Structure de données abstraites (SDA)	Structures arborescentes	**Les Arbres Binaires**

Définition1 : Un arbre **binaire** est un arbre dont chaque nœud a au plus deux fils : le fils gauche et le fils droit.

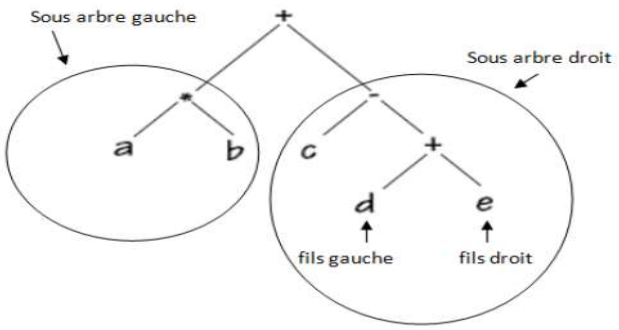
Exercice 1 : Tracer : *Un arbre binaire de taille 5*	*Un arbre binaire de de hauteur 4*	*Tous les arbres binaires ayant 3 nœuds*

Donner des exemples d'arbres binaires dans la vie courante :

-

-

Définition 2 (Sous arbres droits et gauche) :
Soit N un nœud dans un arbre binaire, l'ensemble des descendants du fils gauche est appelé sous arbre gauche, du fils droit est appelé sous arbre droit



L'arbre qui représente l'expression $a \times b + c - (d + e)$ est un arbre binaire.

Les arbres binaires forment une structure de données qui peut **se définir de façon récursive**.
Un arbre binaire est soit vide, soit composé d'une racine portant une étiquette (clé) et d'une paire d'arbres binaires, appelés sous-arbres gauche et droit.

Propriété : La **hauteur** h d'un arbre binaire et sa **taille** n vérifie l'inégalité suivante : $h + 1 \leq n \leq 2^{h+1} - 1$

Exercice 2 : Tracer un arbre binaire qui vérifie $h + 1 = n$	Tracer un arbre binaire qui vérifie $n = 2^{h+1} - 1$ Dans ce cas, on parle d'arbre **parfait**

Les algorithmes importants à connaitre sur les arbres :

La structure d'un arbre binaire incite à choisir des algorithmes récursifs

Calcul de la hauteur de l'arbre :	**Calcul de la taille de l'arbre :**
Fonction : **hauteur** (arbre) : Rôle : Fonction qui prend en paramètre d'entrée un arbre et retourne un entier. Si l'arbre est vide on retournera -1 Début Si …… ……. Sinon …… Fin Fin	Fonction : **taille** (arbre) : Rôle : Fonction qui prend en paramètre d'entrée un arbre et retourne un entier. Si l'arbre est vide on retournera 0 Début Si …… ……. Sinon …… Fin Fin

Il existe différentes façons **d'implémenter** les arbres binaires : (voir les notebooks)

- Un tuple simple ou un tuple de tuples (notebook1)
- Un dictionnaire (notebook2)
- Avec la programmation objet : (script1)

Une façon classique de représenter le type abstrait Arbre binaire est de représenter chaque nœud par un objet d'une **classe Noeud**.

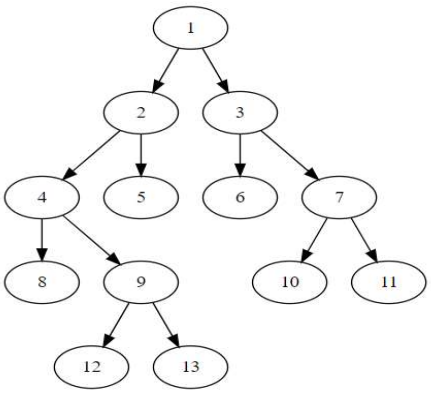
Un objet de cette classe contient trois attributs, donnés dans l'ordre suivant :
- valeur pour la valeur de l'étiquette contenue dans le nœud
- gauche pour le sous-arbre gauche
- droit pour le sous-arbre droit

L'arbre vide est représenté par la valeur None.

On peut alors créer un arbre binaire en appelant autant de fois que nécessaire le constructeur de la classe Noeud.

Pour chaque implémentation, construire une représentation de l'arbre ci-dessous

arbre 2 :



Exercice : a) dessiner l'arbre A construit par les instructions :

n6=Nœud(6)
n5=Noeud(5,n6,None)
n4=Noeud(4,None,n5)
n3=Noeud(3)
n2=Noeud(2,n3,n4)
A=Noeud(1,n2,None)

b) dessiner l'arbre A1 définit par :

```
A1=Noeud(2,Noeud(8,Noeud(4,None,None),Noeud(5,None,None)),Noeud(9,None,Noeud(3,None,None)))
```

*c) Compléter le script1 de la classe **Nœud** avec les méthodes :*

est_vide ,est_feuille, get_valeur, get_fils_gauche, get_fils_droit , hauteur, taille, nb_feuilles,