以遞迴程式設計
列舉字串的所有排列與組合

程式設計與偵錯

s = "1234"
列舉所有可能排列

s = "1234"
列舉所有可能排列

以字元'1'開頭的所有字串

以字元'3'開頭的所有字串

```
['1234', '1243', '1324', '1342', '1423', '1432', '2134', '2143', '2314', '2341', '2413',
 '2431', '3124', '3142', '3214', '3241', '3412', '3421', '4123', '4132', '4213', '4231',
 '4312', '4321']
24
```

```
>>> n = len(ss)
>>> for i in range(n):
...     print(ss[0:i]+ss[i+1:n])
...
234
134
124
123
```

印出不包含第i個字元的所有字串，i從0到n-1，不包括n

s = **"abcde"**

列舉所有可能排列

s = "abcde"
列舉所有可能排列

以字元'a'開頭的所有字串

以字元'c'開頭的所有字串

```
['abcde', 'abced', 'abdce', 'abdec', 'abecd', 'abedc', 'acbde', 'acbed', 'acdbe',
 'acdeb', 'acebd', 'acedb', 'adbce', 'adbec', 'adcbe', 'adceb', 'adebc', 'adecb',
 'aebcd', 'aebdc', 'aecbd', 'aecdb', 'aedbc', 'aedcb', 'bacde', 'baced', 'badce',
 'badec', 'baecd', 'baedc', 'bcade', 'bcaed', 'bcdae', 'bcdea', 'bcead', 'bceda',
 'bdace', 'bdaec', 'bdcae', 'bdcea', 'bdeac', 'bdeca', 'beacd', 'beadc', 'becad',
 'becda', 'bedac', 'bedca', 'cabde', 'cabed', 'cadbe', 'cadeb', 'caebd', 'caedb',
 'cbade', 'cbaed', 'cbdae', 'cbdea', 'cbead', 'cbeda', 'cdabe', 'cdaeb', 'cdbae',
 'cdbea', 'cdeab', 'cdeba', 'ceabd', 'ceadb', 'cebad', 'cebda', 'cedab', 'cedba',
 'dabce', 'dabec', 'dacbe', 'daceb', 'daebc', 'daecb', 'dbace', 'dbaec', 'dbcae',
 'dbcea', 'dbeac', 'dbeca', 'dcabe', 'dcaeb', 'dcbae', 'dcbea', 'dceab', 'dceba',
 'deabc', 'deacb', 'debac', 'debca', 'decab', 'decba', 'eabcd', 'eabdc', 'eacbd',
 'eacdb', 'eadbc', 'eadcb', 'ebacd', 'ebadc', 'ebcad', 'ebcda', 'ebdac', 'ebdca',
 'ecabd', 'ecadb', 'ecbad', 'ecbda', 'ecdab', 'ecdba', 'edabc', 'edacb', 'edbac',
 'edbca', 'edcab', 'edcba']
120
```

```
>>> for i in range(n):
...     print(ss[0:i]+ss[i+1:n])
...
bcde
acde
abde
abce
abcd
```

印出不包含第i個字元的所有字串，i從0到n-1，不包括n

s = “12345"

r= 2

列舉從s中任意取r個字元的所有可能

s = "12345"

r= 2

列舉從s中任意取r個字元的所有可能

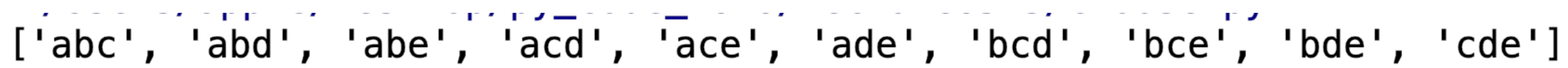
```
['12', '13', '14', '15', '23', '24', '25', '34', '35', '45']
```

s = “**abcde**"

r= 3

列舉從**s**中任意取**r**個字元的所有可能

s = “abcde"
r= 3
列舉從s中任意取r個字元的所有可能



```
['abc', 'abd', 'abe', 'acd', 'ace', 'ade', 'bcd', 'bce', 'bde', 'cde']
```

```
def perm(s):
    if len(s)==1:
        return s
```

字串s的長度為1
只有一種排列可能

遞迴的primitive問題：當len(s)==1時，只有一種排列可能，答案相當明顯，就是字串s本身

s = "1234"

↑

i=0

t = "234"

```
ans = []
for i in range(len(s)):
    t = s[0:i] + s[i + 1:len(s)]
```

列舉出不包含第i個字元的所有字串，i從0到n-1，不包括n

$$s = \text{"}1\underset{\overset{\uparrow}{i=1}}{2}34\text{"}$$

```
ans = []
for i in range(len(s)):
    t = s[0:i] + s[i + 1:len(s)]
```

$$t = \text{"134"}$$

```python
ans = []
for i in range(len(s)):
    t = s[0:i] + s[i + 1:len(s)]
```

s = "1234"

$\uparrow$

i=2

t = "124"

s = "1234"

$\uparrow$

i=3

t = “123"

```
ans = []
for i in range(len(s)):
    t = s[0:i] + s[i + 1:len(s)]
```

s = "1234"

```
ans = []
for i in range(len(s)):
    t = s[0:i] + s[i + 1:len(s)]
    items = perm(t)
    for item in items:
        ans.append(s[i]+item)
return ans
```

i=2

t = "124"

```
['124', '142', '214', '241', '412', '421']
```

$$s = "1234"$$

$i=2$

$$t = "124"$$

```
ans = []
for i in range(len(s)):
    t = s[0:i] + s[i + 1:len(s)]
    items = perm(t)
    for item in items:
        ans.append(s[i]+item)
return ans
```

items

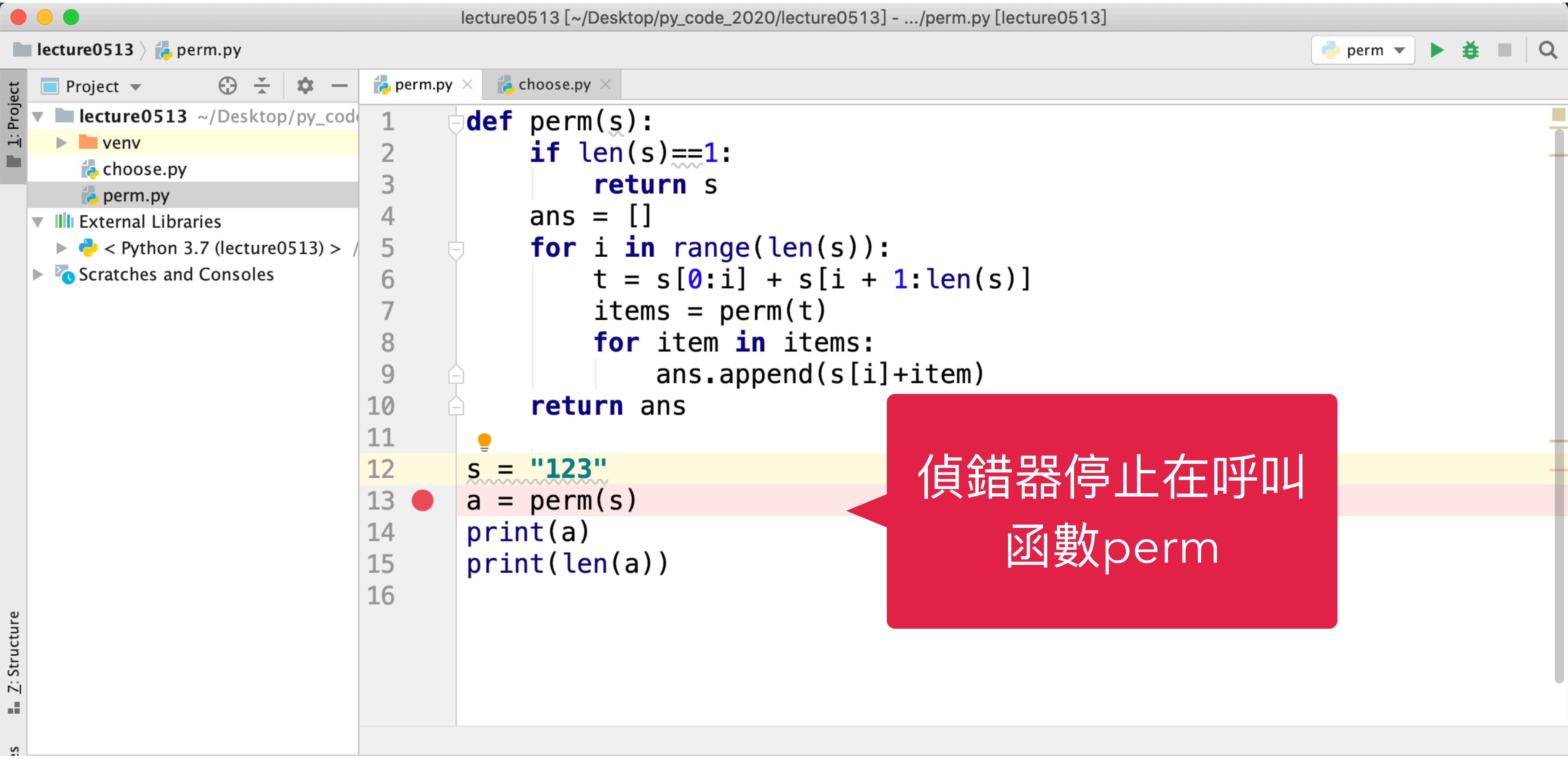
```
['124', '142', '214', '241', '412', '421']
```

ans

```
['3124', '3142', '3214', '3241', '3412', '3421']
```

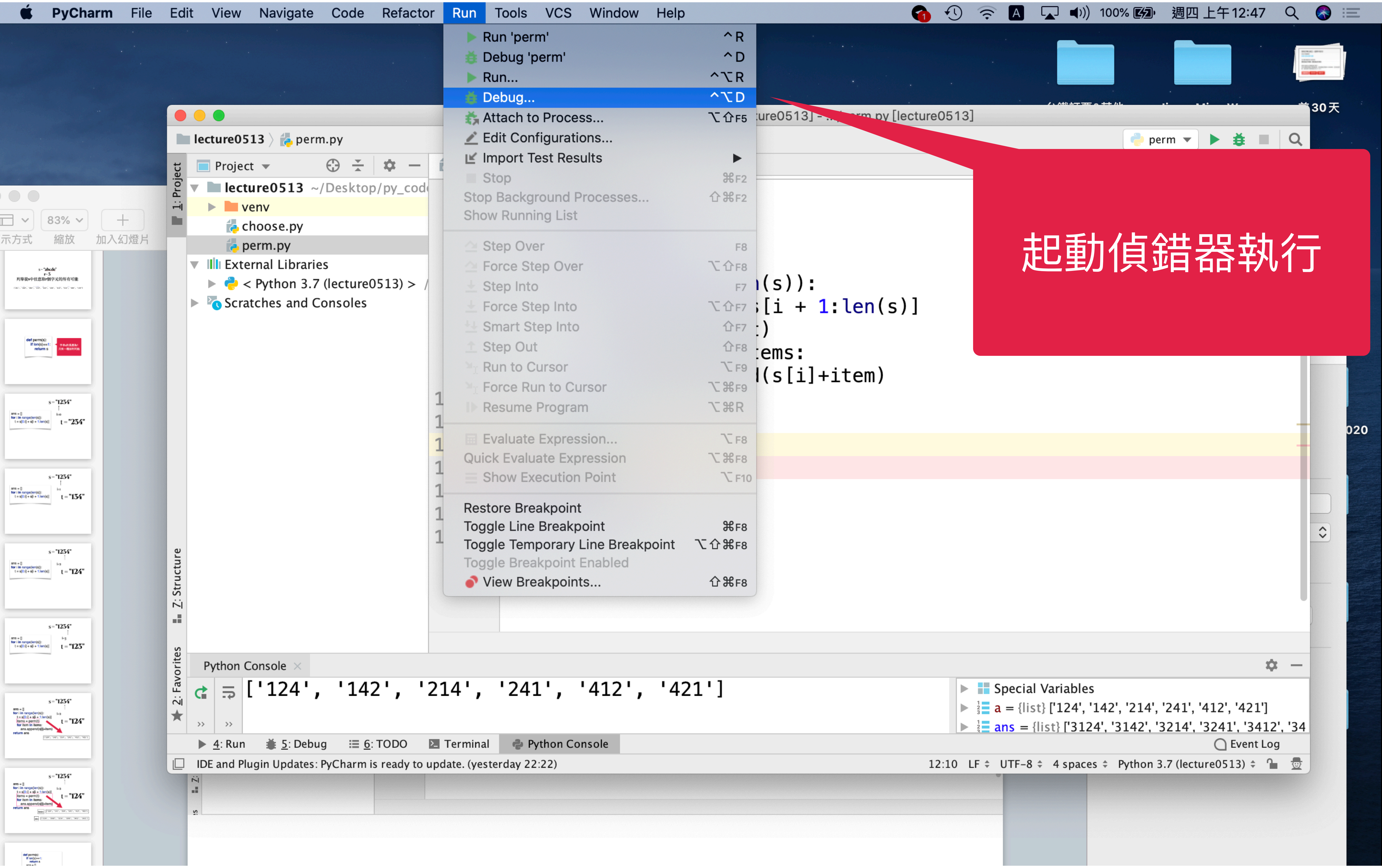
```python
def perm(s):
    if len(s)==1:
        return s
    ans = []
    for i in range(len(s)):
        t = s[0:i] + s[i + 1:len(s)]
        items = perm(t)
        for item in items:
            ans.append(s[i]+item)
    return ans
```

