

## 作业要求

- 作业一：打印乘法口诀表

思路及伪代码

- 使用for循环，两层
- 使用range()函数生成数列

最后我想要的结果：

```
1 * 1 = 1
2 * 1 = 2 2 * 2 = 4
3 * 1 = 3 3 * 2 = 6 3 * 3 = 9
4 * 1 = 4 4 * 2 = 8 4 * 3 = 12 4 * 4 = 16
5 * 1 = 5 5 * 2 = 10 5 * 3 = 15 5 * 4 = 20 5 * 5 = 25
6 * 1 = 6 6 * 2 = 12 6 * 3 = 18 6 * 4 = 24 6 * 5 = 30 6 * 6 = 36
7 * 1 = 7 7 * 2 = 14 7 * 3 = 21 7 * 4 = 28 7 * 5 = 35 7 * 6 = 42 7 * 7 = 49
8 * 1 = 8 8 * 2 = 16 8 * 3 = 24 8 * 4 = 32 8 * 5 = 40 8 * 6 = 48 8 * 7 = 56 8 * 8 = 64
9 * 1 = 9 9 * 2 = 18 9 * 3 = 27 9 * 4 = 36 9 * 5 = 45 9 * 6 = 54 9 * 7 = 63 9 * 8 = 72 9 * 9 = 81
```

## 分析

- 1、因涉及两个数相乘，且两个数都不是常量，故需要两个变量。取名i、j。
- 2、因两个变量都是有规律地变化，简单分析其规律如下：

i	j
1	1
2	1、2
3	1、2、3
4	1、2、3、4
5	1、2、3、4、5
6	1、2、3、4、5、6
7	1、2、3、4、5、6、7
8	1、2、3、4、5、6、7、8
9	1、2、3、4、5、6、7、8、9

分析得到的规律：j的取值固定从1开始，最大与i值相同。

## 技术实现：

第一次for循环实现：

```
for i in range(1,10)
```

第二层for循环实现：

```
for i in range(1,i+1)
```

### 难点：

- 1、确定变量i和变量j的关联关系。
- 2、每次i循环完再换行。

### 正式实现代码V1.0

```
#coding=utf-8
```

```
# 作者： 杨公旺
```

```
# 时间： 2017-11-27
```

```
# 版本： V1.0
```

```
for i in range(1,10):
    for j in range(1,i+1):
        number=i*j
        print "%s * %s = %s"%(i,j,number),
    print " "
```

效果展示：

```
1 * 1 = 1
2 * 1 = 2 2 * 2 = 4
3 * 1 = 3 3 * 2 = 6 3 * 3 = 9
4 * 1 = 4 4 * 2 = 8 4 * 3 = 12 4 * 4 = 16
5 * 1 = 5 5 * 2 = 10 5 * 3 = 15 5 * 4 = 20 5 * 5 = 25
6 * 1 = 6 6 * 2 = 12 6 * 3 = 18 6 * 4 = 24 6 * 5 = 30 6 * 6 = 36
7 * 1 = 7 7 * 2 = 14 7 * 3 = 21 7 * 4 = 28 7 * 5 = 35 7 * 6 = 42 7 * 7 = 49
8 * 1 = 8 8 * 2 = 16 8 * 3 = 24 8 * 4 = 32 8 * 5 = 40 8 * 6 = 48 8 * 7 = 56 8 * 8 = 64
9 * 1 = 9 9 * 2 = 18 9 * 3 = 27 9 * 4 = 36 9 * 5 = 45 9 * 6 = 54 9 * 7 = 63 9 * 8 = 72 9 * 9 = 81
```

### 第一次优化：

1、从上面的输出效果可以看出，每次的输出间隔太小，不便于阅读，可以尝试增加间隔的方式，使输出结果更易读。

2、输出的结果大数字在前、小数字在后，与我们实际使用相反，调整两个数字的位置，以更符合使用习惯。

代码：

```
#coding=utf-8
```

```
# 作者： 杨公旺
```

```
# 时间： 2017-11-27
```

```
# 版本： V1.1
```

```
# 变更内容： 1、输出结果增加空格； 2、调整输出结果值i、j的位置顺序。
```

```
for i in range(1,10):
```

```

for j in range(1,i+1):
    number=i*j
    print "%s * %s = %s" % (j,i,number),
    print " "

