

```

BST.prototype.printTree = function(unused)
{
    this.commands = [];

    if (this.treeRoot != null)
    {
        this.highlightID = this.nextIndex++;
        var firstLabel = this.nextIndex;
        this.cmd("CreateHighlightCircle", this.highlightID,
BST.HIGHLIGHT_CIRCLE_COLOR, this.treeRoot.x, this.treeRoot.y);
        this.xPosOfNextLabel = BST.FIRST_PRINT_POS_X;
        this.yPosOfNextLabel = this.first_print_pos_y;
        this.printTreeRec(this.treeRoot);
        this.cmd("Delete", this.highlightID);
        this.cmd("Step")

        for (var i = firstLabel; i < this.nextIndex; i++)
        {
            this.cmd("Delete", i);
        }
        this.nextIndex = this.highlightID;  /// Reuse objects.  Not
necessary.
    }
    return this.commands;
}

BST.prototype.printTreeRec = function(tree)
{
    this.cmd("Step");
    if (tree.left != null)
    {
        this.cmd("Move", this.highlightID, tree.left.x, tree.left.y);
        this.printTreeRec(tree.left);
        this.cmd("Move", this.highlightID, tree.x, tree.y);

        this.cmd("Step");
    }
    var nextLabelID = this.nextIndex++;
    this.cmd("CreateLabel", nextLabelID, tree.data, tree.x, tree.y);
    this.cmd("SetForegroundColor", nextLabelID, BST.PRINT_COLOR);
    this.cmd("Move", nextLabelID, this.xPosOfNextLabel,
this.yPosOfNextLabel);
    this.cmd("Step");

    this.xPosOfNextLabel += BST.PRINT_HORIZONTAL_GAP;
    if (this.xPosOfNextLabel > this.print_max)
    {
        this.xPosOfNextLabel = BST.FIRST_PRINT_POS_X;
        this.yPosOfNextLabel += BST.PRINT_VERTICAL_GAP;
    }
    if (tree.right != null)
    {
        this.cmd("Move", this.highlightID, tree.right.x,
tree.right.y);
        this.printTreeRec(tree.right);
        this.cmd("Move", this.highlightID, tree.x, tree.y);
    }
}

```

```
        this.cmd("Step");  
    }  
    return;  
}
```