

5.28总结：Python基础

VS操作

ctrl+/ 注释

tab 缩进

shift+tab 往回缩进

数据类型

数字型：bool int float

非数字型：str list tuple set dict

日期型：time datetime

type(n) 查看变量n的类型

```
name="hello"
age=10.5
xb=False #区分大小写
print(name,end="")
#小明学号为00001，年龄为18岁，身高为1.79米
sno=1
age=18
sname="小明"
high=1.786
print(type(name)) #查看name的数据类型

#print打印时，字符数据类型可以使用+与""连接
#其他类型则最好使用f"{变量}其他内容"
print(sname+"你好")
print(sname,"你好") #以空格分隔
print(f"{sname}学号为{sno},年龄为{age}岁,身高为{high}米")
print(f"{sname}学号为{sno:05d},年龄为{age}岁,身高为{high:.2f}米")
#05d: 数字宽度为5位，不足补零（`d`表示十进制整数）
#浮点数保留2位小数（`f`表示浮点数，`.2`控制小数位数）
```

输入

使用 input 函数来接收用户从键盘输入，input输入的内容是字符串型

```
pwd=input("请输入密码：")
print(type(pwd))
print(int(pwd)+5)
```

使用 int()、float()、str() 等对变量进行数据类型的转换

运算符

算数： + - * / %(取余) **(幂) //(取商)

比较： == != > >= < <=

赋值： = += -= *= /= **= //=

逻辑： and or not

流程控制

if 选择结构

```
sex="女"
if sex=="男":
    print("打游戏")
elif sex=="女":
    print("去逛街")
else:
    print("其他")
```

- 1、每个条件后面要使用冒号：，表示接下来是满足条件后要执行的语句块。
- 2、使用缩进来划分语句块，缩进采用 Tab 键或者 4 个空格，相同缩进组成一个语句块。
- 3、在 Python 中没有 case 语句。

while for 循环结构

```
# 输出1-10数字
i= 1
while i<=10:
    print(i)
    i+=1

# 字符串可以直接循环输出里面的字符
names="hello world*你好"
for i in names: #i逐个字符遍历
    print(i)
```

```
#range序列，从0开始，左闭右开
#range(1,10,2) 1开始数字，10结束数字（不包含），步长2
for i in range(1,10) #1,2,3...9
    print(i)
```

索引切片

字符串的索引截取，变量名 [头下标 : 尾下标]
索引值以 0 为开始值，-1 为从末尾的开始位置

从后面索引:	-6	-5	-4	-3	-2	-1
从前面索引:	0	1	2	3	4	5
	R	u	n	o	o	b

```
#索引切片
#abcdef
#012345
s="abcdef"
print(s[2]) #取下标为2的字符，即c

#做切片的时候，是左闭右开的，也就是不包括右边数字对应的下标索引
print(s[0:4]) #取下标0-4的字符（取不到4），即abcd
print(s[2:]) #取下标为2开始直到结束的字符，即cdef
print(s[0:5:2]) #步长为2（即中间隔一个），即ace
```

字符串相关操作

+拼接 *复制 ""保留格式

```
#len() 获取字符串长度
str="today"
l=len(str)

#str.find() 字符查找 ,找到返回索引，没找到返回-1
name = "LionelLionel"
name.find("i") #返回1
print(name.find("i")) #显示还需使用print
name.find("i",3) #开始索引，从下标3的字符开始找
name.find("i",1,2) #代表开始索引和结束索引
```

```

#str.rfind() 从右侧开始查找
name.rfind("i") #返回7

#str.isdigit()
#判断字符串str是否为数字
#全是数字返回True，存在其他字符返回False
sname="he54ll999"
for i in sname:
    if i.isdigit():
        print(i)

i=input("输入字符: ")
if i.isdigit():
    print(i+"是数字") #123
else:
    print(i+"不是数字") #hao123

#str.count() 统计字符串里某个字符或子字符串出现的次数
print(sname.count("l"))

```

list 列表

list（列表）是 Python 中使用最频繁的数据类型，专门用于存储一串数据，可以完成大多数集合类的数据结构实现

存储的数据称为**元素**，元素之间用逗号分隔

列表中元素的类型可以不相同，它支持数字，字符串甚至可以包含列表（所谓嵌套）

列表和字符串一样，索引从 0 开始，列表同样可以被索引和截取，列表被截取后返回一个包含所需元素的新列表

list 可以使用 + 操作符进行拼接

用星号 * 可以将列表中的元素解包为独立的元素，常用于函数传参或合并列表

创建空列表：变量名=[]

列表常用操作

```

list1=[1,2,3,4,5,6]
print(*list1) #输出列表的所有元素，默认以空格隔开
print(*list1,sep="\n") #可使用sep设置分割符

# 增加
list1.append(值) #在列表末尾追加一个元素
list1.extend([值1,值2]) #在列表末尾追加多个元素
list1.insert(下标,值) #将一个元素插入到列表指定位置

```

```

# 删除
list1.remove(值)  #从列表中删除第一次出现在列表中的值
list1.pop(下标)  #删除索引对应的元素，如果不加索引，默认删除最后元素，同时返回删除后列表的最后一个元素
del list1[下标1:下标2:步长]  #按照切片指定索引删除列表元素
list1.clear()  #清空列表

# 修改
list1[下标]=值  #修改指定索引的元素，元素不存在会报错

# 查询
list1[下标]  #根据索引取值，索引不存在会报错
list1.count(值)  #返回列表中包含某个值的个数
list1.sort()  #将列表中的元素进行排序（升序）
list1.sort(reverse=True) #降序排序
list1.reverse()  #列表反转排序

len(list1)  #返回列表长度（元素个数）
max(list1)  #返回列表元素最大值
min(list1)  #返回列表元素最小值
sum(list1)  #返回列表元素的总和

# 索引和元素同时显示
for i in range(len(list1)):
    print(i,list1[i])

# 判断某个元素是否在列表中
if 100 in list1:
    print("在")
else:
    print("不在")

# 列表嵌套列表（多维数组）
list1=[45,60,5,8,"hello",False,4.5,[365,648,"hello"]]
print(list1[-1][-1][0])  #取出h

