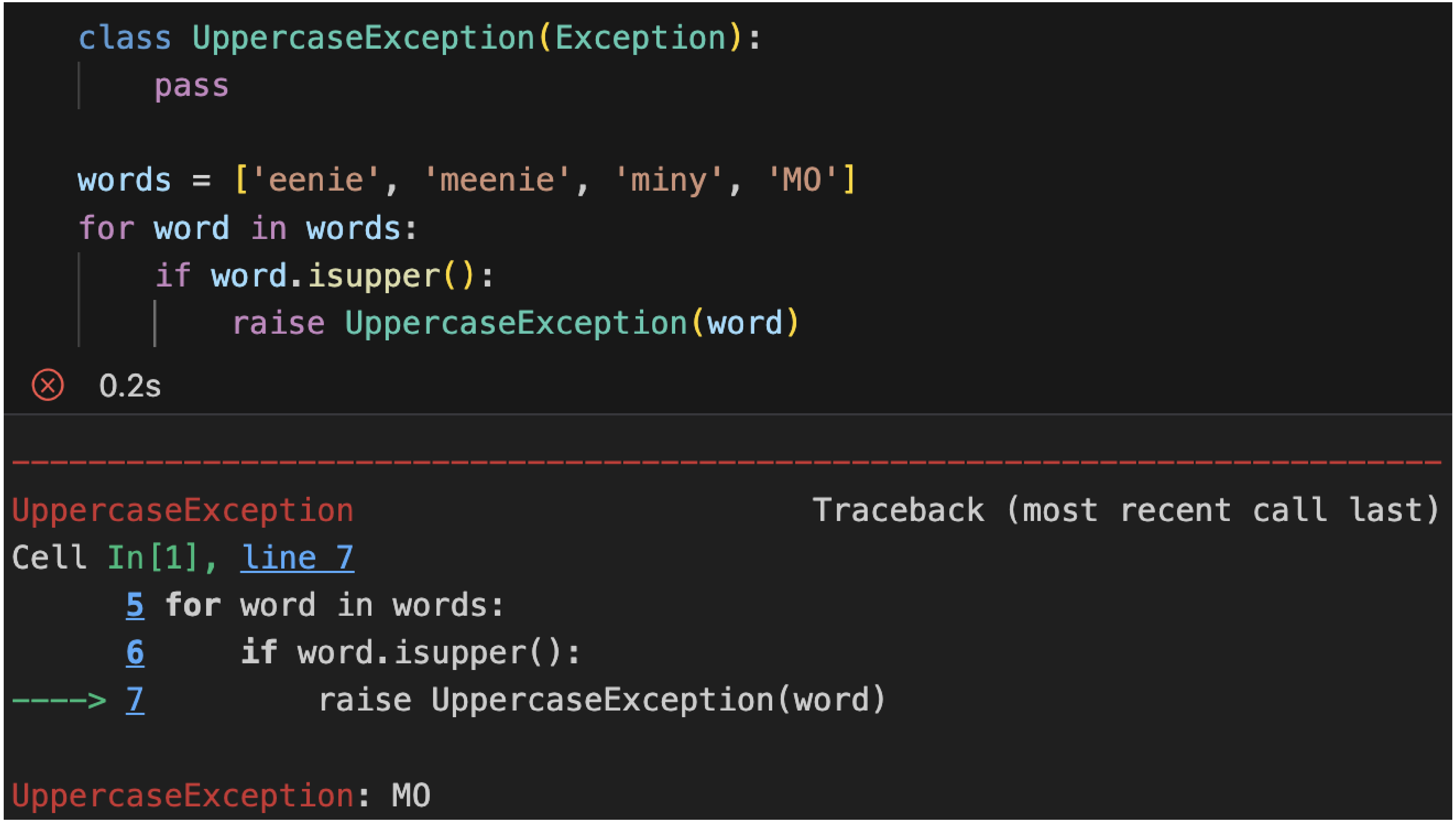
```
Need a position between 0 and 2 but got 5
```

Built-in Exception

亦可定義你自己的ERROR TYPE!



定義自己的例外



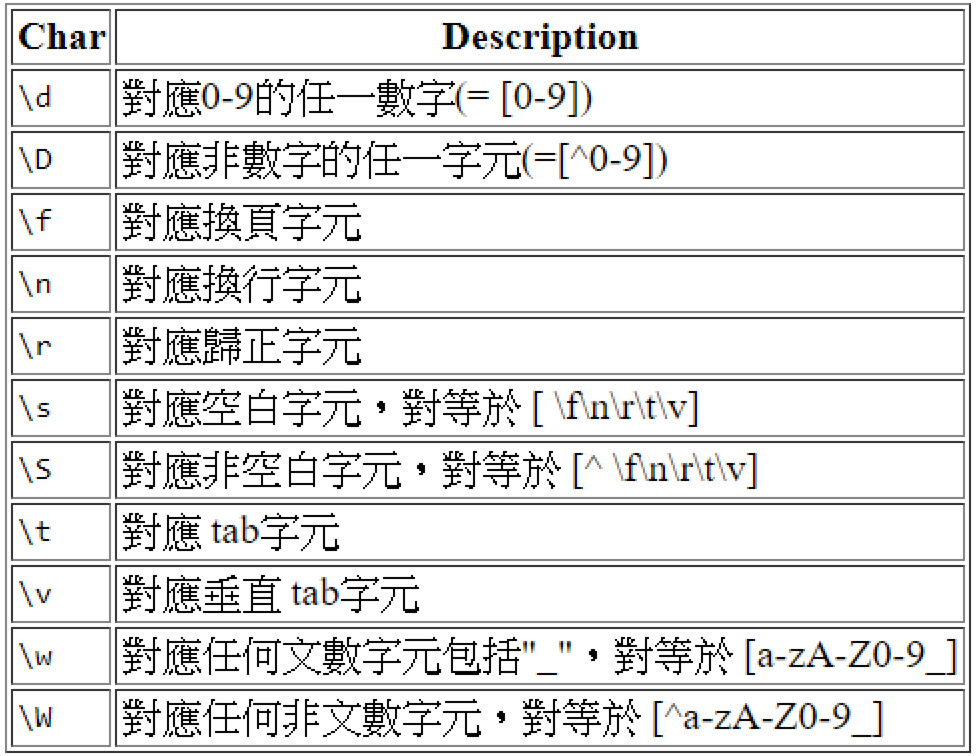
```python
class UppercaseException(Exception):
    pass

words = ['eenie', 'meenie', 'miny', 'MO']
for word in words:
    if word.isupper():
        raise UppercaseException(word)
```

