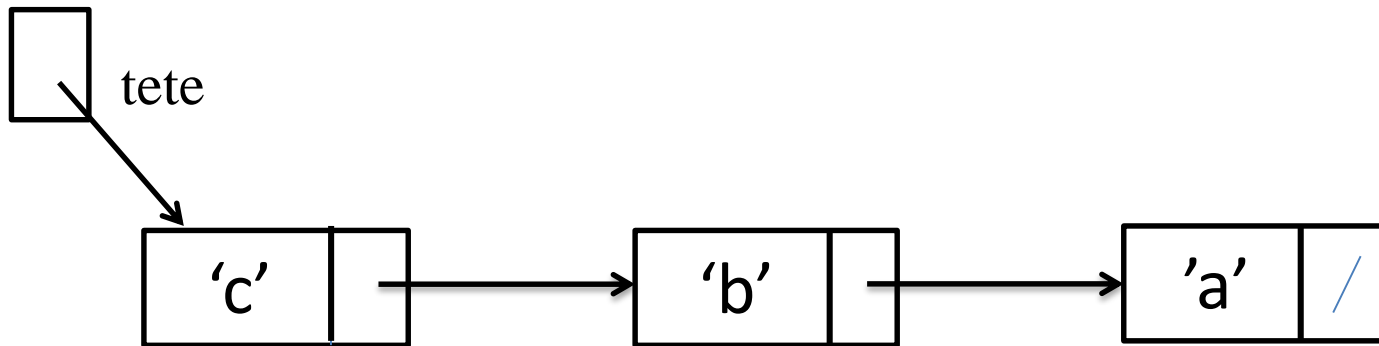
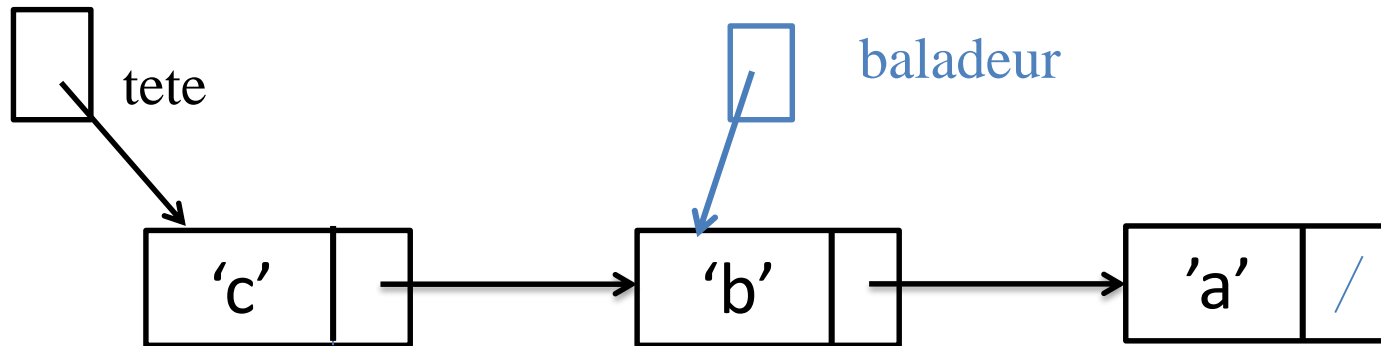


