In [1]:

```
# 默认参数
def cal_0(money, rate=0.1):
    return money + money * rate

print(cal_0(100))
print(cal_0(100,0.2))
```

```
110.0
120.0
```

In [2]:

```
 def cal_1(money, bonus=1000, month=12):
    i = money * month + bonus
    return i

print(cal_1(5000))
print(cal_1(5000, 2000))
print(cal_1(5000, 2000, 10))
```

```
61000
62000
52000
```

In [1]:

```
def draw_triangle(n=5):

    for i in range(n+1):
        print(' '*(n-i),'*'*(2*i-1))

draw_triangle(7)
draw_triangle()
```

```
       *
      ***
     *****
    *******
   *********
  ***********
 *************

     *
    ***
   *****
  *******
 *********
```

In [13]:

```
# 可变长度的参数 元组
def cal_2(kind, *numbers):
    if kind == 'avg':
        n = 0
        for i in numbers:
            n = n + i
        return n / len(numbers)

print(cal_2('avg', 1,2,3,4))
```

```
2.5
```

In [17]:

```
# 可变长度的参数 元组
def cal_2(kind, *numbers):
    if kind == 'avg':
        n = 0
        for i in numbers:
            n = n + i
        return n / len(numbers)
    if kind == 'sum':
        n = 0
        for i in numbers:
            n = n + i
        return n

print(cal_2('avg', 1,2,3,4))
print(cal_2('sum', 1,2,3,4))
```

```
2.5
10
```

In [20]:

```
# 可变长度的参数 元组
def cal_2(kind, *numbers):

    n = 0
    for i in numbers:
        n = n + i

    if kind == 'avg':
        return n / len(numbers)
    if kind == 'sum':
        return n

print(cal_2('avg', 1,2,3,4))
print(cal_2('sum', 1,2,3,4))
```

```
2.5
10
```

In [3]:

```
# 可变长度参数 元组 字典
def cal_3(*numbers, **students):

    for i in numbers:
        print(i)
    for k,v in students.items():
        print(k,v)

numbers = [1,2,3,4,5]
students = {'tom':90, 'jerry':95, 'mary':100}
cal_3(*numbers, **students)
```

```
[1, 2, 3, 4, 5]
jerry 95
tom 90
mary 100
```

In [5]:

```
# base on cal_2
def cal_4(*numbers, kind):

    n = 0
    for i in numbers:
        n = n + i

    if kind == 'avg':
        return n / len(numbers)
    if kind == 'sum':
        return n

print(cal_4(1,2,3,4, kind='avg'))
print(cal_4(1,2,3,4, kind='sum'))
```

```
2.5
10
```

In [3]:

```
# base on cal_2,
# 传递元组和字典参数的最合适写法
def cal_5(*numbers, **kind):

    # 判断是否有 kind 这个 key
    if 'kind' in kind:
        kind_value = kind.get('kind')

    n = 0
    for i in numbers:
        n = n + i

    if kind_value == 'avg':
        return n / len(numbers)
    if kind_value == 'sum':
        return n

print(cal_5(1,2,3,4, kind='avg'))
print(cal_5(1,2,3,4, kind='sum'))
```

```
2.5
10
```

In [4]:

```
# kind 中 增加 max key,
# max = ingnore,  则忽略最大值
def cal_6(*numbers, **kind):

    if 'kind' in kind:
        kind_value = kind.get('kind')

    if 'max' in kind:
        if kind.get('max') == 'ignore':
            numbers = list(numbers)
            numbers.remove(max(numbers))

    n = 0
    for i in numbers:
        n = n + i

    if kind_value == 'avg':
        return n / len(numbers)
    if kind_value == 'sum':
        return n

print(cal_6(1,2,3,4, kind='avg', max='ignore'))
print(cal_6(1,2,3,4, kind='avg'))
print(cal_6(1,2,3,4, kind='sum'))
```

```
2.0
2.5
10
```

In [5]:

```
# kind 中 增加 min key,
# min key = double, 则最小值计算两次
def cal_7(*numbers, **kind):

    numbers = list(numbers)

    if 'kind' in kind:
        kind_value = kind.get('kind')

    if 'max' in kind:
        if kind.get('max') == 'ignore':
            numbers.remove(max(numbers))

    if 'min' in kind:
        if kind.get('min') == 'double':
            numbers.append(min(numbers))

    n = 0
    for i in numbers:
        n = n + i

    if kind_value == 'avg':
        return n / len(numbers)
    if kind_value == 'sum':
        return n

print(cal_7(1,2,3,4, kind='avg', max='ignore', min='double'))
print(cal_7(1,2,3,4, kind='avg'))
print(cal_7(1,2,3,4, kind='sum'))
```

```
1.75
2.5
10
```

In [6]:

```
def add(x, y):
    return x + y

lambda x, y: x + y
```

Out[6]:

```
<function __main__.<lambda>>
```

In [7]:

```
a =  lambda x, y=2 : x + y
a(3)
```

Out[7]:

```
5
```

In [8]:

```
a(3, 5)
```

Out[8]:

```
8
```

In [9]:

```
a = lambda x : x * x +40

print(a(2))
```

```
44
```

In [ ]:

```

```