

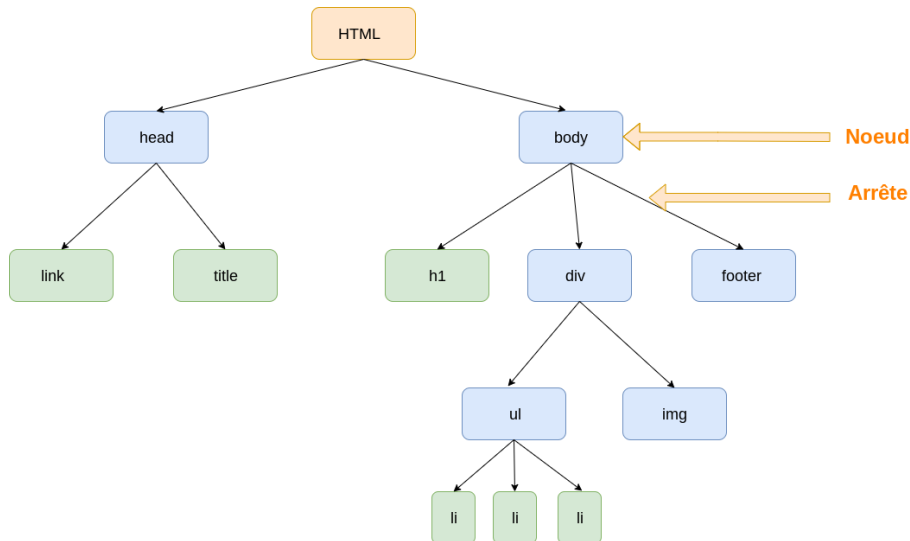
*i* Dans ce cours, les arbres utilisés sont des arbres enracinés.

## Vocabulaire

### Définition

Arbres

Un arbre est un ensemble de noeuds reliés par des arrêtes.

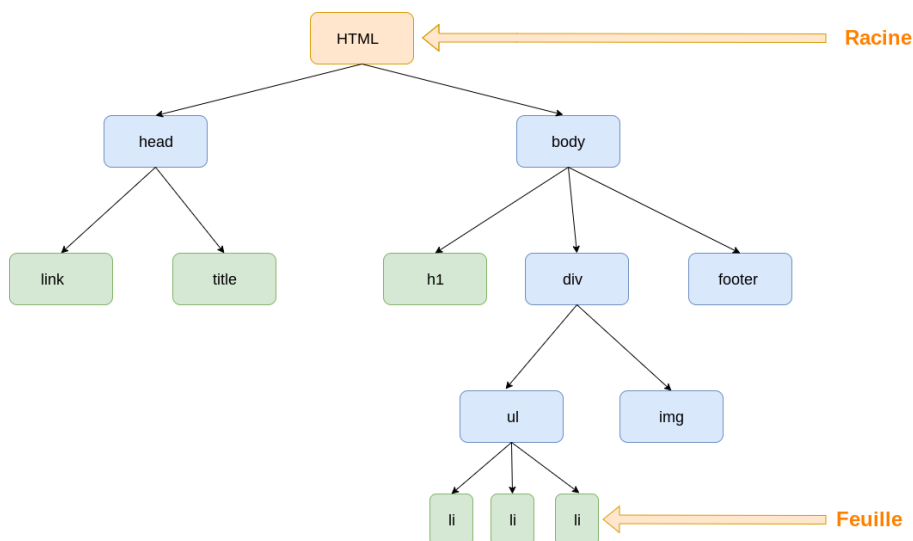


Les noeuds peuvent avoir une valeur, on l'appelle l'**étiquette** du noeud.

### Définition

Racine et feuille

- Un arbre possède une **racine** si un noeud est au départ
- Une **feuille** est un noeud situé à l'extrémité de l'arbre. Il n'a donc pas de descendants.

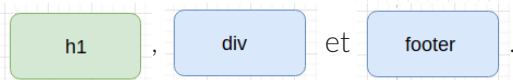


## Définition

Parent & enfants

Dans un arbre chaque noeud possède un parent et des enfants.

