

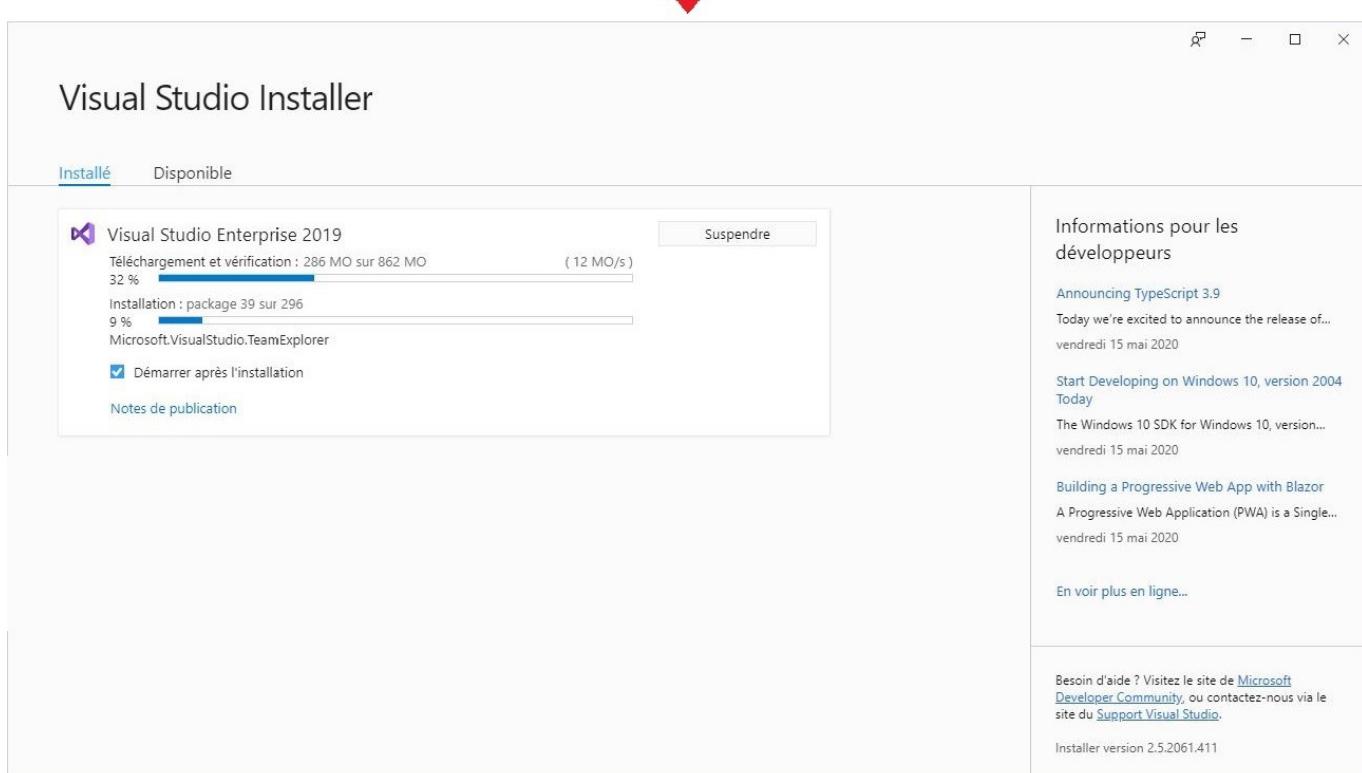
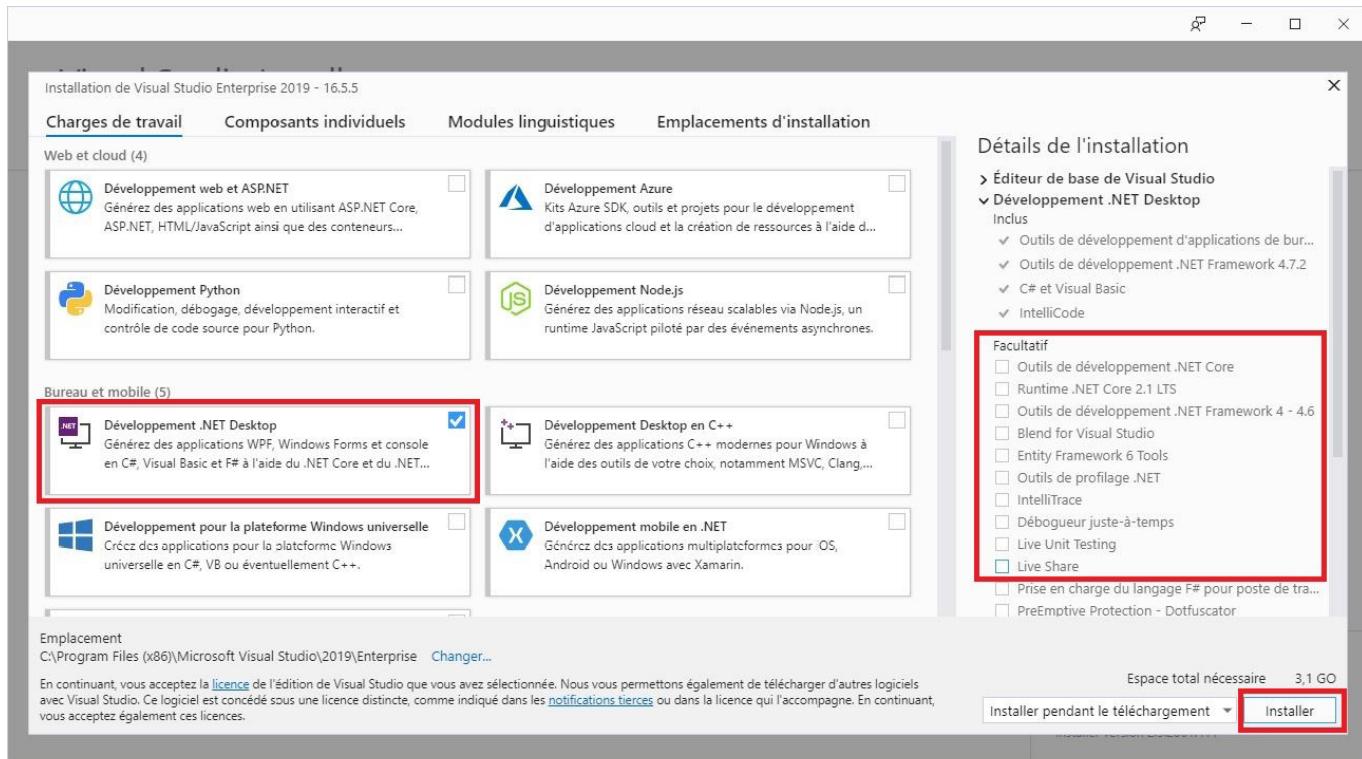
# Fiche savoirs technologiques : guide d'installation de Visual Studio 2022 Entreprise