# Les listes

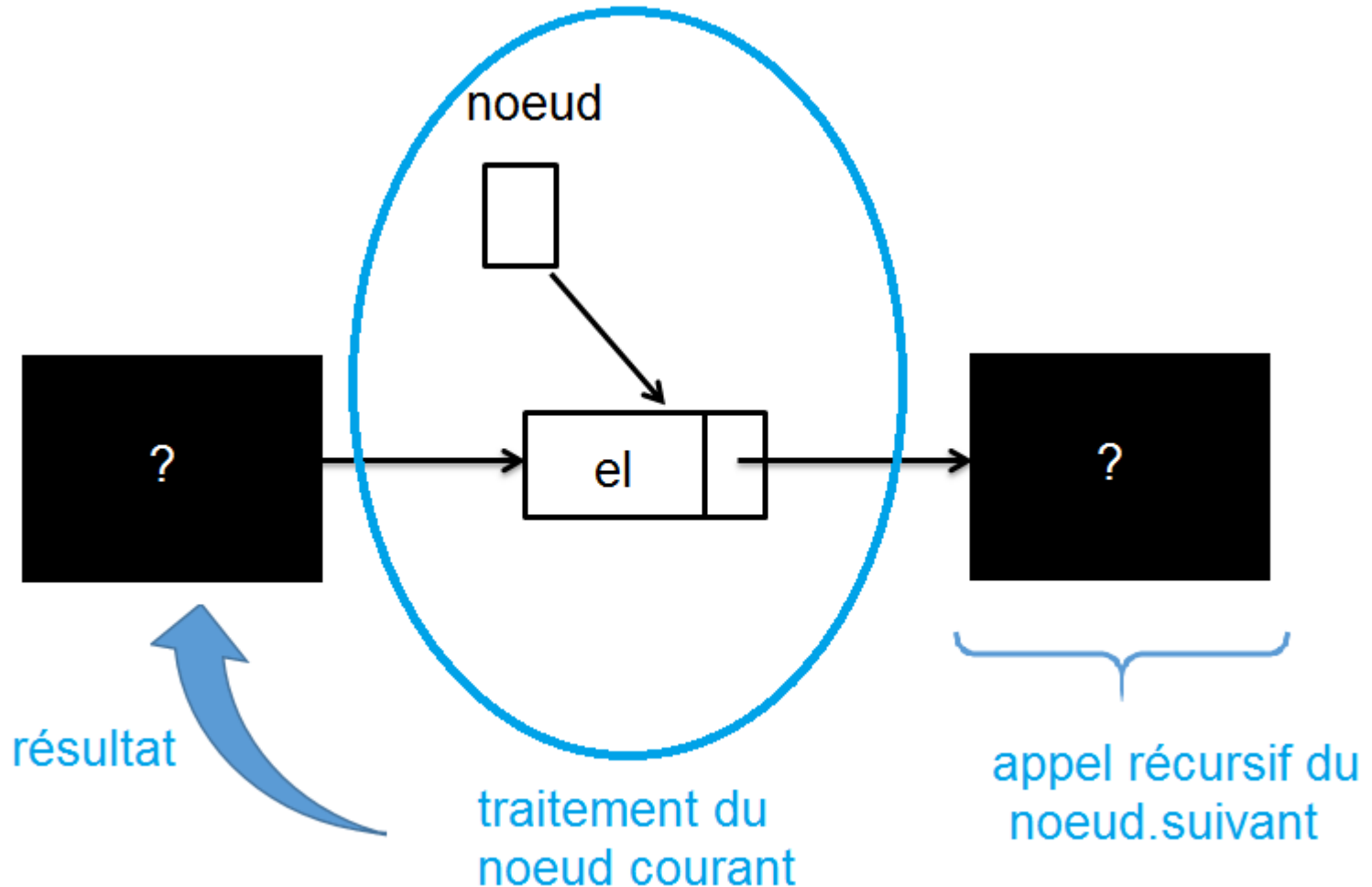
# Implémentation de la liste via une structure chaînée :



Le parcours **itératif** d'une structure chaînée se fait via un « baladeur » :



