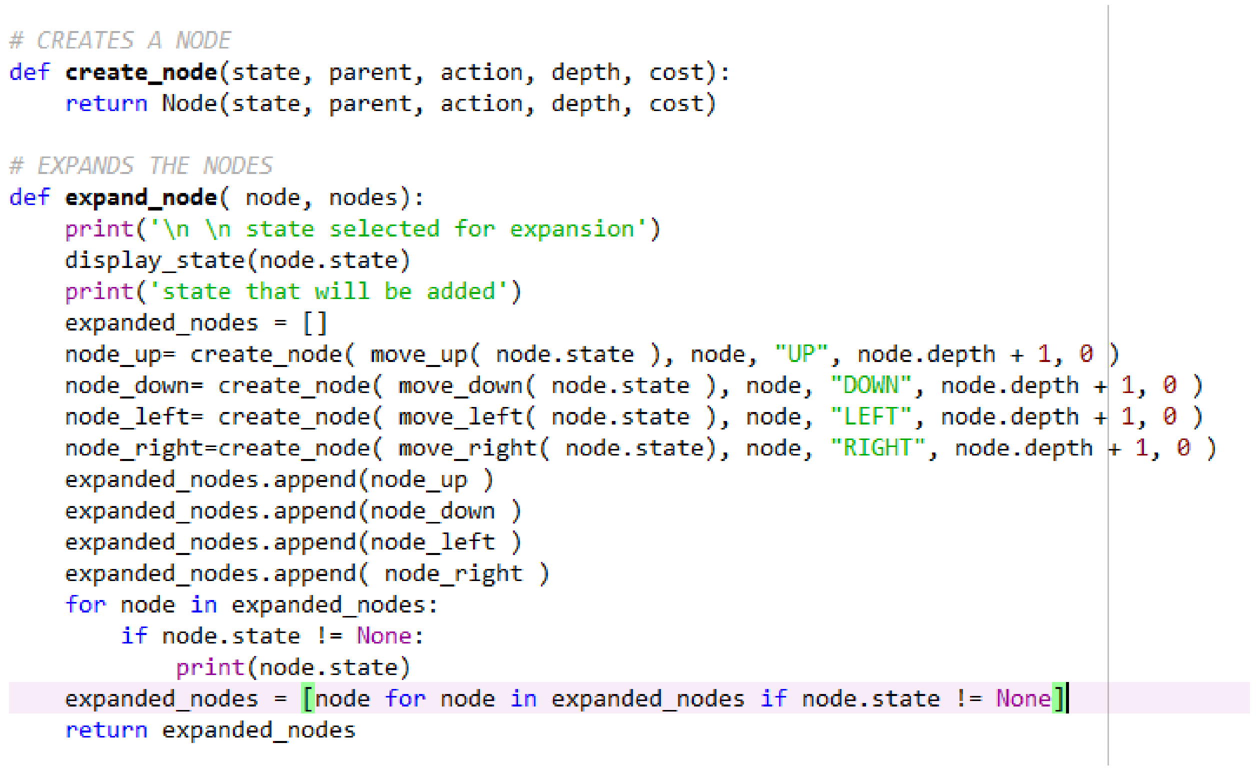
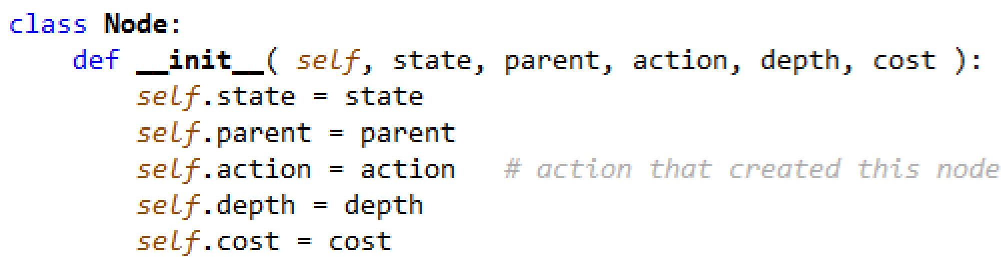
Successor function (expand_node)



```python
# CREATES A NODE
def create_node(state, parent, action, depth, cost):
    return Node(state, parent, action, depth, cost)

# EXPANDS THE NODES
def expand_node( node, nodes):
    print('\n \n state selected for expansion')
    display_state(node.state)
    print('state that will be added')
    expanded_nodes = []
    node_up= create_node( move_up( node.state ), node, "UP", node.depth + 1, 0 )
    node_down= create_node( move_down( node.state ), node, "DOWN", node.depth + 1, 0 )
    node_left= create_node( move_left( node.state ), node, "LEFT", node.depth + 1, 0 )
    node_right=create_node( move_right( node.state), node, "RIGHT", node.depth + 1, 0 )
    expanded_nodes.append(node_up )
    expanded_nodes.append(node_down )
    expanded_nodes.append(node_left )
    expanded_nodes.append( node_right )
    for node in expanded_nodes:
        if node.state != None:
            print(node.state)
    expanded_nodes = [node for node in expanded_nodes if node.state != None]
    return expanded_nodes
```



```python
class Node:
    def __init__( self, state, parent, action, depth, cost ):
        self.state = state
        self.parent = parent
        self.action = action   # action that created this node
        self.depth = depth
        self.cost = cost
```