

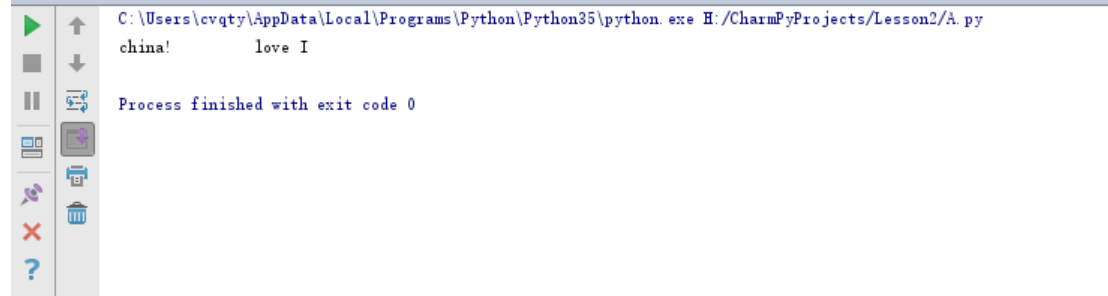
七月在线 python 基础第二节课作业

Author: 粽子

2016/11/03

A. 字符串按单词反转（必须保留所有空格）。'I love china!' 转化为" china! love I '
代码如下：（可能现在有点问题）

```
1 #Lesson2 Homework A
2 #单词反转
3 str = 'I love china!'
4 list_words = list(str.split(' '))
5 list_reverse_words = list(reversed(list_words))
6 str_ans = ' '.join(list_reverse_words)
7 print(str_ans)
```



B. 打印 100000 以内的所有素数（筛法求素数）

```
helloworld.py x A.py x B.py x C.py x D.py x E.py x
1  #Lesson2 Homework B
2  #打印100000以内所有素数
3  import math
4  def prime(n):
5      number = []
6      prime_number = []
7      for i in range(0, n, 1):
8          number.append(True)
9      number[0] = False
10     number[1] = False
11     for i in range(2, int(math.sqrt(n)), 1):
12         for j in range(i*i, n, i):
13             number[j] = False
14     for i in range(0, n, 1):
15         if number[i]:
16             prime_number.append(i)
17     return prime_number
18
19     ans = prime(100000)
20     print(len(ans))
21     print(ans)
22
```

```
Run B
C:\Users\cvqty\AppData\Local\Programs\Python\Python35\python.exe H:/CharmPyProjects/Lesson2/B.py
9592
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157]
Process finished with exit code 0
```

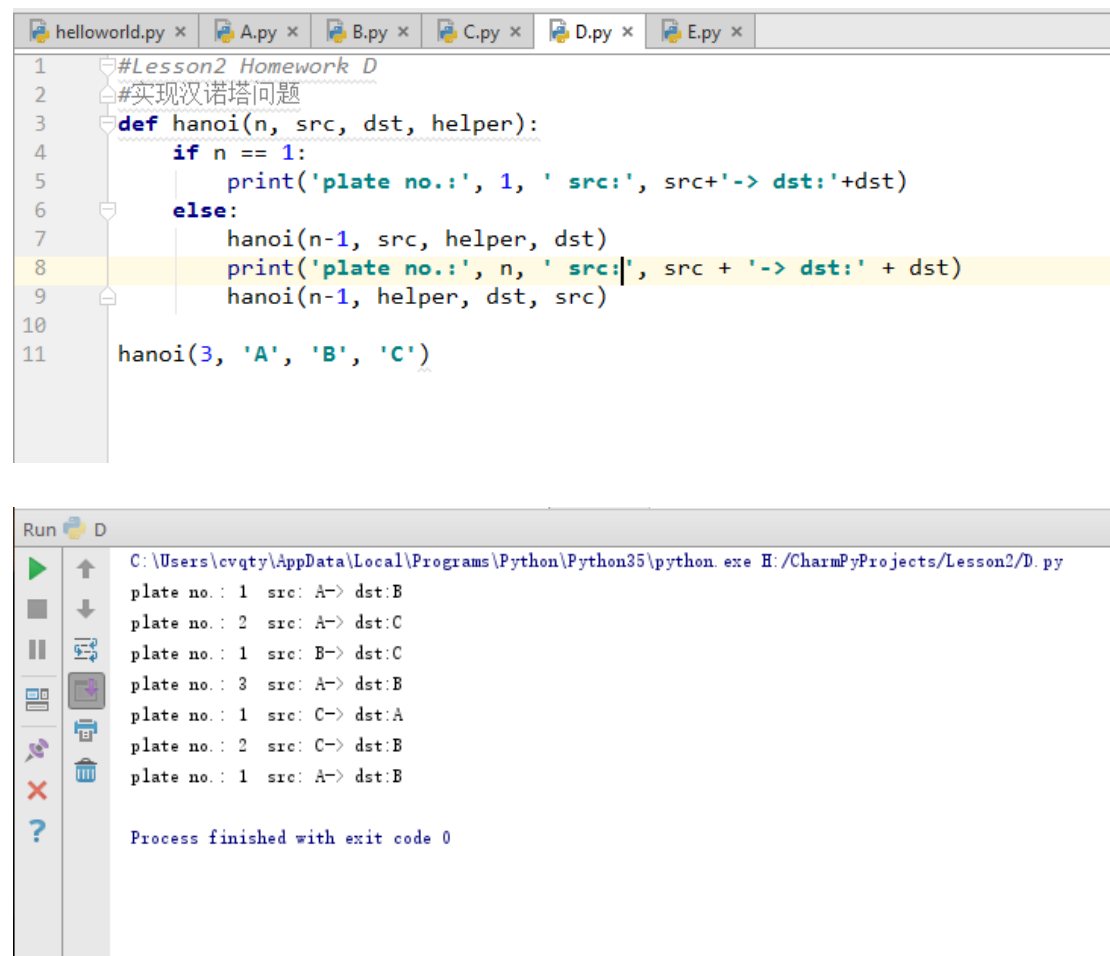
C. 自己实现一个函数支持可变参数

```
helloworld.py x A.py x B.py x C.py x D.py x E.py x
1 #Lesson2 Homework C
2 #写一个参数自己实现可变参数
3 def test_print(*args):
4     print(type(args))
5     print(args)
6     test_print(1, 2, 3, 4, 5, 6, 'what')
7
8 def test_dict(x, y, *name, **kvs):
9     print(x,y)
10    print(name)
11    print(kvs)
12    test_dict(1, 2, 3, 4, 5, wo='wy', json = '66')
13    test_dict(1, 3, 4)
14    #直接以字典的方式, 用 {}
15    test_dict(1, 2, *('a', 'b'), **{'china': 'bj', 'uk': 'london'})
```

```
Run C
C:\Users\cvqty\AppData\Local\Programs\Python\Python35\python.exe E:/CharmPyProjects/Lesson2/C.py
<class 'tuple'>
(1, 2, 3, 4, 5, 6, 'what')
1 2
(3, 4, 5)
{'json': '66', 'wo': 'wy'}
1 3
(4,)
{}
1 2
('a', 'b')
{'uk': 'london', 'china': 'bj'}

Process finished with exit code 0
```