## Téléchargement de Visual Studio 2022 Entreprise

- accédez au portal Azure et authentifiez-vous avec votre compte du lycée ;
- Recherchez le service **Education** puis **Logiciels** ;
- Recherchez le logiciel **Visual Studio Entreprise 2022** ;
- Cliquez sur **Afficher la clé** et notez la clé (vous en aurez besoin pour enregistrer le produit) ;
- Cliquez sur **Télécharger**.

## Installation de Visual Studio 2022 Entreprise

- Double cliquez sur le fichier téléchargé pour lancer l'installation ;
- Dans la première fenêtre, sélectionnez juste **Développement .NET Desktop** ;
- **Désélectionnez** tout ce qui est **facultatif**, à droite ;
- Cliquez sur **Installer**.
- Dans la seconde fenêtre, attendez la fin de l'installation ;
- En fin d'installation, il est demandé de redémarrer l'ordinateur : cliquez sur **Redémarrer**.



## Création d'un projet

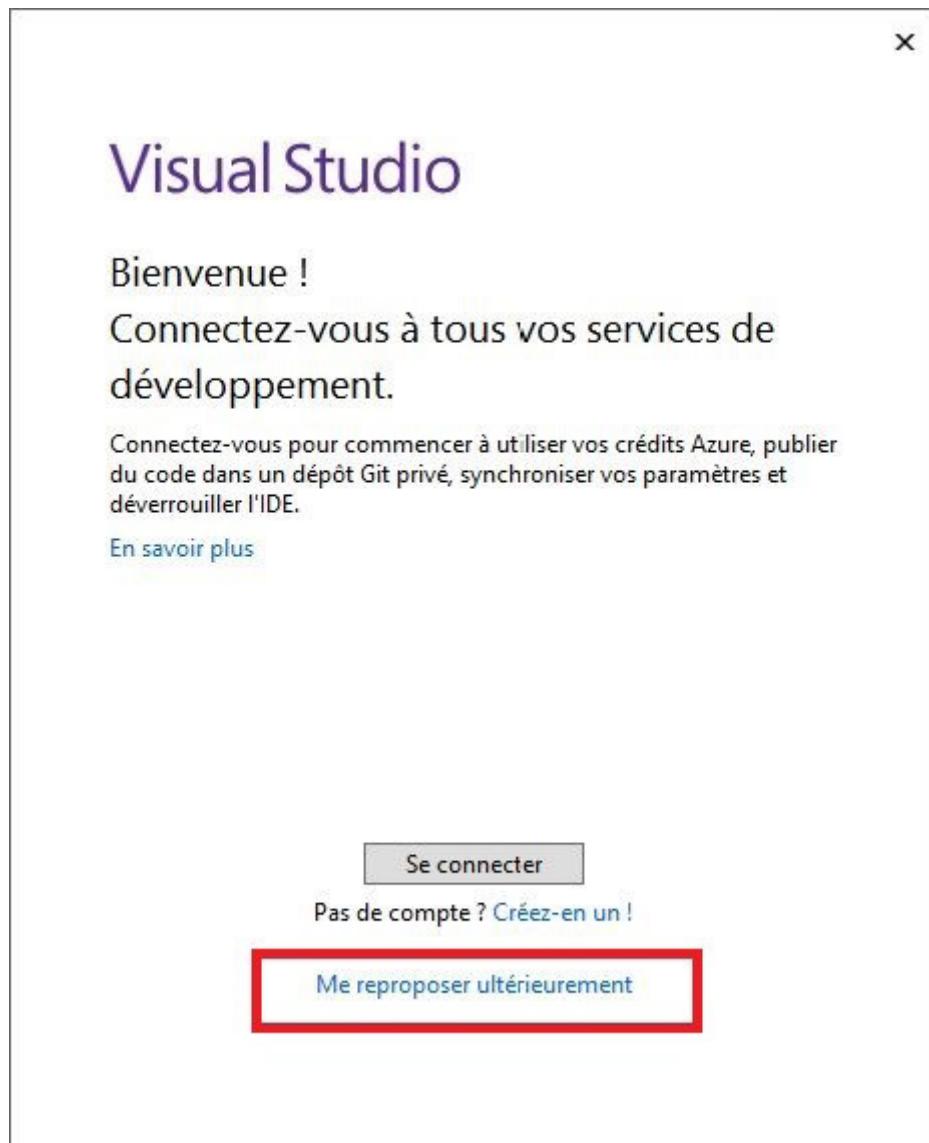
- Lancez Visual Studio 2022 Entreprise et suivez la démarche ci-dessous.

### Étape 1 : écran de bienvenue

Lorsque la première fenêtre de bienvenue apparaît **ne vous connectez pas** pour l'instant et cliquez directement sur **Me proposer ultérieurement**.

