

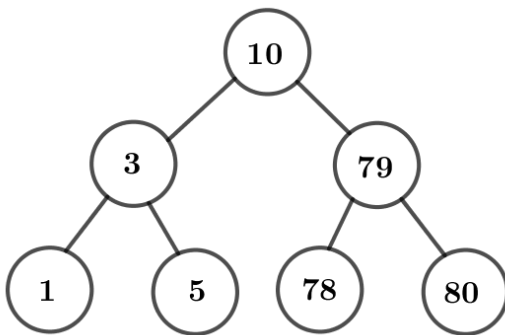
TD : PARCOURS EN PROFONDEUR

RAPPEL : LES PARCOURS EN PROFONDEUR

Ces parcours descendent en profondeur dans l'arbre tant que c'est possible. Pour traiter un nœud n , on traite d'abord tous ses descendants et on remonte ensuite pour traiter le père de n et son autre fils. Ces parcours sont de trois type : le **parcours infixe**, le **parcours préfixe**, le **parcours suffixe** (ou *postfixe*).

- Le **parcours infixe** traite d'abord le sous-arbre gauche, puis le nœud courant puis le sous-arbre droit :
- Le **parcours préfixe** traite d'abord le nœud courant puis le sous-arbre-gauche, puis le sous-arbre droit.
- Le **parcours suffixe** (ou **postfixe**) traite d'abord le sous-arbre gauche, puis le sous-arbre droit, puis le nœud courant.

Par exemple, pour l'arbre :



Parcours en largeur : 10, 3, 79, 1, 5, 78, 80

Parcours infixe : 1, 3, 5, 10, 78, 79, 80

Parcours préfixe : 10, 3, 1, 5, 79, 78, 80

Parcours suffixe : 1, 5, 3, 78, 80, 79, 10

PARCOURS INFIXE :

```
class Noeud:
    def __init__(self, key):
        self.gauche = None
        self.droit = None
        self.val = key

def infixe(Noeud):
    if Noeud==None:
        return
    if Noeud.gauche!=None:
        infixe(Noeud.gauche)
    print(Noeud.val)
    if Noeud.droit!=None:
        infixe(Noeud.droit)
```

TRAVAIL A FAIRE

- 1) Tester la fonction infixe avec l'arbre ci-dessus
- 2) Ecrire et tester avec l'arbre ci-dessus les fonctions préfixe et suffixe.