Dans notre arbre, le noeud `body` a pour parent `HTML` et 3 enfants



## Mesure sur les arbres

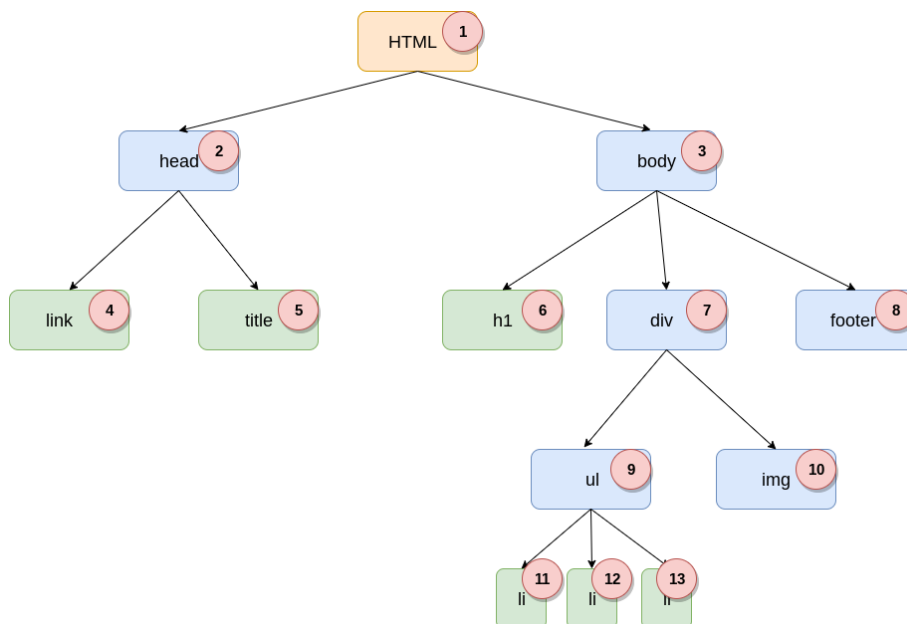
### Définition

taille

La taille d'un arbre est le nombre de noeuds qu'il contient.

### Exemple

Dans l'arbre qui nous sert d'exemple, il y a 13 noeuds



**La taille de l'arbre est 13**

### Définition

Hauteur

La **hauteur** d'un arbre est le nombre de noeuds du plus long chemin qui joint le noeud racine à l'une des feuilles.

On a donc :

- La hauteur d'un arbre vide est 0
- La hauteur d'un arbre réduit à un noeud, c'est-à-dire la racine, est 1.

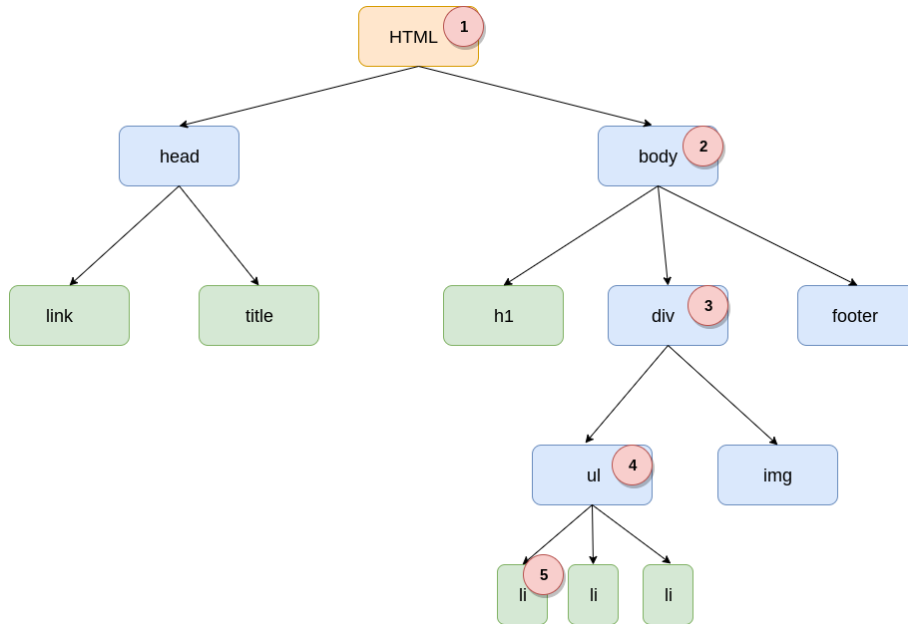
⚠ Cette définition varie en fonction des auteurs. Pour certains la hauteur d'un arbre vide est -1.

Nous utilisons dans ce cours, celle utiliser au baccalauréat 2021.

---

## Exemple

Dans l'arbre qui nous sert d'exemple, le plus long chemin possible se compose de 5 noeuds



**La hauteur de l'arbre est 5.**

# Arbre binaire

## Définition

### Définition

Arbres binaires

Un **arbre binaire** est un arbre dont chaque élément possède au plus deux éléments enfants au niveau inférieur (un fils gauche et un fils droit).