Plus loin, vous verrez comment enregistrer le produit afin de ne plus avoir cette demande et surtout pour ne pas être limité

dans l'utilisation de Visual Studio. En effet, Visual Studio est un produit commercialisé que vous pouvez utiliser grâce à l'abonnement souscrit par le lycée.



[Microsoft Learn : Écrire vos premières lignes de code en C#](#)

## Étape 2 : configuration de l'environnement

La nouvelle fenêtre permet de paramétriser l'environnement.

- Au niveau du combo **Paramètres de développement**, sélectionnez **C#** puisque tous les projets seront faits dans ce langage.

Il vous est possible de choisir l'aspect graphique de la fenêtre. Vous pourrez changer cette configuration par la suite.

- Cliquez ensuite sur **Démarrer Visual Studio**.



**Étape 3 : ouverture ou création d'un projet** À cette étape, il est possible, entre autres, d'ouvrir un projet existant, d'ouvrir un dossier ou bien sûr de créer un nouveau projet.

- Cliquez sur **Créer un nouveau projet**.

X

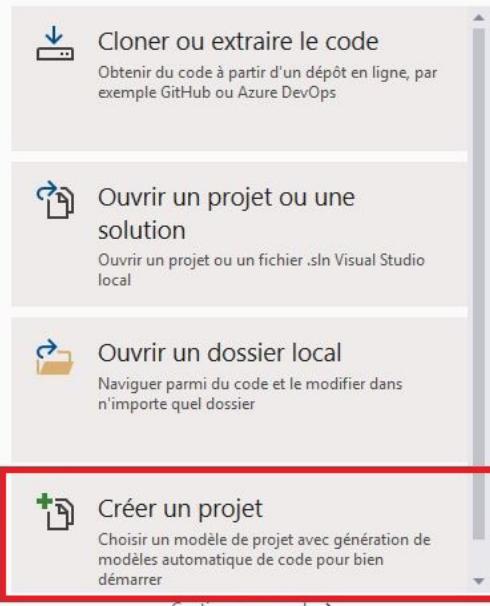
# Visual Studio 2019

## Ouvrir les éléments récents

Quand vous utilisez Visual Studio, tous les projets, dossiers ou fichiers que vous ouvrez s'affichent ici pour vous permettre d'y accéder rapidement.

