



IDE VSCode - Généralités

[Mise à jour le 13/6/23]

- **Sources**
 - Site du logiciel [lien](#)
 - Vidéo YouTube [Développer sous Visual Studio Code. Le meilleur IDE pour Arduino \(v2\)](#)
- **Lectures connexes**
 - Wiki [Arduino](#)
 - Wiki [Python et MicroPython](#)

1. Introduction

Visual Studio Code est un éditeur de code source **léger, mais puissant**, disponible pour Windows, macOS et Linux. Il est livré avec un support intégré pour JavaScript, TypeScript et Node.js et possède des extensions pour d'autres langages (tels qu' **Arduino**, **C++**, C#, Java, **Python**, PHP, Go etc.) et des runtimes (.NET et Unity) .

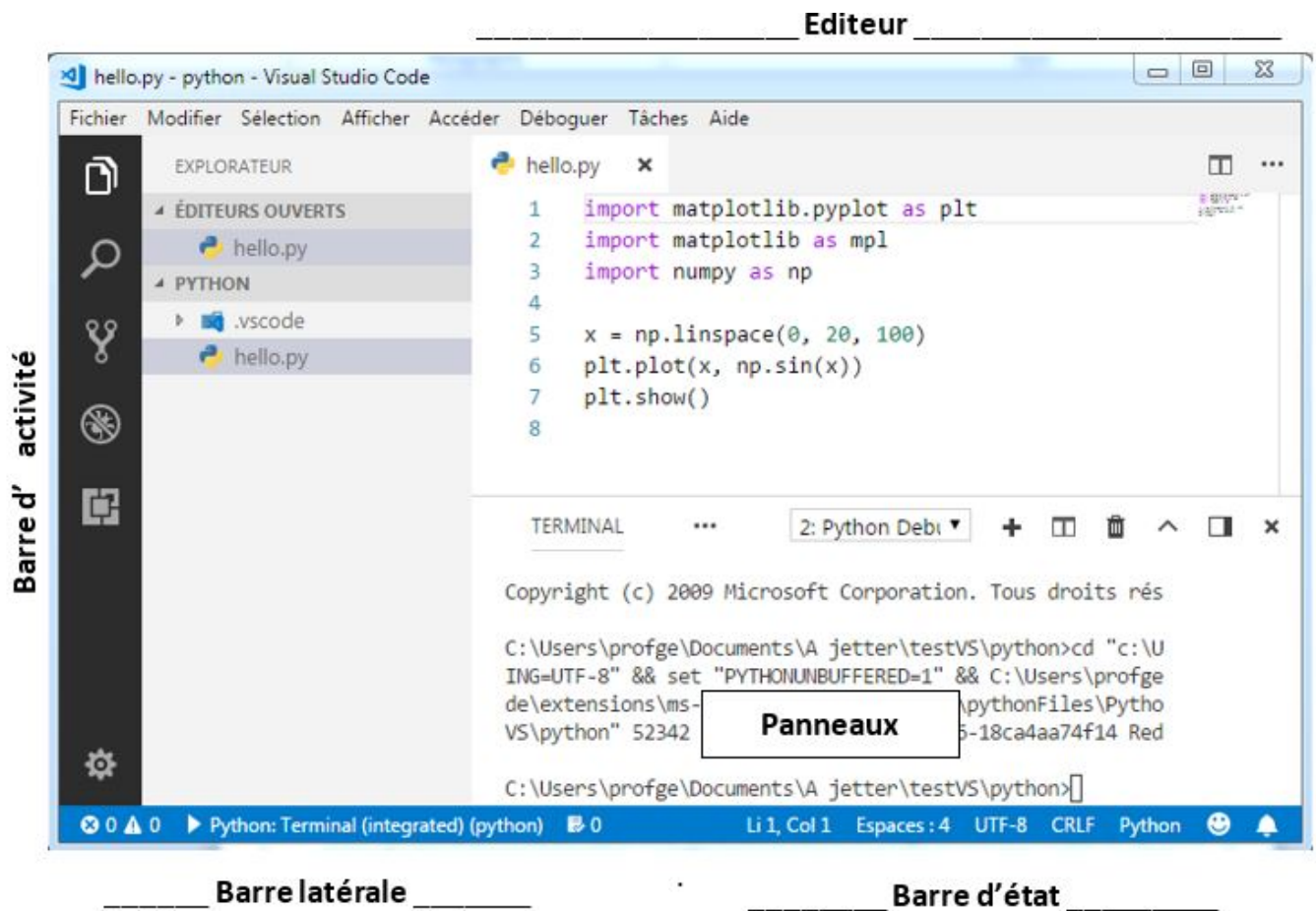


1.1 Caractéristiques

- **IntelliSense**¹⁾ pour CSS, HTML, JavaScript, JSON, Less, Sass, coloration syntaxique pour PHP.
- **Coloration syntaxique**²⁾
- **Code action** : proposition de correction d'erreur.
- **Emmet abréviations**³⁾ en HTML, Razor, CSS, Less, Sass, XML et Jade avec la touche de tabulation [Tab].
- **Débogage**⁴⁾ dans Nodejs.

1.2 L'interface utilisateur

Comme beaucoup d'autres éditeurs, VSCode présente un **explorateur** sur la gauche de son interface montrant tous les fichiers et dossiers auxquels vous avez accès, un **éditeur** sur la droite, montrant le contenu des fichiers que vous avez ouverts.



Cette interface est divisée en cinq zones :

- **La barre d'activité** (Activity Bar) est située à l'extrême gauche. Cette barre vous permet de passer d'une vue à l'autre et vous fournit des indicateurs contextuels supplémentaires, tels que le nombre de changements sortants lorsque Git est activé.
- **La barre latérale** (Side Bar) contient des vues différentes comme l'explorateur pour vous aider tout en travaillant sur votre projet.
- **L'éditeur** (Editor Groups) – C'est la zone principale pour éditer vos fichiers. Vous pouvez ouvrir jusqu'à trois éditeurs côte à côte.
