

# Tutoriel PyCharm

Informatique II

Kévin Huguenin et Timothy Bollé  
20/02/2017

## Installation

Si vous utilisez les machines de la salle informatique, les programmes étant déjà installés, vous pouvez directement passer à l'étape d'utilisation (étape 13).

### Installation de Python

Pour utiliser PyCharm, il est nécessaire que Python soit installé sur votre ordinateur. Si ce n'est pas le cas, Python peut être téléchargé à l'adresse suivante : <https://www.python.org/downloads/>. Assurez-vous de bien noter dans quel dossier vous l'avez installé.

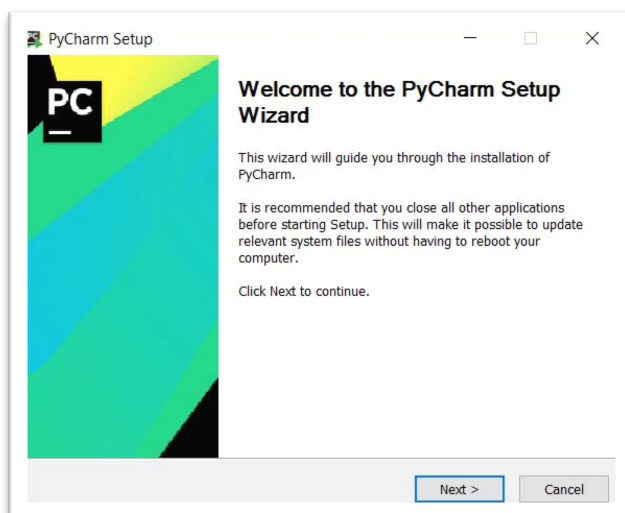
### Installation de PyCharm

PyCharm est un IDE (*Integrated Development Environment*). Il s'agit, tout comme jEdit, d'un logiciel permettant d'intégrer dans une même fenêtre tous les éléments utiles à la programmation en python : un éditeur de texte pour écrire des scripts, une console pour exécuter des programmes, ainsi qu'un explorateur de fichiers pour parcourir le projet en cours.

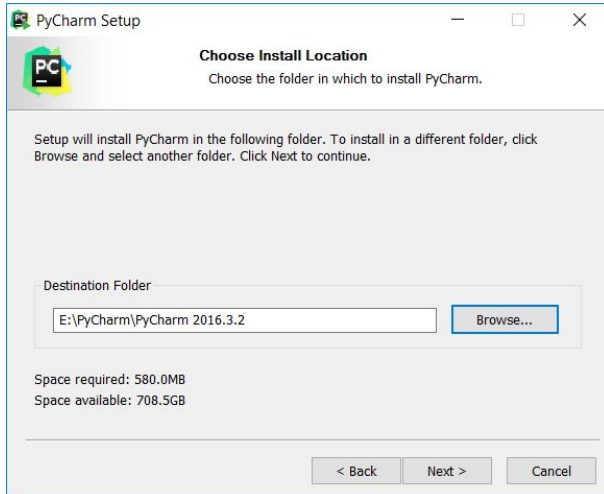
En tant qu'étudiant, vous pouvez utiliser la version professionnelle de PyCharm gratuitement. Pour cela rendez-vous sur la page <https://www.jetbrains.com/student/> et appuyez sur le bouton **Apply Now**. Vous serez amenés sur une page vous demandant votre nom et prénom, ainsi que votre adresse email académique. Il est ici important d'entrer votre **adresse UNIL**. Vous recevrez ensuite un email avec des instructions sur la façon d'activer votre licence. Suivez-les. Il vous sera demandé de créer un compte JetBrains (la société produisant PyCharm). Il vous suffit de créer un identifiant (vous pouvez utiliser votre adresse email de l'Unil comme identifiant facile à retenir), ainsi qu'un mot de passe. Une fois connecté à votre compte, vous pourrez télécharger la version Professionnelle de Pycharm.

Vous pouvez maintenant exécuter l'installeur de PyCharm que vous venez de télécharger en double cliquant dessus. Pour les utilisateurs de Mac, il se peut que vous deviez autoriser l'installation d'applications externes. Pour cela, allez dans l'application *Préférences Système*, puis dans l'onglet *Sécurité et confidentialité*. Ici vous pouvez ouvrir le cadenas et cocher l'option *N'importe où*.