Vous pouvez épinglez tout ce que vous ouvrez fréquemment pour l'afficher toujours en tête de liste.

## Démarrer



## Étape 4 : choix du type de projet

Il existe un grand nombre de types de projet.

Nous n'allons travailler qu'avec deux types de projets :

- d'abord les **applications console** en C#
- et, plus tard, les **applications Windows Forms** en C#.

Pour le moment ne faites aucune sélection et prenez connaissance de la liste des types de projet possibles.

## Créer un projet

### Modèles de projet récents

La liste des modèles récemment consultés sera affichée ici.

The screenshot shows the 'Create a new project' dialog. At the top, there are three dropdown menus: 'Tous les langages', 'Toutes les plateformes', and 'Tous les types de projet'. Below these are several project template cards:

- Bibliothèque de classes (.NET Standard)**: A library project targeting .NET Standard. It includes filters for C#, Android, iOS, Linux, macOS, Windows, and Bibliothèque.
- Bibliothèque de classes (.NET Standard)**: Another library project targeting .NET Standard. It includes filters for Visual Basic, Android, iOS, Linux, macOS, Windows, and Bibliothèque.
- Application Windows Forms (.NET Framework)**: A Windows Forms application. It includes filters for C#, Windows, and Bureau.
- Application WPF (.NET Framework)**: A Windows Presentation Foundation application. It includes filters for C#, Windows, and Bureau.
- WPF App (.NET Core)**: A Windows Presentation Foundation application targeting .NET Core. It includes filters for C#, Windows, and Bureau.

At the bottom right are 'Retour' and 'Suivant' buttons.

**Étape 5 : aide au choix du type de projet** Il est possible de sélectionner un type de projet directement dans la liste, mais il est aussi possible d'affiner la sélection en utilisant les 3 combos.

- Sélectionnez le **langage (C#)**,
- et le type de projet **Console**. Remarquez que maintenant, il n'y a plus qu'un élément dans la liste : **Application console (.NET Framework)**.

## Créer un projet

### Modèles de projet récents

La liste des modèles récemment consultés sera affichée ici.

The screenshot shows the 'Create a new project' dialog with three dropdown filters at the top: 'C#' (selected), 'Windows' (selected), and 'Console' (selected). Below these filters, the list of project templates is updated:

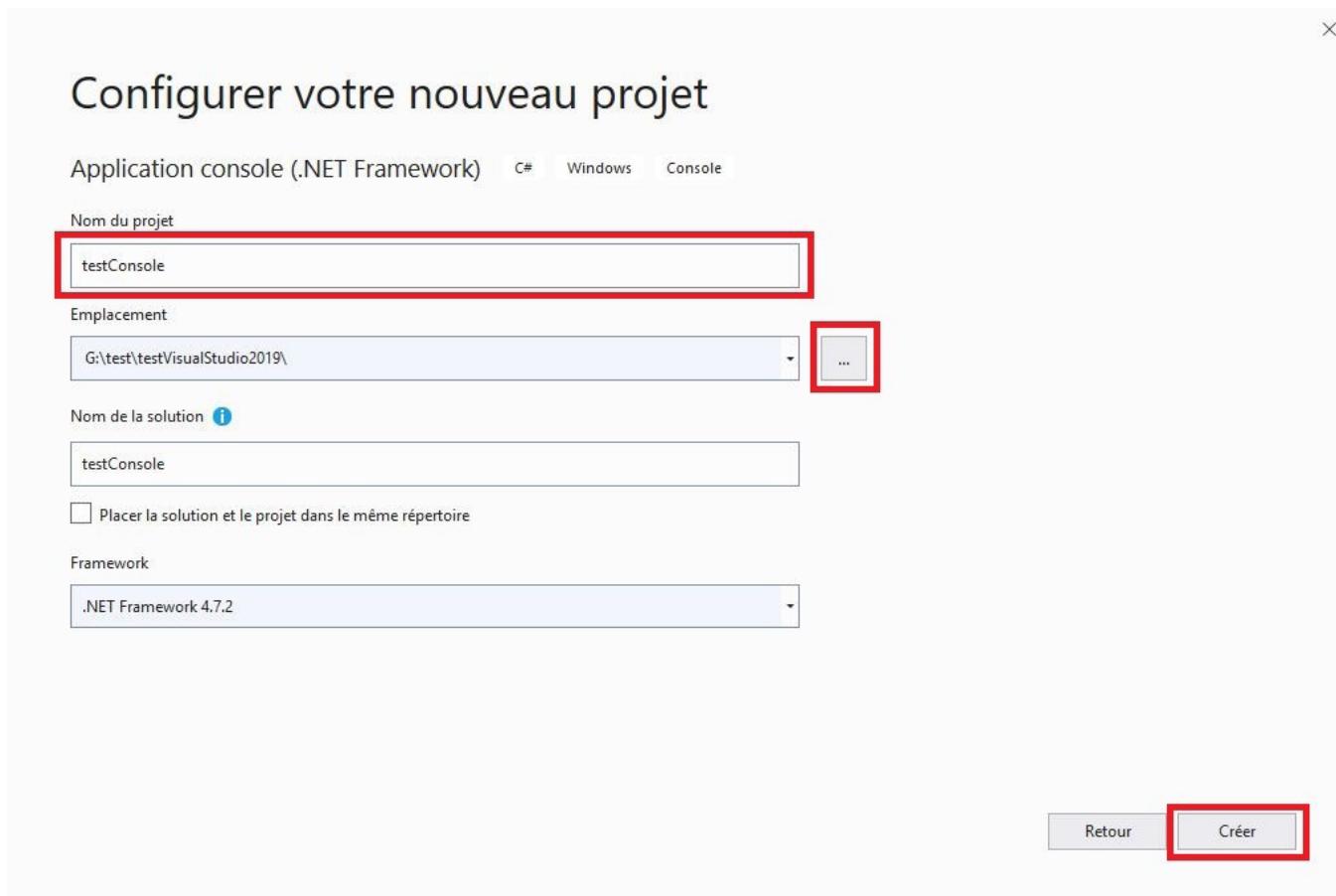
- Application console**: A command-line application targeting .NET Core. It includes filters for C#, Linux, macOS, Windows, and Console.
- Application console (.NET Framework)**: A command-line application targeting .NET Framework. It includes filters for C#, Windows, and Console.

