**Inorder, Preorder, and Postorder in DFS (Depth-First Search)**

Depth-First Search (DFS) is a tree traversal technique used to explore nodes of a tree or graph. In binary trees, DFS can be implemented in three different ways:

1. **Inorder Traversal (Left, Root, Right)**
   * Visit the **left** subtree.
   * Visit the **root** node.
   * Visit the **right** subtree.
   * Used in Binary Search Trees (BST) to retrieve values in sorted order.
2. **Preorder Traversal (Root, Left, Right)**
   * Visit the **root** node.
   * Visit the **left** subtree.
   * Visit the **right** subtree.
   * Useful for copying trees or prefix expressions in expression trees.
3. **Postorder Traversal (Left, Right, Root)**
   * Visit the **left** subtree.
   * Visit the **right** subtree.
   * Visit the **root** node.
   * Used in deleting a tree or evaluating postfix expressions.

**Implementation in Python**

Here is a Python implementation of DFS using these three traversal methods:

class Node:

def \_\_init\_\_(self, key):

self.left = None

self.right = None

self.val = key

# Inorder (Left, Root, Right)

def inorder(root):

if root:

inorder(root.left)

print(root.val, end=" ")

inorder(root.right)

# Preorder (Root, Left, Right)

def preorder(root):

if root:

print(root.val, end=" ")

preorder(root.left)

preorder(root.right)

# Postorder (Left, Right, Root)

def postorder(root):

if root:

postorder(root.left)

postorder(root.right)

print(root.val, end=" ")

# Example Tree

# 1

# / \

# 2 3

# / \

# 4 5

root = Node(1)

root.left = Node(2)

root.right = Node(3)

root.left.left = Node(4)

root.left.right = Node(5)

print("Inorder traversal:")

inorder(root)

print("\nPreorder traversal:")

preorder(root)

print("\nPostorder traversal:")

postorder(root)