

# SublimeText 3

Krystof2so

26 mai 2020

## 1 Juste pour se familiariser

Pour tout ce qui concerne l'utilisation de **SublimeText** l'utilisateur est invité à se rendre dans le répertoire :

```
~/ .config/sublime-text-3
```

Le répertoire **/Packages** contient le répertoire **/User** qui est le répertoire fourre-tout de l'utilisateur ((accessible par **Preferences -> Browse Packages**). On y retrouve les greffons personnalisés, les *snippets*, les macros, etc. Il est à considérer comme votre espace personnel qu'aucune mise à jour de **SublimeText** n'écrasera.

Il est à noter que **SublimeText** est programmable, ce qui permet aux utilisateurs ayant des compétences en programmation d'ajouter leurs propres fonctionnalités, en utilisant le langage **Python**.

**SublimeText** est doté d'une console **Python** que l'on peut ouvrir via **View -> Show Console**. Ainsi, un interpréteur **Python** est intégré à l'éditeur (séparé de l'interpréteur **Python** du système. Cet interpréteur est alors utile pour inspecter les paramètres de l'éditeur et pour tester rapidement des appels d'API lors du développement de greffons. **Capture d'écran**.

Presque tous les aspects du **SublimeText** peuvent être étendus ou personnalisés. La grande flexibilité de **SublimeText** explique pourquoi il y a tant de fichiers de configuration. Les fichiers de configuration de **SublimeText** sont des fichiers texte qui se conforment à une structure ou un format prédéfini : **JSON** prédomine, mais on trouve aussi des fichiers **XML** et **YAML**. Pour les options d'extensibilité plus avancées ce sont des fichiers de code source **Python** qui sont utilisés.

D'autre part **SublimeText** fournit une émulation de **vim** grâce au paquet **Vintage**. Ce paquet est ignoré par défaut et doit être activé par l'utilisateur <sup>1</sup>.

## 2 Pour une utilisation basique

### L'édition

Les fonctions d'édition sont multiples et variées.

---

1. Voir à ce sujet : <https://www.sublimetext.com/docs/3/vintage.html>

## La gestion des fichiers et des projets

Les projets regroupent des ensembles de fichiers et de dossiers qui vous permettent de garder votre travail organisé.

Il y a toujours un projet actif, même si vous n'en avez pas créé ou ouvert un. Dans cette situation, vous travaillez avec un projet anonyme, dont les fonctionnalités sont limitées. Les nouvelles fenêtres utilisent toujours un projet anonyme lorsqu'elles s'ouvrent pour la première fois.

Les métadonnées du projet sont stockées dans des fichiers JSON avec une extension `.sublime-project`. Partout où il y a un fichier `.sublime-project`, vous trouverez également un fichier `.sublime-workspace` auxiliaire. Ce fichier contient des données de session que vous ne devez jamais modifier.

### Créer un projet :

Commencer par un projet anonyme en ouvrant une nouvelle fenêtre ou en fermant tout projet actif avec le menu **Project -> Close project**. Nous pouvons alors ajouter et supprimer des dossiers à un projet à l'aide du menu **Project** ou du menu contextuel de la barre latérale. Faire glisser un dossier sur une fenêtre de **SublimeText** ajoutera également le fichier au projet. Pour enregistrer un projet anonyme, se rendre sur dans **Project -> Save Project As...** Une fois le projet enregistré, il demeure modifiable.

### Ouvrir des projets :

L'onglet **Project** fournit diverses possibilité d'ouverture. De plus, vous pouvez ouvrir un projet en ligne de commande en passant le fichier `.sublime-project` en argument à l'aide de la commande permettant l'ouverture de Sublime Text.