Debugger


lecture0513 [~/Desktop/py_code_2020/lecture0513] - .../perm.py [lecture0513]

```python
def perm(s):
    if len(s)==1:
        return s
    ans = []
    for i in range(len(s)):
        t = s[0:i] + s[i + 1:len(s)]
        items = perm(t)
        for item in items:
            ans.append(s[i]+item)
    return ans

s = "123"
a = perm(s)
print(a)
print(len(a))
```

偵錯器停止在呼叫函數perm

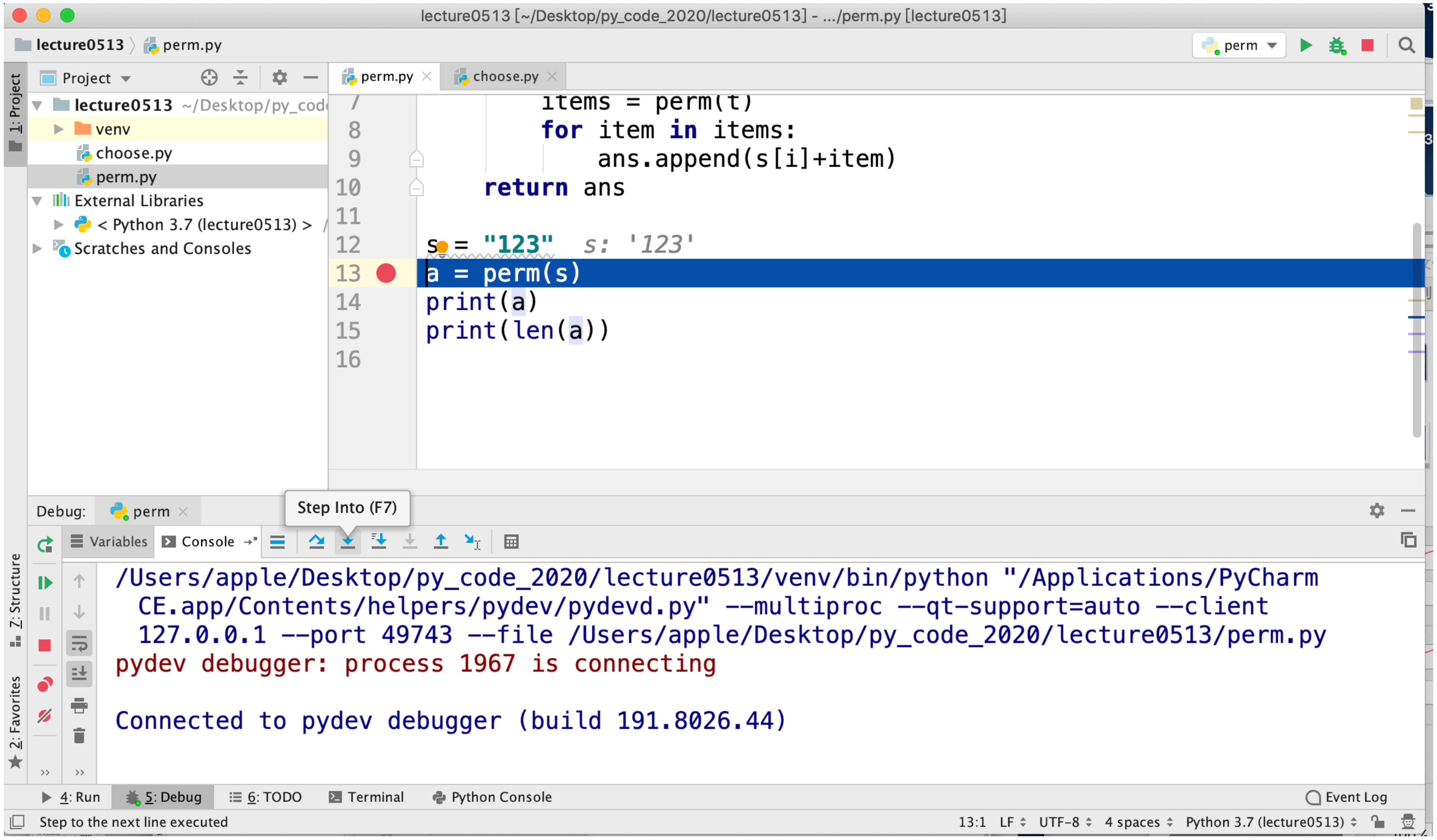


```
['124', '142', '214', '241', '412', '421']
```

```
Special Variables
a = {list} ['124', '142', '214', '241', '412', '421']
ans = {list} ['3124', '3142', '3214', '3241', '3412', '34
```

IDE and Plugin Updates: PyCharm is ready to update. (yesterday 22:22)

起動偵錯器執行



```
        items = perm(t)
        for item in items:
            ans.append(s[i]+item)
    return ans

s = "123"   s: '123'
a = perm(s)
print(a)
print(len(a))
```

```
/Users/apple/Desktop/py_code_2020/lecture0513/venv/bin/python "/Applications/PyCharm
  CE.app/Contents/helpers/pydev/pydevd.py" --multiproc --qt-support=auto --client
  127.0.0.1 --port 49743 --file /Users/apple/Desktop/py_code_2020/lecture0513/perm.py
pydev debugger: process 1967 is connecting

Connected to pydev debugger (build 191.8026.44)
```

```
from perm import perm
s = "abc"
perm(s)
['abc', 'acb', 'bac', 'bca', 'cab', 'cba']
```

s = “12345"
r= 2
列舉從s中任意取r個字元的所有可能

遞迴的primitive問題一：當r==1時，從字串s中選1個字元，答案相當明顯，每一組選擇都只包含字串s中的一個字元

遞迴的primitive問題二：當字串s得長度是r時，答案相當明顯，唯一的選擇就是s

```python
def choose(s,r):
    ans = []
    if r == 1:
        for i in range(len(s)):
            ans.append(s[i])
        return ans
```

s = “abcde"

r=1

```
['a', 'b', 'c', 'd', 'e']
```

遞迴的primitive問題一：當r==1時，從字串s中選1個字元，答案相當明顯，每一組選擇都只包含字串s中的一個字元

```python
def choose(s,r):
    ans = []
    if r == 1:
        for i in range(len(s)):
            ans.append(s[i])
        return ans

    if len(s) == r:
        ans.append(s)
        return ans
```

s = "cd"
r= 2

```
['cd']
```

遞迴的primitive問題二：當字串s得長度是r時，答案相當明顯，唯一的選擇就是s

s = “abcde"

t= “bcde”

s[1:]

從字串s中選r=2個字元可分解成兩個子問題

子問題一：第一個字元必選
從字串s[1:]中選r-1個字元，每一組選擇，都與字串s中的第一個字元"a"合並成r個字元

子問題二：第一個字元不選
從字串s[1:]中選r個字元

```
t = s[1:]
a = choose(t,r-1)
for item in a:
    ans.append(s[0]+item)
b = choose(t, r)
return ans+b
```

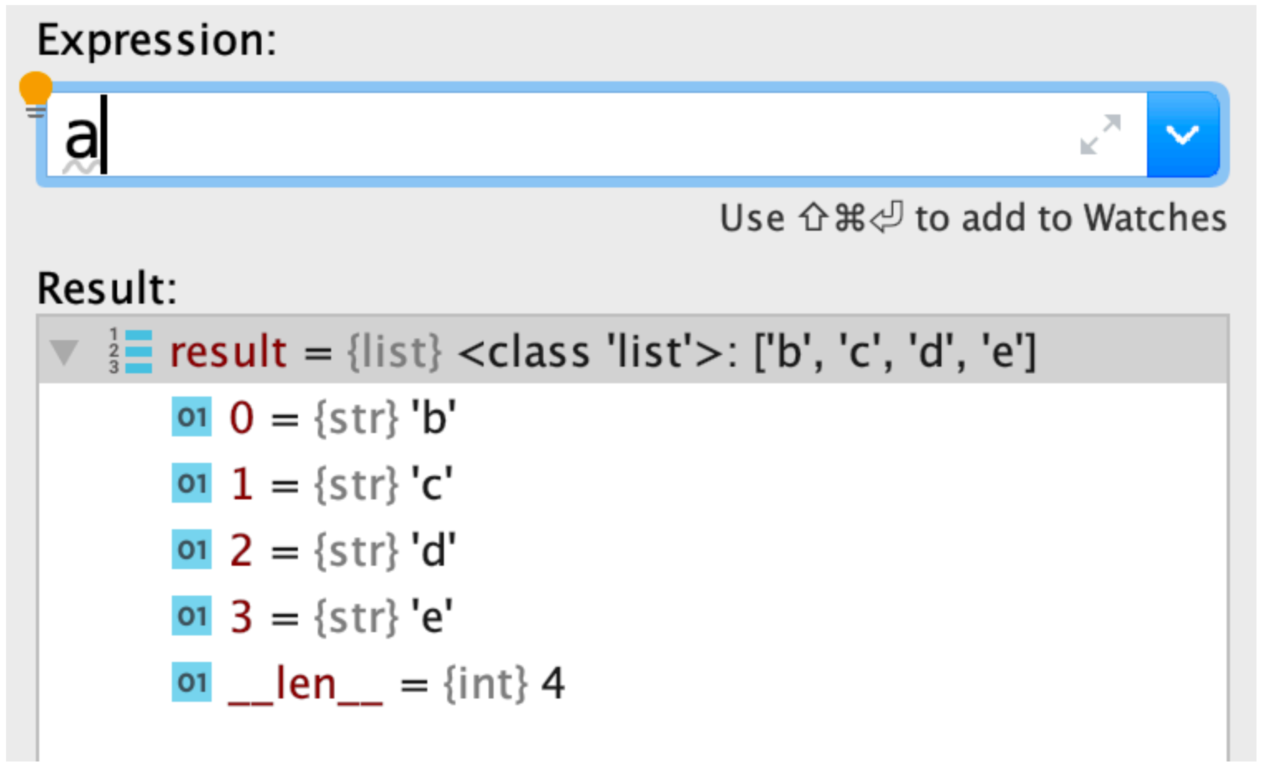
s = “abcde"
t= “bcde"

