# Instructions, expressions et opérateurs (guide de programmation C#)

Le code C# qui comprend une application se compose d’instructions à base de mots clés, d’expressions et d’opérateurs. Cette section contient des informations concernant ces éléments fondamentaux d’un programme C#.

# Instructions (Guide de programmation C#)

L’ordre dans lequel les instructions sont exécutées dans un programme est appelé le flux de contrôle ou le flux d’exécution. Le flux de contrôle peut varier chaque fois qu’un programme est exécuté, selon la manière dont le programme réagit à l’entrée qu’il reçoit au moment de l’exécution.

# Expressions (Guide de programmation C#)

**Présentation C#**

La conception logicielle moderne s’appuie de plus en plus sur les composants logiciels sous la forme de ***packages*** de fonctionnalités autonomes et auto descriptifs.

Point important, ces composants présentent un modèle de programmation avec propriétés, méthodes et événements ; ils ont des attributs qui fournissent des informations déclaratives sur le composant ; et ils intègrent leur propre documentation. C# fournit des constructions de langage qui prennent directement en charge ces concepts.

Plusieurs fonctionnalités de C# participent à la construction d’applications robustes et fiables : le ***garbage collection*** récupère automatiquement la mémoire occupée par les objets inutilisés et inaccessibles

**Instructions, expressions et opérateurs (guide de programmation C#)**

Le code C# qui comprend une application se compose d’instructions à base de mots clés, d’expressions et d’opérateurs. Cette section contient des informations concernant ces éléments fondamentaux d’un programme C#.

**Lexique**

**Package** : Ensemble de logiciels munis d'une documentation, conçus pour répondre à des besoins spécifiques et permettre une utilisation autonome.

**Garbage collection :** gère l’allocation et la libération de mémoire pour votre application. Chaque fois que vous créez un objet, le Common Langage Runtime alloue de la mémoire pour l’objet à partir du tas managé. Aussi longtemps que de l'espace d'adressage est disponible dans le tas managé, le Runtime continue à allouer de l'espace pour de nouveaux objets.

**Programmation de type objet :** En [programmation orientée objet](https://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_objet), un objet est créé à partir d'un modèle appelé [classe](https://fr.wikipedia.org/wiki/Classe_(informatique)) ou [prototype](https://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_prototype), dont il hérite les comportements et les caractéristiques. Les comportements et les caractéristiques sont typiquement basés sur celles propres aux choses qui ont inspiré l'objet : une personne (avec son [état civil](https://fr.wikipedia.org/wiki/%C3%89tat_civil)), un dossier, un produit.