

### 1. Écriture d'une classe Nœud pour un arbre binaire

L'écriture de la classe `Noeud` comme ci-dessous a été définie pour créer un arbre binaire. Dans le constructeur on définit le fils gauche et le fils droit.

**class Noeud:**

```
def __init__(self, valeur, fils_g, fils_d):  
    self.valeur = valeur  
    self.fils_g = fils_g  
    self.fils_d = fils_d
```

L'écriture du nœud est alors la suivante :

```
n1 = Noeud('e', None, None)
```

### 2. Écriture d'une classe Nœud pour un arbre binaire de recherche

Pour créer un arbre binaire de recherche, le fils gauche et le fils droit n'est pas défini au début de la construction car on ne sait pas où va se placer le nœud, par défaut il n'existe pas

Par conséquent si la classe `Noeud` sert seulement à créer un arbre binaire de recherche, il est possible d'écrire la classe tel quel :

**class Noeud:**

```
def __init__(self, valeur):  
    self.valeur = valeur  
    self.fils_g = None  
    self.fils_d = None
```

Ce qui simplifie l'écriture du nœud :

```
n1 = Noeud('e')
```

### 3. Écriture d'une classe Nœud polyvalente

La première écriture de la classe `Noeud` peut être aussi utilisée pour un arbre binaire de recherche mais la création de chaque nœud est plus longue.

L'écriture suivante est aussi polyvalente mais moins fastidieuse.

En absence des attributs `fils_g` et `fils_d`, la valeur qui leur est attribué par défaut `None`.

**class Noeud:**

```
def __init__(self, valeur, fils_g=None, fils_d=None):  
    self.valeur = valeur  
    self.fils_g = fils_g  
    self.fils_d = fils_d
```

L'écriture peut être simplifié si besoin.

```
n1 = Noeud('e')  
n2 = Noeud('e', n1, None)
```