ⓧ 0.2s

```
---------------------------------------------------------------------------
UppercaseException                        Traceback (most recent call last)
Cell In[1], line 7
      5 for word in words:
      6     if word.isupper():
----> 7         raise UppercaseException(word)

UppercaseException: MO
```

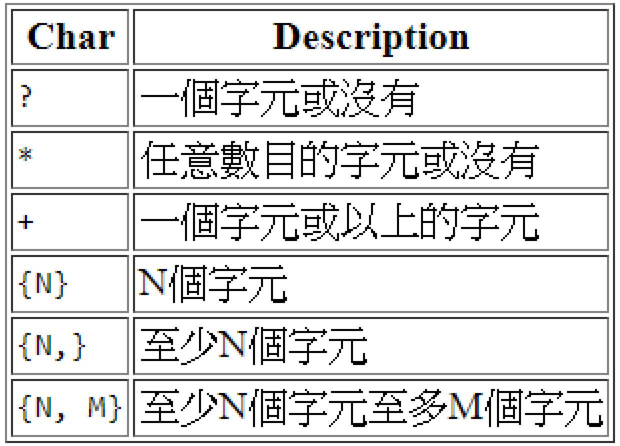
Regular Expression (re) 正規表示式

特殊字元(Metacharacter)—Shorthand Metacharacter



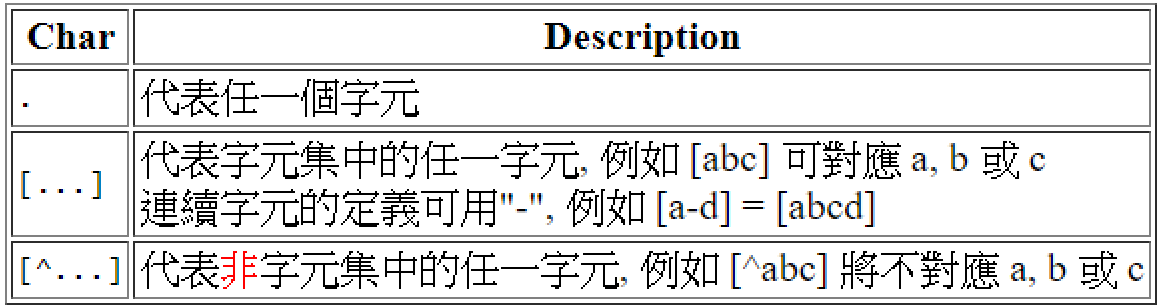
Char	Description
\d	對應0-9的任一數字(= [0-9])
\D	對應非數字的任一字元(=[^0-9])
\f	對應換頁字元
\n	對應換行字元
\r	對應歸正字元
\s	對應空白字元，對等於 [\f\n\r\t\v]
\S	對應非空白字元，對等於 [^ \f\n\r\t\v]
\t	對應 tab字元
\v	對應垂直 tab字元
\w	對應任何文數字元包括"_"，對等於 [a-zA-Z0-9_]
\W	對應任何非文數字元，對等於 [^a-zA-Z0-9_]

數量定義詞(Quantifier)—定義前一個字元的數量

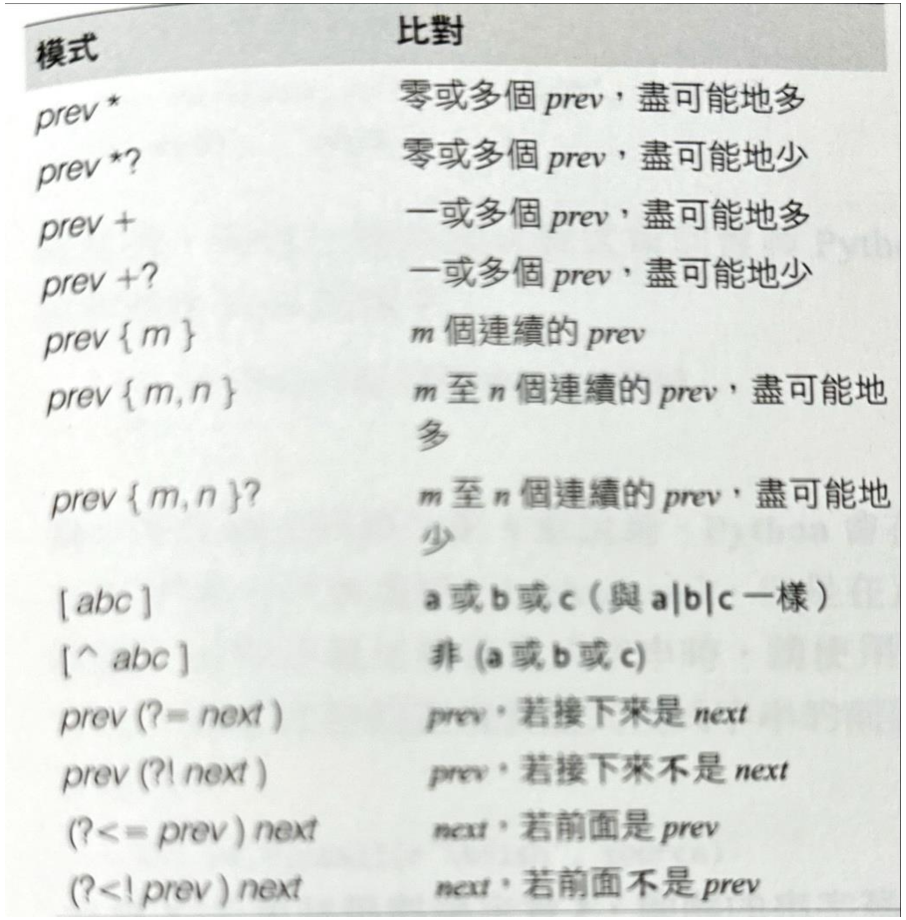


Char	Description
?	一個字元或沒有
*	任意數目的字元或沒有
+	一個字元或以上的字元
{N}	N個字元
{N,}	至少N個字元
{N, M}	至少N個字元至多M個字元

