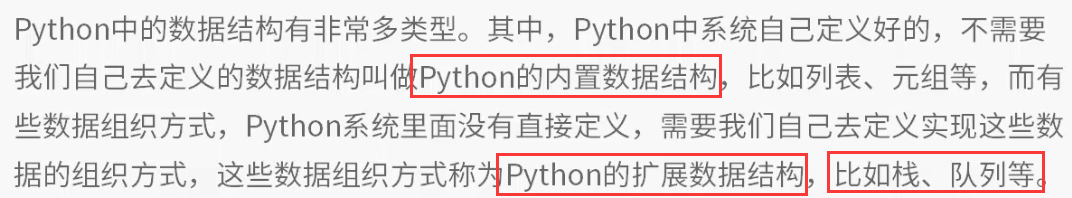
Python中的数据结构

# Python中的数据结构

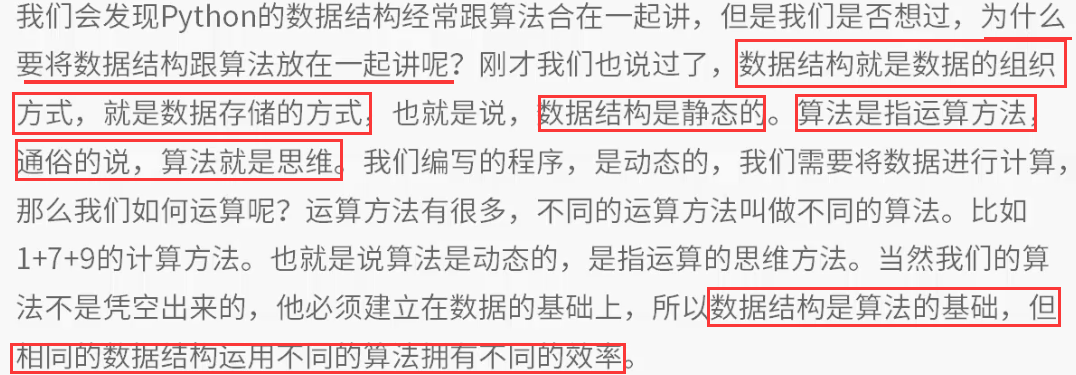
## Python中的内置与扩展数据结构



## 内置数据结构介绍



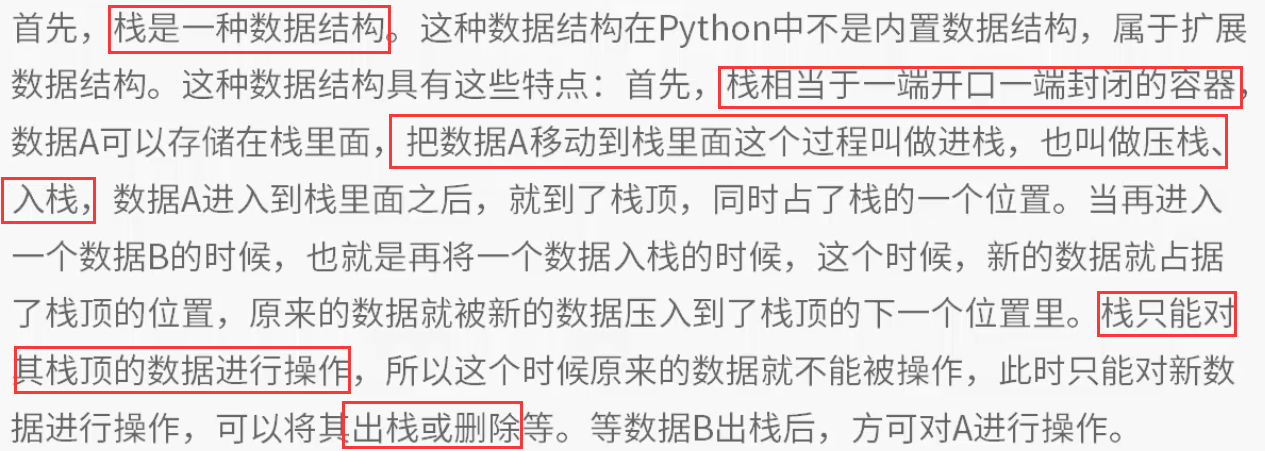
# 数据结构与算法的关系



# 数据结构-栈

## 栈的介绍

**最大的特点是：一端操作，FILO，先进后出**，**重要方法push和pop**。



## 图例



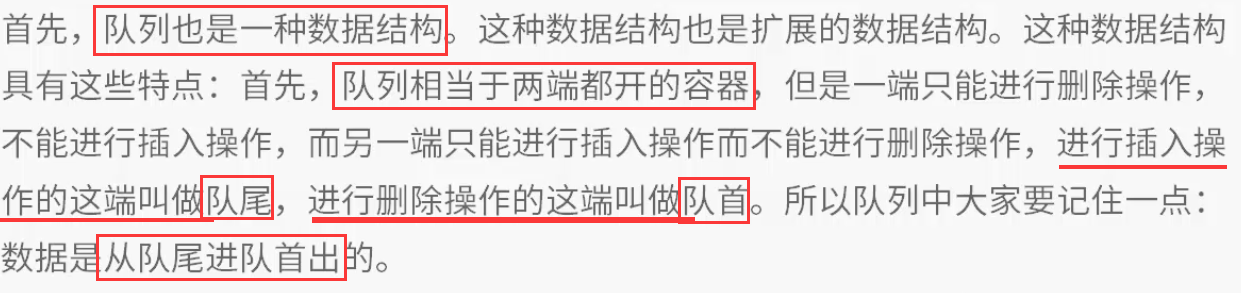
## Python代码对栈数据结构的实现

class Stack():  
 def \_\_init\_\_(self,size=10):  
 *'''  
 构造方法* ***:param*** *size: 栈容量，默认为10  
 '''* self.size = size  
 self.stack = []  
 self.position = -1  
 def push(self,value):  
 *'''  
 向栈中添加元素value* ***:param*** *value:* ***:return****: 返回True表示成功，若已满，则抛出异常  
 '''* if self.position == self.size - 1:  
 raise Exception('栈已满，无法添加元素')  
 else:  
 self.stack.append(value)  
 self.position += 1  
 return True  
 def length(self):  
 *'''  
 现有栈中元素的个数* ***:return****:  
 '''* return len(self.stack)  
  
 def pop(self):  
 *'''  
 返回处于栈顶元素的值，并删除该元素* ***:return****: 栈顶元素值  
 '''* if self.position == -1:  
 raise Exception('栈已空')  
 else:  
 self.position -= 1  
 return self.stack.pop(self.position + 1)  
  
 def isEmpty(self):  
 *'''  
 判断该栈是否为空* ***:return****:  
 '''* if self.position == -1:  
 return True  
 else:  
 return False  
 def delete(self):  
 *'''  
 删除处于栈顶的元素  
 '''* self.stack.pop(self.position)  
 self.position -= 1  
  
stack1 = Stack(4)  
stack1.push('小明')  
stack1.push('小红')  
stack1.push('小亮')  
stack1.push('小明')  
print(stack1.stack)#['小明', '小红', '小亮', '小明']  
stack1.delete()  
print(stack1.stack)#['小明', '小红', '小亮']  
print(stack1.length())#3  
print(stack1.pop())#小亮  
print(stack1.stack)#['小明', '小红']  
print(stack1.pop())#小红  
print(stack1.stack)#['小明']  
print(stack1.pop())#小明  
print(stack1.stack)#[]  
print(stack1.length())#0  
print(stack1.isEmpty())#True