### Les espaces de travail (*Workspaces*) :

Les espaces de travail contiennent des données de session associées à un projet, qui comprennent des informations sur les fichiers ouverts, la disposition des volets, l'historique des recherches et bien plus encore. Un projet peut avoir plusieurs espaces de travail. Les espaces de travail se comportent comme des projets. Pour créer un nouvel espace de travail, sélectionnez **Project -> New Workspace For Project**. Pour enregistrer l'espace de travail actif, sélectionnez **Project -> Save Project As...** Les métadonnées de l'espace de travail sont stockées dans des fichiers JSON et portent l'extension `.sublime-workspace`, que vous n'êtes pas censé modifier. Contrairement aux fichiers `.sublime-project`, les fichiers `.sublime-workspace` ne sont pas destinés à être partagés, vous ne devez donc jamais livrer des fichiers `.sublime-workspace` dans un dépôt de code source.

## La navigation

### Goto Anything : [Ctrl] + T

En utilisant **Goto Anything**, vous pouvez naviguer rapidement dans les fichiers de votre projet. **Goto Anything** accepte divers opérateurs.

Le symbole « : » : [Ctrl] + G

fichier1:100

Cela indique à Sublime Text de rechercher d'abord un fichier dont le chemin d'accès correspond à *fichier1*, puis de se rendre à la ligne 100 dans le dit fichier.

Le symbole : [Ctrl] + R

**@ma\_variable**

Recherche dans le fichier actif le symbole nommé *ma\_variable*. Les symboles comprennent généralement des noms de classe et de fonction.

Le symbole # : [Ctrl] + ;

**#verb**

Recherche les termes dans lequel va figurer l'expression *verb*.

La barre latérale donne un aperçu du projet actif. Les fichiers et dossiers de la barre latérale seront disponibles dans le menu **Goto Anything**. Projets et barre latérale sont étroitement liés. Il est important de noter qu'il y a toujours un projet actif; si vous n'avez pas ouvert de fichier de projet, un projet anonyme sera utilisé à la place. La barre latérale fournit des opérations de base de gestion de fichiers grâce à son menu contextuel.

## Rechercher et remplacer

Sur fichier unique :

**Ouvrir le panneau de recherche** : [Ctrl] + F

**Ouvrir le panneau de recherche progressive** : [Ctrl] + I

**Basculer sur le mode des expressions régulières** : [Alt] + R

**Sensibilité à la casse** : [Alt] + C

**Correspondance exacte** : [Alt] + W

**Ouvrir le panneau de remplacement** : [Ctrl] + H

**Tout remplacer** : [Ctrl] + [Alt] + [Entrée]

Sur plusieurs fichiers :

**Ouvrir le panneau de recherche** : [Ctrl] + [Shift] + F

**Basculer sur le mode des expressions régulières** : [Alt] + R

**Sensibilité à la casse** : [Alt] + C

**Correspondance exacte** : [Alt] + W

Il est également possible de paramétrer la portée de la recherche, en définissant les champs de recherche par l'ajout de répertoires individuels (chemins d'accès de type Unix, même sous Windows), par l'ajout/exclusion de fichiers basés sur des jokers (« \* ») ou l'ajout d'emplacements symboliques (<open folders>, <open files> ...). Il est également possible de combiner ces filtres en utilisant des virgules.

Vous pouvez naviguer dans les résultats de recherche à l'aide des raccourcis suivants :

**Occurrence suivante** : [F4]

**Occurrence précédente** : [Shift] + [F4]

**Se rendre d'une occurrence à une autre** : [Entrée]

**Les expressions régulières (*regex* ou *rege*x) :**

Les expressions régulières permettent de trouver des motifs complexes dans un texte. Pour profiter pleinement des possibilités de recherche et de remplacement, vous devriez au moins apprendre les bases des expressions régulières. Pour les utiliser vous devez d'abord les activer dans les différents panneaux de recherche ([Alt] + R), sinon, les termes recherchés seront interprétés littéralement.

### 3 Personnalisation de Sublime Text

Voyons les options les plus courantes afin de personnaliser entièrement Sublime Text.