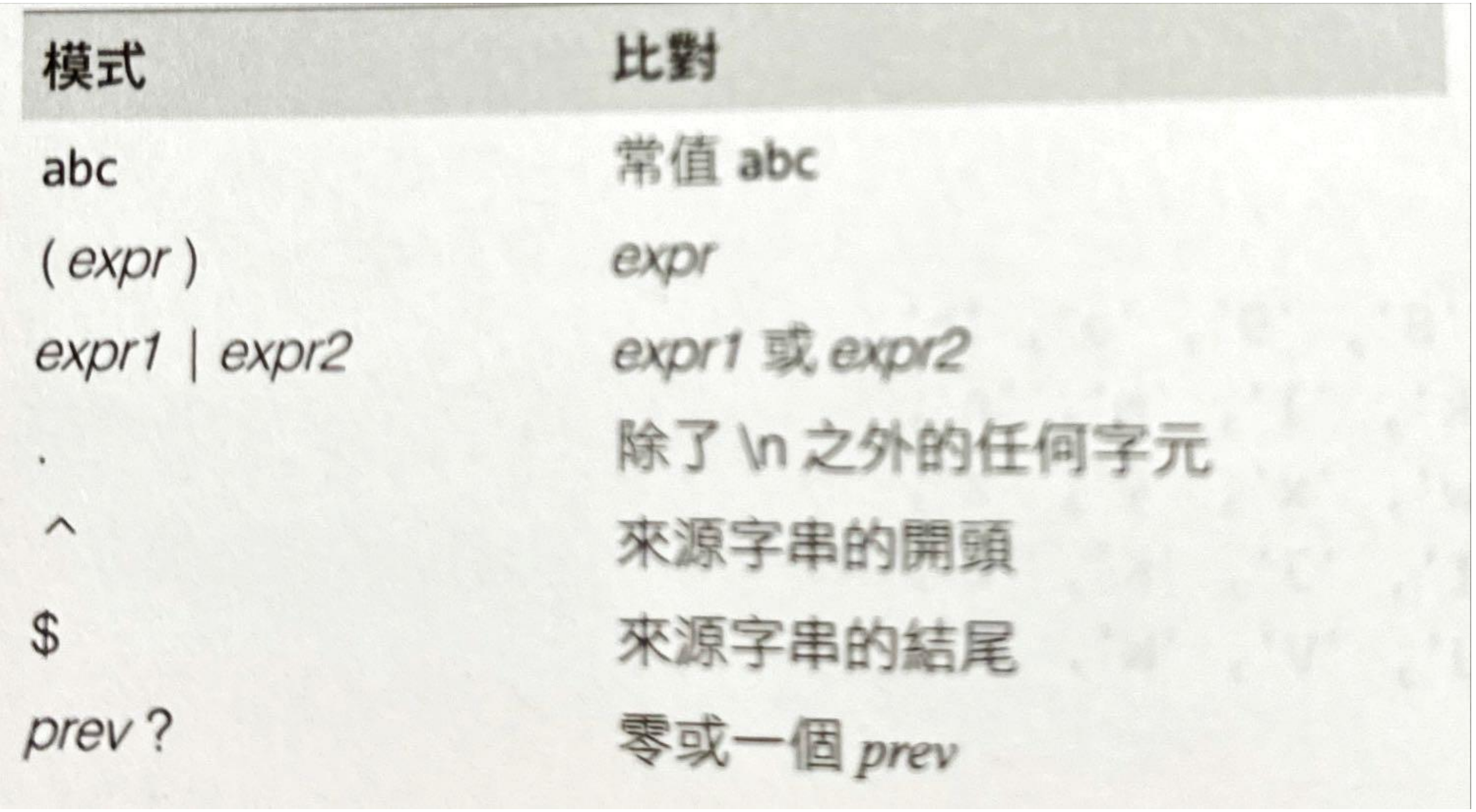
特殊字元(Metacharacter)—Match Metacharacter



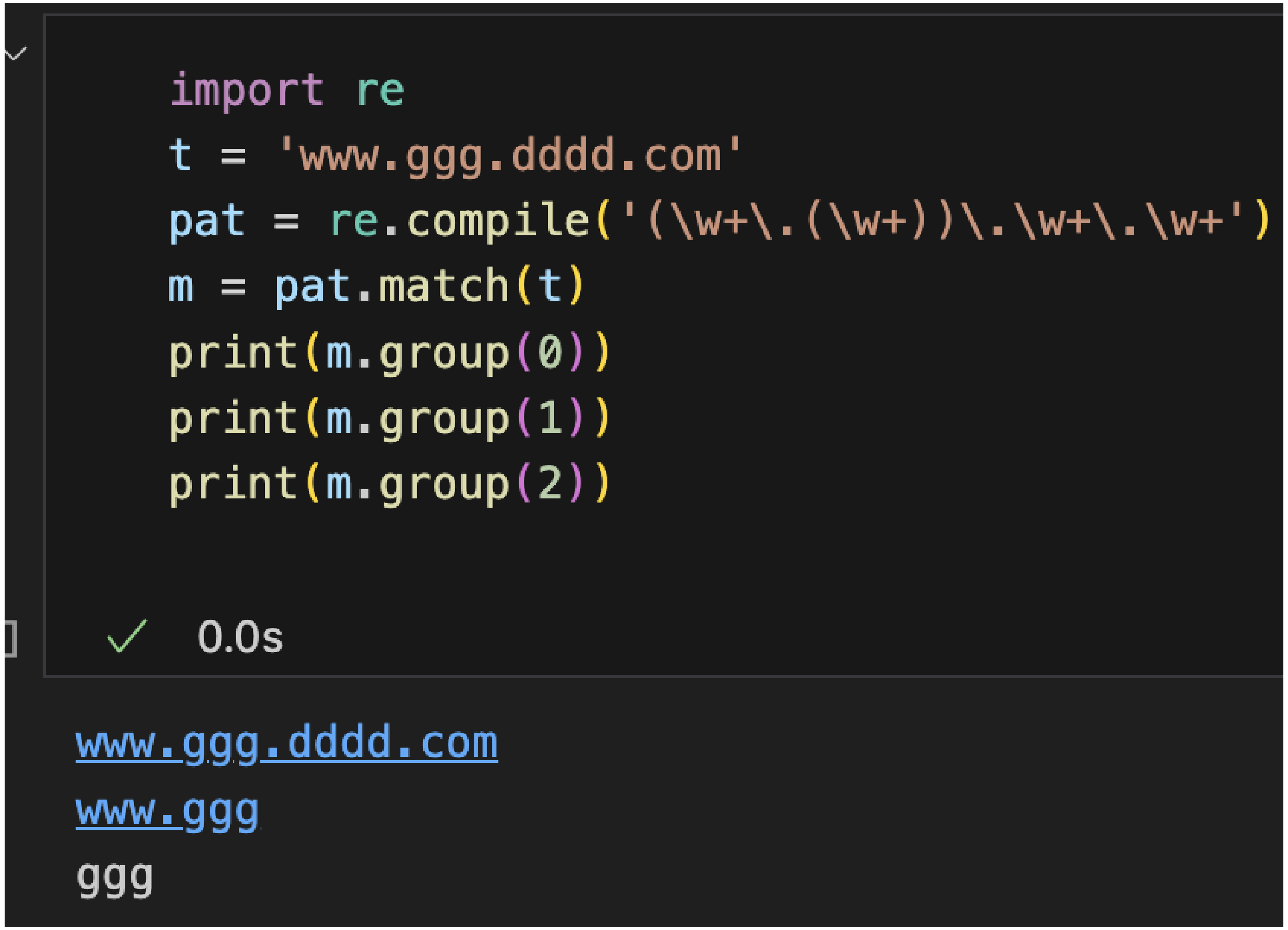
Char	Description
.	代表任一個字元
[...]	代表字元集中的任一字元, 例如 [abc] 可對應 a, b 或 c 連續字元的定義可用"-", 例如 [a-d] = [abcd]
[^...]	代表非字元集中的任一字元, 例如 [^abc] 將不對應 a, b 或 c



模式	比對
prev *	零或多個 *prev*，盡可能地多
prev *?	零或多個 *prev*，盡可能地少
prev +	一或多個 *prev*，盡可能地多
prev +?	一或多個 *prev*，盡可能地少
prev { *m* }	*m* 個連續的 *prev*
prev { *m*, *n* }	*m* 至 *n* 個連續的 *prev*，盡可能地多
prev { *m*, *n* }?	*m* 至 *n* 個連續的 *prev*，盡可能地少
[*abc*]	a 或 b 或 c（與 a|b|c 一樣）
[^ *abc*]	非 (a 或 b 或 c)
