



# Classe ArrayList

Java possède plusieurs classes faisant partie des **collections** permettant de stocker des objets et de proposer des méthodes pour trier, Interdire les doublons,....

La classe **ArrayList** fait partie des collections , elle permet de créer des objets pour stocker **une quantité variable et évolutive d'objets**.



# Classe ArrayList

## Création

- ArrayList non spécialisé, contient des références d'objets de toute nature

```
ArrayList list = new ArrayList(); //
```

- ArrayList spécialisé , ne peut contenir que des références d'objets instances de la même classe.

```
ArrayList <String> otherList= new ArrayList () ;
```



# Classe ArrayList

## Utilisation

```
String chaine1 ="Salut";  
ArrayList <String> list = new ArrayList () ;
```

```
list.add("Hello"); // ajout d'une référence d'objet  
list.add(chaine1);  
list.add(1 , "Hola"); /*Ajout d'une référence d'objet à la position 1  
Correspond à une insertion*/
```

```
int nb = list.size(); //retourne le nombre de références d'objets (3)
```

```
list.clear(); //suppression de tous les éléments
```

```
list.remove(1) // suppression de la référence d'objet à la position1
```



# Classe ArrayList

## Accès aux éléments

```
ArrayList <String> list= new ArrayList () ;
```

```
list.add("Hello");  
list.add("salut");  
list.add("Hola");
```

```
String chaine1 = list.get(1); /*retourne la référence d'objets à la  
                             position1* (chaine1 = "salut")/
```

```
list.set(0,"Bonjour"); //remplace la référence de la position 0
```



# Classe ArrayList

## Parcourir un ArrayList

**Dans une boucle for :**

```
ArrayList <String>list= new ArrayList () ;
```

```
.....// Remplissage
```

```
    //Affichage  
for(int i = 0 ; i < list.size() ; i++){  
  
    System.out.println(list.get(i));  
  
}
```



# Classe ArrayList

## Parcourir un ArrayList

**Dans une boucle for each :**

```
ArrayList <String>list= new ArrayList () ;
```

```
.....// Remplissage
```

```
    //Affichage  
for(String s : list){  
    System.out.println(s);  
}
```



# Classe ArrayList

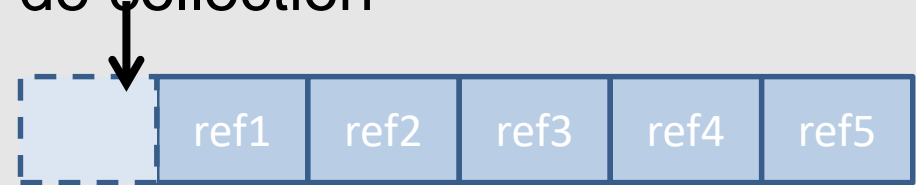
## Parcourir un ArrayList

Utilisation d'un Iterator:

Un Iterator est une Interface (classe particulière) qui permet d'instancier des objets spécialisés dans le parcours de collection

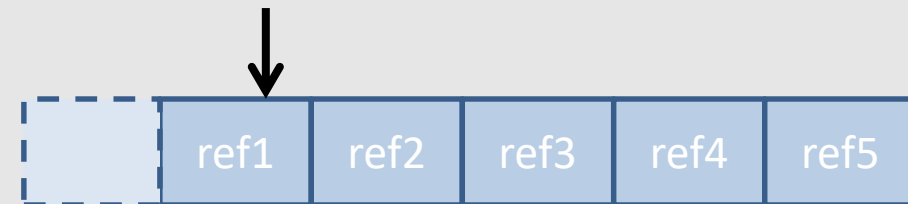
```
Iterator it = list.iterator();  
/*Création
```

Le curseur pointe avant la première référence/\*



```
boolean b = it.hasNext(); //retourne true s'il reste une référence à parcourir
```

```
monObjet = it.next(); /*retourne la référence  
suivante (ref1) et déplace le curseur*/
```





# Classe ArrayList

## Parcourir un ArrayList

**Avec Iterator:**

```
ArrayList <String> list= new ArrayList () ;
```

```
.....// Remplissage
```

```
    //Affichage  
    Iterator it = list.iterator();  
    while(it.hasNext() ){  
        System.out.println(it.next());  
    }
```