- **Les panneaux** (Pannels) permettent d'afficher différents panneaux sous la zone de l'éditeur pour des informations de sortie ou de débogage, des erreurs et des avertissements, ou un terminal intégré. Le panneau peut également être déplacé vers la droite pour plus d'espace vertical.
- **La barre d'état** (Status Bar) donne des informations sur le projet ouvert et les fichiers que vous modifiez.

Chaque fois que vous chargez un projet dans VSCode, il apparaît dans le **même état** que lors de sa dernière fermeture. Le dossier, la disposition et les fichiers ouverts sont conservés.

2. Installation de l'IDE

Se rendre sur la [page](#) du site Visual Studio Code et **double-cliquer** sur le fichier téléchargé.
Des **extensions** ajoutent des fonctionnalités au logiciel.



3. Installation des extensions

3.1 Installations communes à tous les langages

Installer la liste des extensions ci-dessous en entrant leur nom dans la barre de recherche de



| Nom |
|---|
| C/C++ |
| French Language Pack for Visual Studio Code |
| Hex Editor |
| IntelliCode |
| Peacock |
| PrintCode |
| Todo Highlight |
| vsode-icons |
| WSL |

3.2 Extensions pour le Web

Installer la liste des extensions ci-dessous en entrant leur nom dans la
barre de recherche de



| Nom |
|---------------------|
| Lorem Ipsum |
| Markdown All in One |
| Open in browser |


Vous trouverez [ici](#) un **tutoriel** pour préparer VSCode à un projet de site web

3.3 Extension pour les microcontrôleurs (langage C)

La plateforme collaborative professionnelle **PlatformIO** est un environnement de développement intégré convivial et extensible avec un ensemble d'instruments de développement professionnel, offrant des fonctionnalités modernes et puissantes pour accélérer tout en simplifiant la création et la livraison de produits embarqués..



