

编写一个函数，能够接受至少2个参数，返回最小值和最大值

```
import random
def double_values(*nums):
    print(nums)
    return max(nums), min(nums)

print(*double_values(*[random.randint(10,20) for _ in range(10)]))
```

编写一个函数，接受一个参数n，n为正整数，左右两种打印方式。要求数字必须对齐

上三角

```

        1
       2 1
      3 2 1
     4 3 2 1
    5 4 3 2 1
   6 5 4 3 2 1
  7 6 5 4 3 2 1
 8 7 6 5 4 3 2 1
9 8 7 6 5 4 3 2 1
10 9 8 7 6 5 4 3 2 1
11 10 9 8 7 6 5 4 3 2 1
12 11 10 9 8 7 6 5 4 3 2 1
```

思路一

一行行打印，前面追加空格，每一个空格的宽度等于数字字符串的宽度。

```
def triangle_print(n):
    for i in range(1, n+1):
        for j in range(n, 0, -1):
            if i < j:
                print(' '*len(str(j)), end = ' ')
            else:
                print(j, end = ' ')
        print()
```

思路二

右对齐方式。右对齐，最大的问题不知道最后一行多宽？

```
def triangle_print(n):
    tail = " ".join([str(i) for i in range(n,0,-1)])
    width = len(tail)
    for i in range(1,n):
        print("{:>{}}".format(" ".join([str(j) for j in range(i,0,-1)]), width))
    print(tail)
triangle_print(12)
```

先算出最后一行，知道了宽度，采用右对齐打印，最后补上最后一行。

思路三

基于思路二，能够每一行不重新计算，基于tail得到？

```
def triangle_print(n):
    tail = ' '.join(str(i) for i in range(n, 0, -1))
    width = len(tail)
    start = -1
    step = 2
    points = {10**i for i in range(1,3)} # 与其判断两位数、三位数，还不如算好临界值
    for i in range(1, n+1):
        print('{:>{}}'.format(tail[start:], width))
        if i+1 in points:
            step += 1
        start = start - step
```

下三角

```
12 11 10 9 8 7 6 5 4 3 2 1
 11 10 9 8 7 6 5 4 3 2 1
   10 9 8 7 6 5 4 3 2 1
    9 8 7 6 5 4 3 2 1
     8 7 6 5 4 3 2 1
      7 6 5 4 3 2 1
       6 5 4 3 2 1
        5 4 3 2 1
         4 3 2 1
          3 2 1
           2 1
            1
```

```
def showtail(n):
    tail = " ".join([str(i) for i in range(n,0,-1)])
    print(tail)
    # 无需再次生成列表
    for i in range(len(tail)):
        if tail[i] == ' ':
            print(' '*i,tail[i+1:])
showtail(12)
```

核心思想

遇到一个空格，就把前面全部补成空格后面数字和空格全部打印。

由此，可以得到上三角另一种思路

```
def triangle_print(n):
    tail = ' '.join(str(i) for i in range(n, 0, -1))
    width = len(tail)
    for i in range(-1, -width-1, -1):
        if tail[i] == ' ':
            print(' '*(width+i),tail[i+1:])
    print(tail)
```

