

$E \times 1:$

Python Tutor code visualizer: +

python.com/tutor/understandings/play

Python Tutor: Visualize code in Python, JavaScript, C, C++, and Java

```

Python 3.6
(Recent iterations)
1 class Node:
2     def __init__(self,Val):
3         self.Val = Val
4         self.Next = ""
5         self.Prev = ""
6     def append(self,Val):
7         TempNode = self.Next
8         while TempNode != "":
9             if TempNode.Next == "":
10                TempNode.Next = Node(Val)
11            TempNode.Next.Prev = TempNode
12            break
13        else:
14            TempNode = TempNode.Next
15        if self.Next == "":
16            self.Next = Node(Val)
17            self.Next.Prev = self
18
19 header = Node(5)
20 header.append(10)
21 header.append(15)

```

Edit this code

→ line that just executed
→ next line to execute

Done running (35 steps)

Frames

Global frame

header

Objects

Node class

- __init__(self, Val)
- append(self, Val)

Node instance

- Next
- Prev
- Val 5

Node instance

- Next
- Prev
- Val 10

Node instance

- Next
- Prev
- Val 15

 $E \times 2^\circ$ [illegible]

code:

https://www.onlinegdb.com/_CywoFkOYu

Ex3
1

Print output (drag lower right corner to resize)

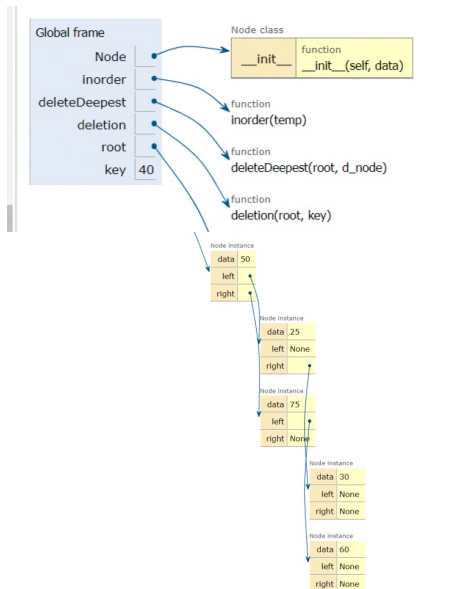
The tree before the deletion:
25 30 40 50 60 75
The tree after the deletion;
25 40 50 60 75

Print output (drag lower right corner to resize)

The tree before the deletion:
25 30 40 50 60 75
The tree after the deletion;
25 30 50 60 40

Print output (drag lower right corner to resize)

The tree before the deletion:
25 30 40 50 60 75
The tree after the deletion;
25 30 50 60 75



Ex 4

```
main.py x +
1 from anytree import Node, RenderTree
2
3 root = Node(1)
4
5 level_1_child_1 = Node(2, parent=root)
6 level_1_child_2 = Node(3, parent=root)
7 level_1_child_3 = Node(4, parent=root)
8 level_2_child_1 = Node(5, parent=level_1_child_1)
9 level_2_child_2 = Node(6, parent=level_1_child_1)
10 level_2_child_1 = Node(7, parent=level_1_child_3)
11
12 ▼ for pre, fill, node in RenderTree(root):
13     print("%s%s" % (pre, node.name))
```



```
graph TD
    1 --- 2
    1 --- 3
    1 --- 4
    2 --- 5
    2 --- 6
    4 --- 7
```