

## 5.30总结：函数

### 命名法

驼峰命名法：首字母小写，后面的首字母大写 helloWorld

帕斯卡命名法：每个首字母都大写（常用于类名） HelloWorld

系统命名法：单词都小写，以下划线\_划分 hello\_world

### 函数

函数代码块以 `def` 关键词开头 `return [表达式]` 结束函数并返回一个或多个值给调用方，不带表达式的 `return` 相当于返回 `None`，多个值为元组

```
# 声明
def sayHello():
    print("你好")

# 调用
sayHello()

# 根据两个数求和
def getSum(n1,n2=18): # n2设置默认值，有默认值的参数必须写在最后
    he=n1+n2
    return he

res=getSum(5,3)
res=getSum(5)
print(f"结果是: {result}")

# 根据两个数求和和差和积
def getJiSuan(n1,n2):
    he=n1+n2
    cha=n1-n2
    ji=n1*n2
    return he,cha,ji #可以返回多个结果，用逗号隔开，以元组形式输出

res=getJiSuan(5,2)
print(res)
print(f"和是: {res[0]}")
```

### 不定长参数

加了星号 \* 的参数会以元组 (tuple) 的形式导入，存放所有未命名的变量参数。加了两个星号 \*\* 的参数会以字典的形式导入

```
def getNumTuple(n1,*n):
    print(n1)
    print(n)
getNumTuple(8,9,90,100,"hello",True,3.5)

def getNumDict(n1,**n):
    print(n1)
    print(n)

getNumDict(8,age=40,name="周杰伦")
```

## 值传递与引用传递

值传递：传递的是数值，适用于实参类型为不可变类型 (int,float,bool,str,tuple)

引用传递：传递的是地址，适用于实参类型为可变类型 (list,set,dict,class)

## main函数

一个规范的代码项目结构如下：

```
# 全局变量
n=1

# 函数
def fun1():
    global n #函数中修改全局变量需要使用global声明
    n += 1
    print(n)

def main():
    print("欢迎")
    fun1()

# 入口
if __name__=="__main__":
    main()
```

## 作业练习

# 1.写函数 , 输入一个数字, 传递到函数中, 输出从 1 到此数

```
num=int(input("输入一个数字: "))
```

```
def func1(n):
```

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

```
        print(i)
```

```
func1(num)
```

# 2.写函数 , 接收 n 个数字 , 返回这些数字的和、最小值、最大值

```
num=input("输入任意个数字(以空格隔开):")
```

```
def func2(num):
```

```
    lst = [float(i) for i in str1.split()]
```

```
    return sum(lst),min(lst),max(lst)
```

```
print(*func2(num))
```

# 3.写函数, 接收一个不定长参数, 为多个字符串, 判断每个字符串的长度是否为大于 5 , 如:

isLen5("hello","world","test") 返回 [False,True,False] 注: 使用推导式

```
def isLen5(*s):
```

```
    return [len(s)>5 for i in s]
```

```
print(isLen5("hello","world","test"))
```

# 4.写函数, 传入两个列表参数 , 返回一个扑克牌列表 , 里面有52项, 每一项都是一个 tuple

# 结果输出如 : [('红心', 'A'), ('梅花', 'A')... ('黑桃', 'k')]

# 注: 使用推导式

```
list1=["红心","梅花","方块","黑桃"]
```

```
list2=["A","2","3","4","5","6","7","8","9","10","j","q","k"]
```

```
def func4(l1,l2):
```

```
    return [(j,i) for i in l2 for j in l1]
```

```
print(func4(list1,list2))
```

# 5.编写printstar(x)函数, (x是大于1的奇数数字), 忽略边框

```
def printstar(x):
```

```
    # 上半
```

```
    for i in range(x//2, 0, -1):
```

```
        print("* " * i + " " * (x-2*i) + "* " * i)
```

```
    # 中间
```

```
    print(' ' * x)
```

```
    # 下半
```

```
    for i in range(1, x//2 + 1):
```

```
        print("* " * i + " " * (x-2*i) + "* " * i)
```

```

x=input("请输入一个大于1的奇数数字:")
if x.isdigit() and int(x)>1 and int(x)%2==1 :
    x=int(x)
    printstar(x)
else:
    print("输入不合法! ")