子問題一：第一個字元必選
從字串s[1:]中選r-1個字元，每一組選擇，都與字串s中的第一個字元"a"合並成r個字元

s = “abcde"
t= “bcde"

```
t = s[1:]
a = choose(t,r-1)
for item in a:
    ans.append(s[0]+item)
b = choose(t, r)
return ans+b
```

從t字串中選出r-1個字元


Expression:

```
a
```

Use ⇧⌘⏎ to add to Watches

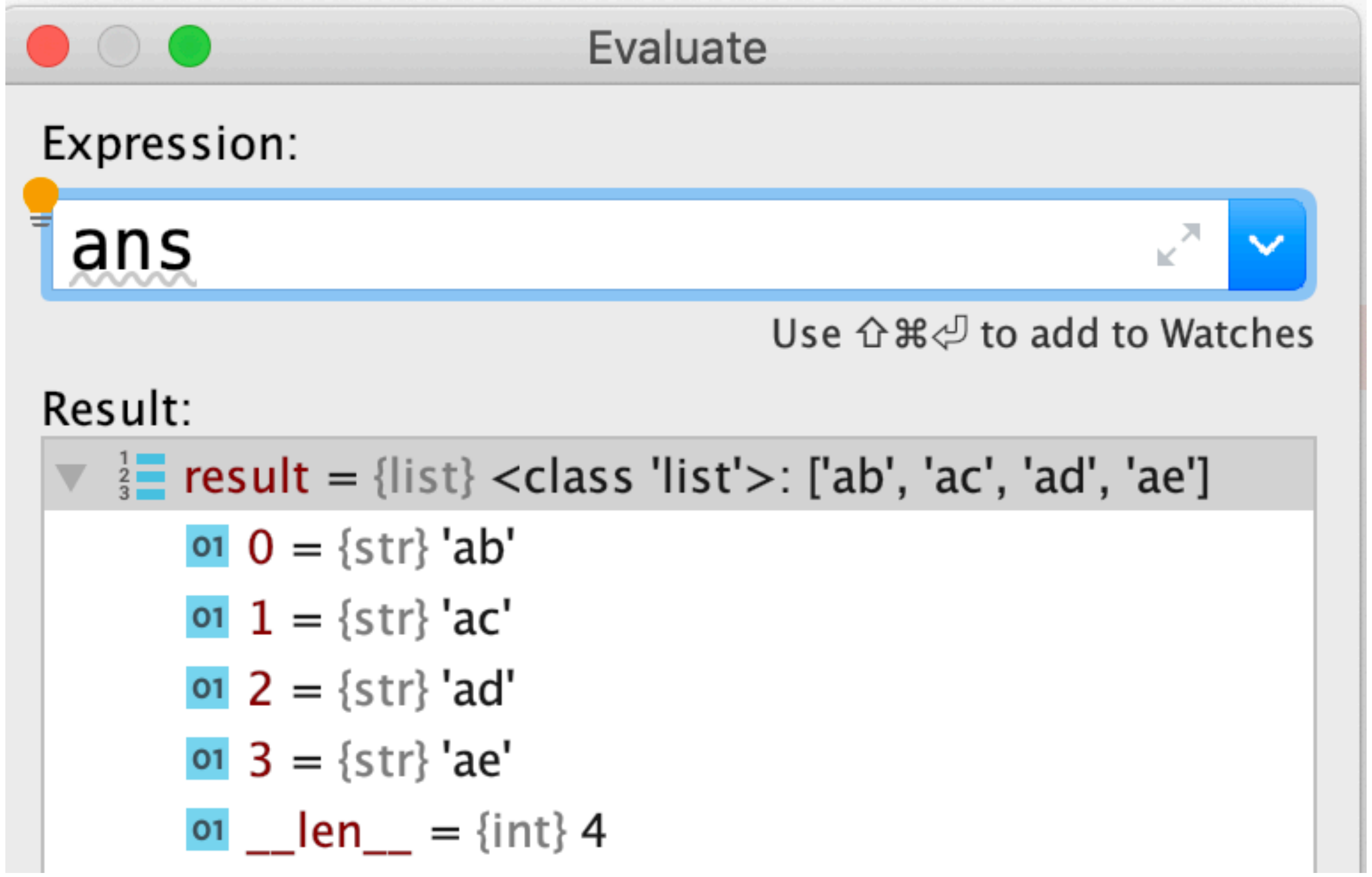
Result:

```
result = {list} <class 'list'>: ['b', 'c', 'd', 'e']
    0 = {str} 'b'
    1 = {str} 'c'
    2 = {str} 'd'
    3 = {str} 'e'
    __len__ = {int} 4
```

s = “abcde"
t= “bcde”
r=2

```
t = s[1:]
a = choose(t,r-1)
for item in a:
    ans.append(s[0]+item)
b = choose(t, r)
return ans+b
```

從t字串中選出r-1個字元
將每一組選擇與s[0]合併



```
ans
```

```
▼ result = {list} <class 'list'>: ['ab', 'ac', 'ad', 'ae']
      0 = {str} 'ab'
      1 = {str} 'ac'
      2 = {str} 'ad'
      3 = {str} 'ae'
      __len__ = {int} 4
```

```
t = s[1:]
a = choose(t,r-1)
for item in a:
    ans.append(s[0]+item)
b = choose(t, r)
return ans+b
```

s = “abcde"
t= “bcde”
r=2

第一個字元沒有被選到，從t字串中選出r個字元

```
['bc', 'bd', 'be', 'cd', 'ce', 'de']
```

s = “abcde"
t= “bcde”
r=2

```
t = s[1:]
a = choose(t,r-1)
for item in a:
    ans.append(s[0]+item)
b = choose(t, r)
return ans+b
```

從t字串中選出r-1個字元
將每一組選擇與s[0]合併

子問題二：第一個字元不選
從字串s[1:]中選r個字元

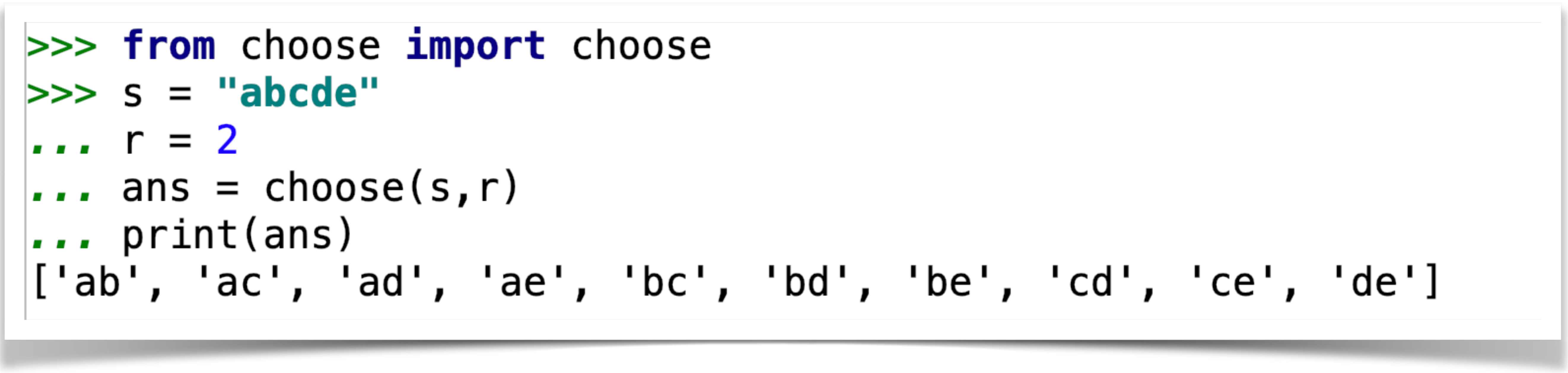
```
t = s[1:]
a = choose(t,r-1)
for item in a:
    ans.append(s[0]+item)
b = choose(t, r)
return ans+b
```

s = “abcde"
t= “bcde”
r=2

第一個字元沒有被選到，從t字串中選出r個字元

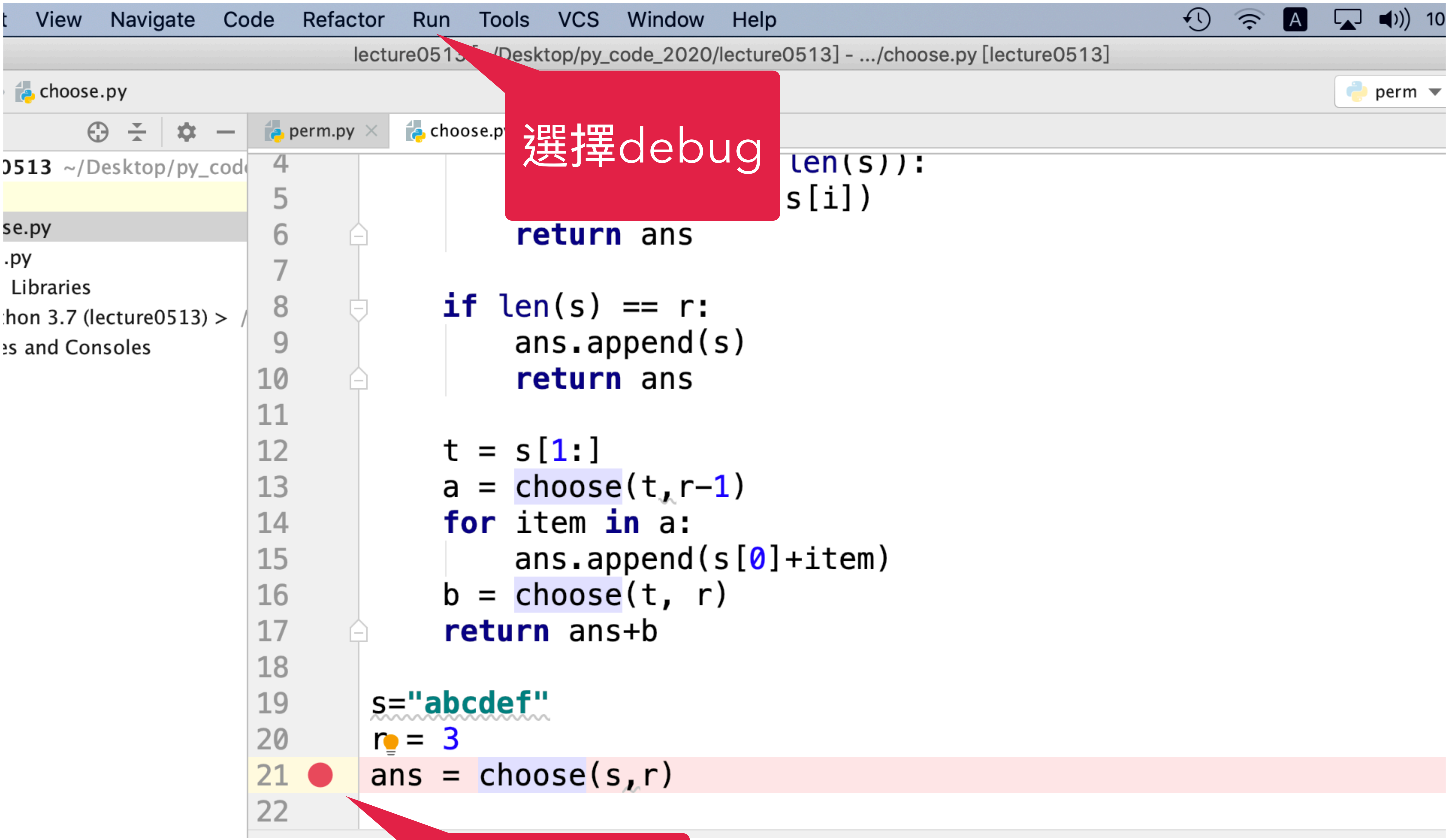
```
['bc', 'bd', 'be', 'cd', 'ce', 'de']
```

```python
def choose(s,r):
    ans = []
    if r == 1:
        for i in range(len(s)):
            ans.append(s[i])
        return ans

    if len(s) == r:
        ans.append(s)
        return ans

    t = s[1:]
    a = choose(t,r-1)
    for item in a:
        ans.append(s[0]+item)
    b = choose(t, r)
    return ans+b
```