D. 自己实现函数解决 hanoi 塔问题



The screenshot displays a Python IDE with a file named `D.py` open. The code implements a recursive function `hanoi` to solve the Hanoi Tower problem. The function takes four arguments: `n` (number of disks), `src` (source peg), `dst` (destination peg), and `helper` (the auxiliary peg). The base case is when `n == 1`, where a single move is printed. For `n > 1`, the function recursively moves `n-1` disks from `src` to `helper`, then moves the `n`th disk from `src` to `dst`, and finally moves the `n-1` disks from `helper` to `dst`. The main part of the code calls `hanoi(3, 'A', 'B', 'C')` to solve the problem with 3 disks, starting at peg A, moving to peg B, and using peg C as the helper.

```
1 #Lesson2 Homework D
2 #实现汉诺塔问题
3 def hanoi(n, src, dst, helper):
4     if n == 1:
5         print('plate no.: 1, ' src:', src+'-> dst:'+dst)
6     else:
7         hanoi(n-1, src, helper, dst)
8         print('plate no.: ' n, ' src:', src + '-> dst:' + dst)
9         hanoi(n-1, helper, dst, src)
10
11 hanoi(3, 'A', 'B', 'C')
```

The Run console shows the execution of the program, displaying the sequence of moves for 3 disks. The output is as follows:

```
C:\Users\cvqty\AppData\Local\Programs\Python\Python35\python.exe H:/CharmPyProjects/Lesson2/D.py
plate no.: 1 src: A-> dst:B
plate no.: 2 src: A-> dst:C
plate no.: 1 src: B-> dst:C
plate no.: 3 src: A-> dst:B
plate no.: 1 src: C-> dst:A
plate no.: 2 src: C-> dst:B
plate no.: 1 src: A-> dst:B
Process finished with exit code 0
```

E. 实现一个 `sort` 函数，通过参数指定比较函数用来实现按不同顺序进行排序。

主要就用了冒泡排序，默认从小到大，如果指定比较参数就从大到小

```

1 #Lesson2 Homework E
2 #通过参数指定比较函数来实现冒泡排序
3 def BubbleSort(nums, size, desc = None):
4     for i in range(1, size, 1):
5         print('the ith time:', i)
6         print('before the ith times:', nums)
7         for j in range(0, size-i, 1):
8             if not desc:
9                 if nums[j] > nums[j+1]:
10                    print('before swap: j=', j, 'nums[j]=', nums[j], 'nums[j+1]=', nums[j+1])
11                    temp = nums[j]
12                    nums[j] = nums[j+1]
13                    nums[j+1] = temp
14                    print('after swap: j=', j, 'nums[j]=', nums[j], 'nums[j+1]=', nums[j+1], 'nums:', nums)
15                else:
16                    pass
17            else:
18                (nums[j], nums[j+1]) = desc(nums[j], nums[j+1])
19        print('after ith time: ', nums)
20        print(nums)
21
22    def desc(x, y):
23        if x < y:
24            return y, x
25        else:
26            return x, y
27
28    BubbleSort([4, 3, 5, 1, 2], 5, desc)

```

```

Run E
C:\Users\evqty\AppData\Local\Programs\Python\Python35\python.exe H:/CharmPyProjects/Lesson2/E.py

the ith time: 1
before the ith times: [4, 3, 5, 1, 2]
after ith time: [4, 5, 3, 2, 1]
the ith time: 2
before the ith times: [4, 5, 3, 2, 1]
after ith time: [5, 4, 3, 2, 1]
the ith time: 3
before the ith times: [5, 4, 3, 2, 1]
after ith time: [5, 4, 3, 2, 1]
the ith time: 4
before the ith times: [5, 4, 3, 2, 1]
after ith time: [5, 4, 3, 2, 1]
[5, 4, 3, 2, 1]

Process finished with exit code 0

```