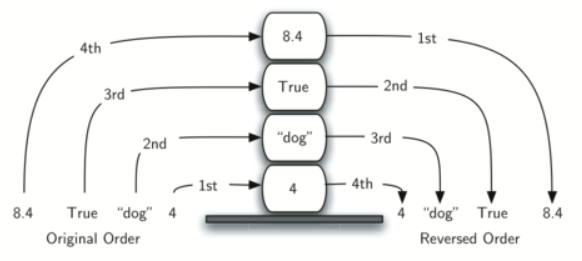
**栈**

1. 定义：栈是一种特殊的列表，栈内的元素只能通过列表的一端访问，这一端称为栈顶。咖啡厅内的一摞盘子是现实世界中常见的栈的例子。只能从最上面取盘子，盘子洗净后，也只能摞在这一摞盘子的最上面。栈被称为一种后入先出（LIFO，last-in-first-out）的数据结构。
2. 栈的操作：对栈的两种主要操作是将一个元素压入栈和将一个元素弹出栈。入栈使用push()方法，出栈使用pop()方法。下图演示了入栈和出栈的过程。

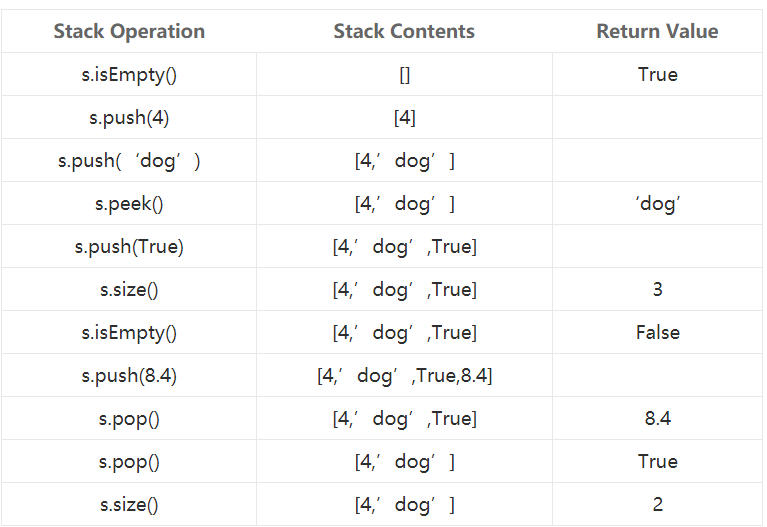


另一个常用的操作是预览栈顶的元素。pop()方法虽然可以访问栈顶的元素，但是调用该方法后，栈顶元素也从栈中被永久性地删除了。peek()方法则只返回栈顶元素，而不删除它。

1. Python中常用的stack操作：

|  |
| --- |
| Stack() 建立一个空的栈对象  push() 把一个元素添加到栈的最顶层  pop() 删除栈最顶层的元素，并返回这个元素  peek() 返回最顶层的元素，并不删除它  isEmpty() 判断栈是否为空  size() 返回栈中元素的个数 |