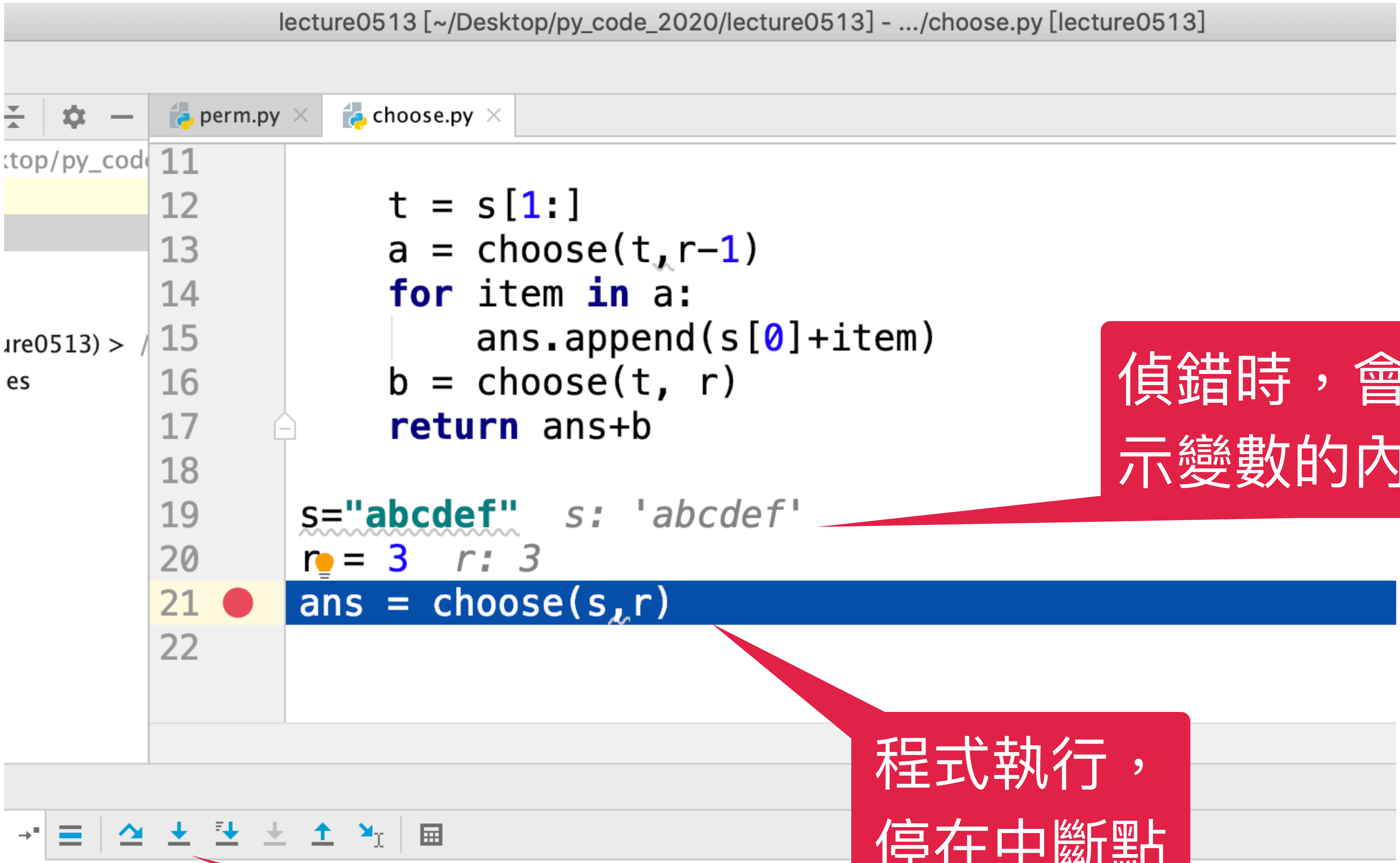


```
>>> from choose import choose
>>> s = "abcde"
... r = 2
... ans = choose(s,r)
... print(ans)
['ab', 'ac', 'ad', 'ae', 'bc', 'bd', 'be', 'cd', 'ce', 'de']
```

Debug of recursive programming
透過偵錯器，了解遞迴程式的執行過程



```
                  len(s)):
                  s[i])
        return ans

    if len(s) == r:
        ans.append(s)
        return ans

    t = s[1:]
    a = choose(t,r-1)
    for item in a:
        ans.append(s[0]+item)
    b = choose(t, r)
    return ans+b

s="abcdef"
r = 3
ans = choose(s,r)
```


lecture0513 [~/Desktop/py_code_2020/lecture0513] - .../choose.py [lecture0513]

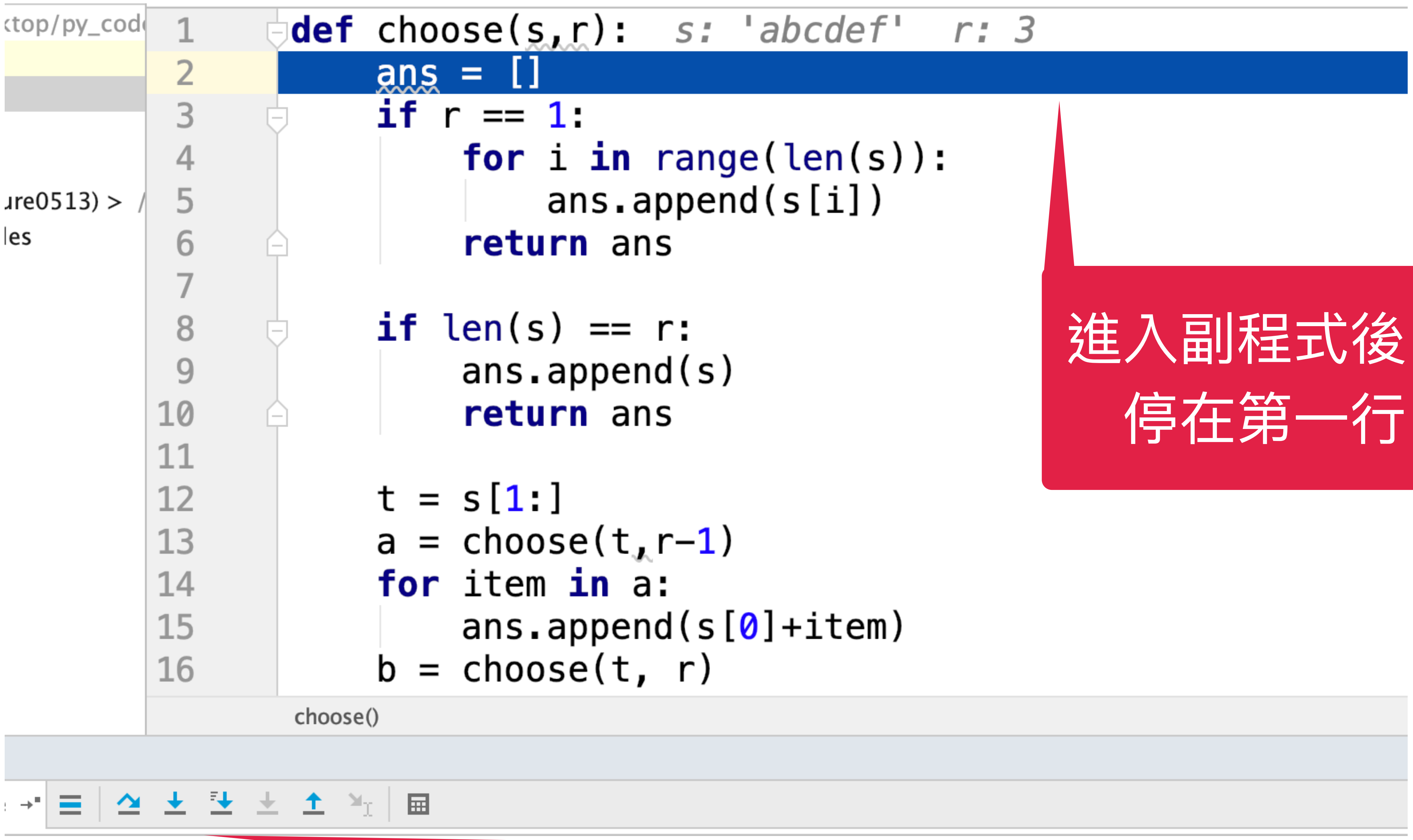
```
        t = s[1:]
        a = choose(t,r-1)
        for item in a:
            ans.append(s[0]+item)
        b = choose(t, r)
        return ans+b

s="abcdef"   s: 'abcdef'
r = 3   r: 3
ans = choose(s,r)
```

偵錯時，會顯示變數的內容

程式執行，停在中斷點

Step over
不進入副程式，跳過副程式執行

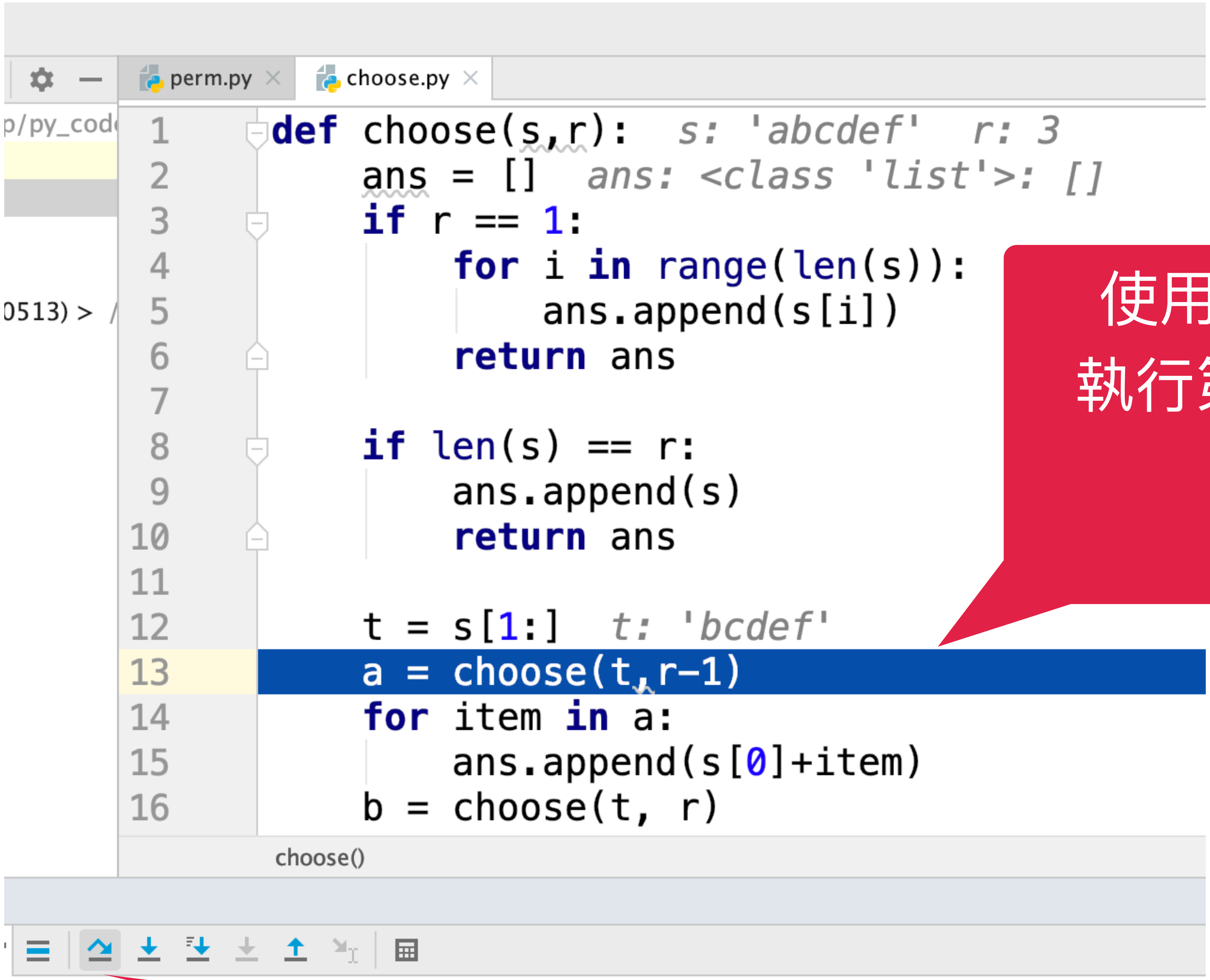
Step in
進入副程式



```
def choose(s,r):   s: 'abcdef'   r: 3
    ans = []
    if r == 1:
        for i in range(len(s)):
            ans.append(s[i])
        return ans

    if len(s) == r:
        ans.append(s)
        return ans

    t = s[1:]
    a = choose(t,r-1)
    for item in a:
        ans.append(s[0]+item)
    b = choose(t, r)
```