## Encadrement de la hauteur d'un arbre

### Propriété

Un arbre filiforme et un arbre complet étant deux cas extrêmes, on peut affirmer que pour un arbre binaire quelconque :

$$\lfloor \log_2(n) \rfloor + 1 \leq h \leq n$$

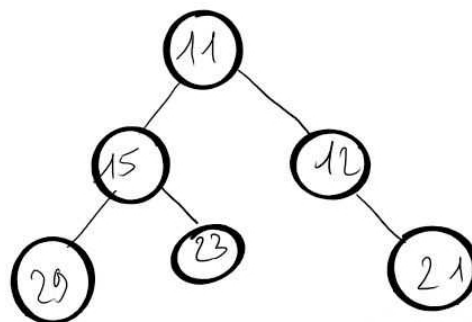
## Implementation

Une manière de représenter un arbre est de créer une classe **Noeud** ayant 3 attributs :

- **valeur** : l'étiquette du noeud.
- **gauche** : le fils gauche du noeud.
- **droit** : le fils droit du noeud.

```
1 class Noeud:
2
3     def __init__(self, valeur, gauche, droit):
4         self.valeur = valeur
5         self.gauche = gauche
6         self.droit = droit
```

Ainsi l'arbre ci dessous :



L'implémentation de cet arbre avec :

```
1 A = Noeud(11,
2     Noeud(15, Noeud(29, None, None), Noeud(23, None, None)),
3     Noeud(12, None, Noeud(21, None, None))
4 )
```

## Algorithmes

Dans cette partie, nous allons étudier l'algorithme de calcul de la taille et la hauteur d'un arbre binaire.

Ces deux mesures se calculent récursivement.

### La taille

- La taille d'un arbre binaire vide vaut 0.
- La taille d'un arbre binaire non vide vaut :  
 $1 + \text{taille}(\text{sous-arbre gauche}) + \text{taille}(\text{sous-arbre droit})$ .

### La hauteur

- La hauteur d'un arbre binaire vide vaut 0.
- La hauteur d'un arbre binaire non vide vaut :  
 $1 + \max(\text{hauteur}(\text{sous-arbre gauche}), \text{hauteur}(\text{sous-arbre droit}))$ .

```
1 def taille(a):
2     if a.valeur == None:
3         return 0
4     else:
5         return 1 + taille(a.gauche) + taille(a.droite)
```

```
1 def hauteur(a):
2     if a.valeur == None:
3         return 0
4     else:
5         return 1 + max(taille(a.gauche), taille(a.droite))
```

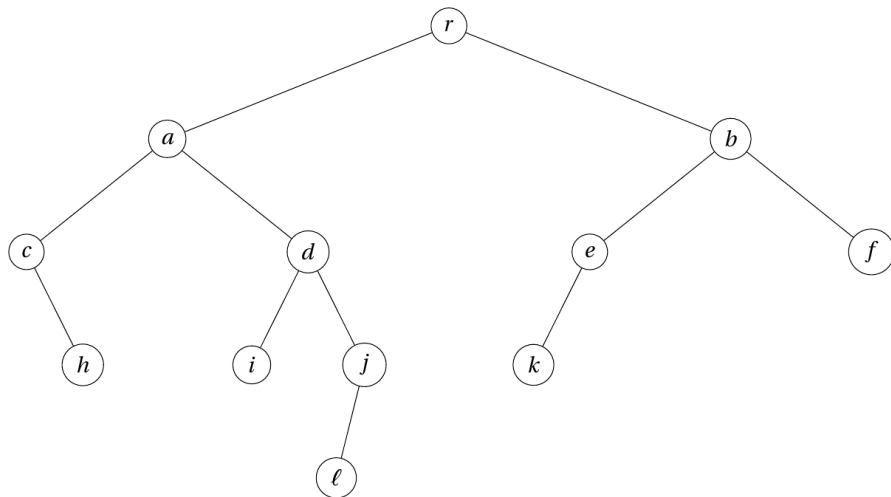
## ● Parcours

*i* Dans cette partie, nous allons voir 4 méthodes de parcours des valeurs d'une arbre.

