**2. Add Two Numbers**

You are given two **non-empty** linked lists representing two non-negative integers. The digits are stored in **reverse order** and each of their nodes contain a single digit. Add the two numbers and return it as a linked list.

You may assume the two numbers do not contain any leading zero, except the number 0 itself.

**Example:**

**Input:** (2 -> 4 -> 3) + (5 -> 6 -> 4)

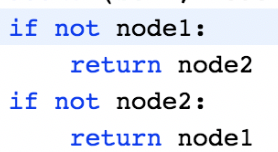
**Output:** 7 -> 0 -> 8

**Explanation:** 342 + 465 = 807.

注意：1.链表长度不一致的情况。2.如何进行进位（carry）

这题是关于链表的，仔细观察链表的构造会发现和之前写的二叉树结构差不多，甚至还要简单，所以自然而然就可以采用递归思路进行解题。

**思路：递归**

1.因为需要注意链表长度不一致的情况，所以这里的递归终点有个小技巧，

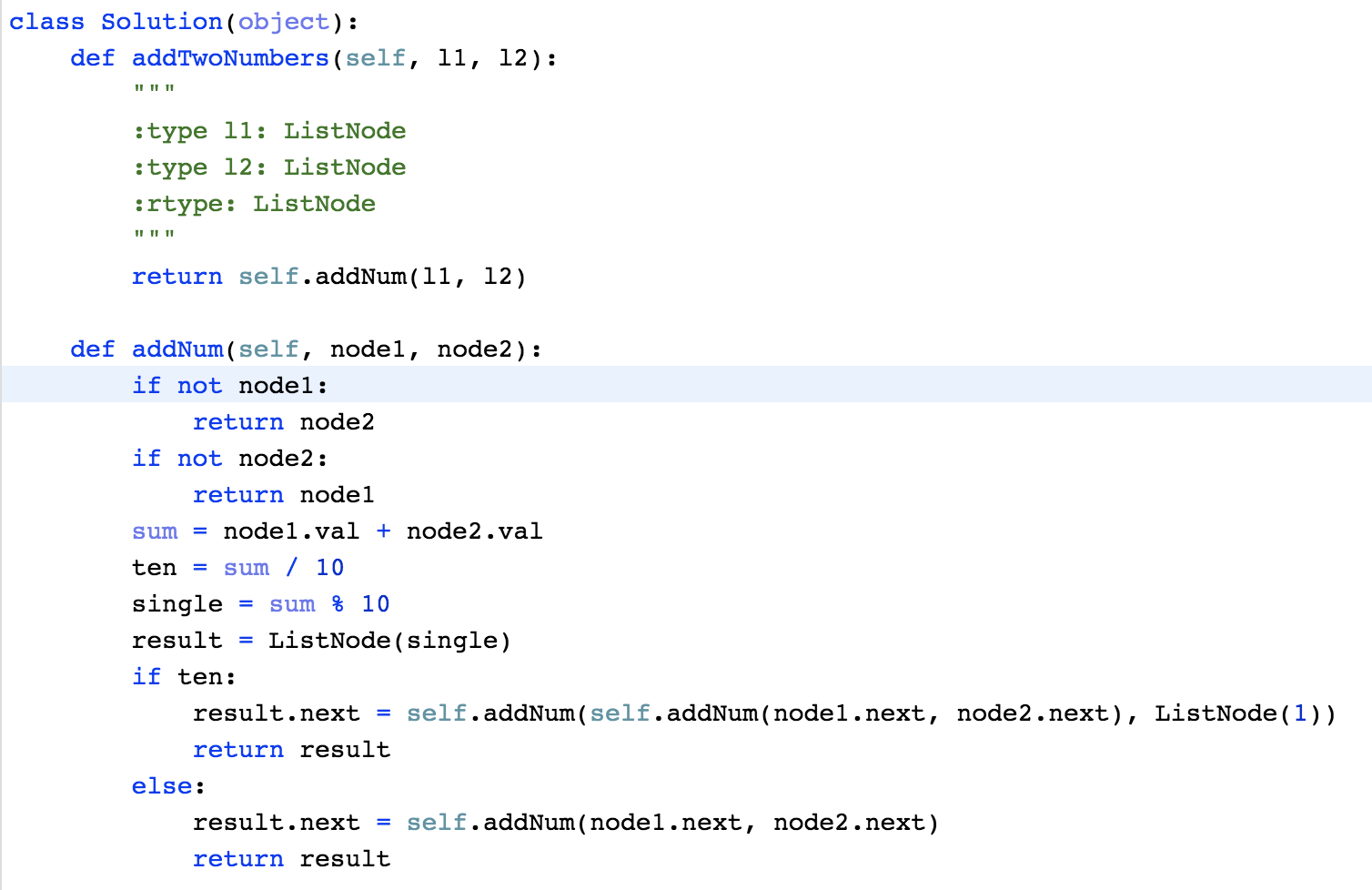
这样就可以解决这个问题。

2.如何解决进位问题：

When: sum >= 10时

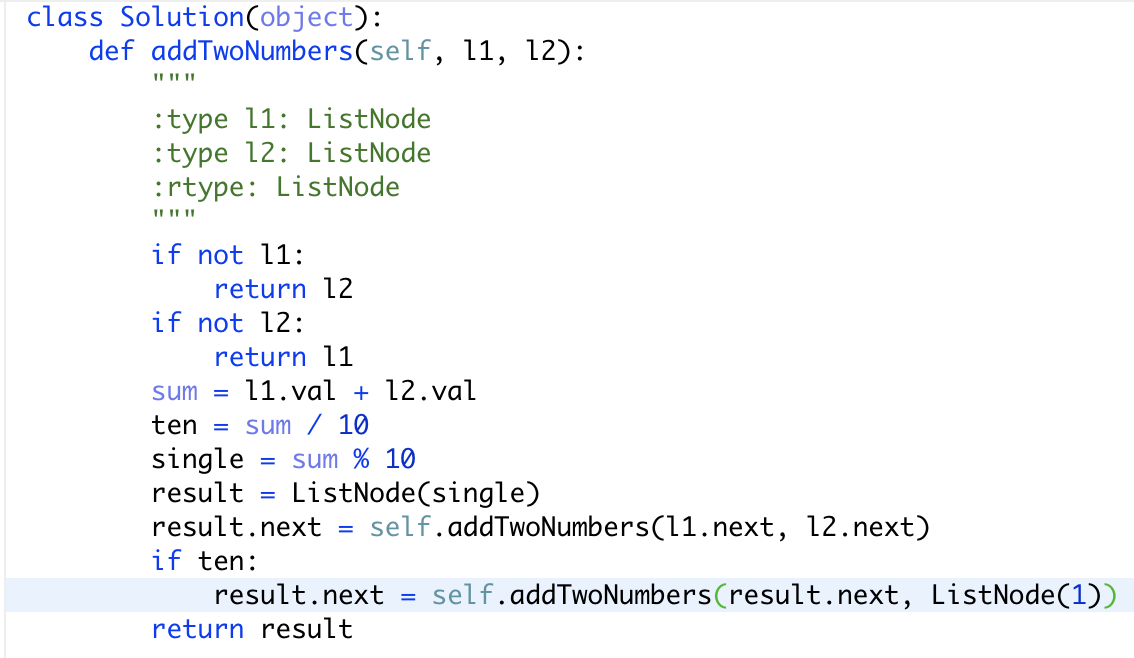
How: %取余，/取整，if语句做判断。

注意：因为采用递归，而递归的函数是用ListNode做变量类型的，所以在进行进位时，要注意类型, 使用ListNode(1)，而不是直接加1.



我们注意到 if, else里面都有这么相同的一步：result.next = self.addTwoNumbers(node1.next, node2.next)，即把下面两个结点相加。同时我也发现写递归可以不用再写一个函数进行递归，所以可以改良一下。

改良版：



Reference：

<https://blog.csdn.net/fuxuemingzhu/article/details/79379626>