```

效果：

```

1 * 1 = 1
1 * 2 = 2    2 * 2 = 4
1 * 3 = 3    2 * 3 = 6    3 * 3 = 9
1 * 4 = 4    2 * 4 = 8    3 * 4 = 12    4 * 4 = 16
1 * 5 = 5    2 * 5 = 10   3 * 5 = 15    4 * 5 = 20    5 * 5 = 25
1 * 6 = 6    2 * 6 = 12   3 * 6 = 18    4 * 6 = 24    5 * 6 = 30    6 * 6 = 36
1 * 7 = 7    2 * 7 = 14   3 * 7 = 21    4 * 7 = 28    5 * 7 = 35    6 * 7 = 42    7 * 7 = 49
1 * 8 = 8    2 * 8 = 16   3 * 8 = 24    4 * 8 = 32    5 * 8 = 40    6 * 8 = 48    7 * 8 = 56    8 * 8 = 64
1 * 9 = 9    2 * 9 = 18   3 * 9 = 27    4 * 9 = 36    5 * 9 = 45    6 * 9 = 54    7 * 9 = 63    8 * 9 = 72    9 * 9 = 81

```

## 第二次优化：

经过第一次优化，每一列的显示相对清晰了很多，但是，第二行、第三行的出现了未对齐的情况，如下图。因此，需要进一步调整格式，以使每一列整齐排列。

```

1 * 1 = 1
1 * 2 = 2    2 * 2 = 4
1 * 3 = 3    2 * 3 = 6    3 * 3 = 9
1 * 4 = 4    2 * 4 = 8    3 * 4 = 12    4 * 4 = 16
1 * 5 = 5    2 * 5 = 10   3 * 5 = 15    4 * 5 = 20    5 * 5 = 25
1 * 6 = 6    2 * 6 = 12   3 * 6 = 18    4 * 6 = 24    5 * 6 = 30    6 * 6 = 36
1 * 7 = 7    2 * 7 = 14   3 * 7 = 21    4 * 7 = 28    5 * 7 = 35    6 * 7 = 42    7 * 7 = 49
1 * 8 = 8    2 * 8 = 16   3 * 8 = 24    4 * 8 = 32    5 * 8 = 40    6 * 8 = 48    7 * 8 = 56    8 * 8 = 64
1 * 9 = 9    2 * 9 = 18   3 * 9 = 27    4 * 9 = 36    5 * 9 = 45    6 * 9 = 54    7 * 9 = 63    8 * 9 = 72    9 * 9 = 81

```

优化后的代码：

#coding=utf-8

# 作者：杨公旺

# 时间：2017-11-27

# 版本：V1.2

# V1.1变更内容：1、输出结果增加空格；2、调整输出结果值i、j的位置顺序。

# V1.2变更内容：1、对输出进行判断，通过增加空格的方式使输出结果对齐。

```

for i in range(1,10):
    for j in range(1,i+1):
        number=i*j
        if(i==3 and j==3):
            print " %s * %s = %s" % (j,i,number),
        elif(i==4 and j==3):
            print " %s * %s = %s" % (j,i,number),
        else:
            print "%s * %s = %s" % (j,i,number),
        print " "

```

效果：

```
1 * 1 = 1
1 * 2 = 2   2 * 2 = 4
1 * 3 = 3   2 * 3 = 6   3 * 3 = 9
1 * 4 = 4   2 * 4 = 8   3 * 4 = 12   4 * 4 = 16
1 * 5 = 5   2 * 5 = 10  3 * 5 = 15   4 * 5 = 20   5 * 5 = 25
1 * 6 = 6   2 * 6 = 12  3 * 6 = 18   4 * 6 = 24   5 * 6 = 30   6 * 6 = 36
1 * 7 = 7   2 * 7 = 14  3 * 7 = 21   4 * 7 = 28   5 * 7 = 35   6 * 7 = 42   7 * 7 = 49
1 * 8 = 8   2 * 8 = 16  3 * 8 = 24   4 * 8 = 32   5 * 8 = 40   6 * 8 = 48   7 * 8 = 56   8 * 8 = 64
1 * 9 = 9   2 * 9 = 18  3 * 9 = 27   4 * 9 = 36   5 * 9 = 45   6 * 9 = 54   7 * 9 = 63   8 * 9 = 72   9 * 9 = 81
```



Generic File