# DFS Tree Traversals (Inorder, Preorder, Postorder)

Tree traversal means visiting each node of a tree in some order. Depth First Search (DFS) has three common types of traversals:

1. Inorder  
2. Preorder  
3. Postorder

## 1. Inorder Traversal (Left → Root → Right)

Steps:  
- Visit the Left child  
- Visit the Root  
- Visit the Right child  
  
Inorder traversal of a Binary Search Tree gives nodes in sorted order.

Example Tree:  
 A  
 / \  
 B C  
 / \  
 D E  
  
Inorder = D → B → E → A → C

## 2. Preorder Traversal (Root → Left → Right)

Steps:  
- Visit the Root  
- Visit the Left child  
- Visit the Right child  
  
Preorder traversal is useful for making a copy of a tree.

Example Tree:  
 A  
 / \  
 B C  
 / \  
 D E  
  
Preorder = A → B → D → E → C

## 3. Postorder Traversal (Left → Right → Root)

Steps:  
- Visit the Left child  
- Visit the Right child  
- Visit the Root  
  
Postorder traversal is useful for deleting nodes in a tree because children are visited before the parent.

Example Tree:  
 A  
 / \  
 B C  
 / \  
 D E  
  
Postorder = D → E → B → C → A

## Python Code Example

class Node:  
 def \_\_init\_\_(self, value):  
 self.value = value  
 self.left = None  
 self.right = None  
  
def inorder(root):  
 if root:  
 inorder(root.left)  
 print(root.value, end=" ")  
 inorder(root.right)  
  
def preorder(root):  
 if root:  
 print(root.value, end=" ")  
 preorder(root.left)  
 preorder(root.right)  
  
def postorder(root):  
 if root:  
 postorder(root.left)  
 postorder(root.right)  
 print(root.value, end=" ")  
  
root = Node("A")  
root.left = Node("B")  
root.right = Node("C")  
root.left.left = Node("D")  
root.left.right = Node("E")  
  
print("Inorder: ")  
inorder(root)  
  
print("\nPreorder: ")  
preorder(root)  
  
print("\nPostorder: ")  
postorder(root)