prev (?= *next*)	*prev*，若接下來是 *next*
prev (?! *next*)	*prev*，若接下來不是 *next*
(?<= *prev*) *next*	*next*，若前面是 *prev*
(?<! *prev*) *next*	*next*，若前面不是 *prev*



模式	比對
abc	常值 abc
(*expr*)	*expr*
expr1 | *expr2*	*expr1* 或 *expr2*
.	除了 \n 之外的任何字元
^	來源字串的開頭
$	來源字串的結尾
prev ?	零或一個 *prev*



```
import re
t = 'www.ggg.dddd.com'
pat = re.compile('(\w+\.(\w+))\.\w+\.\w+')
m = pat.match(t)
print(m.group(0))
print(m.group(1))
print(m.group(2))
```

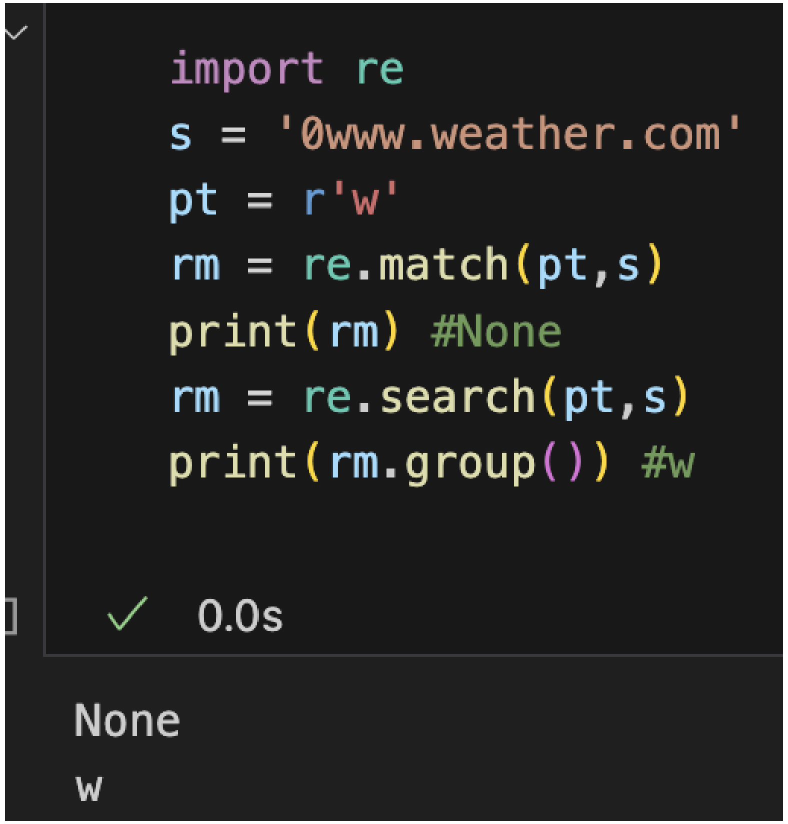
✓ 0.0s

```
www.ggg.dddd.com
www.ggg
ggg
```

re package

- search()
- match()
- findall()
- split()