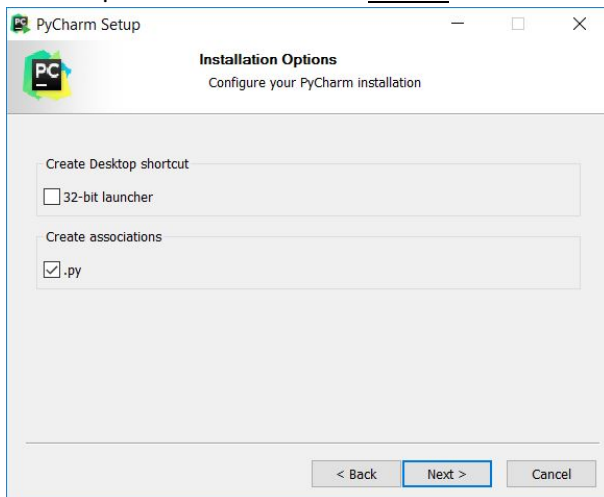
1. Cliquez sur **Next**.



2. Ici vous pouvez choisir le dossier où vous voulez installer PyCharm. **Next.**



3. Ici vous pouvez cocher la case **Create associations .py**. Ensuite, faites **Next.**



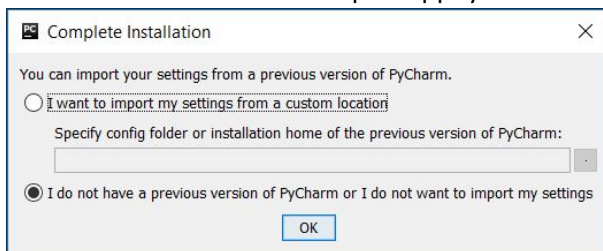
4. Cliquez ensuite sur **Install** puis sur **Finish**, une fois l'installation terminée.

## Utilisation

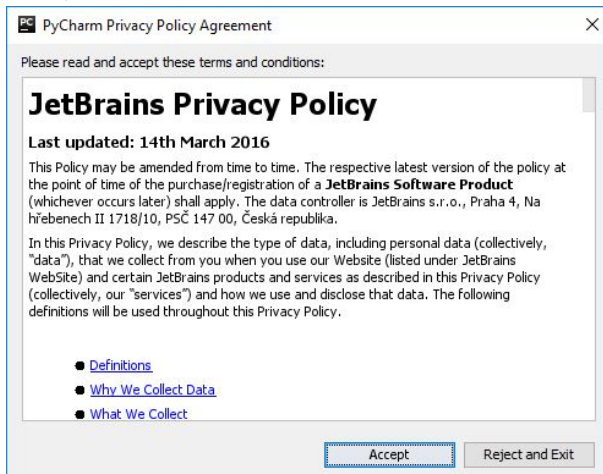
### Configuration initiale

Pour cette première utilisation, nous allons configurer PyCharm. Une documentation complète de PyCharm est disponible à l'adresse <https://www.jetbrains.com/pycharm/documentation/>.

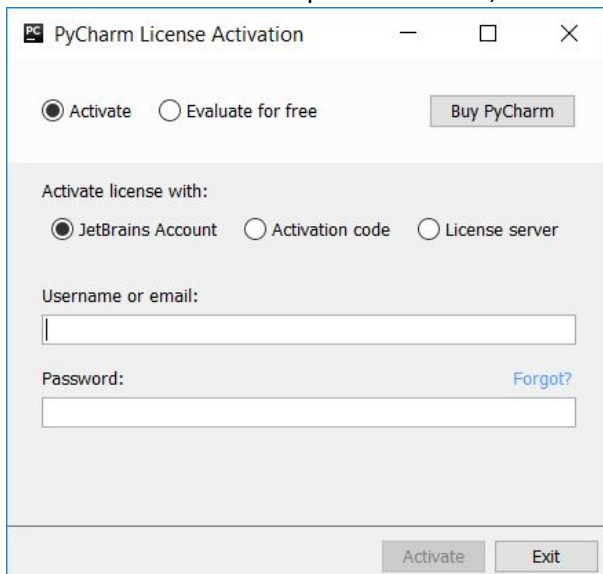
5. Démarrez PyCharm
6. Sélectionnez **I do not have...** puis appuyez sur **OK**



7. Acceptez les conditions d'utilisation



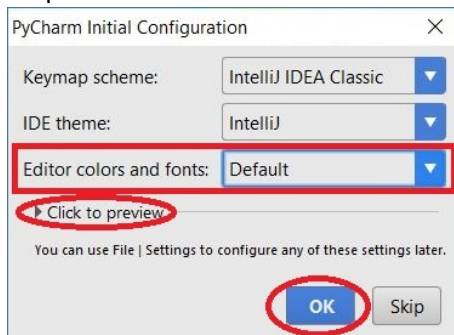
8. Si vous utilisez la version professionnelle, **entrez vos identifiants** puis cliquez sur **Active**.