進入副程式後，
停在第一行

Step in
進入副程式

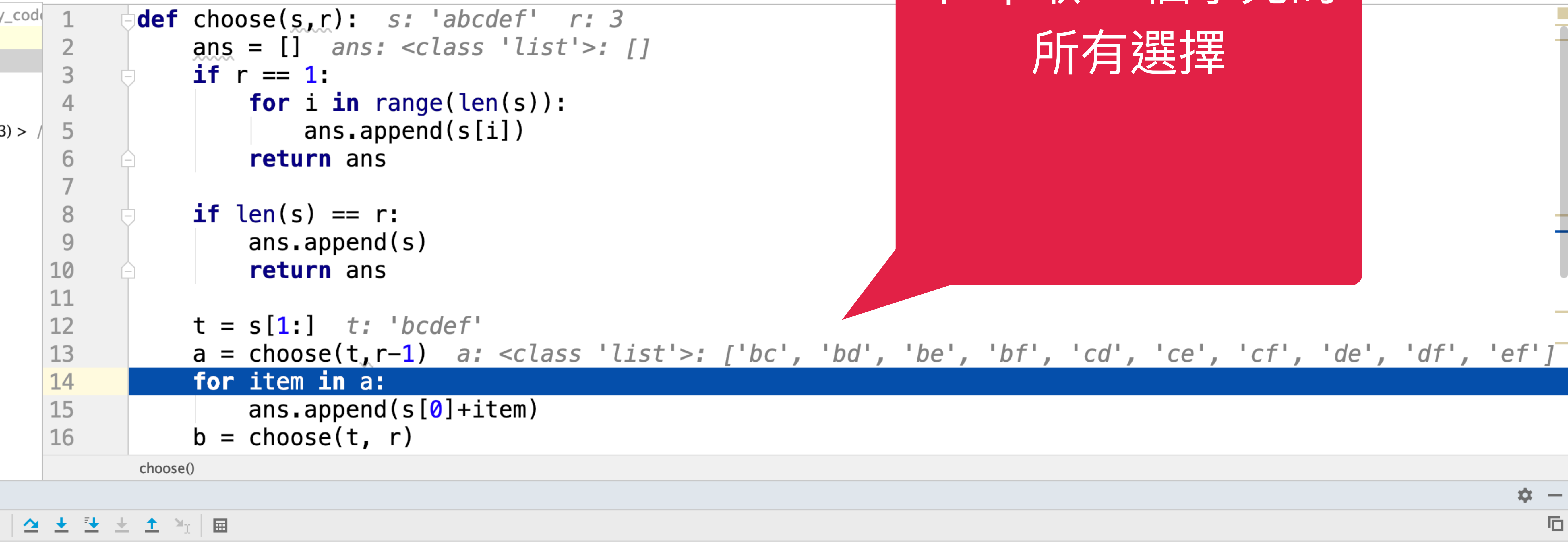


```
def choose(s,r):   s: 'abcdef'   r: 3
    ans = []  ans: <class 'list'>: []
    if r == 1:
        for i in range(len(s)):
            ans.append(s[i])
        return ans

    if len(s) == r:
        ans.append(s)
        return ans

    t = s[1:]  t: 'bcdef'
    a = choose(t,r-1)
    for item in a:
        ans.append(s[0]+item)
    b = choose(t, r)
```

使用Step over
執行第13行程式

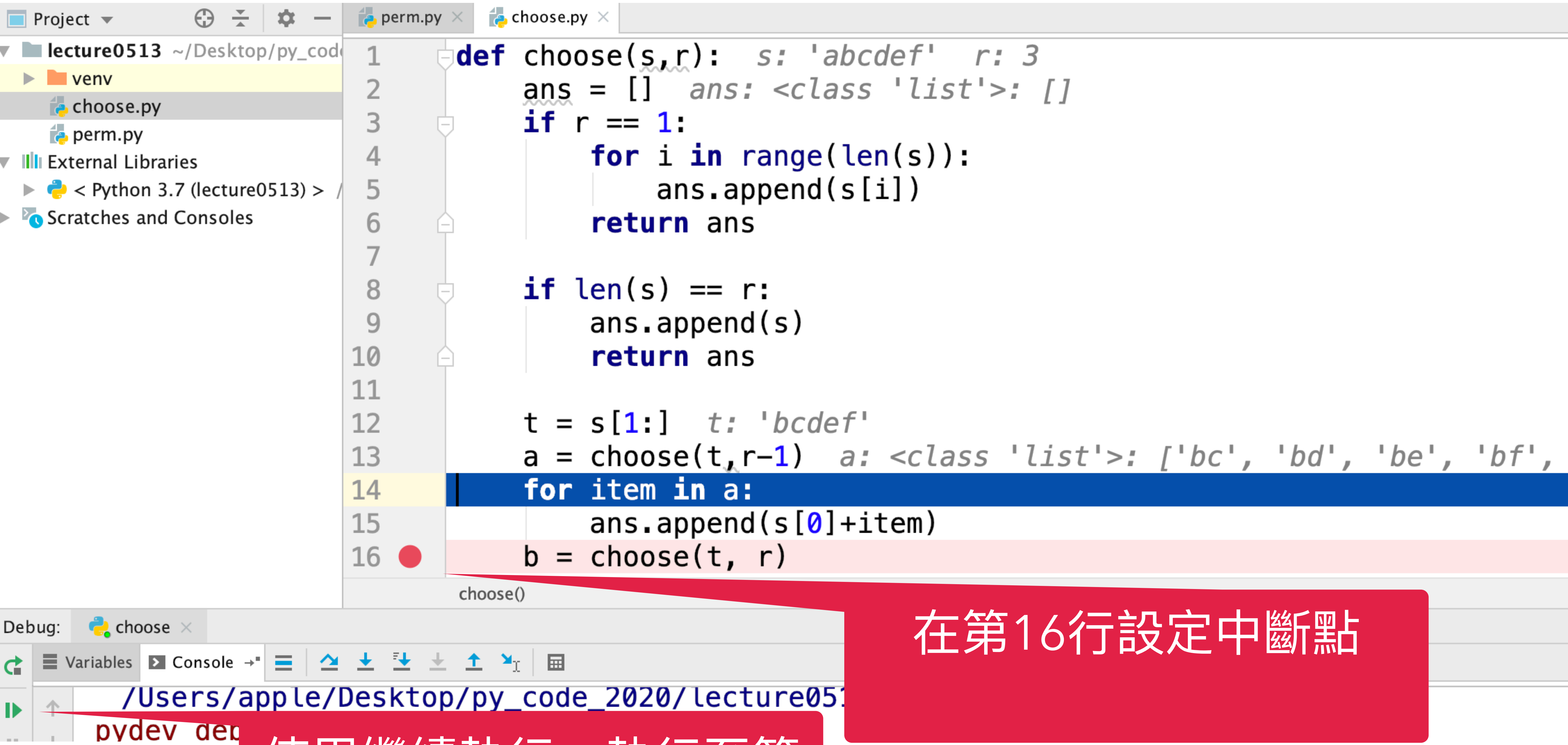
Step over
跳過副程式

執行第13行程式後，串列a中包含：從字串t中取r-1個字元的所有選擇



```
 1  def choose(s,r):   s: 'abcdef'   r: 3
 2      ans = []   ans: <class 'list'>: []
 3      if r == 1:
 4          for i in range(len(s)):
 5              ans.append(s[i])
 6          return ans
 7
 8      if len(s) == r:
 9          ans.append(s)
10          return ans
11
12      t = s[1:]   t: 'bcdef'
13      a = choose(t,r-1)   a: <class 'list'>: ['bc', 'bd', 'be', 'bf', 'cd', 'ce', 'cf', 'de', 'df', 'ef']
14      for item in a:
15          ans.append(s[0]+item)
16      b = choose(t, r)
```

choose()

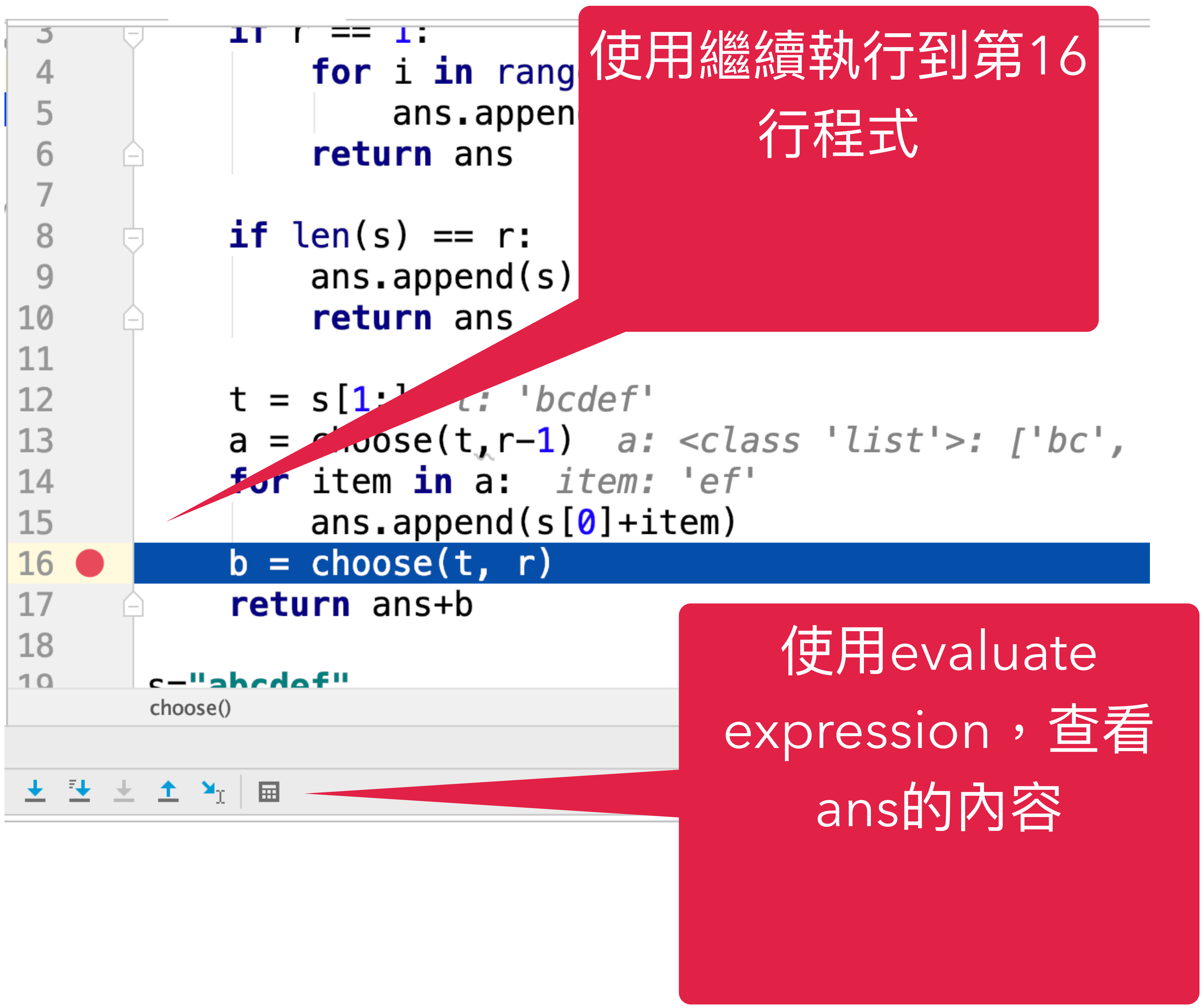


```
def choose(s,r):   s: 'abcdef'   r: 3
    ans = []   ans: <class 'list'>: []
    if r == 1:
        for i in range(len(s)):
            ans.append(s[i])
        return ans

    if len(s) == r:
        ans.append(s)
        return ans

    t = s[1:]   t: 'bcdef'
    a = choose(t,r-1)   a: <class 'list'>: ['bc', 'bd', 'be', 'bf',
    for item in a:
        ans.append(s[0]+item)
    b = choose(t, r)
```

```
/Users/apple/Desktop/py_code_2020/lecture05
pydev deb
```

