**Python常用函数总结**

**一、输入输出函数**

**1、print()函数 输出函数**

print("hello world")

print()函数可以同时输出多个字符串，用逗号“,”隔开

print("hello","how","are","you")

python中print之后是默认换行的，要实现不换行要加end参数表明

print("n =",n,end=' ')

**2、input()函数**

输入函数

Python提供了一个input()函数，可以让用户输入字符串，并且存放在变量中，比如输入用户名

name = input()

**二、进制转换函数**

**1、bin(),oct(),hex()进制转换函数**

使用bin(),oct(),hex()进行转换的时候的返回值均为字符串,且带有0b, 0o, 0x前缀.

十进制转换为二进制

bin(10) '0b1010'

十进制转为八进制

oct(12) '014'

十进制转为十六进制

hex(12) '0xc'

**2、’{0:b/o/x}’.format()进制转换函数（不带前缀）**

十进制转换为二进制

'{0:b}'.format(10)'1010'

十进制转为八进制

'{0:o}'.format(12)'14'

十进制转为十六进制

'{0:x}'.format(12)'c'

注意:hex函数比格式化字符串函数format慢,不推荐使用.

**3、int(’’,2/8/16)转化为十进制函数（不带前缀）**

二进制转为十进制

int('1010',2)10

八进制转为十进制

int('014', 8)12

十六进制转十进制

int('0xc',16)12

**4、’{0:d}’.format()进制转换为十进制函数**

二进制转十进制

'{0:d}'.format(0b11)'3'

八进制转十进制

'{0:d}'.format(0o14)'12'

十六进制转十进制

'{0:d}'.format(0x1f)'31'

**5、eval()进制转为十进制函数**

二进制转十进制

eval('0b11')'3'

八进制转十进制

eval('0o14')'12'

十六进制转十进制

eval('0x1f')'31'

注意:eval函数比int函数慢,不推荐使用

**三、求数据类型函数**

**1、type()**

n = "hello world"

n = type(n)

print(n)

输出：<class 'str'>

**2、isinstance()**

功能：判断变量是否属于某一数据类型，可以判断子类是否属于父类。

**四、删除变量/对象函数**

**1、del() 函数**

变量一旦删除，就不能引用，否则会报错

**五、数学函数**

**1、abs(num) 返回num的绝对值**

print(abs(-3))

输出：3

**2、max(num1,num2,…,numn) 返回给定参数的最大值**

**num1 = 10**

num2 = 20

print(num1 > num2)

print(max(num1,num2,56))

输出：False 56

**3、min(num1,num2,…,numn) :返回给定参数的最小值**

print(min(12,3,34,0))

输出：0

**4、pow(x,y) : 求x的y次方，x^y**

print(pow(2,3))

输出：8

**5、round(num,n) : 四舍五入，**

​ 参数一：需要进行四舍五入的数据；  
参数二：保留小数的位数。若n不写，默认为0

print(round(123.486,2))

输出：123.49

**6、range()函数**

range([start,] stop [,step])  
实质：创建了一个可迭代对象；一般情况下与for循环一起连用  
1、start 可以不写，默认值是0，若给定则从start开始  
2、stop 必须给定；  
3、取值范围[start,stop)  
4、step：步长，若不给则默认为1

**六、字符串函数**

**1、eval(str)函数**

功能：将字符串转成有效的表达式来求值或者计算结果

可以将字符串转化成列表list，元组tuple，字典dict，集合set

注意：生成了一个新的字符串，没有改变原本的字符串

**2、len(str)函数**

功能：获取字符串长度

str1 = "you are good man"

print(len(str1))

输出：16

**3、str.lower()函数**

功能：返回一个字符串中大写字母转化成小写字母的字符串

注意：生成了一个新的字符串，没有改变原本的字符串

str1 = "You are good Man"

print(str1.lower())

print(str1)

输出：

you are good man

You are good Man

**4、str.upper()函数**

功能：返回一个字符串中小写字母转化成大写字母的字符串

注意：生成了一个新的字符串，没有改变原本的字符串

str1 = "You are good man"print(str1.upper())print(str1)

输出：

YOU ARE GOOD MAN

You are good man

**5、str.swapcase()函数**

功能：返回字符串中的大写字母转小写，小写字母转大写的字符串

注意：生成了一个新的字符串，没有改变原本的字符串

str1 = "You are Good man"

print(str1.swapcase())

print(str1)