- Un éditeur de code source multiplateforme léger mais puissant.
- Complétions de code intelligentes basées sur des types de variables, des définitions de fonctions et des dépendances de bibliothèque.
- Flux de travail multiprojets avec une navigation facile autour de la base de code du projet, plusieurs volets et prise en charge des thèmes.
- Intégration transparente avec PlatformIO Home (UI) avec les gestionnaires de cartes et de bibliothèques.
- Assistant de projet intuitif et un large éventail d'exemples de projets.
- Terminal intégré avec PlatformIO Core (CLI) et puissant moniteur de port série.

Installer l'extension **PlatformIO** en entrant son nom dans la barre de recherche des 

Une **prise en main** de l'extension **PlatformIO** pour VSCode est proposée sur [RANDOM NERD TUTORIALS](#).

3.4 Extensions pour le langage Python


- **Source** : [Getting Started with Python in VSCode](#)
- **Lectures connexes**
 - Wiki [Python et MicroPython](#)

3.4.1 Présentation

*“Travailler avec Python dans Visual Studio Code, en utilisant l'extension **Microsoft Python**, est **simple**, **amusant** et **productif**. Cette extension fait de VSCode un excellent IDE et s'applique à n'importe quel système d'exploitation avec une variété d'interpréteurs Python. Il exploite toute la puissance de VSCode pour fournir des fonctions de complétion automatique et IntelliSense, de lintérisation, de débogage et de test unitaire, ainsi que la possibilité de basculer facilement entre les environnements Python, y compris les environnements virtuels et [conda](#). [Site VSCode](#)*



3.4.2 Installations

- **Anaconda**
 - **Installer** une distribution **Anaconda** en suivant le document [Outils - Installer et utiliser Anaconda avec VSCode](#)
- **Extensions Python pour VSCode**
 - **Installer** la liste des extensions ci-dessous en entrant leur nom dans la barre de recherche de 

| Nom |
|---------|
| Jupyter |
| Pylance |
| Python |



3.5 Extension pour SQLite

- **Installer SQLite** en entrant son nom dans la barre de recherche.

3.6 Extensions pour PHP

| Nom |
|------------------|
| PHP Debug |
| PHP Intelephense |

4. Réglages


Il est facile de configurer Visual Studio Code grâce à ses différents paramètres. Les différentes parties de l'éditeur et de l'interface utilisateur de VSCode comportent des options que vous pouvez modifier.

VSCode propose deux paramétrages :

- **Paramètres utilisateur** (User Settings) - Les paramètres s'appliquent à toutes les instances de VSCode.
- **Paramètres de l'espace de travail** (Workspace Settings) - Les paramètres sont stockés dans votre espace de travail et ne s'appliquent que lorsque l'espace de travail est ouvert.

Les paramètres de l'espace de travail remplacent les paramètres utilisateur.

4.1 Accéder aux paramètres

On accède aux paramètres de réglage de l'éditeur à partir de **Fichier → Préférences → Paramètres** ou  → **Paramètres** ou (Ctrl+,)

4.2 Vider l'IDE au démarrage

- **Pour avoir un espace de travail vide au démarrage**
 - *Barre de recherche des paramètres* : entrer **window** puis sélectionner **none** pour **Window:Restore Windows**

4.3 Complétion par tabulation

- Barre de recherche des paramètres : entrer **editor.tabCompletion** puis **on**.

