

# TABLEAUX VS LISTES

---

# Tableaux / Listes

- Choix entre tableaux et listes:
  - Tableau : taille fixe et multidimensionnel

=> utile pour des constantes ou données paramètres dont le nombre est fixé. Par exemple : jour de la semaine, tarifs

- Liste : taille variable

=> utile pour des listes d'informations variables : liste de produits, de clients ...

# Méthodes pour les tableaux

Méthodes statiques de classe Array rangées dans l'espace de nom System :

- int **IndexOf**( Array array, Object value ) : retourne la position de la 1ere occurrence de la valeur au sein du tableau
- int **LastIndexOf**( Array array, Object value ) : retourne l'index de la dernière occurrence de la valeur au sein du tableau
- void **Reverse** (Array) : inverse
- void **Sort** (Array) : trie

Si le type des éléments est comparable

```
String [ ] jours = new String [ ] { "Lundi", "Mardi", "Mercredi", ... "Dimanche" };  
int posMercredi = Array.IndexOf(jours, "Mercredi");  
Console.WriteLine("Mercredi c'est le jour " + (posMercredi+1));
```



```
Array.Sort(jours);
```

# Listes : List <T>


<T> signifie qu'on mettre tout type

- Initialisation lors de la déclaration : taille non obligatoire

```
List<String> prenom = new List<String> ();  
ou  
List<String> prenom = new List<String> (10);  
Ou  
List<String> prenom = new List<String>{ "Franck", "Aimerick", ... };
```

- Déclaration puis initialisation à l'aide de la méthode Add:

```
List<String> prenom = new List<String>(10); // création de la liste
```

```
  
prenom.Add("Franck");  
prenom.Add("Aimerick");  
prenom.Add("Elodie");
```

# Listes : list <T>

- Avec des variables structurées

```
List<Rectangle> lesCibles = new List< Rectangle >();  
// création de la liste
```

```
lesCibles.add( new Rectangle (0,0, 20,20) );  
lesCibles.add ( new Rectangle (60,60,40,40) );
```

# Parcours de liste

- Pour le lire : Ex: List<String> prenom;

```
foreach (String prenom in prenom)  
    Console.WriteLine(prenom);
```

```
for (int i = 0; i < prenom.Count; i++)  
    Console.WriteLine( prenom[i] );
```

```
List<Rectangle> lesCibles ; //...  
foreach (Rectangle r in lesCibles)  
    Console.WriteLine(r);
```

- Pour le modifier :

```
for (int i = 0; i < prenom.Count; i++)  
    prenom[i] = prenom[i] + "s";
```

Même syntaxe que  
pour un tableau !

# Méthodes pour les listes : list <T>

Méthodes pour les objets de classe List :

- int IndexOf( T value ) : retourne la position de la 1ere occurrence de la valeur au sein du tableau
- int LastIndexOf( T value ) : retourne l'index de la dernière occurrence de la valeur au sein du tableau
- void Sort () : trie
- void Insert( int index, T item )
- ...

```
List<String> prenom = new List<String>(); // création de la liste  
prenom.Add("Franck");  
prenom.Add("Aimerick");  
prenom.Add("Elodie");
```



**prenom.Sort**();