**6、str.capitalize()函数**

功能：返回字符串中的首字母大写，其余小写的字符串

注意：生成了一个新的字符串，没有改变原本的字符串

tr1 = "you Are good man"

print(str1.capitalize())

print(str1)

str2 = "You are a good Man"

print(str2.capitalize())

输出：

You are good man

you Are good man

**7、str.title()函数**

功能：返回一个每个单词首字母都大写的字符串

注意：生成了一个新的字符串，没有改变原本的字符串

str1 = "you Are good man

print(str1.title())

print(str1)

str2 = "You are a good Man"

print(str2.title())

输出：

You Are Good Man

you Are good man

You Are A Good Man

**8、str.center(width[,fillchar])函数**

功能：返回一个指定宽度的居中字符串

参数一：指定的参数【必须有】

参数二：fillchar填充的字符，若未指定，则默认使用空格

注意：生成了一个新的字符串，没有改变原本的字符串

str1 = "you Are good man"

print(str1.center(20,"\*"))

print(str1)

输出：

you Are good man \*\*you Are good man\*\*

you Are good man

**9、str.ljust(width[,fillchar])函数**

功能：返回一个指定宽度左对齐的字符串

参数一：指定字符串的宽度【必须有】

参数二：填充的字符，若不写则默认为空格

注意：生成了一个新的字符串，没有改变原本的字符串

str1 = "you Are good man"print(str1.ljust(20,"\*"))print(str1)

输出：

you Are good man\*\*\*\*

you Are good man

**10、str.rjust(width[,fillchar])函数**

功能：返回一个指定宽度右对齐的字符串

参数一：指定字符串的宽度【必须有】

参数二：填充的字符，若不写则默认为空格

注意：生成了一个新的字符串，没有改变原本的字符串

str1 = "you Are good man"

print(str1.rjust(20,"\*"))

print(str1)

输出：\*\*\*\*you Are good man

you Are good man

**11、str.zfill(width)函数**

功能：返回一个长度为width的字符串，原字符右对齐，前面补0

注意：生成了一个新的字符串，没有改变原本的字符串

str1 = "you Are good man"

print(str1.zfill(20))

print(str1)

输出：

0000you Are good man

**12、str2.count(str1,start,end])函数**

功能：返回str1在str2中出现的次数，可以指定一个范围，若不指定则默认查找整个字符串

区分大小写

注意：生成了一个新的字符串，没有改变原本的字符串

str1 = "hello"

str2 = "Hello hello1 Hello2 hi haha helloa Are good man"

print(str2.count(str1,0,20))

输出：1

**13、str2.find(str1,start,end)函数**

功能：从左往右检测str2，返回str1第一次出现在str2中的下标

若找不到则返回-1，可以指定查询的范围，若不指定则默认查询整个字符串

注意：生成了一个新的字符串，没有改变原本的字符串

str1 = "hello"

str2 = "Hello hello1 Hello2 hi haha helloa Are good man"print(str2.find(str1,5,20))

输出：6

**14、str2.rfind(str1,start,end)函数**

功能：从右往左检测str2，返回str1第一次出现在str2中的小标，若找不到则返回-1，可以指定查询的范围，若不指定则默认查询整个字符串

注意：生成了一个新的字符串，没有改变原本的字符串

str1 = "hello"

str2 = "Hello hello1 Hello2 hi haha helloa Are good man"

print(str2.rfind(str1,10,35))

输出;28

**15、str2.index(str1,start,end)函数**

功能：和find()一样，不同的是若找不到str1，则会报异常

注意：生成了一个新的字符串，没有改变原本的字符串

str1 = "hello"

str2 = "Hello hello1 Hello2 hi haha helloa Are good man"

print(str2.index(str1,2,25))

print(str2.index(str1,24,25))

输出：6

**16、str.lstrip(char)函数**

功能：返回一个截掉字符串左侧指定的字符，若不给参数则默认截掉空字符: \n \r \t 空格

注意：生成了一个新的字符串，没有改变原本的字符串

str3 = " \n\r \t ni hao ma"

print(str3)

print(str3.lstrip())

str4 = "\*\*\*\*ni hao ma\*\*\*\*"

print(str4.lstrip('\*'))