#### Les paramètres (*Settings*)

Sublime Text stocke les données de configuration dans des fichiers portant l'extension `.sublime-settings`. La flexibilité s'obtient au prix d'un système légèrement complexe s'agissant des paramètres. Il faudra veiller à toujours placer ses fichiers de configuration personnels, afin de garantir qu'ils auront la priorité sur tout autre fichier de configuration, dans le répertoire :

```
~/.config/sublime-text-3/Packages/User
```

Ces fichiers de paramétrage sont au format JSON. Le nom de chaque fichier de type `.sublime-settings` détermine sa finalité. Les noms peuvent être descriptifs (comme `Preferences.sublime-settings` ou `Minimap.sublime-settings`, ou ils peuvent être liés aux paramètres de contrôle. Les fichiers syntaxiques propres aux langages de programmation sont aussi reconnaissables (exemple : `Python.sublime-settings`). En outre, certains fichiers de paramètres ne s'appliquent qu'à des plates-formes spécifiques. Cela peut se déduire à partir du nom du fichier, comme par exemple `Preferences ({Linux}).sublime-settings`. À noter que de tels fichiers de réglages spécifiques à une plate-forme situés dans le répertoire `Packages/User` sont ignorés. De cette façon, vous pouvez être sûr qu'un seul fichier de ce type a priorité sur tous les autres. Enfin, sachez que les modifications de paramètres sont généralement prises en compte par Sublime Text en temps réel, mais vous devrez peut-être redémarrer l'éditeur afin de charger de nouveaux fichiers de paramètres.

Pour accéder aux fichiers de paramètres et les modifier il faudra passer par le menu **Preferences -> Settings**. Seront alors ouverts en parallèle les fichiers :

```
Preferences.sublime-settings User
```

et

```
Preferences.sublime-settings Default.
```

L'édition de ce dernier est fortement déconseillée, mais vous pouvez utiliser ce fichier à titre de référence puisqu'il contient les commentaires expliquant la finalité de tous les paramètres globaux disponibles.

Le même fichier de paramètres (tel que `Python.sublime-settings`) peut apparaître à plusieurs endroits. Tous les paramètres définis dans des fichiers de nom identique seront fusionnés et écrasés selon des règles prédéfinies. Ci-dessous, vous pouvez voir l'ordre dans lequel **Sublime Text** traiterait une hiérarchie hypothétique de paramètres pour les fichiers Python :

- `Packages/Default/Preferences.sublime-settings`
- `Packages/Default/Preferences (votre_système).sublime-settings`
- `Packages/User/Preferences.sublime-settings`
- `Packages/Python/Python.sublime-settings`
- `Packages/User/Python.sublime-settings`

## Les raccourcis claviers (*Key Bindings*)

Les combinaisons de touches sont définies au format JSON et stockées dans des fichiers portant l'extension `.sublime-keymap`. Dans le même répertoire, des fichiers *keymap* séparés pour Linux, OSX et Windows peuvent co-exister pour une meilleure intégration au système d'exploitation. Exemple :

```
[
  { "keys": ["ctrl+shift+n"], "command": "new_window" },
  { "keys": ["ctrl+o"], "command": "prompt_open_file" }
]
```

**Sublime Text** est livré avec des raccourcis clavier par défaut. Afin de pouvoir les remplacer ou d'en ajouter de nouveaux, utilisez un fichier *keymap* séparé avec une priorité plus élevée en le plaçant dans le répertoire `Packages/User`.

Les raccourcis clavier simples consistent en une séquence d'une ou plusieurs touches associées à une commande. Cependant, il existe des syntaxes plus complexes pour passer des arguments aux commandes et pour restreindre les raccourcis clavier à des contextes spécifiques. Exemple :

```
[
  { "keys": ["shift+enter"], "command": "insert", "args":
    { "characters": "\n" } }
]
```

Dans cet exemple, « `\n` » est passé à la commande d'insertion à chaque fois que l'on presse les touches `[Shift]` + `[Entrée]`.

Vous pouvez créer raccourcis composés de plusieurs combinaisons de raccourcis. Exemple :

```
{ "keys": ["ctrl+k", "ctrl+v"], "command":
  "paste_from_history" }
```

Ici, pour déclencher la commande *paste\_from\_history*, vous devez d'abord presser les touches `[Ctrl] + K`, puis relâcher la touche `K`, et enfin appuyer sur la touche `V`. Cet exemple est une liaison de touches par défaut que vous pouvez donc essayer.