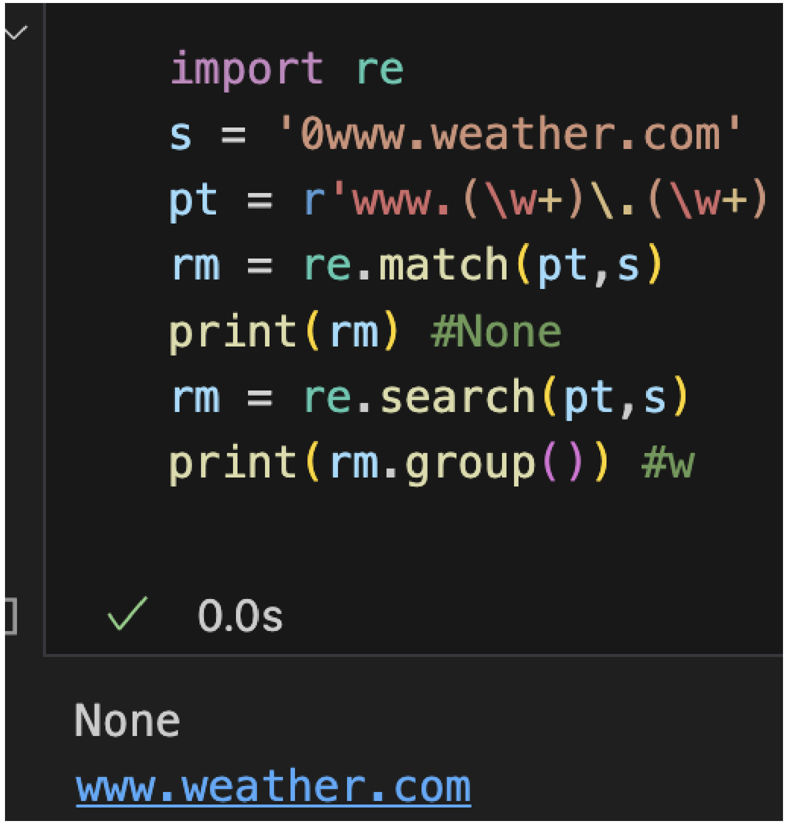
Match vs. Search



```
import re
s = '0www.weather.com'
pt = r'w'
rm = re.match(pt,s)
print(rm) #None
rm = re.search(pt,s)
print(rm.group()) #w
```

✓ 0.0s

```
None
w
```

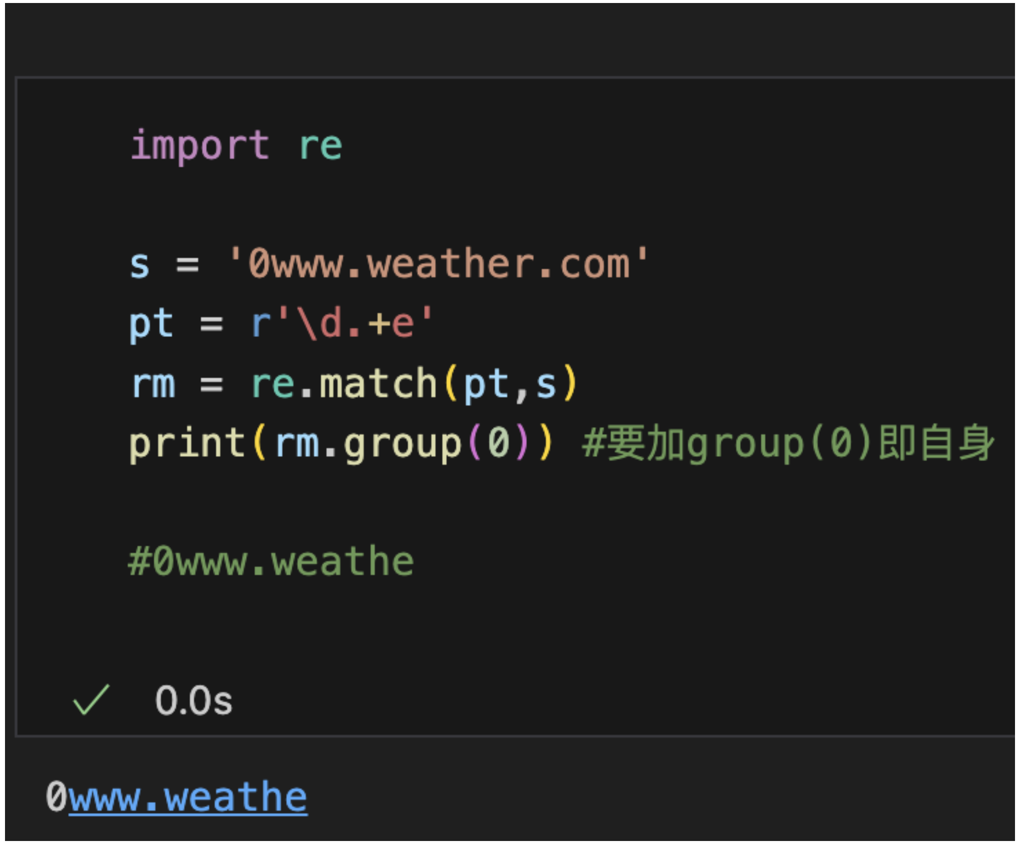


```
import re
s = '0www.weather.com'
pt = r'www.(\w+)\.(\w+)
rm = re.match(pt,s)
print(rm) #None
rm = re.search(pt,s)
print(rm.group()) #w
```