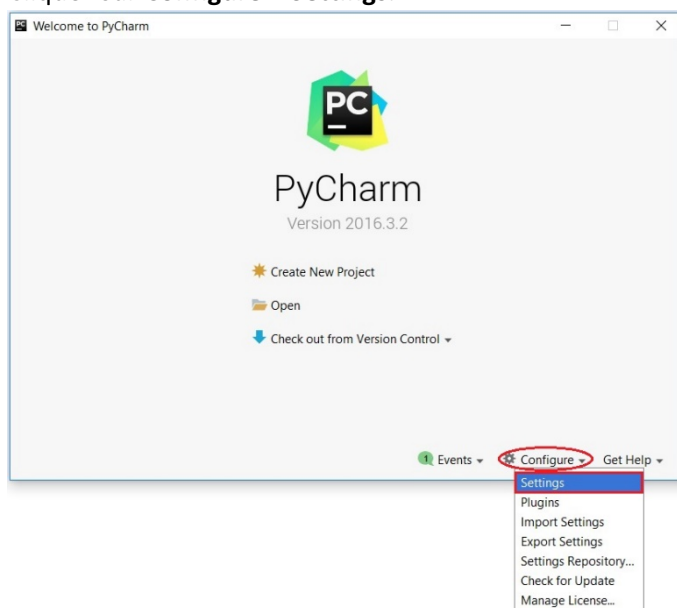
Sur la fenêtre suivante, vous pourrez choisir les thèmes d'affichage des fenêtres et de l'éditeur. Ce sont des réglages esthétiques à effectuer selon vos préférences. Vous pouvez regarder les

différents thèmes proposés pour l'éditeur (*Editor colors and fonts*) et voir à quoi cela ressemble en cliquant sur le bouton *Click to preview*.

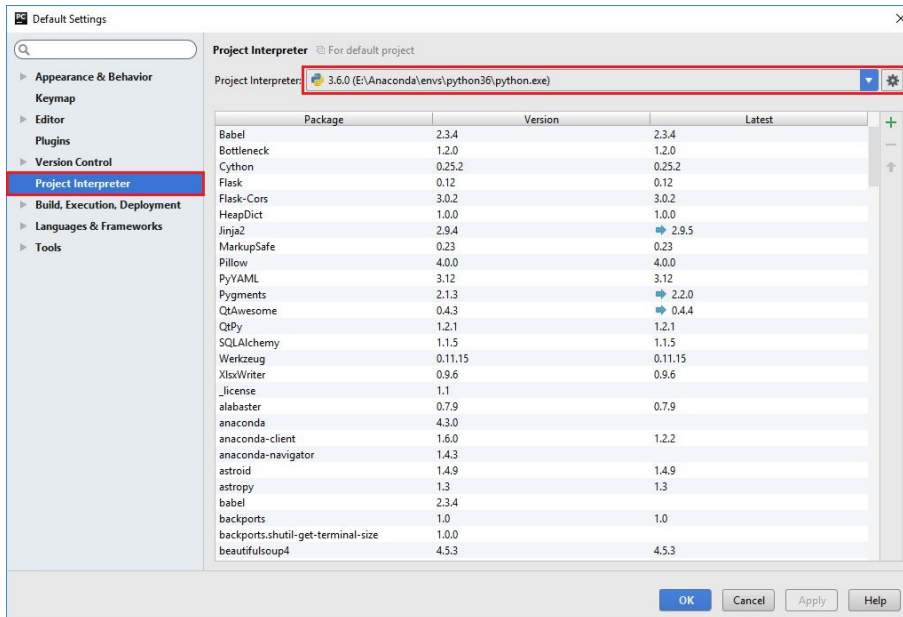
9. Cliquez sur **OK**.



