1. **Python的输入和输出**

**第一部分**

**一、Python的输入**

Python提供了input() 内置函数从标准键盘输入读入一行文本。input 可以接收一个Python表达式作为输入，并将运算结果返回。

str="please input your text:"

print(int(input(str)))

其中的int()方法是把str转化为整数。

**二、Python的输出**

2.1、%

#%s表示输出一个字符串

name="zhengfeng chen"

print("My name is %s" %(name))

2.2 %d表示输出一个整数

age=23

print("My age is %d" %(age))

2.3 %f表示输出一个浮点数,默认小数点6位，多出的数据四舍五入

price=23.5623425252

print("the price of the cup is %f" %(price))

2.4 %.3f表示输出一个浮点数,小数点有三位，多出的数据四舍五入

price=23.5627425252

print("the price of the cup is %.3f" %(price))

2.5 指定占位符宽度,右对齐

print("My age is %8d,My name is %20s" %(age,name))

age=2

name="czf"

print("My age is %8d,My name is %20s" %(age,name))

2.6指定占位符宽度,左对齐

name="zhengfeng chen"

print("My age is %-8d,My name is %-20s" %(age,name))

age=2

name="czf"

print("My age is %-8d,My name is %-20s" %(age,name))

输出的结果如下：

My name is zhengfeng chen

My age is 23

the price of the cup is 23.562343

the price of the cup is 23.563

My age is 23,My name is zhengfeng chen

My age is 2,My name is czf

My age is 2 ,My name is zhengfeng chen

My age is 2 ,My name is czf

**三、format**

3.1、{}

name="zhengfeng chen"

print("My name is {}".format(name))

print('{},{},{}'.format('zhangk','boy',32))

age=23

print("My age is {0}".format(age))

My name is zhengfeng chen

zhangk,boy,32

My age is 23

3.2、0和1表示占位符的位置

print('{0},{1}'.format('zhangk', 32))

zhangk,32

3.3、关键字参数

print('{name},{sex},{age}'.format(age=32,name='zhangk',sex='male'))

zhangk,male,32

3.4、{0:f}中的f表示输出一个浮点数,默认小数点6位，多出的数据四舍五入，0表示第一位占位符

price=23.5623425252

print("the price of the cup is {0:f}".format(price))

the price of the cup is 23.562343

3.5 {0:.3f}表示输出一个浮点数,小数点有三位，多出的数据四舍五入

price=23.5627425252

print("the price of the cup is {0:.3f}".format(price))

the price of the cup is 23.563

3.6 {0:8}中的8指定占位符宽度,左对齐

print("My age is {0:8},My name is {1:20}".format(age,name))

age=2

name="czf"

print("My age is {0:8},My name is {1:20}".format(age,name))

3.7 ^、<、>分别是居中、左对齐、右对齐，后面带宽度

print("My age is {0:^8},My name is {1:^20}".format(age,name))

print("My age is {0:<8},My name is {1:<20}".format(age,name))

print("My age is {0:>8},My name is {1:>20}".format(age,name))

My age is 23,My name is zhengfeng chen

My age is 2,My name is czf

My age is 2 ,My name is czf

My age is 2 ,My name is czf

My age is 2,My name is czf

3.8 主要就是进制了，b、d、o、x分别是二进制、十进制、八进制、十六进制

print('{:b}'.format(27))

print('{:d}'.format(27))

print('{:o}'.format(27))

print('{:x}'.format(27))

11011

27

33

1b

3.9用逗号还能用来做金额的千位分隔符

print('${:,}'.format(123456789))

输出结果：

$123,456,789

如果你希望输出的形式更加多样，可以使用 str.format() 函数来格式化输出值。因为 str.format() 比较新的函数，大多数的 Python 代码仍然使用 % 操作符。但是因为这种旧式的格式化最终会从该语言中移除, 应该更多的使用 str.format()。

**第二部分**

**四、值转化成转成字符串，**

有两种方案：repr() 或 str() 函数

4.1 repr()或 tr() 函数的区别

 str()： 函数返回一个用户易读的表达形式。

 repr()： 产生一个解释器易读的表达形式。

a="hello,Python!"

#打印一个字符串，给用户看的，所以不带引号

print(str(a))

#repr()打印出来的值是给python理解的，这里python理解为字符串，所以带着引号

print(repr(a))

b=1000L

#用户看到的当然是一个数字1000，而不是字符串1000L

print(str(b))

#repr()多了一个L,表示它是长整形，便于计算机处理

print(repr(b))

执行结果：

hello,Python!

'hello,Python!'

1000

1000L

4.2 repr()函数不转义字符串中的特殊字符，str()转义

a="hello,Python\n Hello!"

print(str(a))

print("#########")

print(repr(a))

执行结果：

hello,Python

Hello!

#########

'hello,Python\n Hello!'

这些特殊符号包括\n,\t,\r等

4.3 repr() 的参数可以是 Python 的任何对象

**五、字符的常见对齐方法**

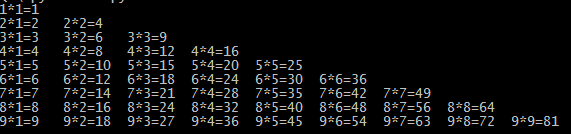
例子：打印九九乘法表

for row in range(1,10):

for column in range(1,row+1):

print(repr(row).ljust(1)+"\*"+repr(column).ljust(1)+"="+repr(row\*column).ljust(3),end=" ")

print()



其中的ljust()是左对齐，字符串对象的 rjust() 方法, 它可以将字符串靠右, 并在左边填充空格。

还有类似的方法, 如 ljust() 和 center()。