在第16行設定中斷點

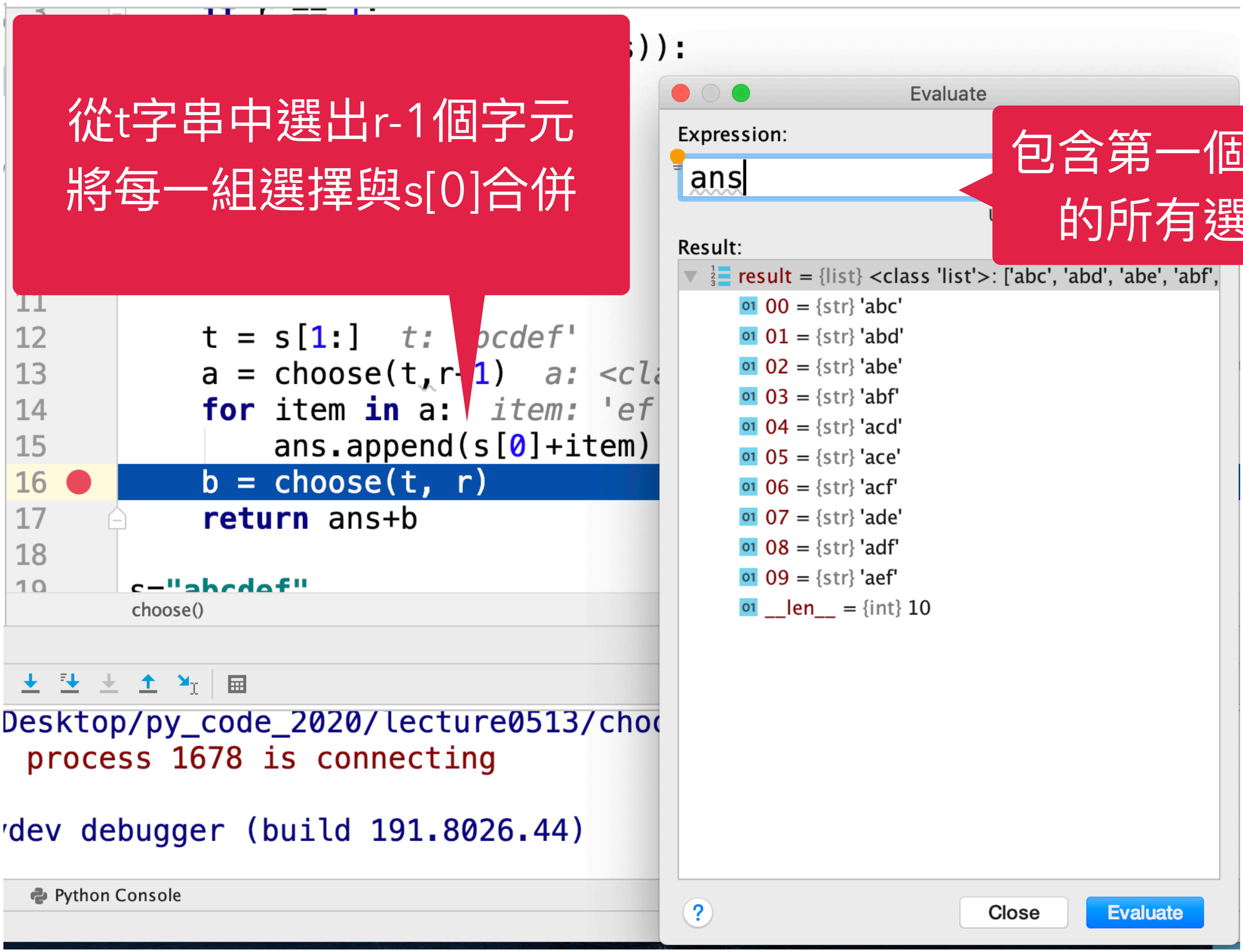
使用繼續執行，執行至第16行



```
    if r == 1:
        for i in rang
            ans.appen
        return ans

    if len(s) == r:
        ans.append(s)
        return ans

    t = s[1:]   t: 'bcdef'
    a = choose(t,r-1)   a: <class 'list'>: ['bc',
    for item in a:   item: 'ef'
        ans.append(s[0]+item)
    b = choose(t, r)
    return ans+b

s="abcdef"
```

使用繼續執行到第16行程式

使用evaluate expression，查看ans的內容



從t字串中選出r-1個字元
將每一組選擇與s[0]合併

包含第一個字元
的所有選擇

```
        t = s[1:]   t: 'bcdef'
        a = choose(t,r-1)   a: <cla
        for item in a:   item: 'ef
            ans.append(s[0]+item)
        b = choose(t, r)
        return ans+b

s="abcdef"
```

```
Desktop/py_code_2020/lecture0513/choo
 process 1678 is connecting

dev debugger (build 191.8026.44)
```

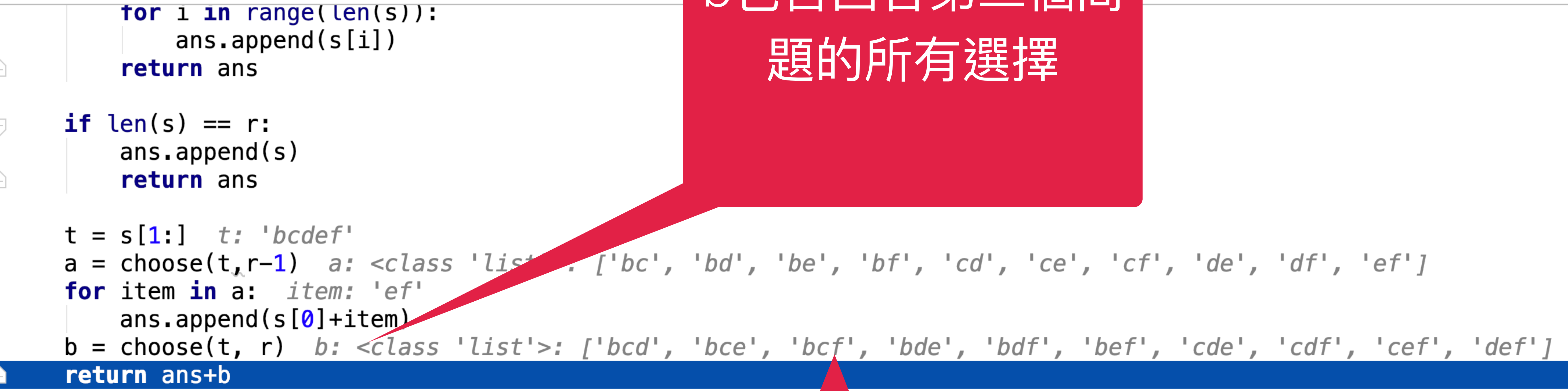
Evaluate

Expression:

ans

Result:

```
result = {list} <class 'list'>: ['abc', 'abd', 'abe', 'abf',
    00 = {str} 'abc'
    01 = {str} 'abd'
    02 = {str} 'abe'
    03 = {str} 'abf'
    04 = {str} 'acd'
    05 = {str} 'ace'
    06 = {str} 'acf'
    07 = {str} 'ade'
    08 = {str} 'adf'
    09 = {str} 'aef'
    __len__ = {int} 10
```



```
    for i in range(len(s)):
        ans.append(s[i])
    return ans

if len(s) == r:
    ans.append(s)
    return ans

t = s[1:]  t: 'bcdef'
a = choose(t,r-1)  a: <class 'lis  . ['bc', 'bd', 'be', 'bf', 'cd', 'ce', 'cf', 'de', 'df', 'ef']
for item in a:  item: 'ef'
    ans.append(s[0]+item)
b = choose(t, r)  b: <class 'list'>: ['bcd', 'bce', 'bcf', 'bde', 'bdf', 'bef', 'cde', 'cdf', 'cef', 'def']
return ans+b
```

b包含回答第二個問題的所有選擇

不包含"a"，從字串t中選擇r個字元

Project Study: translate records in a text file to table in a cvs file

INPUT: Recods in a text file

Dec 3
16 PLA aircraft, 6 PLAN vessels and 3 official ships operating around Taiwan were detected up until 6 a.m. (UTC+8) today. 10 of the aircraft crossed the median line and entered Taiwan’s northern, southwestern and southeastern ADIZ. We have monitored the situation and responded.

Dec 2
15 PLA aircraft, 7 PLAN vessels and 2 official ships operating around Taiwan were detected up until 6 a.m. (UTC+8) today. 11 of the aircraft crossed the median line and entered Taiwan’s southwestern and eastern ADIZ. We have monitored the situation and responded accordingly.

Dec 1
9 PLA aircraft, 7 PLAN vessels and 1 official ship operating around Taiwan were detected up until 6 a.m. (UTC+8) today. 3 of the aircraft crossed the median line and entered Taiwan’s northern and southwestern ADIZ. We have monitored the situation and responded accordingly.