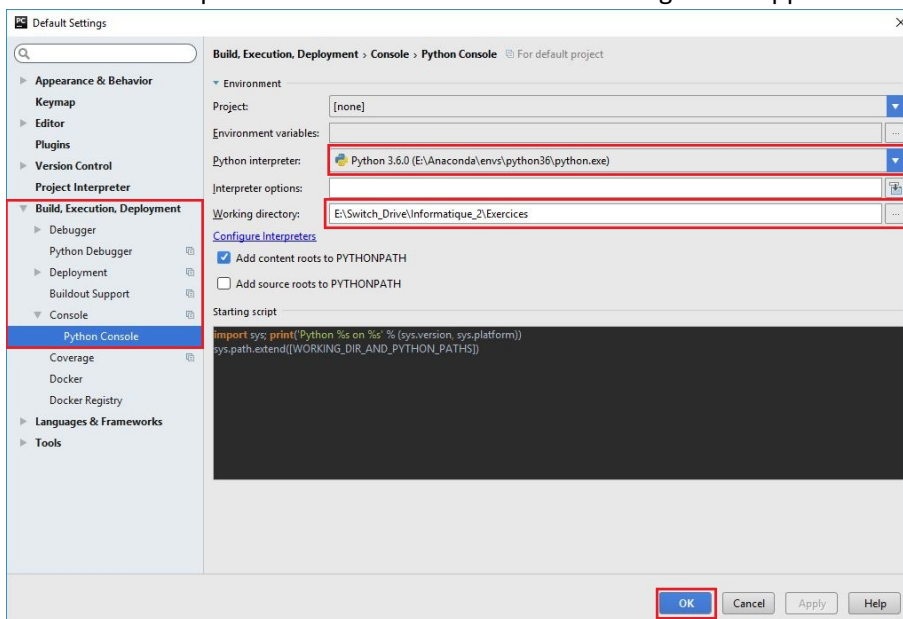
10. Cliquez sur **Configure > Settings**.



11. Dans la nouvelle fenêtre, allez dans l'onglet **Project Interpreter** puis cliquez sur la petite flèche en haut à droite. Dans le menu déroulant, il devrait y avoir un chemin vers votre exécutable python python.exe. Sélectionnez le. Il est normal de ne pas observer la même liste dans l'exemple de tableau ci-dessous.



12. Allez ensuite dans l'onglet **Build, Execution, Deployment > Console > Python Console**. Sélectionnez à nouveau le chemin vers votre exécutable python à l'aide de la flèche dans la ligne **Python interpreter**. Sélectionnez ensuite le dossier où vous souhaitez faire vos exercices dans la ligne Working directory, à l'aide du bouton sur la droite. Nous vous conseillons de **faire vos exercices dans un sous dossier de votre répertoire SwitchDrive**. Cela vous permettra d'y accéder depuis n'importe quel ordinateur ayant un accès à votre SwitchDrive. Cliquez ensuite sur **OK**. Une barre de chargement apparaît. Laissez la finir.



Ces quelques opérations nous ont permis d'indiquer à PyCharm quel Python nous souhaitons utiliser et où nous souhaitons travailler.

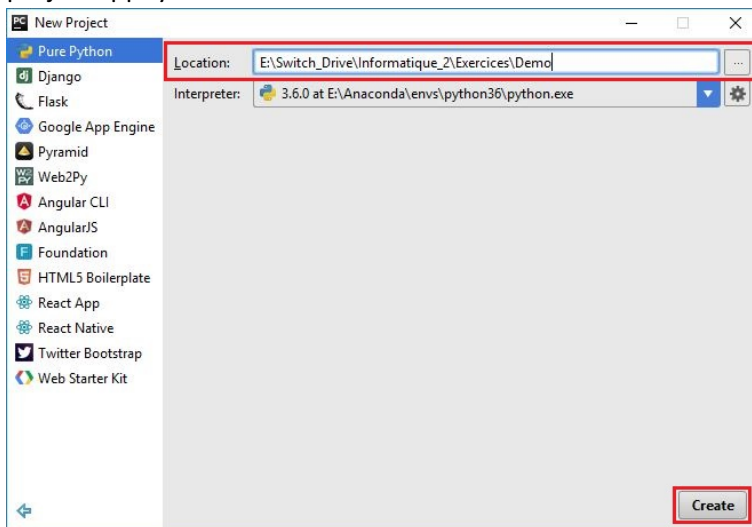
## Travailler sur un projet

Nous allons commencer par créer un nouveau projet.

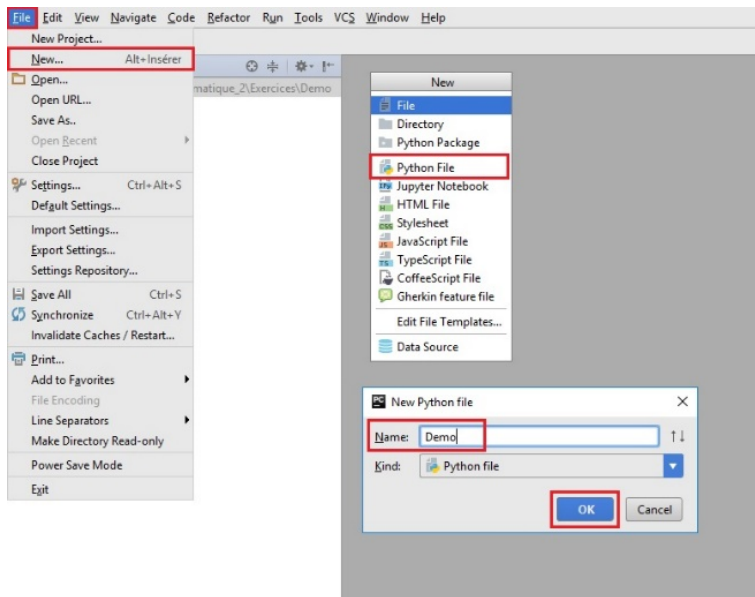
### 13. Appuyez sur **Create New Project**.



