



Méthode et Fonction pour Array





Length

La propriété Length retourne le nombre d'éléments dans un tableau.

```
int[] myArray = new int[10];  
Console.WriteLine(myArray.Length); // Affiche 10
```



Rank

La propriété Rank retourne le nombre de dimension d'un tableau.

```
int[,] myArray = new int[5,5];  
Console.WriteLine(myArray.Rank); // Affiche 2
```



IndexOf()

La méthode IndexOf() recherche un élément dans un tableau et retourne l'index de la première occurrence de cet élément.

IndexOf(Array, Obj) tel que :

```
int[] myArray = {1, 2, 3, 4, 5, 6, 7, 8, 5};  
Console.WriteLine(myArray.IndexOf(5)); // Affiche 4
```



Clone()

La méthode Clone() créer une copie profonde d'un tableau.

Remarque :

Pour cloner un array, nous devons recaster l'array que l'on veut cloner lors de l'attribution de celui-ci a un nouvel array.

```
int[] myArray = {1, 2, 3, 4, 5, 6, 7, 8, 9};  
int[] myArrayClone = (int[])myArray.Clone(); // nous recaston myArray en (int[])  
// pour l'attribuer a myArrayClone  
Console.WriteLine(myArrayClone[0]); // Affiche 1
```



Copy()

La méthode Copy() copie un tableau dans un autre tableau.

Cette méthode s'appelle avec le type `Array` et non avec une variable.

Elle prend comme argument :

- 📌 Le tableau a copier
- 📌 Le tableau dans lequel on veut copier les elements
- 📌 Le nombre d'éléments a copier



```
int[] myArray = {1, 2, 3, 4, 5, 6, 7, 8, 9};  
int[] myArrayCopy = new int[5];  
Array.Copy(myArray, myArrayCopy, 5);  
for (int i = 0; i < myArrayCopy.Length; i++)  
{  
    Console.WriteLine(myArrayCopy[i]); // Affiche 12345  
}
```



Copy()

Copy peut fonctionner de différente manière.

Elle peut prendre un index de départ pour le tableau a copier,

Elle peut prendre un index de départ pour le tableau a remplir en plus du nombre d'elements a copier.

```
int[] myArray = {1, 2, 3, 4, 5, 6, 7, 8, 9};  
int[] myArrayCopy = new int[5];  
Array.Copy(myArray, 3, myArrayCopy, 0, 5);  
for (int i = 0; i < myArrayCopy.Length; i++)  
{  
    Console.WriteLine(myArrayCopy[i]); // Affiche 45678  
}
```




Clear()

La méthode `Clear()` remplace toutes les valeurs d'un tableau par une valeur par défaut.

Elle prend comme argument le tableau à effacer, l'index de départ, et le nombre d'éléments à effacer.

```
int[] myArray = {1, 2, 3, 4, 5, 6, 7, 8, 9};  
  
Array.Clear(myArray, 0, 1);  
  
for (int i = 0; i < myArray.Length; i++)  
{  
    Console.WriteLine(myArray[i]); // Affiche 023456789  
}
```



Find()

La méthode Find() recherche un élément dans un tableau et retourne la première occurrence de cet élément.

Elle prend comme paramètres l'array dans lequel on recherche l'élément et la condition, qui définira l'élément recherché.

```
int[] myArray = {1, 2, 3, 4, 5, 6, 7, 8, 9};  
  
int myInt = Array.Find(myArray, element => element > 5);  
  
Console.WriteLine(myInt); // Affiche 6
```



Sort()

La méthode Sort() trie les éléments d'un tableau.

```
int[] myArray = {9, 8, 6, 7, 5, 4, 3, 2, 1};  
  
Array.Sort(myArray);  
for (int i = 0; i < myArray.Length; i++)  
{  
    Console.WriteLine(myArray[i]); // Affiche 123456789  
}
```



Sort()

Il existe aussi une version de Sort qui prend en paramètres un tableau et un index de départ et un nombre d'elements a trier.

```
int[] myArray = {9, 3, 13, 12, 45, 6, 7, 8, 9};  
  
Array.Sort(myArray, 3, 3);  
for (int i = 0; i < myArray.Length; i++)  
{  
    Console.WriteLine(myArray[i]); // Affiche 9 3 13 6 12 45 7 8 9  
}
```