A red box highlights the 'Application console (.NET Framework)' entry. At the bottom right are 'Retour' and 'Suivant' buttons.

Vous n'arrivez pas à trouver ce que vous cherchez ?  
[Installer plus d'outils et de fonctionnalités](#)

**Étape 6 : choix du nom et du chemin du projet** Il reste enfin l'étape où il faut donner un nom au projet et préciser son emplacement.

- Dans la zone de texte **Nom du projet**, tapez **testConsole** ;
- dans **Emplacement**, cliquez sur ... et sélectionnez un dossier, préalablement créé sur votre disque, dans lequel vous allez placer vos projets ;
- Ne modifiez pas les autres informations et cliquez sur **Créer**.



## Fenêtre initiale du projet

Après la création du projet, la fenêtre de travail apparaît.

### Contenu de la fenêtre

La fenêtre contient :

- \* une zone principale contenant le code du fichier **Program.cs** ;
- une zone à droite **Explorateur de solutions** contenant la liste des fichiers de la solution (on peut voir le fichier Program.cs) ;
- une zone à droite en bas **Propriétés** qui contiendra les propriétés des objets sélectionnés (pour le moment, on ne s'en occupera pas) ;
- en haut, les menus et icônes de raccourcis qui seront présentés au fur et à mesure des besoins.

Les différentes zones peuvent être déplacées, ouvertes, fermées, adaptées en taille.

### Notion de bloc de code

Une partie du code a déjà été automatiquement générée. Les différents éléments d'un projet seront expliqués en temps utile, mais pour le moment, sachez juste qu'un programme se découpe en blocs, repérables par un entête de bloc et des accolades (ouvrantes et fermantes) pour déterminer le contenu du bloc.

### Notion d'indentation

Chaque bloc peut contenir d'autres blocs : dans ce cas, vous pouvez remarquer des décalages vers la droite. Cela permet de mieux repérer qui contient quoi.