输出;

ni hao ma

ni hao ma

ni hao ma\*\*\*\*

**17、str.rstrip()函数**

功能：返回一个截掉字符串右侧指定的字符，若不给参数则默认截掉空字符: \n \r \t 空格

注意：生成了一个新的字符串，没有改变原本的字符串

str3 = " ni hao ma \n\r \t"print(str3.rstrip())

str4 = "\*\*\*\*ni hao ma\*\*\*\*"print(str4.rstrip('\*'))

输出：

ni hao ma\*\*\*\*ni hao ma

**18、str2.split(str1,num) 分离字符串**

功能：返回一个列表，列表的元素是以str1作为分隔符对str2进行切片，  
若num有指定值，则切num次，列表元素个数为num+1  
若不指定则全部进行切片  
若str1不指定，则默认为空字符（空格、换行\n、回车\r、制表\t）

注意：生成了一个新的字符串，没有改变原本的字符串

str2 = "22hello nihao hi hello haha ello2 hello3 hello"

print(str2.split(' ',3))

str3 = ["1257309054@qq.com"](mailto:\"1257309054@qq.com\")

print(str3.split('@'))

list1 = str3.split('@')

print(list1[1].split('.'))

输出：['22hello', 'nihao', 'hi', 'hello haha ello2 hello3 hello']['1257309054', 'qq.com']['qq', 'com']

**19、str2.splitlines()**

功能:返回一个列表，列表的元素是以换行为分隔符，对str2进行切片

注意：生成了一个新的字符串，没有改变原本的字符串

**20、str1.join(seq)函数 字符串连接**

功能：以指定字符串作为分隔符，将seq中的所有元素合并成为一个新的字符串

**21、ord() 求字符的ASCLL码值函数**

print(ord("a"))

输出：97

**22、chr() 数字转为对应的ASCLL码函数**

print(chr(97))

输出：a

**23、 max(str) min(str)获取最大最小字符**

功能: 返回字符串str中最大的字母

str1 = "how are you , i am fine thank you"

print(max(str1))

输出：

y

min(str)

功能：返回字符串str中最小字母

str1 = "how are you , i am fine thank you"

print(min(str1))

输出：' '

**24、str.replace(old , new [, count]) 字符串的替换**

str.replace(old , new [, count])

功能：使用新字符串替换旧字符串，若不指定count，则默认全部替换，  
若指定count，则替换前count个

str1 = "you are a good man"

print(str1.replace("good","nice"))

输出：

you are a nice man

**25、str.startswith(str1,start.end) 判断字符串的开头**

str.startswith(str1,start.end)  
功能：判断在指定的范围内字符串str是否以str1开头，若是就返回True，否则返回False  
若不指定start，则start默认从开始，  
若不指定end，则默认到字符串结尾

str1 = "hello man"

print(str1.startswith("h",0,6))

输出：True

**26、str.endswith(str1,start.end) 判断字符串的结尾**

功能：判断在指定的范围内字符串str是否以str结束，若是就返回True，否则返回False  
若不指定start，则start默认从开始，  
若不指定end，则默认到字符串结尾

str1 = "hello man"print(str1.endswith("man"))

输出：True

**27、str.encode(编码格式)**

对字符串进行编码 默认是utf-8  
编码：str.encode()  
解码：str.encode().decode()  
注意：encode()的编码格式与decode()的编码格式必须保持一致

**28、str1.isalpha() 字符串为字母**

功能：判断字符串【至少含有一个字符】中的所有的字符是否都是字母【a~z A~Z 汉字】  
若符合条件则返回True，否则返回False

str5 = "hello你二"print(str5.isalpha())

str5 = "hello "print(str5.isalpha())

输出：TrueFalse

**29、str5.isalnum()**

功能：判断字符串【至少含有一个字符】中的所有字符都是字母或者数字【0-9，a-z，中文】

**30、str.isupper()**

功能：判断字符串中所有字符是不是大写字符

print("WWW".isupper())

print("wWW".isupper())

print("123".isupper())

print("一二三".isupper())

输出;TrueFalseFalseFalse

**31、str.islower()**

功能：判断字符串中所有字符是不是小写字符

print("WWW".islower())

print("wWW".islower())

print("123".islower())

print("一二三".islower())

print("qwww".islower())