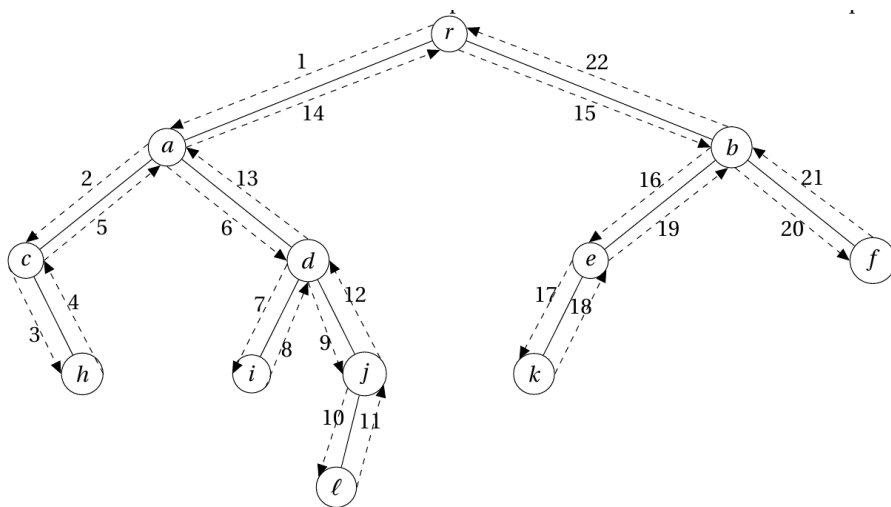
- Parcours en profondeur :
  - Parcours préfixe
  - Parcours infixe
  - Parcours suffixe ou postfixe
- Parcours en largeur

### ●● Parcours en profondeur

Nous allons considérer l'arbre suivant.



Une balade sur l'arbre :



A partir de ce contour, on définit trois parcours des sommets de l'arbre :

- l'ordre **préfixe** : on liste chaque sommet la première fois qu'on le rencontre dans la balade (c'est à dire dès qu'on passe à sa gauche),
- l'ordre **postfixe** ou **suffixe** : on liste chaque sommet la dernière fois qu'on le rencontre. (c'est à dire dès qu'on passe à sa droite),
- l'ordre **infixe** : on liste chaque sommet ayant un fils gauche la seconde fois qu'on le voit et chaque sommet sans fils gauche la première fois qu'on le voit. (c'est à dire dès qu'on passe à en dessous).

---

### Exercice 1

À partir de l'arbre ci-dessus.

1. Déterminer le parcours **préfixe** .

.....  
.....

2. Déterminer le parcours **postfixe** de l'arbre ci-dessus.

.....  
.....

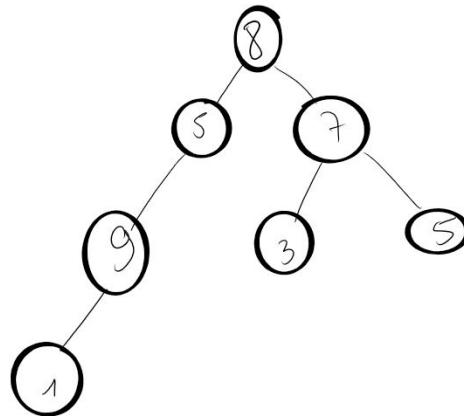
3. Déterminer le parcours **infixe** de l'arbre ci-dessus.

.....  
.....

---

### Exercice 2

À partir de l'arbre ci-dessous.



1. Déterminer le parcours **préfixe** .

.....  
.....

2. Déterminer le parcours **postfixe** de l'arbre ci-dessous.

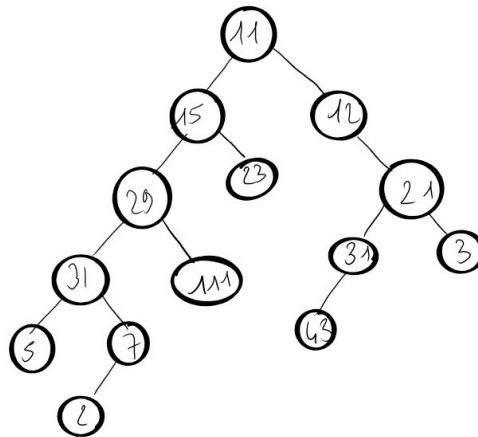
.....  
.....

3. Déterminer le parcours **infixe** de l'arbre ci-dessous.

.....  
.....

### Exercice 3

À partir de l'arbre ci-dessous.



1. Déterminer le parcours **préfixe** de cet arbre.

.....  
.....

2. Déterminer le parcours **suffixe** de cet arbre.

.....  
.....

3. Déterminer le parcours **infixe** de cet arbre.

.....  
.....

### ● Implémentation des parcours en profondeur

*i*

Encore une fois, on va implémenter les parcours de manière **récursive**

On peut aussi définir ces 3 modes de parcours de cette façon.