# Le parcours **récuratif** d'une structure chaînée :

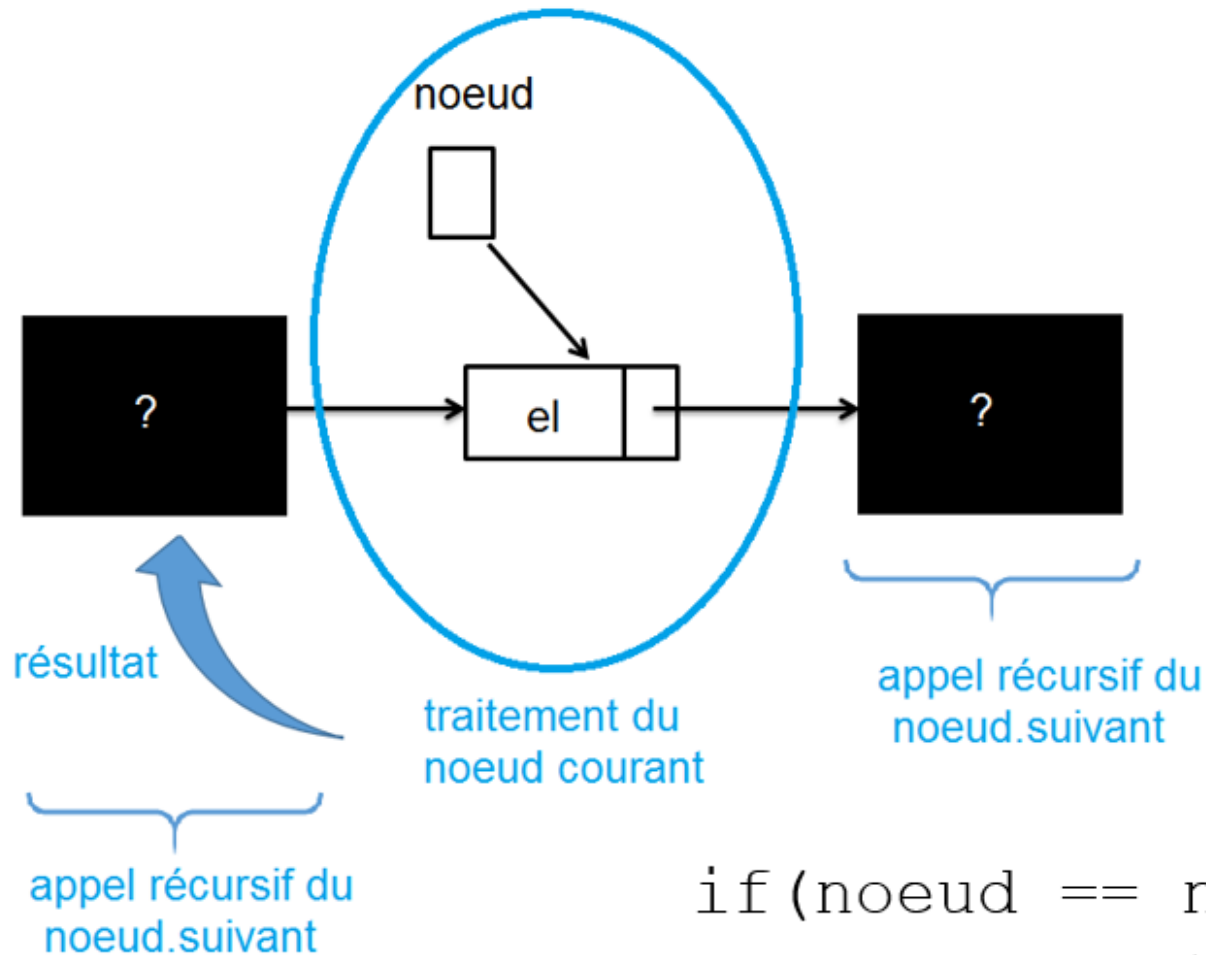


# Exemple 1 : somme()

```
public int somme() {  
    return somme(tete);  
}
```

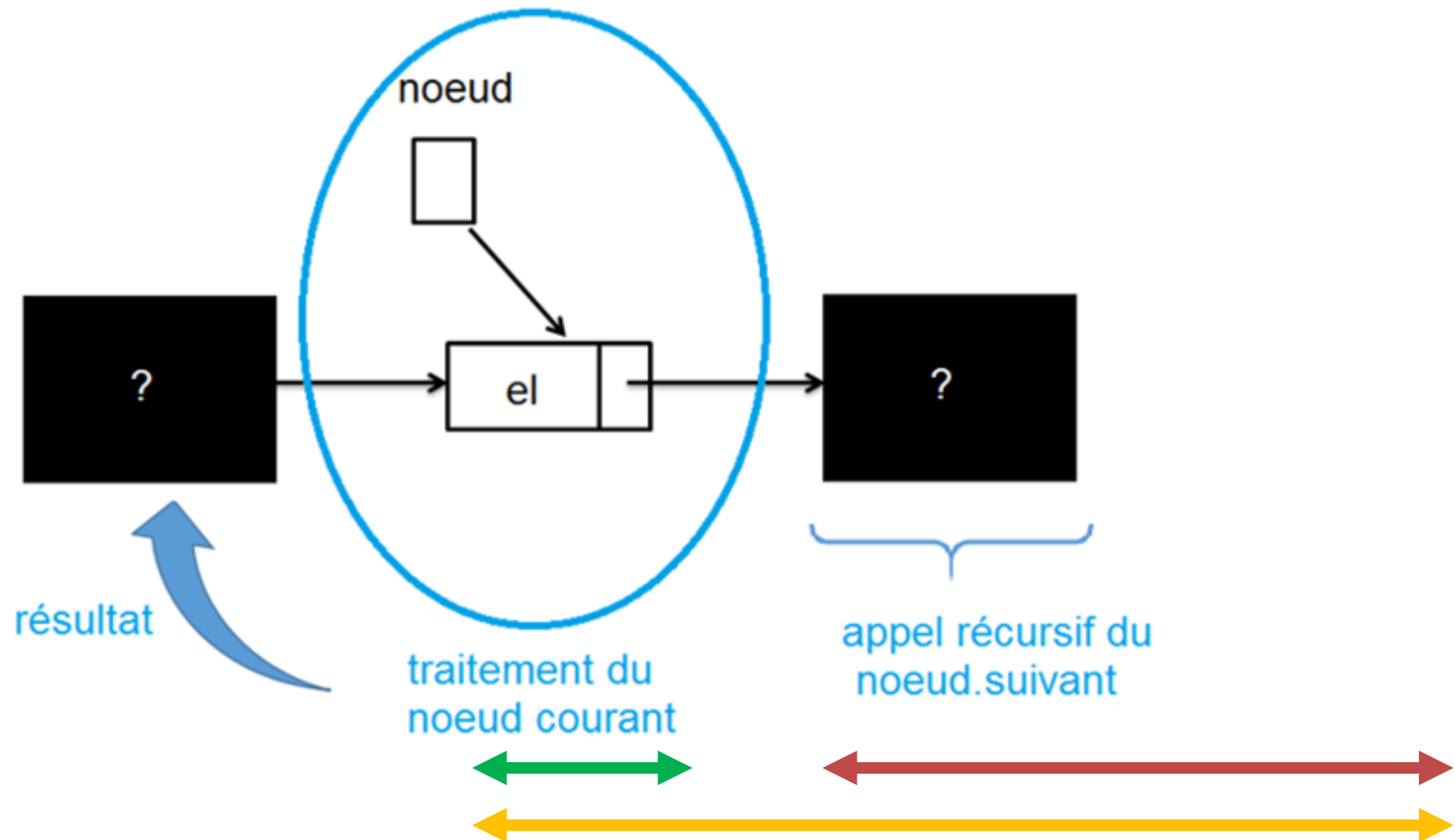
```
private int somme(Noeud noeud) {  
    if (noeud == null)  
        return 0;  
    return noeud.entier +  
        somme (noeud.suivant);  
}
```

# Exemple 1 : somme()



```
if (noeud == null)
    return 0;
```

# Exemple 1 : somme()



```
return noeud.entier +  
        somme (noeud.suivant) ;
```

## Exemple 2 : contient()

```
public boolean contient(int entier) {  
    return contient(tete, entier);  
}  
  
private boolean contient(Noeud noeud,  
                           int entier) {  
    if (noeud == null)  
        return false;  
    if (noeud.entier == entier)  
        return true;  
    return contient(noeud.suivant, entier);  
}
```



## Exemple 3 : toString()

```
public String toString() {  
    return "[" + toString(tete) + "];"  
}  
  
private String toString(Noeud noeud) {  
    if(noeud == null)  
        return "";  
    return " " + noeud.entier +  
        toString(noeud.suivant);  
}
```