输出：FalseFalseFalseFalseTrue

**32、str.istitle()**

功能：判断字符串是否是标题化字符串【每个首字母大写】

print("U Wss".istitle())

print("wWW ".istitle())

print("123 ".istitle())

print("一二三".istitle())

print("qwww".istitle())

输出：TrueFalseFalseFalseFalse

**33、 str.isdigit()**

True: Unicode数字，byte数字（单字节），全角数字（双字节）  
False: 汉字数字， ，罗马数字

**34、str.isspace()**

功能：判断字符串中是否只含有空格

print("ddd".isspace())

print("".isspace())

print("a ddd".isspace())

print(" aaa".isspace())

print(" ".isspace())

输出;FalseFalseFalseFalseTrue

**35、str.isnumeric()**

功能：若字符串中只包含数字字符，则返回True，否则返回False

True: Unicode数字，全角数字（双字节），汉字数字

False: 罗马数字，

Error: byte数字（单字节）

**36、str.isdecimal()**

功能：检查字符串是否只包含十进制字符【0，9】，如果是返回True，否则返回False

True: Unicode数字，，全角数字（双字节）

False: 罗马数字，汉字数字

Error: byte数字（单字节）

print("123".isdecimal())

print("123z".isdecimal())

结果TrueFalse

**八、list列表函数**

**1、list.append(元素)**

功能：在列表末尾添加新的元素，只要是python中的数据类型都可以添加，如列表，元组、字典等

list1 = [1,2,3,4]

list2 = ["good","nice","beautiful"]

list1.append("hello")print(list1)

list1.append(list2)print(list1)

输出：[1, 2, 3, 4, 'hello', ['good', 'nice', 'beautiful']]

**2、list1.extend()**

功能：在列表的末尾一次性追加另一个列表中的多个值

注意：extend()中的值，只能是列表、元组、字符串、字典（可迭代的对象）

list1 = [1,2,3,4]

list2 = ["good","nice","beautiful"]

list1.extend(list2)print(list1)

list1.extend({'h':2,'e':3})print(list1)

输出：[1, 2, 3, 4, 'good', 'nice', 'beautiful'][1, 2, 3, 4, 'good', 'nice', 'beautiful', 'h', 'e']

什么时候使用append，什么时候用extend?

当我们需要在原本的列表中追加像number类型后者Boolean类型的时候，

可以使用append，

或者是我们需要把另外一个列表当成一个元素追加到原本的列表中去的时候，

这时候也可以使用append

当我们需要在原本的列表中插入一个新的列表中的所有的元素的时候，

这时候我们需要使用extend

**3、str.insert(下标值，object)**

功能：在下标处插入元素，不覆盖原本的数据，原数据向后顺延  
它与append非常类似，不同之处：append默认把新的元素添加在列表的末尾  
而insert可以指定位置进行添加【插入】

list1 = [1,2,3,4]

list2 = ["good","nice","beautiful"]

list1.insert(0,list2)

print(list1)

输出：[['good', 'nice', 'beautiful'], 1, 2, 3, 4]

**4、list.pop()**

功能：移除列表最后一个元素，并且返回移除元素的值  
list.pop(index)  
index:下标值  
功能：移除指定下标处的元素，并且返回移除元素的值  
注意：pop一次，list元素个数减1，index的取值为[0,len(list))，若超出取值范围则会报错  
IndexError: pop index out of range

list2 = ["good","nice","beautiful"]

list2.pop()print(list2)

list1 = [1,2,3,4]

list2.pop(0)print(list2)

list2.extend(list1)print(list2)

输出：['good', 'nice']['nice']['nice', 1, 2, 3, 4]

**5、list.remove(元素)**

功能：移除列表中指定元素的第一个匹配成功的结果  
没有返回值

list2 = [2,"good","nice","beautiful",2]

list2.remove(2)

print(list2.remove(2))

print(list2)

输出:['good', 'nice', 'beautiful']

**6、list.clear()**

功能：清除列表中的所有元素，但不删除列表，没有返回值

list2 = [2,"good","nice","beautiful",2]

list2.clear()print(list2.clear())print(list2)

输出：None[]

**7、list.index(元素,start,end)**

功能：返回从指定的范围内[start,end)的列表中查找到第一个与元素匹配的元素的下标  
若不指定范围，则默认为整个列表。  
注意：若在列表中查不到指定的元素，则会报错