- **Préfixe** : on lit d'abord la Racine (RGD) puis le sous arbre Gauche et ensuite le sous-arbre Droit
- **Suffixe** ou **postfixe** : on lit la Racine à la fin (GDR)
- **Infixe** : on lit la Racine au milieu de la séquence (GRD), infixe comme intérieur.



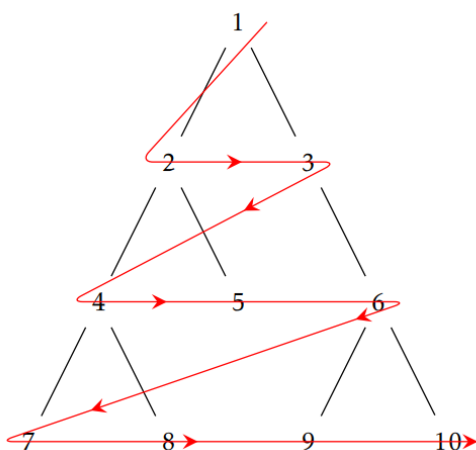
• Définition du parcours en largeur

Définition

Dans le cadre d'un parcours en largeur, on commence par la racine et ensuite de énumère les noeuds de d'une même profondeur de droite à gauche.

Exemple

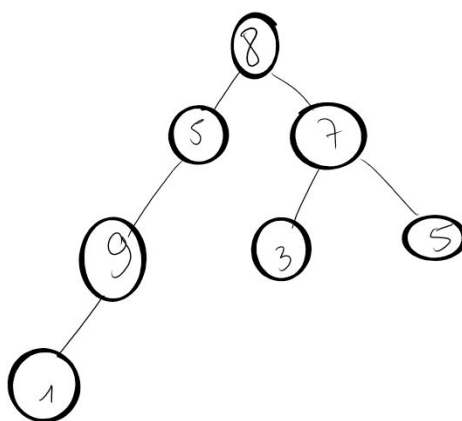
On considère l'arbre ci-dessous.



Le parcours en largeur de l'arbre ci-dessou, nous donnerais 1, 2, 3, 4, 5, 6, 7, 8, 9 et 10.

Exercice 4

À partir de l'arbre ci-dessous.



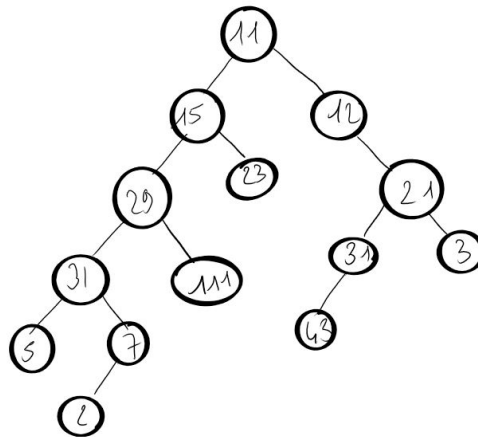
1. Déterminer le parcours en **largeur** de l'arbre ci-dessous.

.....  
.....

---

**Exercice 5**

À partir de l'arbre ci-dessous.



1. Déterminer le parcours en largeur de cet arbre.

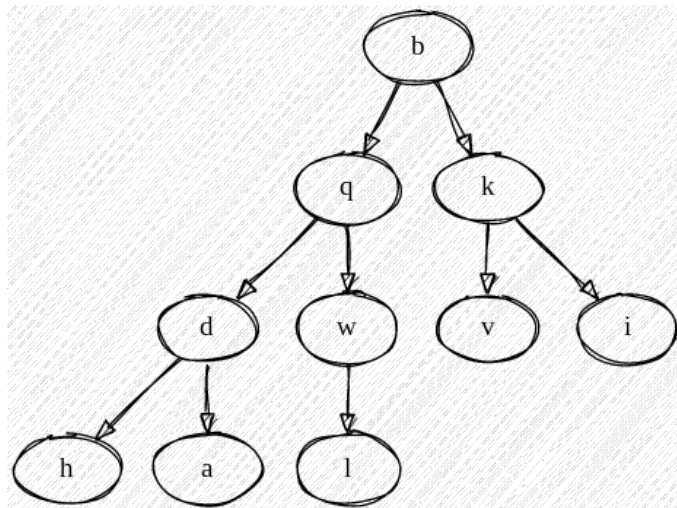
.....

.....

---

**Exercice 6**

À partir de l'arbre ci-dessous.



1. Déterminer le parcours en largeur de cet arbre.

.....

.....

---

**Implémentation du parcours en largeur**