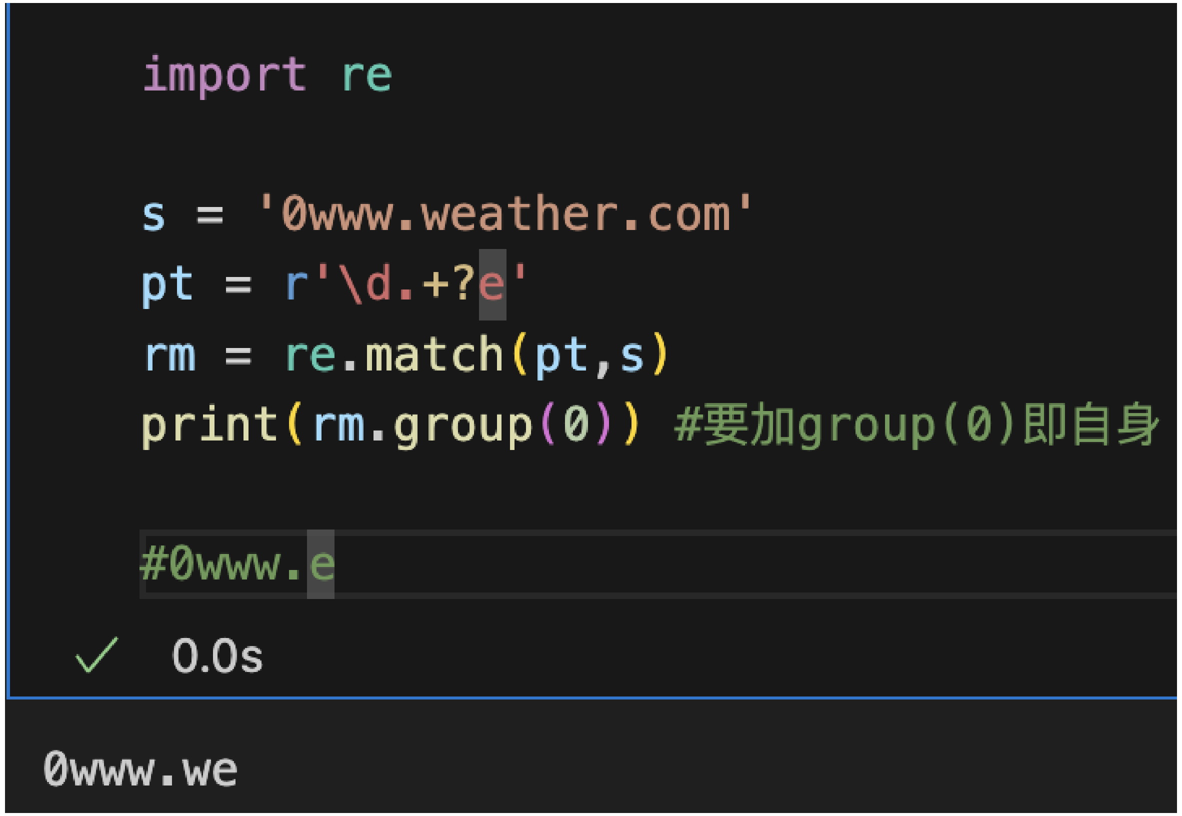
✓ 0.0s

```
None
www.weather.com
```

Greedy vs. Non-greedy

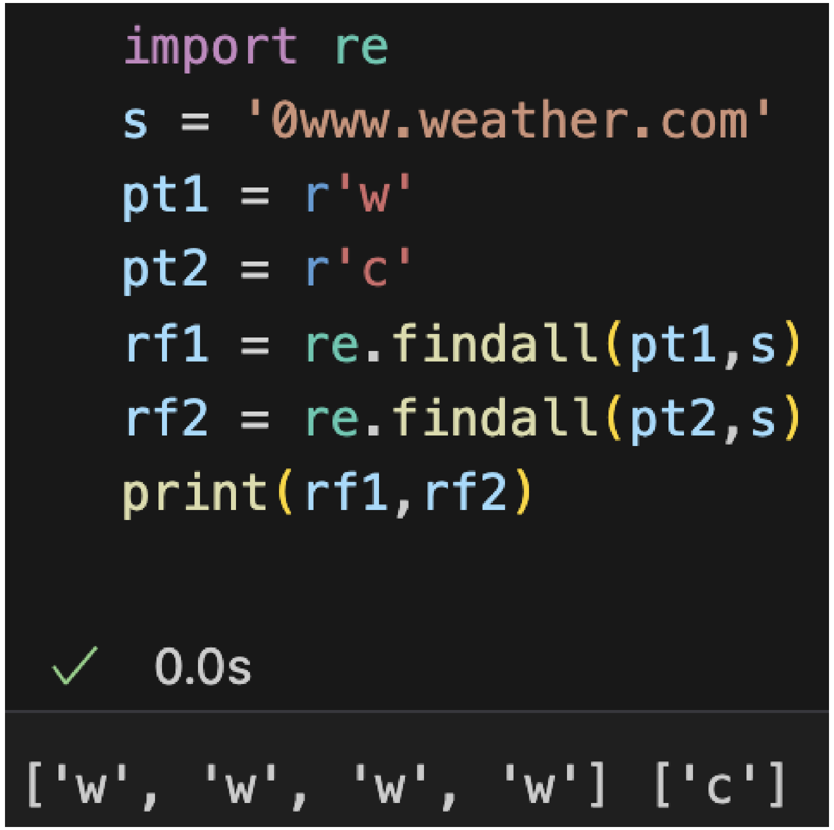


```
import re

s = '0www.weather.com'
pt = r'\d.+e'
rm = re.match(pt,s)
print(rm.group(0)) #要加group(0)即自身

#0www.weathe

✓ 0.0s

0www.weathe
```



```
import re

s = '0www.weather.com'
pt = r'\d.+?e'
rm = re.match(pt,s)
print(rm.group(0)) #要加group(0)即自身

#0www.e

✓ 0.0s

0www.we
```

