

```

#格式化字符串
#全局变量，局部变量，作用域
# a=10 #全局变量
# def fun1():
#     global a #声明此变量为全局变量,如果不声明，则函数内的变量都是局部变量
#     a=9 #局部变量
#     return a
# #函数必须调用才能生效，不调用的情况下，函数不会执行，下面两句话的顺序如果反了，就会报错
# print(fun1())
# print(a)
#假设你要在控制台显示a+b=a+b,应该怎么处理?
# a=12
# b=90
# print(str(a)+'+'+str(b)+'='+str(a+b))

#方案一,%的方式
# print('%s+%s=%s'%(a,b,a+b))
# info='我叫%5s,你叫%-5s,他叫%5s, 编号是%05d'%( '张三', '李四', '王五', 20)
#方案一的前后参数的数量要一致
# print(info)
#默认都是右对齐，如果想改成左对齐，在位数的前面加负号
#%5s,%是固定写法，5表示最少显示多少位，s表示显示的是字符串,d表示显示的数字
#如果需要在数字前补0，则写成%0nd,n表示最少显示多少位
# info='我叫%-5s,你叫%-5s,他叫%-5s, 编号是%05s'%( '高手名字不能太长', '李四', '王五', 20)
# print(info)
#6.2f最少显示6位，其中2位小数，如果没有指定小数点后的位数，则默认保留6位小数
# number1=('%6.2f'%(3.1))
# print(number1)
#方案二
#可以用:n表示最少显示多少位
# print('My name is{:6},Your name is{:6}'.format('刘德华','张学友'))
#对齐方式的修改,>n表示右对齐,<n表示左对齐,^表示居中对齐
# print('My name is{:>6},Your name is{:^6}'.format('刘德华','张学友'))
#前面的{}和后面的参数个数不一致，会发生什么? ,如果后面的参数比前面的多，并不会报错
# print('My name is{:>6}'.format('刘德华','张学友'))
#如果前面的{}数量比后面的参数多，就会报错
# print('My name is{:>6},Your name is{:^6}'.format('刘德华'))
#方案二中， 字符串默认是左对齐，数字默认是右对齐。 ,n前面写个0一样表示补位数
# print('My name is{:6},Your name is{:06}'.format('刘德华',23)) #23是数字形式
#print('My name is{:6},Your name is{:6}'.format('刘德华','23')) #23是字符串形式
#可以用下标法来表示
# print('My name is{1},Your name is{0}'.format('刘德华',23))
#如果本来就要打印{}怎么办?
# print('My name is{}}},Your name is{}}}'.format('刘德华',23))
#在Python3.6之后的版本中，可以使用f'的写法,f表示格式化字符串的format
# a='刘德华'
# b='张学友'
# print(f'My name is {a},your name is {b}')

```