# 数据结构-队列

## 队列：

**最大特点：两端操作，队尾进，队首出，先进先出FIFO。**



## 图例：



## Python代码对队列数据结构的实现

设计说明：

**队列需要有两个指针，一个对应队首，一个对应队尾；**

**起初，默认队首指针为0，队尾指针为-1，此时元素个数为0；**

**当添加一个元素时，队尾指针加1，队首指针不变；**

**队列中元素个数=队尾指针-队首指针+1;当队首指针等于队尾指针时，队列中元素个数为1；当队首指针比队尾指针大1时，队列元素个数为0.**

class Queue():  
 def \_\_init\_\_(self,size = 10):  
 self.size = size  
 self.queue = []  
 self.head = 0  
 self.tail = -1  
  
 def addLast(self,value):  
 if self.lenth() >= self.size:  
 raise('队列已满')  
 self.queue.append(value)  
 self.tail += 1  
 return True  
  
 def getValue(self):  
 if self.lenth() != 0:  
 self.head += 1  
 return self.queue[self.head-1]  
 else:  
 raise Exception('获取异常')  
  
 def lenth(self):  
 if self.tail < self.head-1:  
 raise Exception('队列指针异常')  
 elif self.tail - self.tail == -1:  
 return 0  
 else:  
 return self.tail - self.head + 1  
  
 def isEmpty(self):  
 if self.lenth() == 0:  
 return True  
 else:  
 return False  
  
queue1 = Queue(4)  
print('head:{0};tail:{1};length:{2}'.format(queue1.head,queue1.tail,queue1.lenth()))  
# head:0;tail:-1;length:0  
queue1.addLast('北京大学')  
queue1.addLast('清华大学')  
queue1.addLast('北京邮电大学')  
queue1.addLast('北京师范大学')  
print(queue1.queue)#['北京大学', '清华大学', '北京邮电大学', '北京师范大学']  
# queue1.addLast('北京理工大学')#异常：队列已满  
print('head:{0};tail:{1};length:{2}'.format(queue1.head,queue1.tail,queue1.lenth()))  
# head:0;tail:3;length:4  
print(queue1.getValue())#北京大学  
print(queue1.getValue())#清华大学  
print('head:{0};tail:{1};length:{2}'.format(queue1.head,queue1.tail,queue1.lenth()))  
# head:2;tail:3;length:2  
print(queue1.getValue())#北京邮电大学  
print(queue1.isEmpty())#False  
print(queue1.getValue())#北京师范大学  
print(queue1.isEmpty())#True  
print('head:{0};tail:{1};length:{2}'.format(queue1.head,queue1.tail,queue1.lenth()))  
# head:4;tail:3;length:0