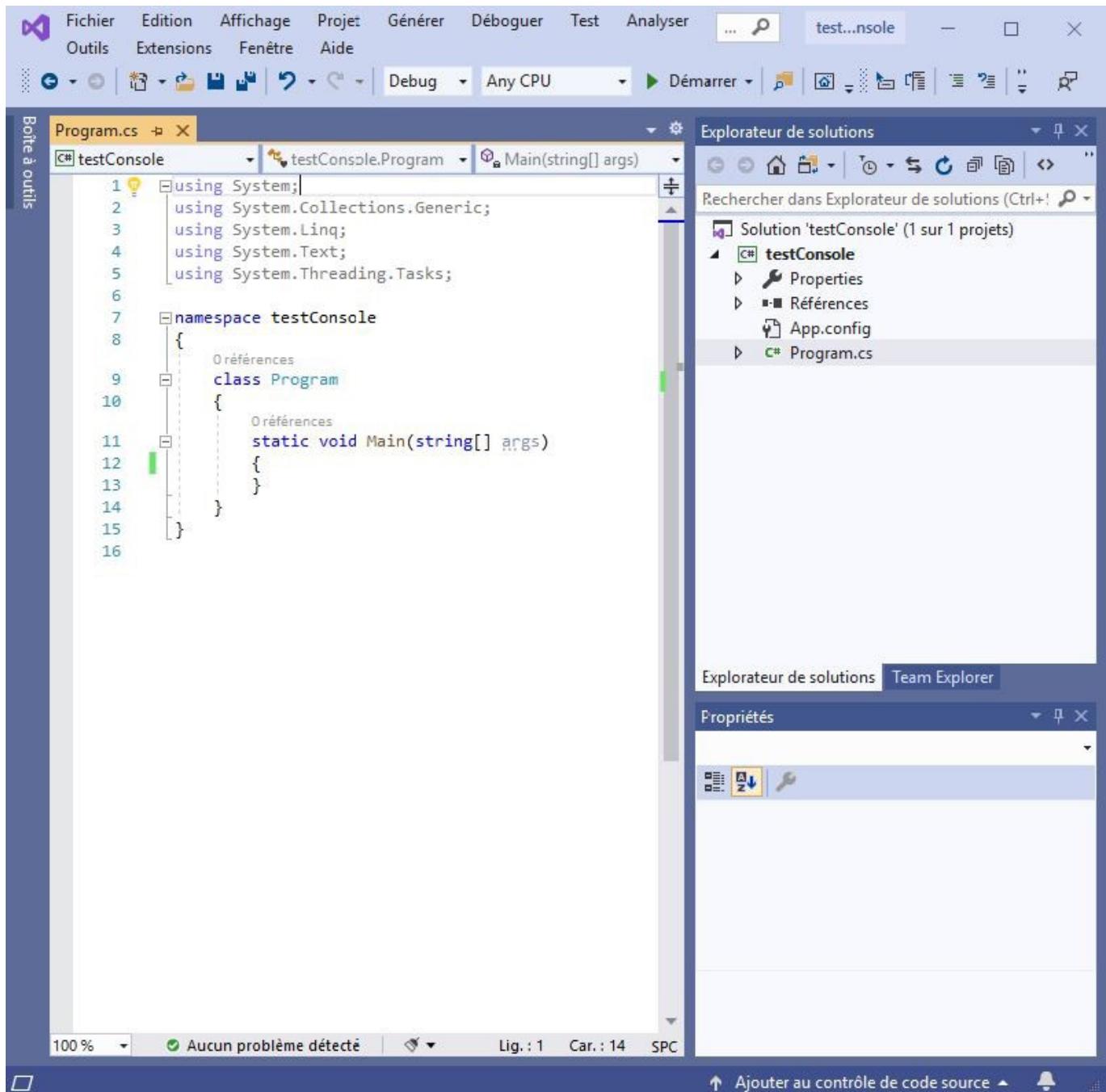
En programmation, ces décalages s'appellent aussi des **indentations**. Normalement l'IDE les gère automatiquement, mais il faudra

cependant surveiller qu'elles sont toujours correctes.

L'ordinateur ne fait pas attention aux indentations quand il exécute un programme (sauf pour certains langages comme Python), mais il est important de les respecter car sinon un programme devient vite illisible pour un programmeur.

## Où écrire votre code

À partir du code généré, le code que vous allez ajouter doit être écrit dans les accolades du bloc **static void Main(string[] args)**.



## Enregistrement du produit

Avant de poursuivre, il est conseillé d'enregistrer la clé de Visual Studio :

- Allez dans le menu **Aide > Incrire Visual Studio** ;
- Cliquez sur **Déverrouiller avec une clé de produit (Product Key)**,
- vous allez obtenir la petite fenêtre du milieu.
- Copiez la clé que vous avez précédemment récupérer sur Azure, au moment du téléchargement puis cliquez sur **Appliquer** ;
- Normalement, vous allez obtenir la fenêtre du bas, avec le message **Licence : Clé de produit appliquée** ;
- Cliquer sur **Fermer**.

## Connectez-vous à Visual Studio

Visual Studio vous permet de planifier vos projets, de collaborer avec votre équipe et de gérer votre code en ligne quel que soit l'emplacement où vous vous trouvez.

[En savoir plus](#)

Connectez-vous pour commencer à utiliser vos crédits Azure, publier du code dans un dépôt Git privé, synchroniser vos paramètres et déverrouiller l'IDE.

