



Dictionnaire





Classe Dictionary

La classe `Dictionary` permet de créer un tableau de `clef / valeur`.
Les avantages sont :

- 📌 les méthodes de base a dispositions.
- 📌 Le faites que le "tableau" soit extensible.
- 📌 La recherche des éléments rapide.



Déclaration

Un dictionnaire se déclare avec 2 types;
Un type pour la **clef**, un pour la **valeur**.

```
Dictionary<int , string> dict = new Dictionary<string, int>(); // int --> clef , string --> valeur
```



Clef Valeur

Un dictionnaire associe une clef a une valeur.

Une clef est unique.

Pour rechercher une valeur, nous utiliserons sa clef comme index.

```
Dictionary<int, string> dictionnaire = new Dictionary<int, string>();  
dictionnaire.Add(1, "un");  
dictionnaire.Add(2, "deux");  
dictionnaire.Add(3, "trois");
```

```
Console.WriteLine(dictionnaire[1]); // affichera un  
Console.WriteLine(dictionnaire[2]); // affichera deux  
Console.WriteLine(dictionnaire[3]); // affichera trois
```



Fonction recherche

La fonction `ContainsKey` permet de rechercher si une clef existe dans le dictionnaire.

```
Dictionary<int, string> dictionnaire = new Dictionary<int, string>();  
dictionnaire.Add(1, "un");  
dictionnaire.Add(2, "deux");  
dictionnaire.Add(3, "trois");  
  
if (dictionnaire.ContainsKey(1)) {  
    Console.WriteLine("La clef 1 existe dans le dictionnaire");  
}
```



Méthode Dictionary



Add : permet d'ajouter une clef / valeur

```
Dictionary<string, int> dict = new Dictionary<string, int>();

// Ajoute une paire clé-valeur au dictionnaire
dict.Add("un", 1);
dict.Add("deux", 2);
dict.Add("trois", 3);

// Ajoute une paire clé-valeur au dictionnaire à l'aide d'un objet KeyValuePair
KeyValuePair<string, int> kvp = new KeyValuePair<string, int>("quatre", 4);
dict.Add(kvp);

// Ajoute une paire clé-valeur au dictionnaire avec une clé existante (lèvera une exception)
dict.Add("un", 10);
```



Remove : permet de supprimer une clef / valeur

```
using System.Collections.Generic;

ConsoleApplication.Program.Main();
namespace ConsoleApplication
{
    class Program
    {
        public static void Main()
        {
            // Création d'un dictionnaire qui associe des chaînes de caractères (les clés) à des entiers (les valeurs)
            Dictionary<string, int> scores = new Dictionary<string, int>();

            // Ajout de quelques éléments au dictionnaire
            scores.Add("Alice", 10);
            scores.Add("Bob", 20);
            scores.Add("Charlie", 30);
            scores.Add("Dave", 40);

            // Suppression de l'élément associé à la clé "Bob"
            scores.Remove("Bob");

            // Vérification que l'élément a bien été supprimé
            if (scores.ContainsKey("Bob"))
            {
                // L'élément associé à la clé "Bob" existe toujours
                Console.WriteLine("Bob's score is still in the dictionary!");
            }
            else
            {
                // L'élément associé à la clé "Bob" a bien été supprimé
                Console.WriteLine("Bob's score has been removed from the dictionary.");
            }
        }
    }
}
```



Clear : permet de vider le dictionnaire

```
public static void Main()
{
    // Création d'un dictionnaire qui associe des chaînes de caractères (les clés) à des entiers (les valeurs)
    Dictionary<string, int> wordLengths = new Dictionary<string, int>();

    // Ajout de quelques mots et de leur longueur dans le dictionary
    wordLengths.Add("chat", 4);
    wordLengths.Add("chien", 5);
    wordLengths.Add("oiseau", 6);
    wordLengths.Add("loup", 4);

    // Affichage du contenu du dictionary avant l'utilisation de la méthode Clear
    Console.WriteLine("Contenu du dictionary avant Clear:");
    foreach (KeyValuePair<string, int> kvp in wordLengths)
    {
        Console.WriteLine($"Mot: {kvp.Key}, Longueur: {kvp.Value}");
    }

    // Utilisation de la méthode Clear pour supprimer tous les éléments du dictionary
    wordLengths.Clear();

    // Affichage du contenu du dictionary après l'utilisation de la méthode Clear
    Console.WriteLine("\nContenu du dictionary après Clear:");
    foreach (KeyValuePair<string, int> kvp in wordLengths)
    {
        Console.WriteLine($"Mot: {kvp.Key}, Longueur: {kvp.Value}");
    }
}
```




ContainsKey : permet de vérifier si une clef existe

```
Dictionary<string, int> ages = new Dictionary<string, int>();
ages.Add("Alice", 25);
ages.Add("Bob", 30);
ages.Add("Charlie", 35);

// Vérification de la présence d'une clé
bool aliceExists = ages.ContainsKey("Alice"); // aliceExists vaudra true
bool daveExists = ages.ContainsKey("Dave"); // daveExists vaudra false

// Utilisation de la méthode ContainsKey dans une condition
if (ages.ContainsKey("Alice"))
{
    Console.WriteLine("Alice a " + ages["Alice"] + " ans.");
}
else
{
    Console.WriteLine("Alice n'est pas présente dans le dictionnaire.");
}
```



ContainsValue : permet de vérifier si une valeur existe

```
Dictionary<string, double> fruitPrices = new Dictionary<string, double>();
fruitPrices.Add("pomme", 0.99);
fruitPrices.Add("banane", 0.59);
fruitPrices.Add("mangue", 1.29);

// Vérification si le dictionnaire contient la valeur 0.99
if (fruitPrices.ContainsValue(0.99))
{
    Console.WriteLine("Le dictionnaire contient une valeur de 0.99.");
}
else
{
    Console.WriteLine("Le dictionnaire ne contient pas de valeur de 0.99.");
}

// Vérification si le dictionnaire contient la valeur 1.50
if (fruitPrices.ContainsValue(1.50))
{
    Console.WriteLine("Le dictionnaire contient une valeur de 1.50.");
}
else
{
    Console.WriteLine("Le dictionnaire ne contient pas de valeur de 1.50.");
}
```



Count : permet de connaître le nombre d'éléments du dictionnaire

```
Dictionary<string, int> ages = new Dictionary<string, int>();  
ages.Add("Alice", 25);  
ages.Add("Bob", 30);  
ages.Add("Charlie", 35);  
  
// Affichage du nombre d'éléments du dictionnaire  
Console.WriteLine("Le dictionnaire contient " + ages.Count + " éléments.");
```