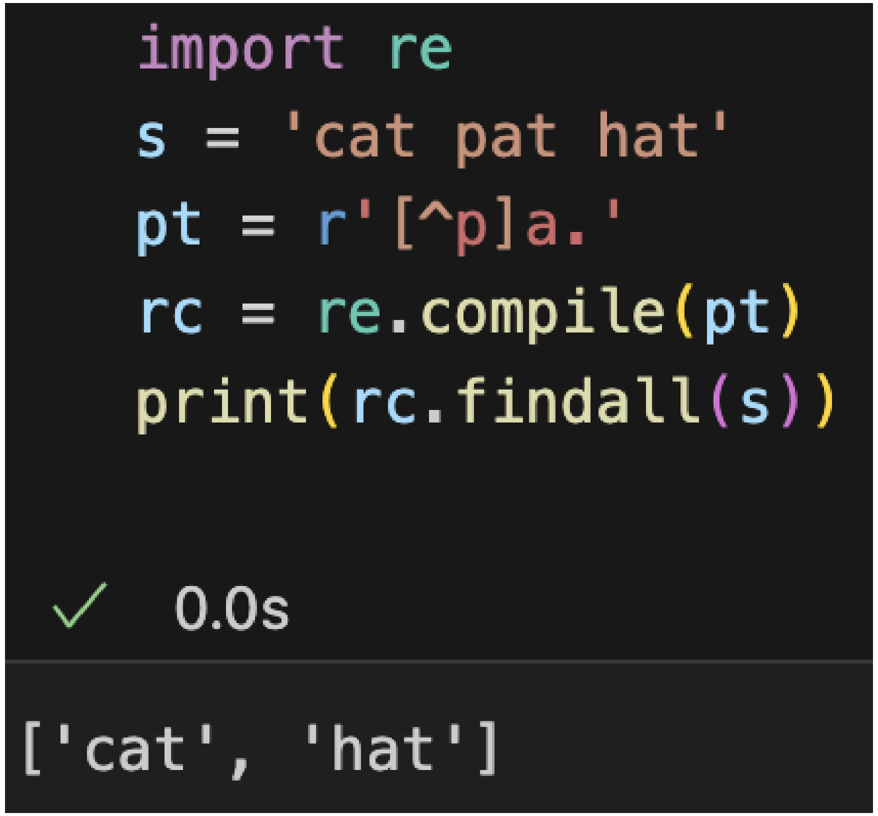
Findall



```
import re
s = '0www.weather.com'
pt1 = r'w'
pt2 = r'c'
rf1 = re.findall(pt1,s)
rf2 = re.findall(pt2,s)
print(rf1,rf2)
```

✓ 0.0s

```
['w', 'w', 'w', 'w'] ['c']
```



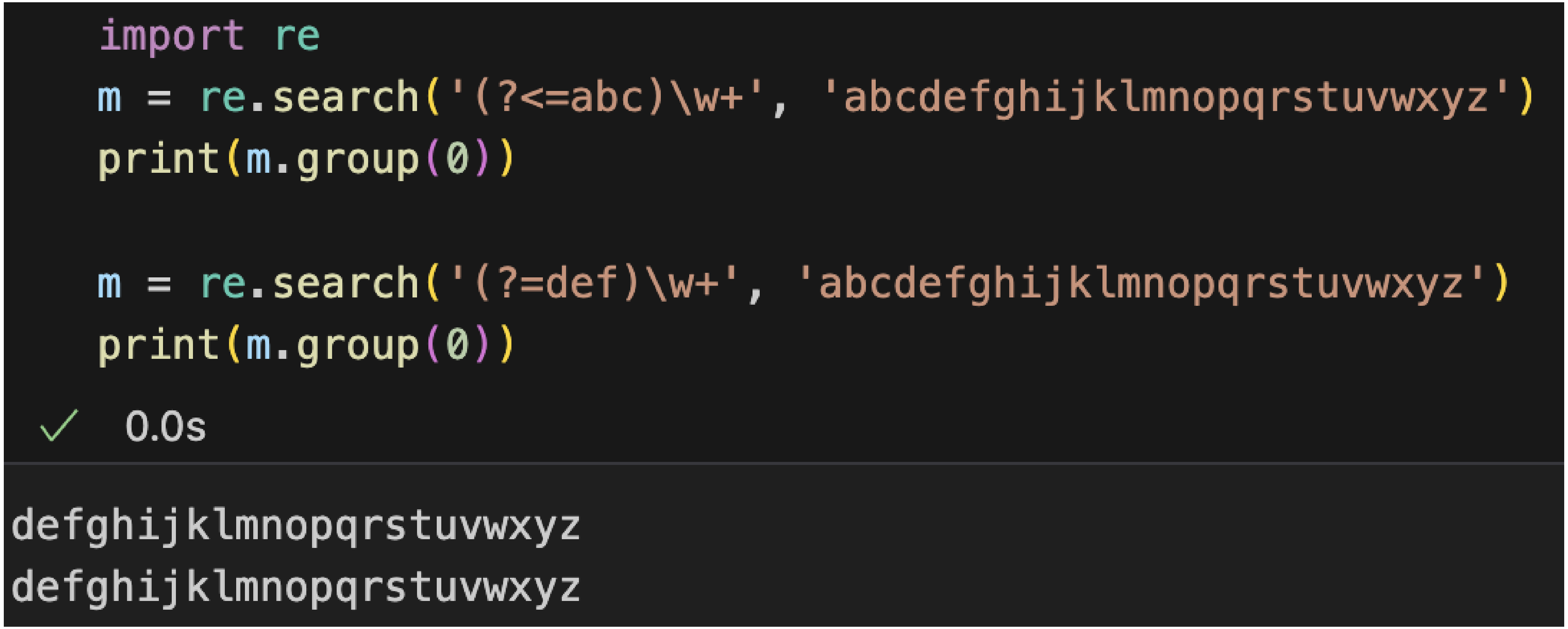
```
import re
s = 'cat pat hat'
pt = r'[^p]a.'
rc = re.compile(pt)
print(rc.findall(s))
```