[Se connecter](#)

Pas de compte ? Créez-en un !

## Visual Studio

Enterprise 2019

Licence : Version d'essai 30 jours (pour évaluation seulement)  
La période d'évaluation a expiré.

⚠ Votre version d'essai a expiré. Connectez-vous pour prolonger la version d'essai ou obtenir une licence à l'aide d'un compte en ligne ou d'une clé de produit.

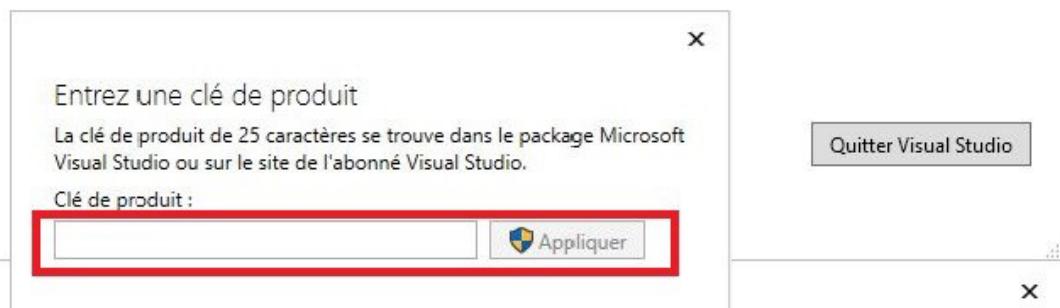
[Rechercher une licence mise à jour](#)

[Déverrouiller avec une clé de produit \(Product Key\)](#)

## Tous les comptes

[Se connecter...](#)

16.0.30104.148 D16.5



## Connectez-vous à Visual Studio

Visual Studio vous permet de planifier vos projets, de collaborer avec votre équipe et de gérer votre code en ligne quel que soit l'emplacement où vous vous trouvez.

[En savoir plus](#)

Connectez-vous pour commencer à utiliser vos crédits Azure, publier du code dans un dépôt Git privé, synchroniser vos paramètres et déverrouiller l'IDE.

[Se connecter](#)

Pas de compte ? Créez-en un !

## Visual Studio

Enterprise 2019

Licence : Clé de produit appliquée

## Tous les comptes

[Se connecter...](#)

16.0.30104.148 D16.5

[Fermer](#)

## Premier test

## Indentation

- Placez le curseur à droite de l'accolade ouvrante sous **static void Main(string[] args)** et validez.

Normalement une ligne a été insérée et le curseur s'est automatiquement décalé vers la droite pour respecter l'indentation (on est à l'intérieur d'un bloc).

## Autocomplétion

- Écrivez le code montré dans la capture ci-contre.

Remarquez qu'en tapant le code, l'éditeur propose au fur et à mesure différentes possibilités pour vous aider lors de la frappe : c'est **l'autocomplétion**.

```
0 références
class Program
{
    0 références
    static void Main(string[] args)
    {
        Console.WriteLine("premier test");
        Console.ReadLine();
    }
}
```

## Sensibilité à la casse

C# est sensible à la casse : il fait la distinction entre minuscule et majuscule. Exemple : **Console** est différent de **console**.

## Explication du code

`Console.WriteLine(...)` permet de demander un affichage.

`Console.ReadLine()` permet d'attendre une saisie au clavier (ici, c'est juste pour éviter que la fenêtre de commande se ferme juste après l'affichage).

Le ; en fin de ligne marque la fin d'une instruction. Chaque bloc est en effet composé d'autres blocs et/ou d'instructions.

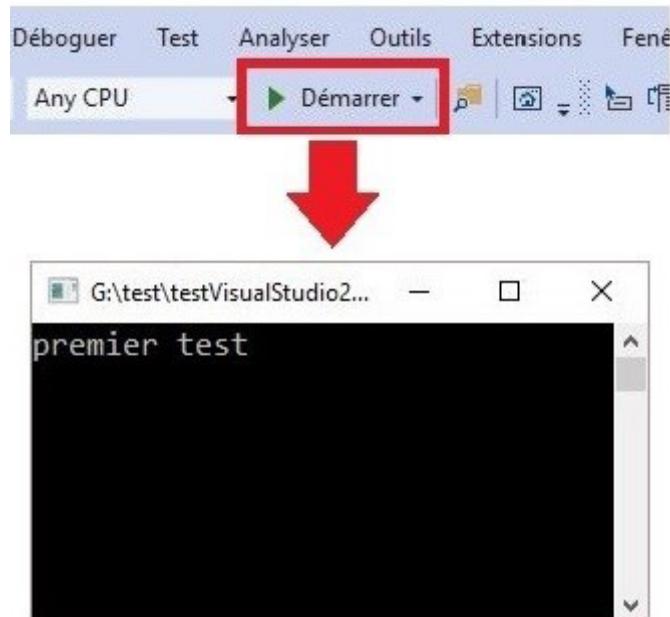
## Sauvegarde

Pensez à régulièrement enregistrer votre travail : menu **Fichier > Enregistrer (Program.cs, ou enregistrer tout)** ou plus rapidement en utilisant le raccourci clavier **CTRL + S**.