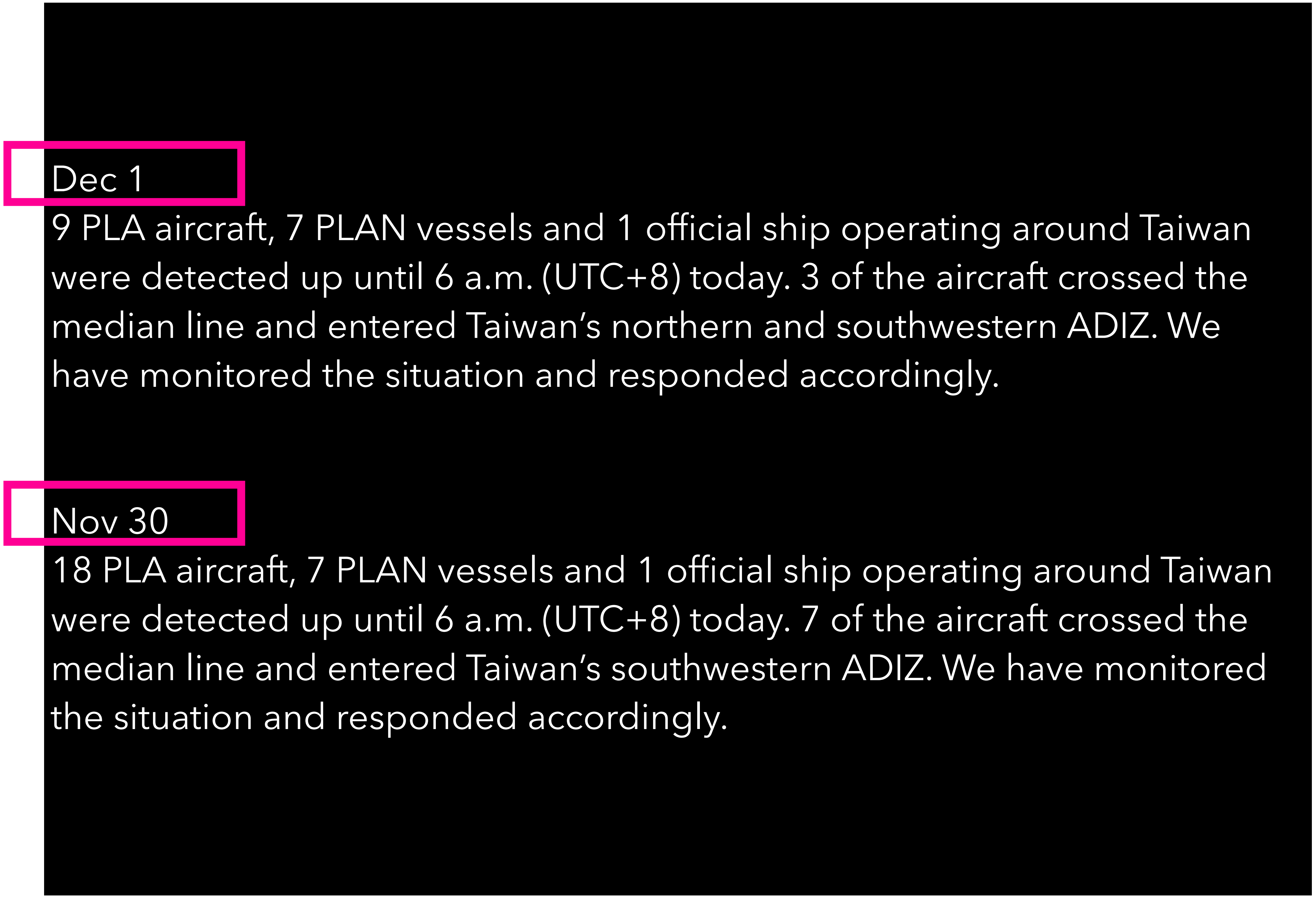
Nov 30
18 PLA aircraft, 7 PLAN vessels and 1 official ship operating around Taiwan were detected up until 6 a.m. (UTC+8) today. 7 of the aircraft crossed the median line and entered Taiwan’s southwestern ADIZ. We have monitored the situation and responded accordingly.

Nov 29
33 PLA aircraft and 8 PLAN vessels operating around Taiwan were detected up until 6 a.m. (UTC+8) today. 21 of the aircraft crossed the median line and entered Taiwan’s northern, southwestern and eastern ADIZ. We have monitored the situation and responded accordingly.

Nov 28
13 PLA aircraft and 7 PLAN vessels operating around Taiwan were detected up until 6 a.m. (UTC+8) today. 9 of the aircraft crossed the median line and entered Taiwan’s northern and southwestern ADIZ. We have monitored the situation and responded accordingly.

Each Recod : date and three numbers

The first line is for date



Dec 1

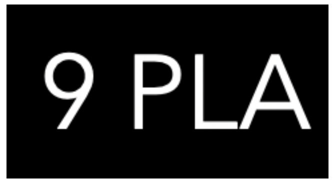
9 PLA aircraft, 7 PLAN vessels and 1 official ship operating around Taiwan were detected up until 6 a.m. (UTC+8) today. 3 of the aircraft crossed the median line and entered Taiwan's northern and southwestern ADIZ. We have monitored the situation and responded accordingly.

Nov 30

18 PLA aircraft, 7 PLAN vessels and 1 official ship operating around Taiwan were detected up until 6 a.m. (UTC+8) today. 7 of the aircraft crossed the median line and entered Taiwan's southwestern ADIZ. We have monitored the situation and responded accordingly.

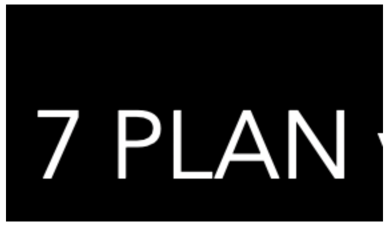
Keywords: PLA, PLAN, of the aircraft

- Case 1: the word before “PLA” is the first number



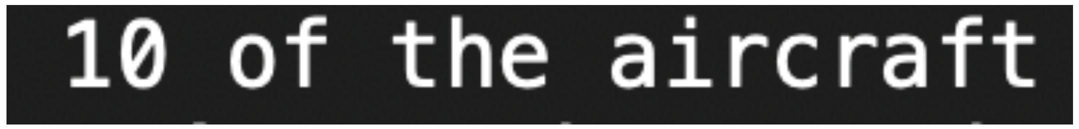
9 PLA

- Case 2: the word before PLAN is the second desired number



7 PLAN

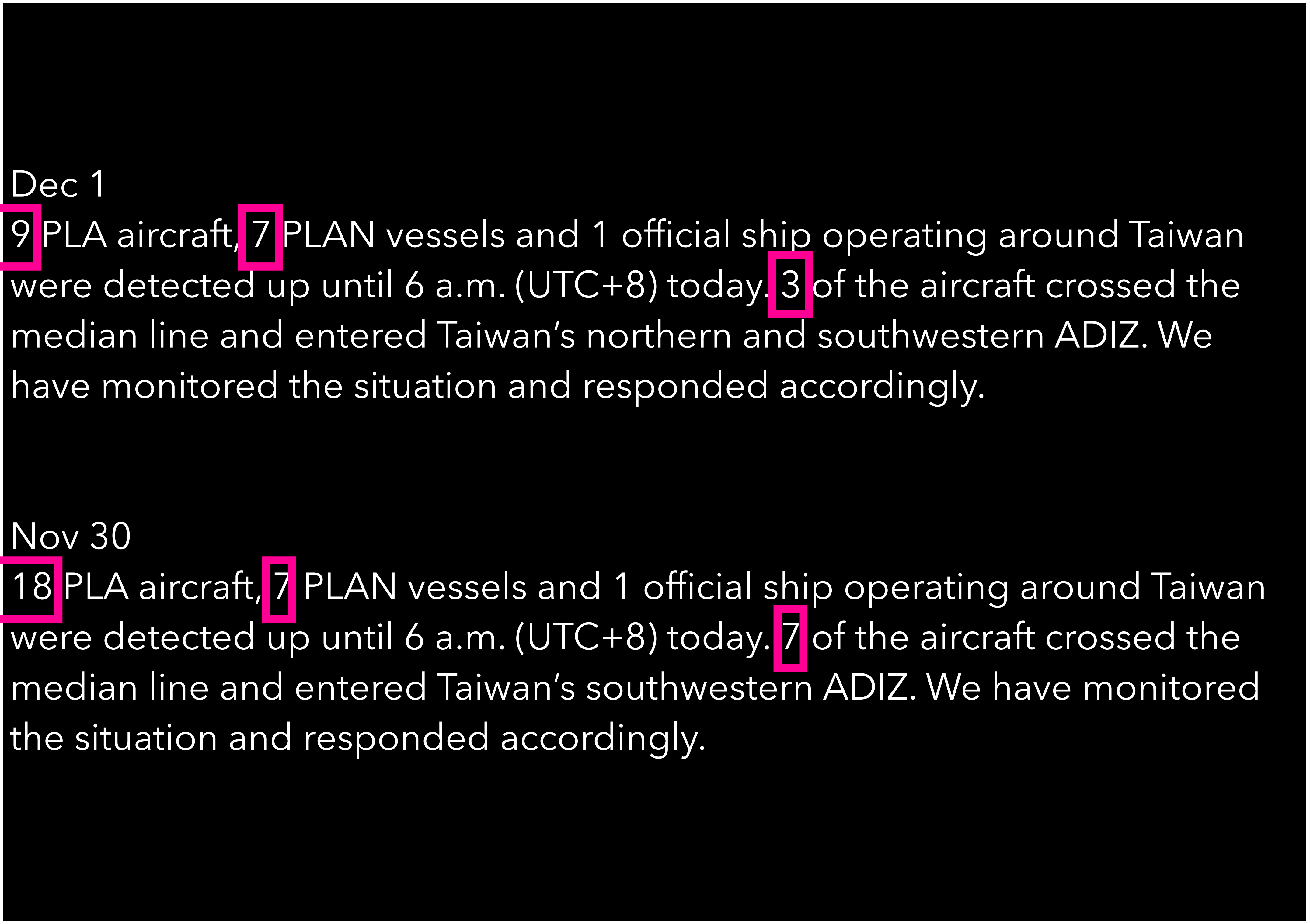
- Case 3: the word before “of the aircraft” is the third desired number



```
10 of the aircraft
```

Each Recod : date and three numbers

The second line is for three desired numbers



Dec 1

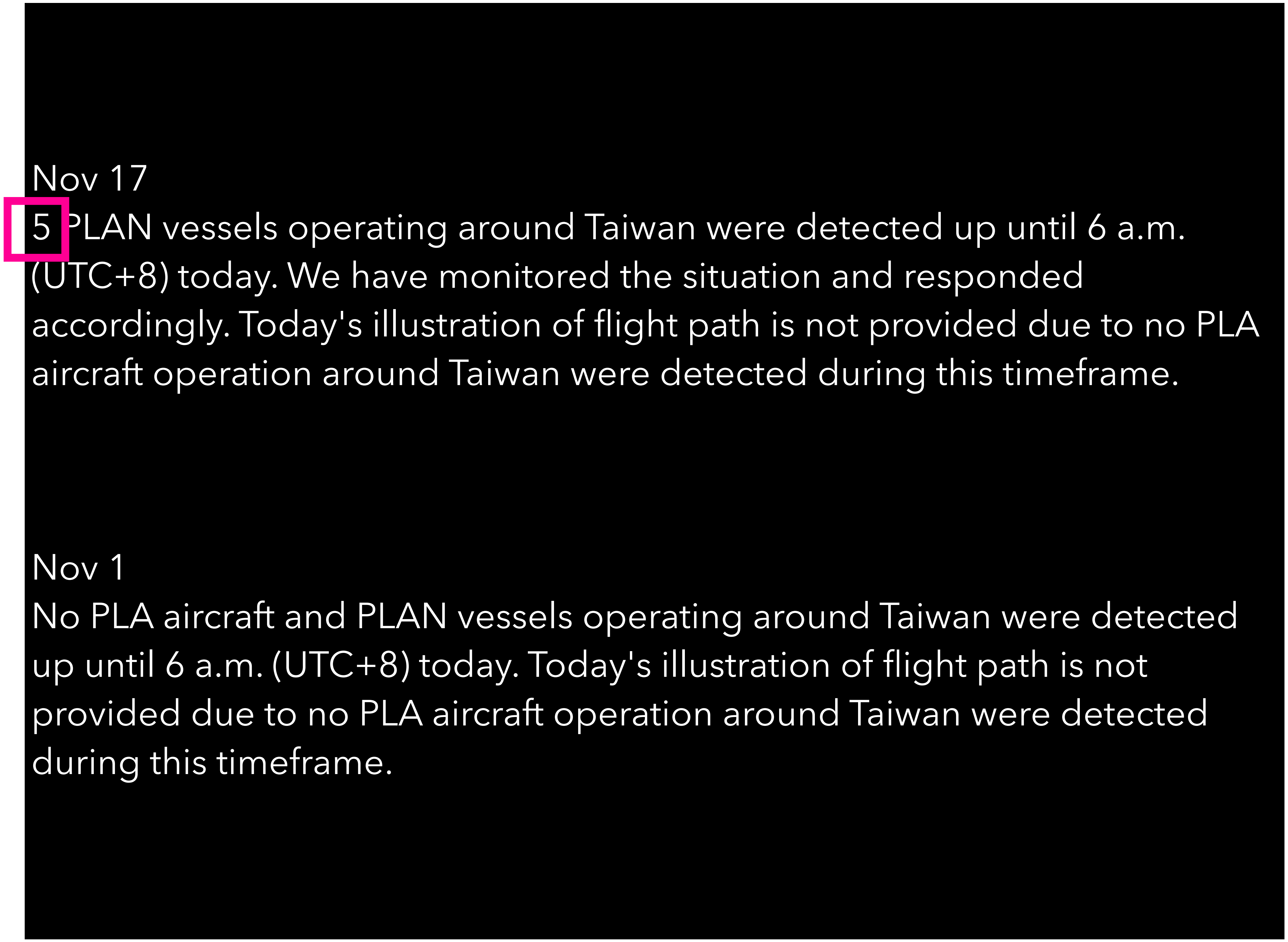
$\boxed{9}$ PLA aircraft, $\boxed{7}$ PLAN vessels and 1 official ship operating around Taiwan were detected up until 6 a.m. (UTC+8) today. $\boxed{3}$ of the aircraft crossed the median line and entered Taiwan's northern and southwestern ADIZ. We have monitored the situation and responded accordingly.

