# Return Binary In-order Traversal

**Question**: Given a binary tree, return the inorder traversal of its nodes' values.

For example:

Given binary tree {1,2,3},

```
   1
    \
     2
    /
   3
```

return [1,3,2].

**Solutions:**

```
class TreeNode:
    def __init__(self, x):
        self.val = x
        self.left = None
        self.right = None
```

```
class Solution:
    def inorderTraversal(root):
        stack = []
        node = root
        solution = []
        while node!= None or len(stack)>0:
            if node != None:
                stack.append(node)
                node = node.left
            else:
                node = stack.pop()
                solution.append(node.val)
                node = node.right
        return solution

if __name__ == '__main__':
    BT, BT.right, BT.right.left = TreeNode(1), TreeNode(2), TreeNode(3)
    print ( Solution.inorderTraversal(BT) )
```