## Exécution

- Exécuter le programme en cliquant sur la flèche verte suivie du mot **Démarrer**.

Une fenêtre de commande doit s'ouvrir avec **premier test** affiché. Appuyez sur n'importe quelle touche, la fenêtre va se fermer.



## Erreur de compilation

Visual Studio est capable de repérer des erreurs faites lors de la saisie du code, lorsque celui-ci ne respecte pas la syntaxe du langage.

### Avertissement dans le code

Imaginons que vous ayez fait une erreur, en oubliant le point-virgule de la fin de la seconde ligne (enlevez-le). Vous remarquez qu'un petit soulignement rouge en forme de vague est apparu à l'endroit de l'erreur.

### Avertissement dans la liste des erreurs

De même, dans la fenêtre du bas, le message d'erreur apparaît ; **attendu**, en précisant le nom du fichier et du projet.

Si vous double-cliquez sur cette ligne d'erreur, le curseur se place automatiquement sur la ligne de code concernée.

### Avertissement lors de l'exécution

Sans corriger l'erreur, enregistrez et lancez l'exécution. Vous allez obtenir la fenêtre d'information comme ci-contre, vous avertissant de la présence d'erreurs.

Vous pouvez soit exécuter la version précédente qui n'avait pas d'erreur (avec **oui**) soit refuser d'exécuter (avec **non**).

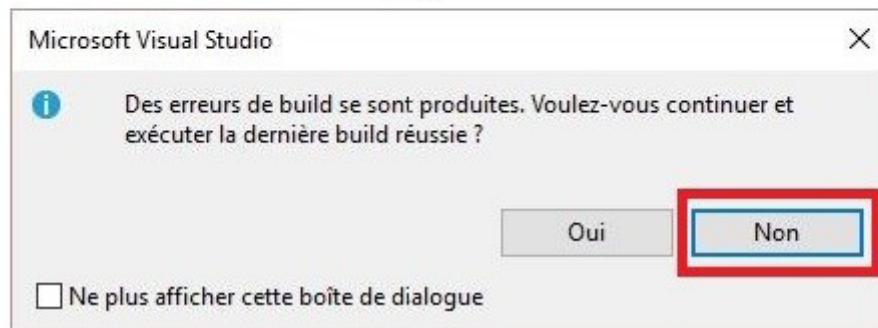
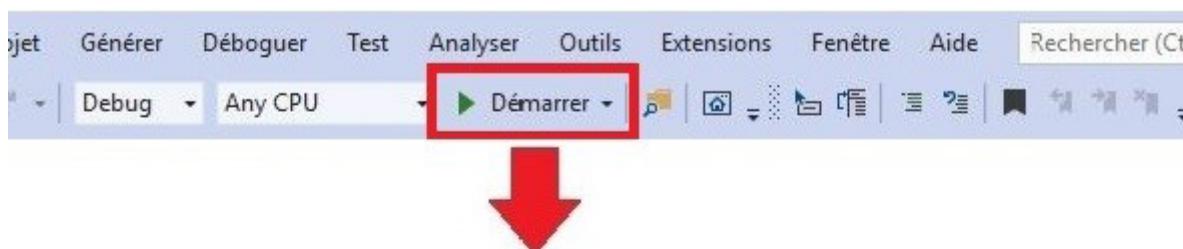
**Il faut toujours choisir Non** afin de corriger les erreurs avant de tester à nouveau. Remettez le point-virgule manquant, enregistrez et faites un nouveau test. Normalement tout doit fonctionner.

The screenshot shows the Visual Studio IDE. In the top pane, there is a code editor with the following C# code:

```
11  O références
12
13  static void Main(string[] args)
14  {
15      Console.WriteLine("premier test");
16      Console.ReadLine();
17  }
18
```

In the bottom pane, the 'Liste d'erreurs' (Error List) shows one error:

Solution complète	Code	Description	Projet	Fichier
		CS1002 ; attendu	testConsole	Program.cs



From:

[/- Les cours du BTS SIO](#)

Permanent link:

</doku.php/bloc1/guideinstallvisualstudio>

Last update: **2023/10/08 21:54**

