

Fonction Filter

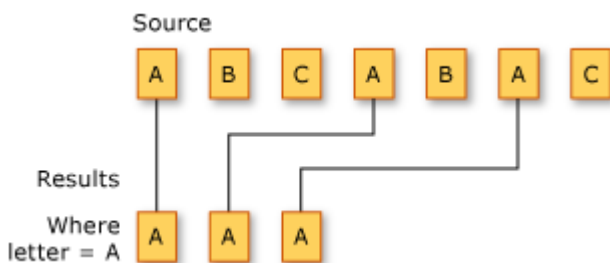
La fonction filter est un concept important en programmation qui permet de sélectionner des éléments d'une collection en fonction d'une condition spécifique.

Définition

La fonction filter est une fonction qui prend deux arguments :

- Une collection (par exemple, une liste, un tableau, un ensemble, etc.)
- Une condition (par exemple, une fonction, un lambda, un expression, etc.)

La fonction filter retourne une nouvelle collection qui contient les éléments de la collection originale qui répondent à la condition.



Exemples basiques

Exemple 1 : Filtrage d'une liste de nombres

```
List<int> numbers = new List<int> { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
List<int> evenNumbers = numbers.Where(x => x % 2 == 0).ToList();
```

Dans cet exemple, la fonction filter est utilisée pour récupérer les nombres pairs d'une liste de nombres.

Exemple 2 : Filtrage d'une liste de personnes

```
List<Person> people = new List<Person> {
    new Person { Name = "John", Age = 25 },
    new Person { Name = "Mary", Age = 30 },
    new Person { Name = "Jane", Age = 20 },
    new Person { Name = "Bob", Age = 35 }
};

List<Person> adults = people.Where(p => p.Age >= 18).ToList();
```

Dans cet exemple, la fonction filter est utilisée pour récupérer les personnes adultes d'une liste de personnes.

Exemple 3 : Filtrage d'une liste de chaîne de caractères

```
List<string> words = new List<string> { "hello", "world", "abc", "def", "ghi" };  
  
List<string> wordsStartingWithA = words.Where(w => w.StartsWith("a")).ToList();
```

Dans cet exemple, la fonction `filter` est utilisée pour récupérer les mots qui commencent par "a" d'une liste de mots.

Filtrages avancés

Au cas où cela ne coulerait pas de source avec ce qui a été présenté sur les fonctions d'ordre supérieur, il est intéressant de noter que les filtres peuvent être combinés soit directement avec des opérateurs de LINQ ou alors avec n'importe quelle expression du langage C# avec une fonction dédiée.

3 différentes manières de filtrer avec plusieurs critères

```
numbers.Where(n => n > 2).Where(n => n % 2 == 0); // 2 where chaînés en lambda  
  
numbers.Where(n => n > 2 && n % 2 == 0); //condition avec && logique en lambda  
  
//version avec une fonction complète  
numbers.Where(n => {  
    return n > 2 && n % 2 == 0;  
});
```

Avantages et inconvénients

La fonction `filter` offre plusieurs avantages, notamment :

- **Flexibilité** : La fonction `filter` permet de créer des filtres complexes en utilisant des conditions logiques et des fonctions.
- **Réutilisation** : La fonction `filter` peut être utilisée avec différentes collections et conditions.
- **Simplification** : La fonction `filter` peut simplifier le code en permettant de regrouper des opérations complexes en une seule fonction.

En contrepartie, elle a l'inconvénient de pouvoir être plus difficile à relire lorsqu'on revient à froid sur le projet...

Comparaison avec le javascript

En javascript, le concept de *filter* s'appelle même du même nom...

Voici, à titre d'exemple, quelques snippets en javascript.

Exemple 1 : Filtrage d'un tableau de nombres

```
const numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];
const evenNumbers = numbers.filter(x => x % 2 === 0);
console.log(evenNumbers); // [2, 4, 6, 8, 10]
```

Dans cet exemple, la fonction filter est utilisée pour récupérer les nombres pairs d'un tableau de nombres.

Exemple 2 : Filtrage d'un tableau de personnes

```
const people = [
  { name: "John", age: 25 },
  { name: "Mary", age: 30 },
  { name: "Jane", age: 20 },
  { name: "Bob", age: 35 }
];

const adults = people.filter(p => p.age >= 18);
console.log(adults); // [{ name: "John", age: 25 }, { name: "Mary", age: 30 }, {
name: "Bob", age: 35 }]
```

Dans cet exemple, la fonction filter est utilisée pour récupérer les personnes adultes d'un tableau de personnes.

Exemple 3 : Filtrage d'un tableau de chaîne de caractères

```
const words = ["hello", "world", "abc", "def", "ghi"];

const wordsStartingWithA = words.filter(w => w.startsWith("a"));
console.log(wordsStartingWithA); // ["abc"]
```

Conclusion

La fonction filter est un concept important en programmation qui permet de sélectionner des éléments d'une collection en fonction d'une condition spécifique. Les exemples ci-dessus montrent comment utiliser la fonction filter en C# (et en javascript) pour filtrer des données.

Source

Inspiré de

- <https://learn.microsoft.com/en-us/dotnet/csharp/linq/standard-query-operators/filtering-data>
- [groq ai inference](#) avec le prompt suivant :

théorie sur la fonction filter de manière générale et exemples en C# pour programmeurs avancés et en 800 mots maximum