Nov 30

$\boxed{18}$ PLA aircraft, $\boxed{7}$ PLAN vessels and 1 official ship operating around Taiwan were detected up until 6 a.m. (UTC+8) today. $\boxed{7}$ of the aircraft crossed the median line and entered Taiwan's southwestern ADIZ. We have monitored the situation and responded accordingly.

Each Recod : date and three numbers

There may be only one number or no numbers in a record

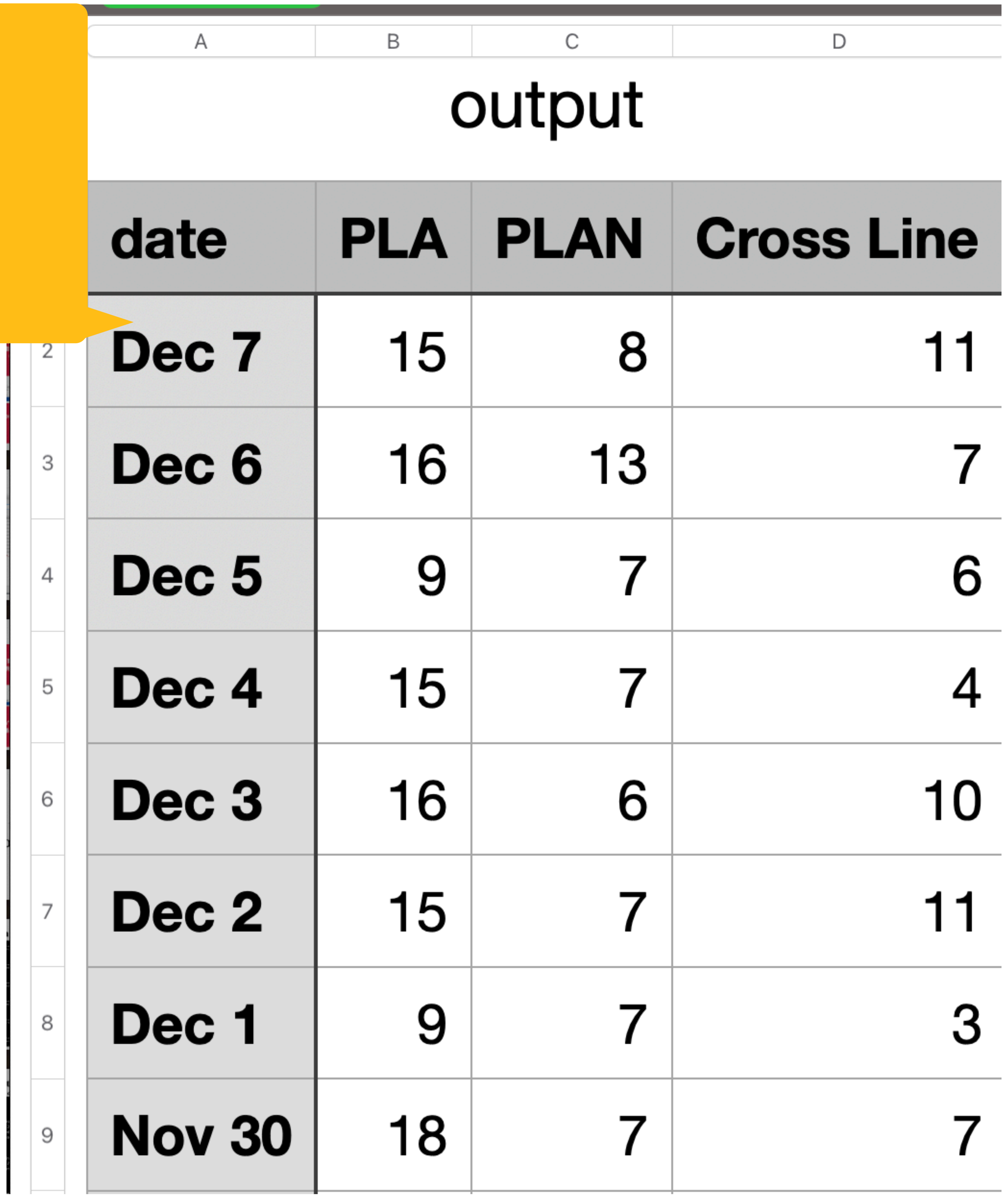


Nov 17
5 PLAN vessels operating around Taiwan were detected up until 6 a.m. (UTC+8) today. We have monitored the situation and responded accordingly. Today's illustration of flight path is not provided due to no PLA aircraft operation around Taiwan were detected during this timeframe.

Nov 1
No PLA aircraft and PLAN vessels operating around Taiwan were detected up until 6 a.m. (UTC+8) today. Today's illustration of flight path is not provided due to no PLA aircraft operation around Taiwan were detected during this timeframe.

OUTPUT: A table in a csv file

第1欄：日期



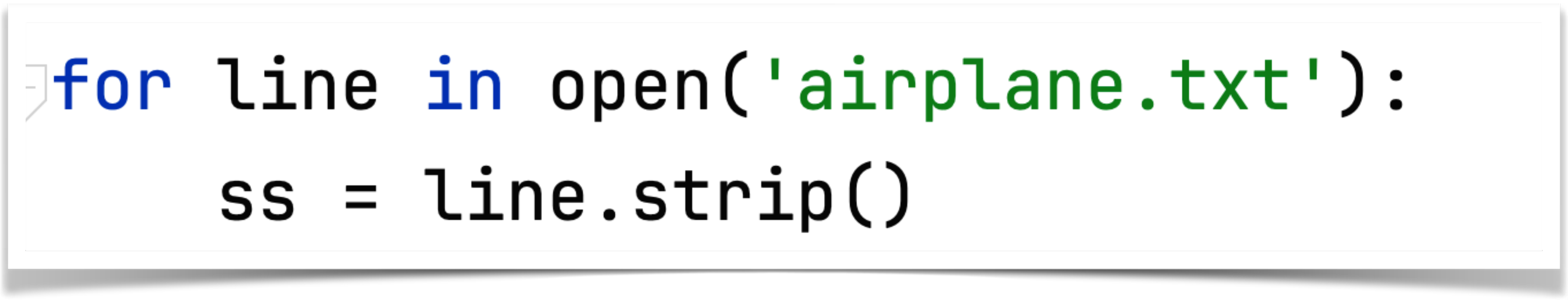
output

date	PLA	PLAN	Cross Line
Dec 7	15	8	11
Dec 6	16	13	7
Dec 5	9	7	6
Dec 4	15	7	4
Dec 3	16	6	10
Dec 2	15	7	11
Dec 1	9	7	3
Nov 30	18	7	7

第2-4欄：3個數字

Subproblems for solving this problem

- 開啟文字檔，應用for 迴圈將每一行儲存在字串ss

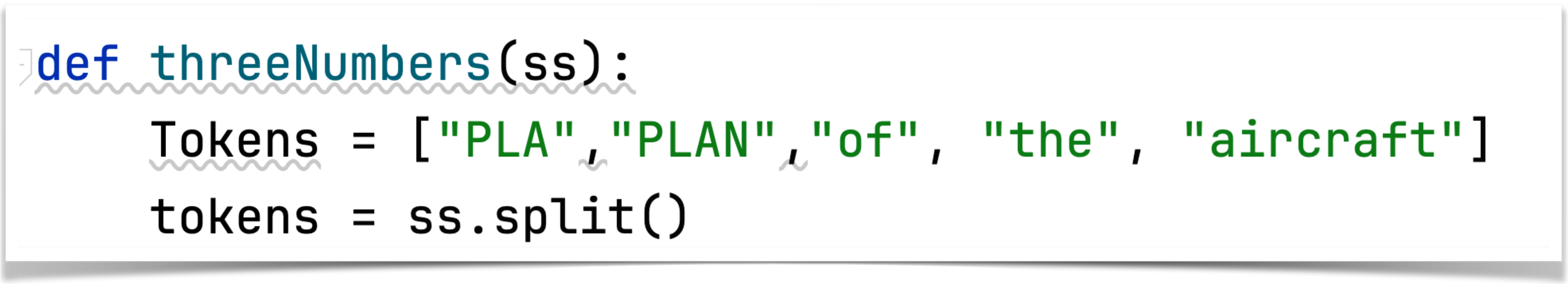


```python
for line in open('airplane.txt'):
    ss = line.strip()
```

- 使用if設定條件，分辨ss包含日期或是包含三個數字
  - 設計月份串列，以偵測ss的開頭3個字元是否為月份
- 設計日期串列dateRec，將代表日期的串列ss附加在串列dateRec

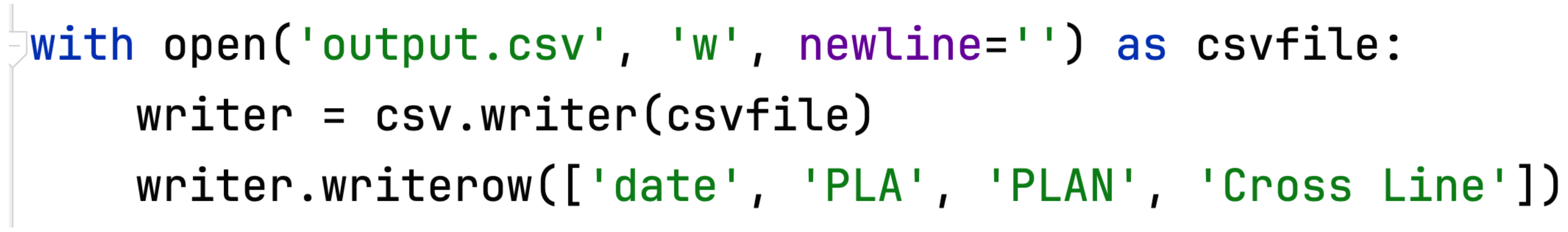
Subproblems for solving this problem

- 定義方法threeNumbers從字串ss中，擷取3個關鍵數字，並回傳



```python
def threeNumbers(ss):
    Tokens = ["PLA","PLAN","of", "the", "aircraft"]
    tokens = ss.split()
```

- 匯入csv套件，開啟輸出檔案，使用writerow方法，將資料一行一行寫入csvfile



```python
with open('output.csv', 'w', newline='') as csvfile:
    writer = csv.writer(csvfile)
    writer.writerow(['date', 'PLA', 'PLAN', 'Cross Line'])
```