

LISTE (LIST)

Une liste est une suite d'objets de même type, possédant un ordre bien précis, et dont le nombre est variable

(Une liste peut être vide!)

Interface ListSimple<E> extends Iterable<E>

int taille()

boolean estVide()

String toString()

Interface ListeSimple<E> extends Iterable<E>

```
boolean contient(E element)
```

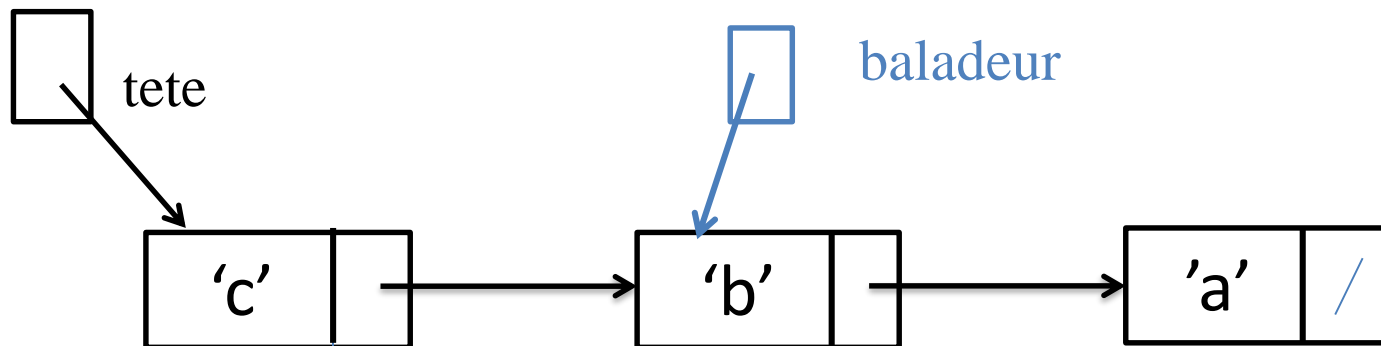
```
E premier()
```

```
void insererEnTete(E element)
```

```
boolean insererApres(E e, E eAInserer)
```

```
boolean supprimer(E element)
```

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



Exemple :

```
public String toString() {  
    String aRenvoyer="";  
    Noeud baladeur=tete;  
    while (baladeur!=null) {  
        aRenvoyer+=" " + baladeur.element;  
        baladeur=baladeur.suivant;  
    }  
    return aRenvoyer;  
}
```

ET JAVA?

ListeSimpleImpl<E>	LinkedList<E>
<code>int taille()</code>	<code>int size()</code>
<code>boolean estVide()</code>	<code>boolean isEmpty()</code>
<code>String toString()</code>	<code>String toString()</code>
<code>Iterator<E> iterator()</code>	<code>Iterator<E> iterator()</code>
<code>boolean contient(E element)</code>	<code>boolean contains(Object o)</code>

ET JAVA?

ListeSimpleImpl<E>	LinkedList<E>
<code>E premier()</code>	<code>E getFirst()</code>
<code>void insererEnTete(E e)</code>	<code>void addFirst(E e)</code>
<code>boolean supprimer(E e)</code>	<code>boolean remove(Object o)</code>
<code>boolean insererApres(E e, E eAInserer)</code>	<code>int indexOf(Object o)</code> <code>void add(int index, E e)</code>