Généralités sur les arbres

.

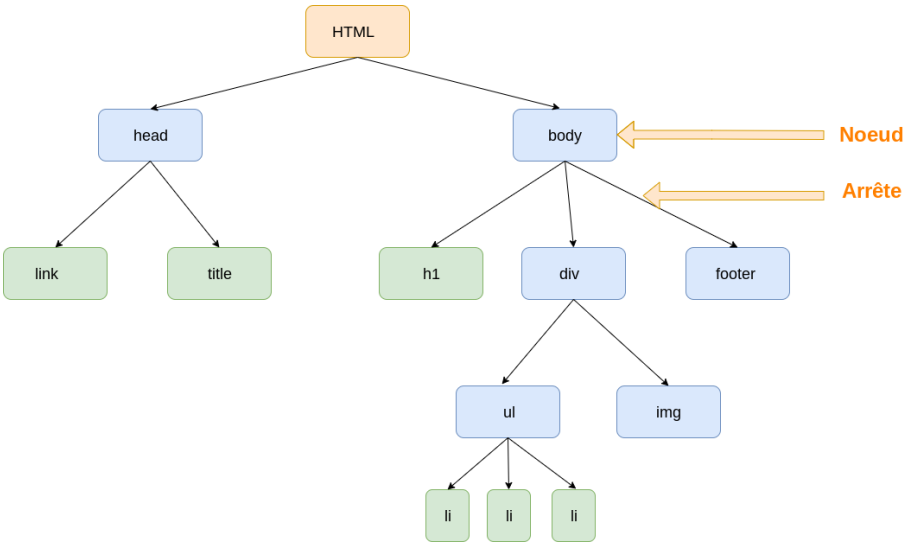
i Dans ce cours, les arbres utilisés dont des arbres enracinés.

Vocabulaire

Définition

Arbres

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

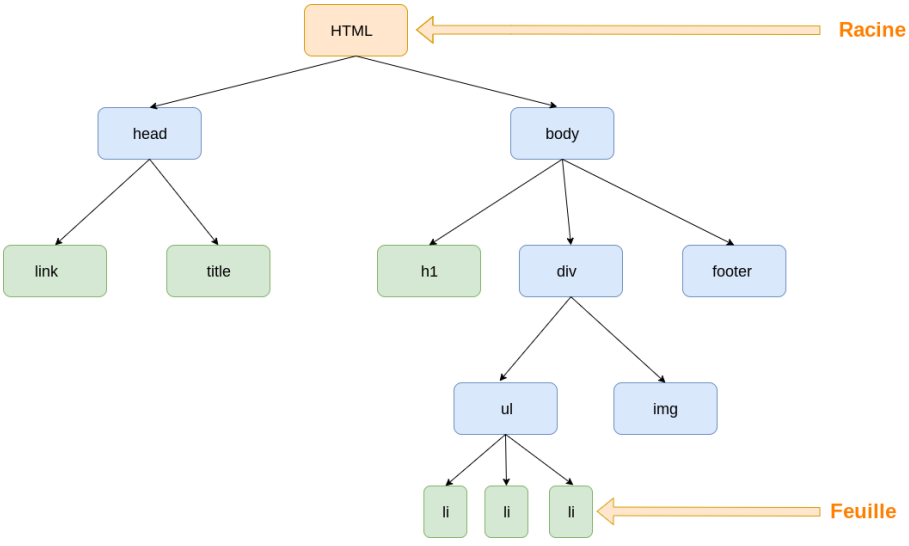


Les noeuds peuvent avoir une valeur, on l'appelle **`l'étiquette`** du noeud.

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 h1, div et footer.