```

# 扩展题：输入s1和s2两个字符串，用于判断s1是否是s2经过删除一些（也可以不删除）字符而不改变剩余字符相对位置形成的新字符串。

# 3. 输入s1和s2两个字符串，用于判断s1是否是s2经过删除一些（也可以不删除）字符而不改变剩余字符相对位置形成的新字符串。

```

str3=input("请输入两个字符串（以空格隔开）： ")
str1,str2=str3.split()

```

```

def sCheck(s1,s2):
    for i in s1:
        for j in s2:
            if i == j :
                if i == s1[-1]:
                    return True
                s2="".join(s2.split(j,1)[1:])
                break
        return False
print(sCheck(str1,str2))

```

# ATM机

```

data=[{'id':'1001','passwd':'123456','bat':'1000'},
      {'id':'1002','passwd':'123456','bat':'2000'},
      {'id':'1003','passwd':'123456','bat':'3000'},
      {'id':'1004','passwd':'123456','bat':'4000'}]
userdata={}

```

```

def CaoZuoYeMian():
    menu="""-----操作页面-----
1.存款
2.取款
3.查询余额
4.转账
5.退出
-----"""
    print(menu)
    while True:
        ctrl = input("请输入操作选项:")
        if ctrl == '1':

```

```

        CunKuan()
    elif ctrl == '2':
        QuKuan()
    elif ctrl == '3':
        ChaXun()
    elif ctrl == '4':
        ZhuanZhang()
    elif ctrl == '5':
        TuiChu()
    else:
        print("无效的操作选项，请重新输入！")

```

```

def CunKuan():
    global userdata
    money=input("请输入存款金额:")
    money=checkMoney(money)
    userdata["bat"]=str(float(userdata["bat"])+money)
    print(f"您已成功存款{money}，您当前的余额为: {userdata['bat']}")
    inputToReturn()

```

```

def QuKuan():
    global userdata
    money=input("请输入取款金额:")
    money=checkMoney(money)
    if float(userdata["bat"]) >= money:
        userdata["bat"]=str(float(userdata["bat"])-money)
        print(f"您已成功取款{money}，您当前的余额为: {userdata['bat']}")
        inputToReturn()
    else:
        print("您的余额不足！")
        inputToReturn()

```

```

def ChaXun():
    global userdata
    print(f"您当前的余额为: {userdata['bat']}")
    inputToReturn()

```

```

def ZhuanZhang():
    global userdata
    skzh=input("请输入收款账号:")
    if skzh==userdata['id']:
        print("收款账号不能是自己！")
        inputToReturn()
    for i in data:
        if i['id']==skzh:

```

```

        money=input("请输入转帐金额: ")
        money=checkMoney(money)
        if float(userdata["bat"]) > money:
            userdata["bat"]=str(float(userdata["bat"])-money)
            # i["bat"]=str(float(i["bat"])+money)
            print(f"您已成功转帐{money}, 您当前的余额为: {userdata['bat']}")
            inputToReturn()
        else:
            print("您的余额不足! ")
            inputToReturn()

print("收款账号不存在! ")
inputToReturn()

def TuiChu():
    global userdata
    esc=input("请问是否退出? y/n:")
    if esc.lower() == 'y' or esc.lower() == 'yes':
        # for i in data:
        #     if i['id']==userdata['id']:
        #         i=userdata
        #         break
        print("已成功退出, 请取回您的银行卡")
        exit()
    else:
        CaoZuoYeMian()

def checkMoney(m):
    try:
        m=float(m)
        if m <=0 :
            print("金额必须大于0!")
            inputToReturn()
        return m
    except ValueError:
        print("输入不合法! ")
        inputToReturn()

def inputToReturn():
    input("输入任意键返回主菜单..")
    CaoZuoYeMian()

def main():
    global userdata

```

```
while True:
    print("-----欢迎使用ATM机-----")
    userid=input("请输入账号:")
    pwd=input("请输入密码:")
    isLogin=False
    for i in data:
        if i['id']==userid and i['passwd']==pwd:
            isLogin=True
            userdata=i
            break

    if isLogin==True:
        print("登录成功, 请进行下列操作: ")
        CaoZuoYeMian()
    else:
        print("账号或密码错误! ")

if __name__ == "__main__":
    main()
```