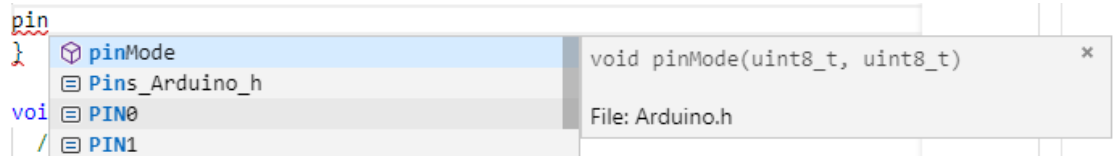
4.4 L'extension C/C++

• Intellisense

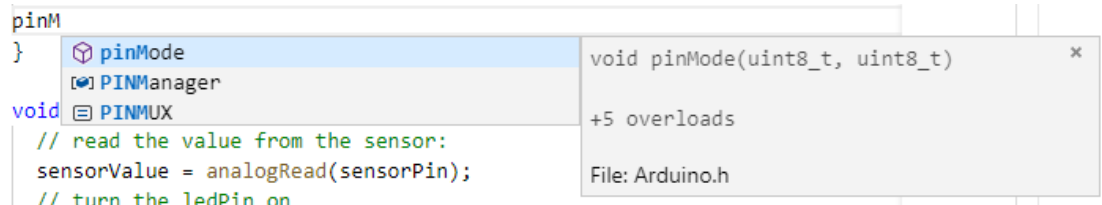
- Barre de recherche des paramètres des paramètres : entrer Intelli Sense Engine →

C_Cpp: Intelli Sense Engine :

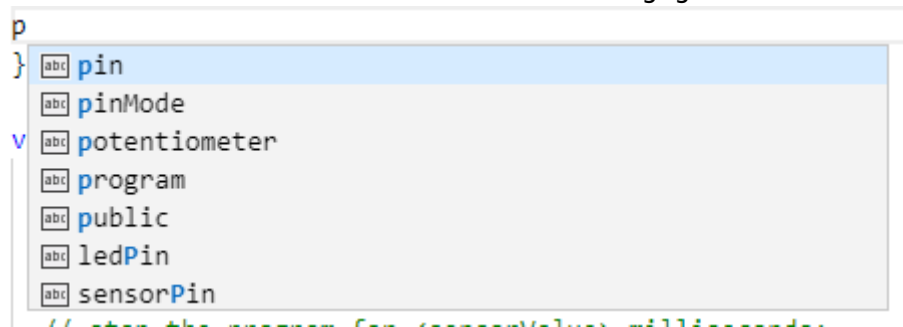
- **Default** fournit des résultats contextuels (à choisir si on ne connaît pas les noms des attributs, méthodes de la classe)



- **Tag Parser** fournit des résultats "flous" qui ne sont pas sensibles au contexte (à choisir si on connaît le nom des attributs, méthodes de la classe).



- **Disabled** désactive les fonctionnalités du service de langage C / C ++.



4.5 L'extension PHP Intelephense

• Intellisense

- Barre de recherche des paramètres

- entrer **suggest basic** et décocher PHP: Suggest:Basic
- entrer **format** et cocher **Editor: Format on save** après avoir sélectionné **Editeur de texte**



4.6 L'extension PHP Debug

L'utilisation de cette extension nécessite l'installation de xdebug. Voir [Live coding](#) et l'exemple de

modification du fichier [php.ini](#).

5. Utilisation de l'éditeur

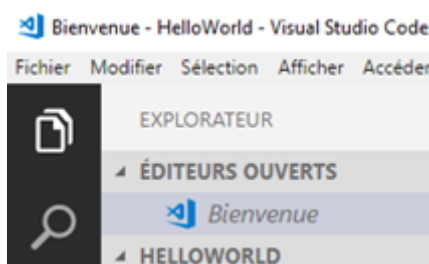
Quel que soit le langage utilisé, la création d'un projet suit les étapes suivantes :

1. créer un répertoire pour le projet,
2. placer ce répertoire dans VSCode,
3. ajouter un ou plusieurs fichiers,
4. ajouter un ou plusieurs sous-répertoires.

5.1 Créer un dossier pour le projet

Créer un dossier destiné à recevoir le projet en dehors de VSCode (par exemple sur le bureau ou sur un serveur). Intégrer ce dossier à VSCode à partir de **Fichier → Ouvrir le dossier** ou par un **glisser-déposer**.

Exemple : Projet HelloWorld



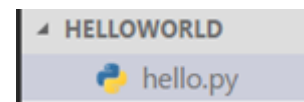
5.2 Ajouter un fichier dans le projet

Placer la souris sur le nom du projet. Les icônes



apparaissent. Cliquez sur  pour créer un fichier.


Exemple : fichier hello.py



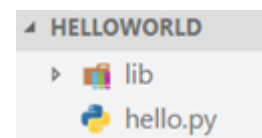
5.3 Ajouter un répertoire dans le projet

Placer la souris sur le nom du projet. Les icônes



apparaissent. Cliquez sur  pour créer un dossier.