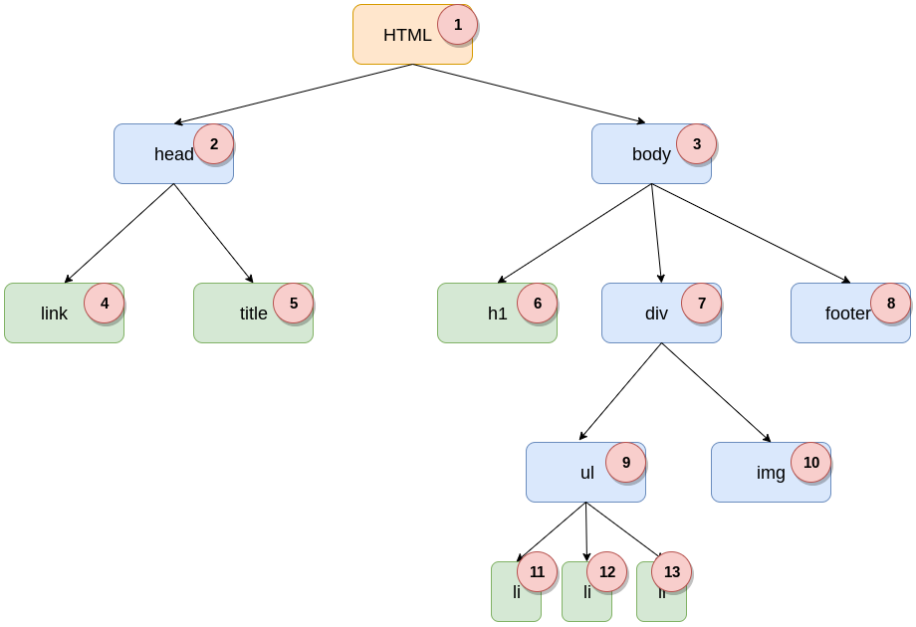
Mesure sur les arbres

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.

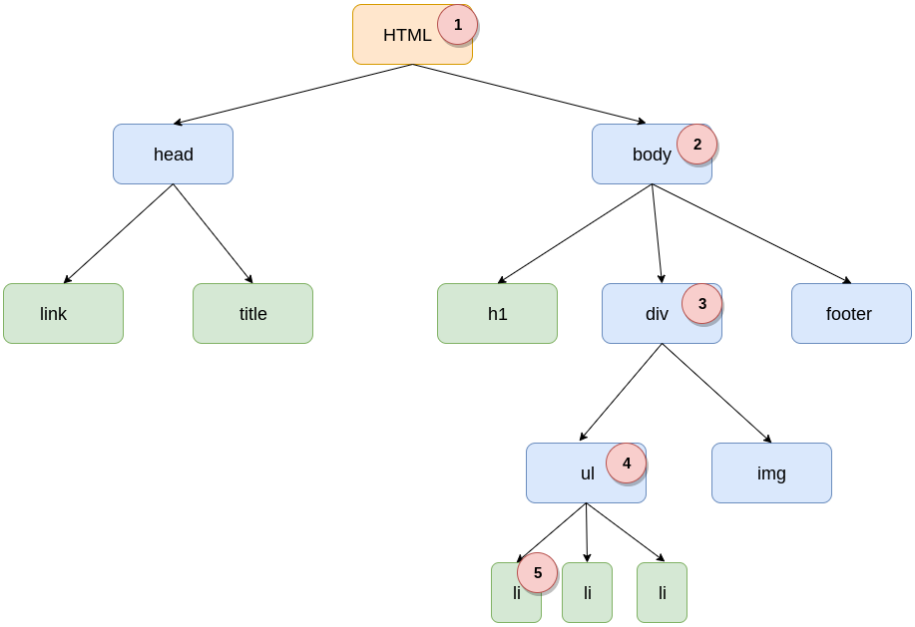
On a donc :

- 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

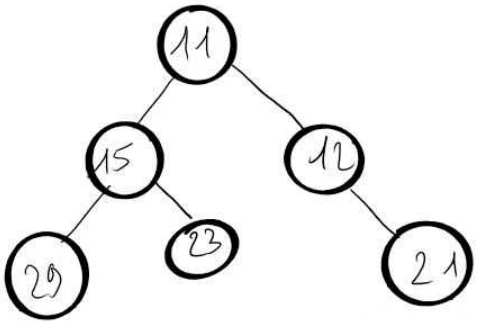
Une manière de représenter un arbre est de créer une classe **Noeud** ayant 3 attributs :

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

```
class Noeud:

    def __init__(self, valeur, gauche, droit):
        self.valeur = valeur
        self.gauche = gauche
        self.droit  = droit
```

Ainsi l'arbre ci dessous :



L'implémentation de cet arbre avec :

```
A = Noeud(11,
        Noeud(15, Noeud(29, None, None), Noeud(23, None, None)),
        Noeud(12, None, Noeud(21, None, None))
    )
```

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 valent 0.
- La taille d'un arbre binaire non vide vaut :

  1 + taille(sous-arbre gauche) + taille(sous-arbre droit).

La hauteur

- La hauteur d'un arbre binaire vide valent 0.
- La hauteur d'un arbre binaire non vide vaut :

  1 + max(hauteur(sous-arbre gauche), hauteur(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.gauche)
```

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