例子：



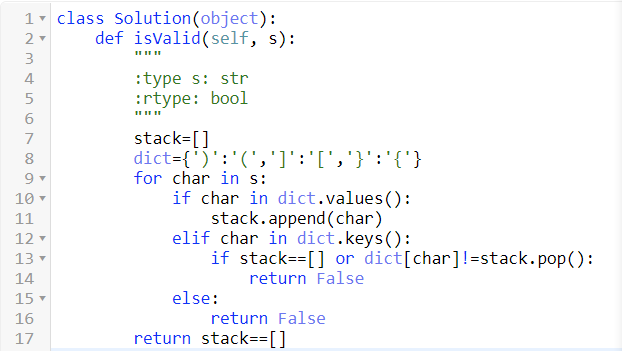
**习题**

1. **有效括号（题号20，Valid Parentheses）**

题目：给出一组字符串，包括括号结构，判断括号是否有效

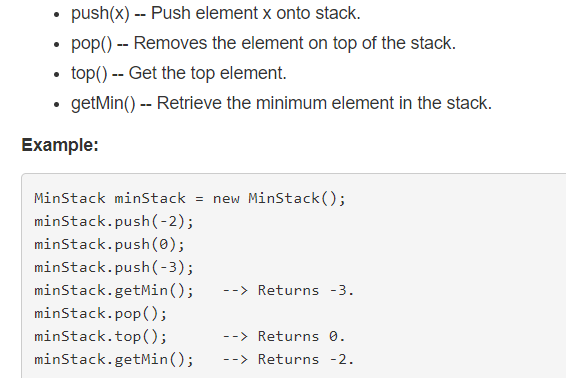
C:\Users\zhaolei\AppData\Local\Temp\1521077377(1).png

思路：将括号构成字典，这里巧妙地是把左括号作为值，右括号作为键，因为如果字符串中是正确的括号，一定会先出现左括号，因此在值里找左括号，然后把它加入栈，等着右边的来吻合，然后如果字符在键中也就是出现右边的括号则，则判断它的值是否能与栈顶的元素匹配，若不匹配则返回错误，如匹配只能说它暂时是对的，但还没有遍历完字符串，不能认定后边是对的，如果字符串的值不是左括号也不是右括号，返回false

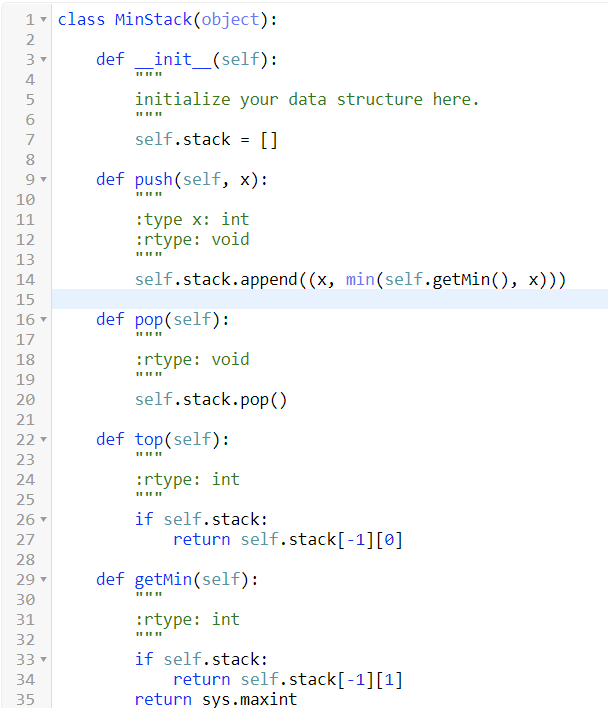


1. 最小栈（题号155，Min Stack）

题目：设计一个栈，支持push、pop、top，并在常量时间内检索最小元素。



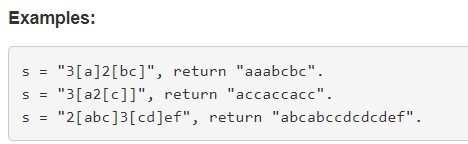
思路：构建栈,push里放输入元素和当前最小值组成的元组，便于获得最小值



Sys.maxint求的是最大值

1. **解码字符串（题号394，String）**

题目：给定字符串，解码，数字代表后边括号内字母重复的次数



思路：创建一个类似元组的栈，元组第一个位置放括号里的字母，第二个位置放它前边的数字，检测数字以它自身为准，而检测字母以括号为开始和结束，当字符为左括号时，代表将要出现字母，此时入栈一个空着字母位置的元组，当出现字母时，把它放进刚创建元组的空字母位置，当出现右括号是，代表当前字母输入结束，元组创建成功，此时出栈这个元组，并把字母及其出现分别次数赋值，然后把第一个位置的字母处放入上述二者的积，然后把这个位置返回，就实现了解码。栈是后进先出，因此每遇到左括号就入栈，右括号就出栈，每遇到一个完整的括号就实现了一次入栈及出栈，使栈始终只有栈底元素，刚好利用它可以存放解码，空间利用很巧妙