ValueError: 4 is not in list

list1 = ['h','e','l','l','o']print(list1.index('e'))

输出：1

**8、list.count(元素)**

功能：返回元素在列表中出现的次数

list1 = [1,2,3,4,5,1]

print(list1.count(1))

输出：2

**9、len(list)**

功能：返回列表元素的个数

list1 = [1,2,3,4,5,[1,3,4]]

print(len(list1))

输出：6

**10、max(list)、min(list)**

功能;返回列表中的最大值，若是字符串比较ASCII码  
注意：数据类型不同的不能进行比较  
min(list)  
功能: 返回列表中的最小值，若是字符串则比较ASCII码

list1 = [1,2,3,4,5]

print(max(list1))

print(min(list1))

list2 = ['hello','he','hl','hl','ho']

print(max(list2))

print(min(list2))

输出：51

ho

he

**11、list.reverse()**

功能：列表倒叙  
注意：操作的是原本的列表

list3 = [2,1.22,3,6,33]

list4 = ['hello','nihao','how are you']

list3.reverse()

list4.reverse()

print(list3)

print(list4)

输出：[33, 6, 3, 1.22, 2]['how are you', 'nihao', 'hello']

**12、list.sort()**

功能：列表排序，默认升序  
注意：操作的是原本的列表

list4 = ['hello','nihao','how are you']

list4.sort()

print(list4)

list4 = [2,3,1,2,4]

list4.sort()

print(list4)

输出：['hello', 'how are you', 'nihao'][1, 2, 2, 3, 4]

降序：list.sort(reverse=True)

list1 = [1,3,2,4,5,6]

list1.sort(reverse=True)

print(list1)

输出：[6, 5, 4, 3, 2, 1]

**13、浅拷贝、深拷贝**

list1 = list2  
注意：浅拷贝是引用拷贝，类似于快捷方式  
深拷贝【内存拷贝】  
list3 = list1.copy()  
注意：重新开辟了一个新的内存空间，存储的数据于list1相同

list1 = ['hello','nihao','how are you']

list2 = list1print(id(list1))print(id(list2))

list3 = list1.copy()

print(id(list3))

输出;167042630400816704263040081670426233032

**14、list(元组)**

功能：将元组转为列表。

list1 = list((1,2,3))

print(list1)

输出:[1, 2, 3]

**九、dict字典函数**

**1、value= 字典名.get(key)**

获取字典中关键字对应的值，如果key不存在，返回None

dict1 = {"key1":1,"key2":2}

print(dict1.get("key1"))

输出：1

**2、dict.pop(key) 删除元素**

通过key删除元素，返回被删除元素的值，pop一次，dict长度减1

dict1 = {"key1":1,"key2":2}

print(dict1.pop("key1"))

print(dict1.pop("key2"))

print(dict1)

输出:12{}

**十、set集合函数**

**1、set1.add(元素) 添加元素**

可以添加重复的元素但是没有效果  
set1 = {'key1','key2'}

set1.add('key1')

set1.add('key3')

print(set1)

输出;{'key1', 'key2', 'key3'}

**2、set.update() 添加seq元素**

功能：插入整个list【一维】，字符串打碎插入

set1 = {'key1','key2'}

set1.update(['key3','key4'])

set1.update(('hello',True))

set1.update('key5')

print(set1)

输出;{'key3', True, '5', 'key1', 'y', 'key4', 'key2', 'k', 'hello', 'e'}

**3、set.remove(元素)删除元素**

功能：删除集合中的元素，若元素不存在则会报错，删除一次，set长度减1

set1 = {'key1','key2'}

set1.remove('key1')print(set1)

set1.remove('key3')

输出;{'key2'}

报错：KeyError: 'key3'

**十一、栈和队列**

**1、 栈 stack 先进先出**

mystack = []#压栈[向栈中存数据]

mystack.append(1)

print(mystack)

mystack.append(2)

print(mystack)

mystack.append(3)

print(mystack)

#出栈[从栈中取数据]

mystack.pop()

print(mystack)

mystack.pop()

print(mystack)

**2、 队列 queue**

特点: 先进先出

#导入数据结构的集合

import collections

queue = collections.deque([1, 2, 3, 4, 5])

print(queue)

#入队[存数据]

queue.append(8)

print(queue)

queue.append(9)

print(queue)

#取数据

print(queue.popleft())

print(queue)