Exemple : dossier lib



5.4 Quelques raccourcis

- Sélection du mot sous le curseur ou de sa prochaine occurrence (**Ctrl+D**)
- Curseur multiple (**Alt+Click**)
- Remplacement multiple (**Ctrl+F2**)
- Complétion de code (**Ctrl+ESPACE**) ou "." pour la sélection, (ENTER) pour la validation
- Go to définition (**F12**) si supporté par le langage
- Go to symbol (**Ctrl+shift+O**)
- Survol (Pour les langages qui le supportent comme CSS, des informations complémentaires sont délivrées)
- Affichage des erreurs et des avertissements (**Ctrl+Shift+M**)
- [Snippets](#) utilisateur (à créer : Fichier → préférences) et [Emmet intégrés](#)

5.5 La complétion de code (IntelliSense)









• 5.5.1 Généralités

IntelliSense est un terme général pour une variété de fonctionnalités d'édition de code, y compris : la complétion du code, les informations sur les paramètres, les brèves et les listes de membres. Les fonctionnalités d'IntelliSense sont parfois appelées "complétion de code", "aide au contenu" et "astuce de code".

Dans VSCode, IntelliSense est fourni pour JavaScript, TypeScript, JSON, HTML, CSS, Less et Sass. VSCode prend en charge les complétions basées sur des mots pour n'importe quel langage de programmation, mais peut également être enrichi en installant une extension de langue.

• Types de complétion

- **Source** : [IntelliSense dans VSCode](#)

| Symbole | Description | Symbole | Description |
|---|------------------------------------|---|-----------------------|
|  | Méthodes, fonctions, constructeurs |  | Mots-clés |
|  | Variables et champs |  | Propriétés, attributs |
|  | Classes |  | Interfaces |
|  | Valeurs, énumérations |  | Références |

5.6 Quelques raccourcis clavier

* **Source** : [Édition de base](#)

| Raccourci | Description |
|----------------------------------|-------------------------|
| F1 ou ctrl-shift-P | Editeur de commande |
| F2 | Renommer |
| Ctrl+F2 | Renommer multi-curseurs |
| Shift+F12 | Recherche de référence |

| Raccourci | Description |
|---------------------|---|
| Alt+F12 | Définition |
| Ctrl+Tab | Sélection d'un fichier pour l'afficher dans la fenêtre active |
| Ctrl+Shift+O | Définition des symboles Remarque : faire suivre @ de : pour les grouper par catégories |
| Ctrl+G | Atteindre une ligne dans un fichier |
| Ctrl+Shift+F | Recherche d'un texte dans tous les fichiers du projet |
| Ctrl+Shift+M | Voir le récapitulatif des erreurs et avertissements |
| Ctrl+Click | Atteindre une définition |

1)

IntelliSense ou encore complétion ou **autocomplétion**, est une fonctionnalité informatique permettant à l'utilisateur de limiter la quantité d'informations qu'il saisit avec son clavier, en se voyant proposer un complément qui pourrait convenir à la chaîne de caractères qu'il a commencé à taper.

2)

La **coloration syntaxique** est une fonctionnalité informatique proposée par certains éditeurs de texte, qui consiste à formater automatiquement chacun des éléments du texte affiché en utilisant une couleur et une fonte caractéristiques de son type.

3)

Emmet est un plug-in permettant de créer rapidement une structure HTML/CSS, existant sous forme d'extension pour tous les principaux éditeurs de texte et environnements de développement.

4)

Le **débogage** est un processus de diagnostic, de localisation et d'élimination des erreurs des programmes informatiques; ce processus permet aussi la vérification (autrement dit le test) du programme en cours d'élaboration.

From:

<http://webge.fr/dokuwiki/> - **WEBGE Wikis**

Permanent link:

<http://webge.fr/dokuwiki/doku.php?id=outils:vscode:generalites&rev=1686662916>Last update: **2023/06/13 15:28**