✓ 0.0s

```
['cat', 'hat']
```

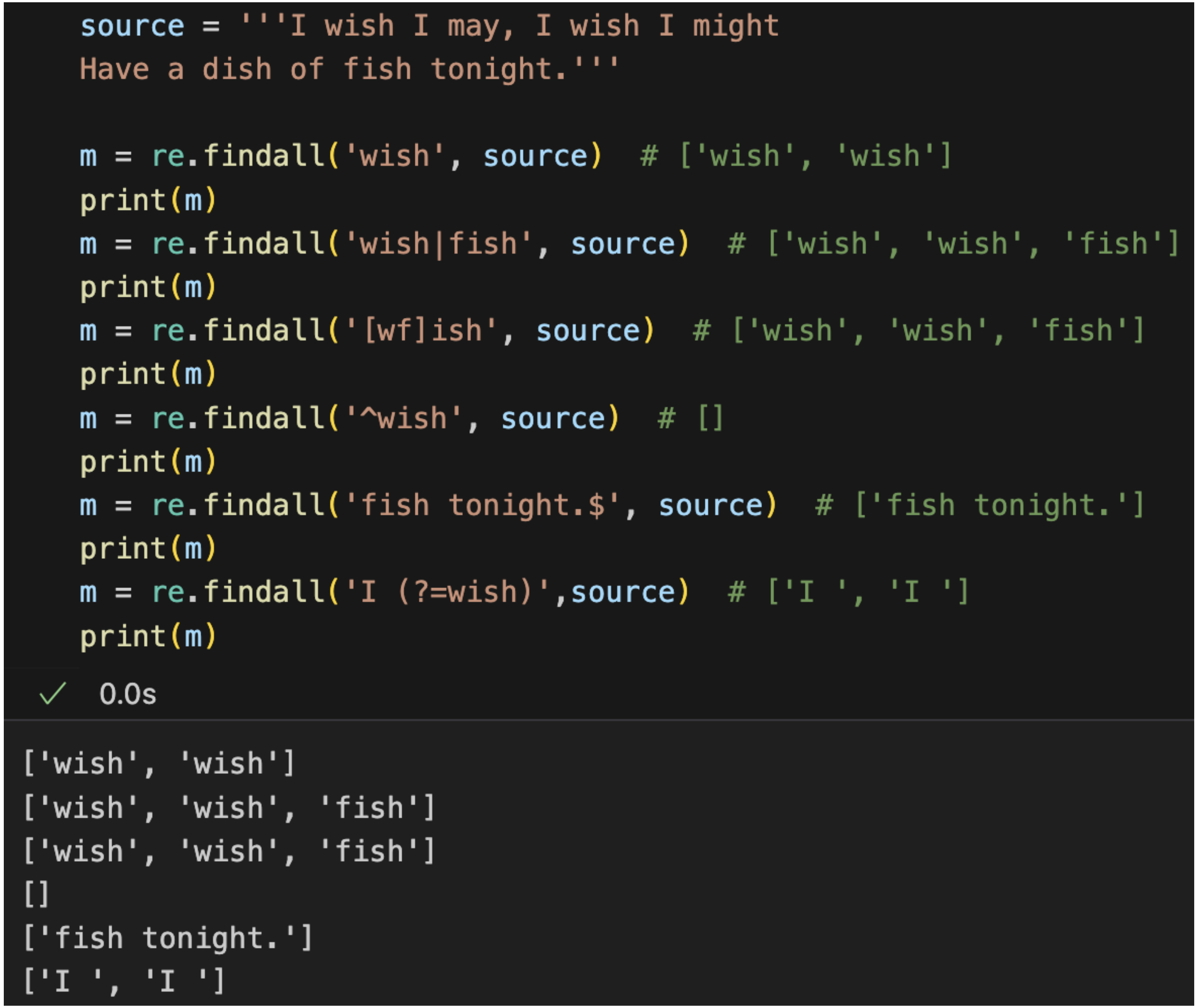
Lookahead、Lookbehind

- Pattern1(?:Pattern2)符合 Pattern1 並且右方符合 Pattern2 才算符合全部的 Pattern (return Pattern1與Pattern2)
- Pattern1(?=Pattern2)符合 Pattern1 並且右方符合 Pattern2 才算符合全部的 Pattern (return Pattern1)
- Pattern1(?!Pattern2)
- (?<=Pattern1)Pattern2: 符合 Pattern2 並且左方符合 Pattern1 才算符合 Pattern2
  白話文: 我要找 Pattern2，其左方必須為 Pattern1
- (?<!Pattern1)Pattern2: 符合 Pattern2 但左方不符合 Pattern1 才算符合 Pattern2
  白話文: 我要找 Pattern2，但左方不能為 Pattern1



```
import re
m = re.search('(?<=abc)\w+', 'abcdefghijklmnopqrstuvwxyz')
print(m.group(0))

m = re.search('(?=def)\w+', 'abcdefghijklmnopqrstuvwxyz')
print(m.group(0))
```

```
defghijklmnopqrstuvwxyz
defghijklmnopqrstuvwxyz
```



```python
source = '''I wish I may, I wish I might
Have a dish of fish tonight.'''

m = re.findall('wish', source)  # ['wish', 'wish']
print(m)
m = re.findall('wish|fish', source)  # ['wish', 'wish', 'fish']
print(m)
m = re.findall('[wf]ish', source)  # ['wish', 'wish', 'fish']
print(m)
m = re.findall('^wish', source)  # []
print(m)
m = re.findall('fish tonight.$', source)  # ['fish tonight.']
print(m)
m = re.findall('I (?=wish)',source)  # ['I ', 'I ']
print(m)
```

```
['wish', 'wish']
['wish', 'wish', 'fish']
['wish', 'wish', 'fish']
[]
['fish tonight.']
['I ', 'I ']
```

HW7

1. 建立一個名為zoo.py的檔案。在裡面定義一個名為hours()的函式，用它來印出字串’Open 9-5 daily’接著寫一個主程式main.py
    a. 匯入zoo為menagerie並呼叫他的函式hours()
    b. 直接從zoo匯入hours()函式並呼叫它

2. 寫一個程式
    a. 印出在你目前目錄的檔案
    b. 印出在你的上一層目錄中的檔案
    c. 將字串’This is a test of the emergency text system’指派給變數test1並將其寫到test.txt 檔案
    d. 打開test.txt檔案，並將他的內容讀入字串test2。test1與test2一樣嗎？

3. 定義一個稱為OopsException的例外。發出這個例外，接著寫一段程式來捕捉這個例外，並印出’Caught an oops.’

4. 讀取檔案’mammoth.txt’

a.匯入 re 模組來使用Python 的正規表達式函式。使用re.findall（ ）來印出c開頭的所有單字。

b. 找出所有c開頭的四字母單字。

c. 找出所有以r結束的單字。

d. 尋找有連續三個母音的所有單字。

5. 請撰寫一程式，計算檔案所包含的字元數、單字數、以及行數，單字之間是以空白隔開（輸入mammoth.txt）