```

字符串与列表转换

split 分割字符串为列表

```

str1="hello world test"
# 将字符串以空格分割存入列表
list1=str1.split(" ")  #在""内设置分割符

```

join 拼接列表为字符串

```
list1=["hello","world","test"]
# 将列表元素以_拼接
str2="_".join(list1)  #在"_"内设置拼接符
```

作业练习

```
# 上机练习1
# 1.定一个浮点型，将其转成int类型，观察取整后截断情况
n=int(3.14)
print(n)

# 2.定义一个整型，将其转成float，再计算其平方
n=float(3)
print(n**2)

# 3.不换行输出两个字符串（print方法的end参数）我喜欢 编程

print("我喜欢", end="编程")

# 4.输入一个数字，判断是奇数还是偶数
n=input("请输入一个数字")
if n.isdigit():
    if int(n)%2==0 :
        print(f"{n}是偶数")
    else:
        print(f"{n}是奇数")
else:
    print("输入的不是数字！")

# 5.输出8到88的偶数
for i in range(8,89,2):
    print(i)

# 6.已知字符串，s="哈佛在等我呢"
s="哈佛在等我呢"
# 获取第4至第5个字符
print(s[3:5])
# 获取左起第3个起所有的字符
print(s[2:])
# 字符串做*运算，拷贝字符串5次
print(s*5)
```

7.从键盘接收一个字符串，查找它在 `s="aforsadvj happy so thank"` 中是否存在，如果存在就打印 " 找到你了 "，并且显示出现的次数，不存在就打印 " 怎么没有你 "

```
s="aforsadvj happy so thank"
fs=input("请输入要查找的字符串：")
count=s.count(fs)
if count>0 :
    print("找到你了")
    print(f"出现次数：{count}")
else:
    print("怎么没有你")
```

8.从键盘接收一个字符串，取出字符串中的数字，并且统计字符串的总长度及所有数字出现的个数。

```
s=input("请输入字符串：")
cd=0
for i in s:
    if i.isdigit():
        cd+=1
print(f"总长度：{len(s)}")
print(f"所有数字数量：{cd}")
```

9.根据完整路径从路径中取出文件路径，文件扩展名py

```
str2="D:\ 软件 \python\python39\Tools\scripts\mitype.py"
n1=str2.rfind("\\")
n2=str2.rfind(".")+1
print(str2[0:n1])
print(str2[n2:])
# 输出结果为：
# D:\ 软件 \python\python39\Tools\scripts
# py
```

上机练习2

1.请将列表 `l1=["hello","world","test"]` 中的每个元素通过 "_" 链接起来。 输出结果：

```
hello_world_test
l1=["hello","world","test"]
str1="_".join(l1)
print(str1)
```

2.有如下两个列表

```
list1 = ["M", "na", "i", "Ke"]
list2 = ["y", "me", "s", "lly"]
#处理后保存到新列表中
list3 = []
for i in range(len(list1)):
    s=list1[i]+list2[i]
    list3.append(s)
print(list3)
```

```
#输出结果: ['My', 'name', 'is', 'Kelly']
```

```
# 3.生成 100 个卡号 , 存储到列表中打印输出; 卡号以 6102009 开头 , 后面 3 位依次是 (001,002,003,...,100 )
```

```
list1=[]
for i in range(1,101):
    n=f"6102009{i:03d}"
    list1.append(n)
print(*list1,sep="\n")
```

```
#直接赋值写法
```

```
list1 = [
    f"6102009{i:03d}" #:03d表示3位整数, 不足补0
    for i in range(1,101)
]
print(*list1,sep="\n")
```

```
# 4.已知字符串
```

```
str1="hello c sharp"
#将其中的单词以反序输出
list1=str1.split(" ")
list1.reverse()
print(*list1)
# str2=" ".join(list1)
# print(str2)
#输出结果: sharp c hello
```

```
# 5.给定一个列表,将每一位数字变成它的平方
```

```
alist = [1,2,3,4,5,6,7]
for i in range(len(alist)):
    alist[i]**=2
print(alist)
#输出结果: [1, 4, 9, 16, 25, 36, 49]
```

```
# 6.重新使用split和join的方法做1.9题:根据完整路径从路径中取出文件路径, 文件扩展名py
```

```
str2="D:\ 软件 \python\python39\Tools\scripts\mitype.py"
list2=str2.split("\\")
print(list2)
list3=list2[-1].split(".")
list2.pop()
str3="\\".join(list2)
print(str3)
print(list3[-1])
# 输出结果为:
# D:\ 软件 \python\python39\Tools\scripts
# py
```



```
# 7.把下面的列表里面的数字提取出来生成一个新的列表
list1 = [13, 2, 55, [11, 32, "eee"], ["ssd", 55, "ggg"]]
list2 = []
def func(l):
    for i in l:
        if type(i) == int:
            list2.append(i)
        elif type(i) == list:
            func(i)

func(list1)
print(list2)
# 输出: [13, 2, 55, 11, 32, 55]
```