原来是什么切片之后还是什么

# 字符串

1. msg = "My name is Alex , I'm 22 years old!" 这个就需要单双引号配合。
2. msg = """ 今天我想写⾸首⼩小诗，

歌颂我的同桌，

你看他那乌⿊黑的短发，

好像⼀一只炸⽑毛鸡。 """

想多⾏行行赋值⼀一个字符串串，就需要三引号。

1. 数字类型有 +-\*/ 字符串串有么？字符串串只有 + *\*(+是字符串拼接，\**多次输出)
2. 字符串复制给变量 这个变量可以指向任意的字符串（即变量的指向可以修改，但是字符串一旦定义好就不能修改 字符串的+操作很耗费内存）
3. 字符串占位符 a. print("%s你好" %"wdd") 只有一个占位符的时候可以不用括号括 起来

b. print("%s你好%s" %("wdd","wd"))

c.字符串中一但%作为占位符使用，所有的%都当做占位符使用（包括%2，这个会报错，必须用%%）

1. 新玩法

s="我叫{},我今年{}".format("wdd",18)

s=”我叫{1},我今年{0}”.format(“wdd”,18)

s=”我叫{name},我今年{age}”.format(name=“wdd”,age=18)

1. 索引 字符串索引从0开始，整个字符串可以看成首位相连的环，最后一个字符的索引可以用-1表示

只切最后两个字符s[-2:] 不能s[-2:0]这样的话0在左-2在右切不了

**总结s[start:end:step] index看成一个环 包括start不包括end start和end不管正负（但start和end必须同为正或同为负）**

**step为正则从左往右切start必须小于end才切出值来**

**step为负则从右往左切start必须大于end才切出值来**

1. **常用操作（因为字符串一旦定义就不可以修改，一般都是操作后返回一个新字符串）**

**tmp=s.capitalize()#首字母大写**

**tmp=s.lower()**

**tmp=s.upper()**

**tmp=s.swpcase()#大小写互换**

**tmp=s.casefold()#转换成小写，lower()的区别是lower()对某些字符支持不是太好，casefold支持所有**

**tmp=s.title()#s="asdfa d dfa你adsf " Asdfa D Dfa你Adsf#被特殊字符（空格或中文）隔开的首字母大写**

**tmp=s.center(8,”\*”)**

**tmp=s.strip()**#去掉字符串左右两段所有指定的部分（注意strip只去左右两段的内容，指定的内容重复也都去了）

**tmp=s.lstrip()**

**tmp=s.rstrip()**

**tmp=s.replace(“old”,”new”)**

list=s.split(“，”) #（如果你的刀贴着边切回产生空字符）

b=s.startswith(“&&&”)

b=s.endwith(“###”)

tmp=s.count(“a”)

index=s.find(“a”)##返回下标（以字符串出项的第一个位置为准，没有的话就返回-1）

s.isalnum()#字符串是否是数字字母组成

s.isdigit()#字符串是否是数字组成，不包括小数点

s.isalpha()#字符串是否是字母组成

s.isnumeric()#即使是汉子数字也认识，数字加汉子组合也认识（和isdigit的区别）

len(s)和s.\_len\_()一样是求长度的（len实际执行的是s.\_len\_()）

1. 切片查找

s.find(“a”,3)#从第3个位置开始找

s.index(“a”)#如果内容不存在直接报错（和find的区别）

1. tmp=str.join(t)#把字符串str插入到str2中（把str2拆成每个字符在每个字符中间添加str）

join可以把列表变成字符串，把字符串变成列表.split（）

-

# 列表

list和str是不一样的，lst可以发生改变，所以直接在原来的对象上进行操作

1. 增： a list.append()

b list.insert(index,”aaa”)

c list.extend(list\_tmp)

1. 删除： a tmp=list.pop() #删除最后一个并返回

b list.remove(“aa”)#删除不存在的元素会报错

c list.clear()

d del list[1:3]

1. 修改 a list[1]=”啊啊“

b 迭代修改 切片切除的个数要与给出的list的个数不一定要一致，如果给出的不是list是str的话会将str转成list，

c 切片修改 (前提赋值的是数组)如果切⽚片没有步⻓长或者步⻓长是1. 则不不⽤用关⼼心个数

1. c=list.count(“aa”)

list.sort(reverse=True)

list.reverse()

1. 列表的嵌套，将列表当成数组做

# 元祖

俗称不可变的列表，又被称为只读列表（就把列表的方括号换成圆括号）

元祖可以查询，可以循环，可以切片，但就是不能改

坑：tu=(108) 这是个int(当元祖的个数少于一个需要加，空元祖这样定义tu=tuple())

1. 查询 tu = (1, "太⽩白", "李李⽩白", "太⿊黑", "怎么⿊黑")

print(tu[0])

1. 切片 print（tu[2:5]）
2. 循环 for el in tu:

print(el)

1. 尝试修改元祖

# tu[1] = "⻢马⻁虎疼" # 报错 'tuple' object does not support item assignment

tu = (1, "哈哈", [], "呵呵")

# tu[2] = ["fdsaf"] # 这么改不不⾏行行

tu[2].append("麻花藤") # 可以改了了. 没报错

tu[2].append("王剑林林")

print(tu)

总结：这里元祖的不可变的意思是子元素不可变，而子元素内部的子元素是可以变得，这取决于子元素是否是可变对象

# 字典

1. key可以是数字，bool,字符串，元祖

key不可以是list,dict,set

1. dict[‘a’]=”a” #赋值操作 key可以是重复的，重复会替换原来的数据

ret=dict.setdefault(‘马融’，“aa”)#添加操作，key不可以重复（注意两者的区别:key不存在才会把key-value添加进去，key存在的话就不起效果，返回的是最终key对应的value值），会返回value（setdefault两个执行步骤：首先判断字典有没有key，如果没有执行新增，然后用这个key去字典查询，返回查到的结果）

1. ret=dic.pop(“aa”)#返回key对应的value
2. del dic[“joy”]
3. ret=popitem()#返回的是元祖，第一个是key,第二个是value
4. dic.clear()
5. dic.get(“aa”,”key=aa不存在就返回默认值”)##如果key不存在返回none
6. dic.keys(),dic.values(),dic.items()
7. s=dic.fromkeys("王健林","思聪")（原来不管什么字典都没有影响，产生新字典（深坑，考试））

# set

注意：set集合中的元素必须是可哈希的，但是set本身是不可hash的，set是可变的（set里面不能装set,可以使用s=frozenset()冻结set）

1. s.add(“c”)#重复的内容不会添加到set集合中去
2. s.update("麻花藤") # 迭代更更新
3. item=s.pop()#随机删除一个
4. s.remove(“aa”)#删除指定，不存在则会报错
5. item=s.pop()#随机删除一个
6. set集合中的数据没有索引，也没有办法定位一个元素，所以没有办法进行修改。

我们可以先删除指定元素在添加来完成修改操作