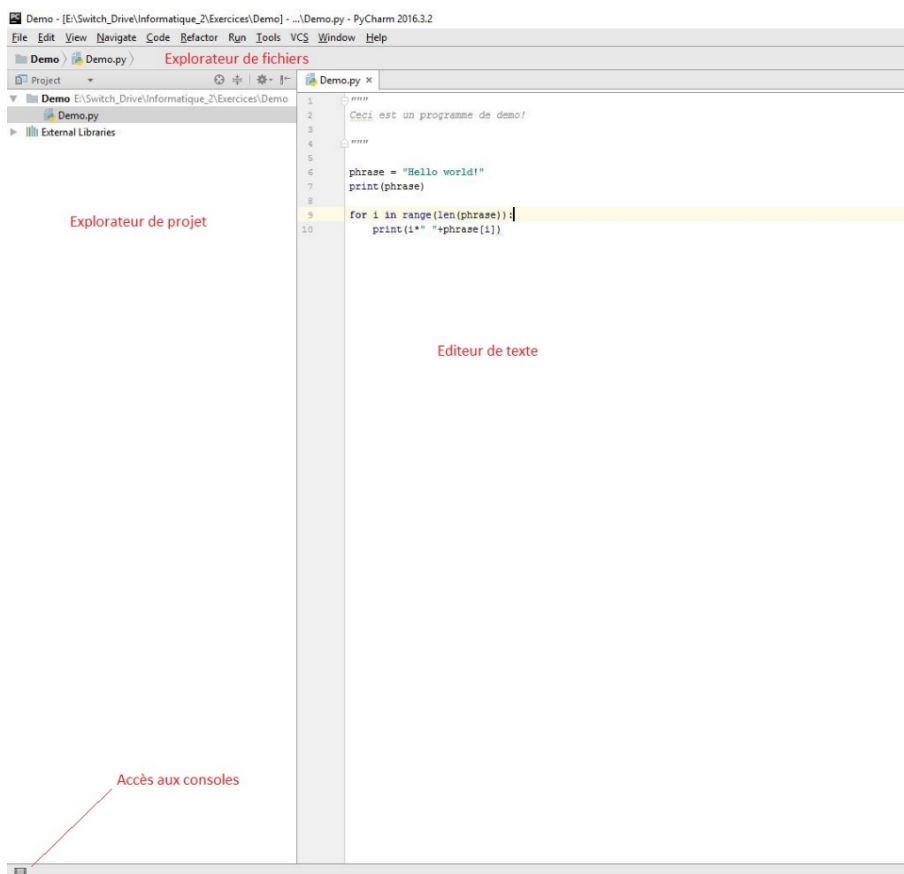
### 14. Sélectionnez ensuite l'onglet Pure Python et assurez-vous que le dossier de travail est celui désiré (sous dossier SwitchDrive par exemple). Vous pouvez donner un nom au dossier du projet. Appuyez ensuite sur **Create**.



### 15. Une fois arrivé sur l'écran principal de PyCharm, vous pouvez créer votre premier script python. Pour cela, en haut à gauche, cliquez sur **File > New... > Python File**.

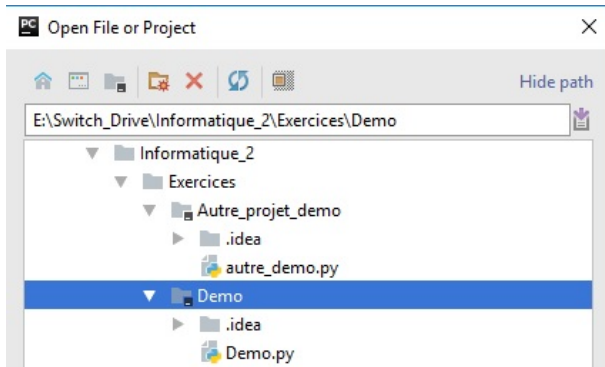


16. Le fichier sera automatiquement placé dans le dossier de votre projet.

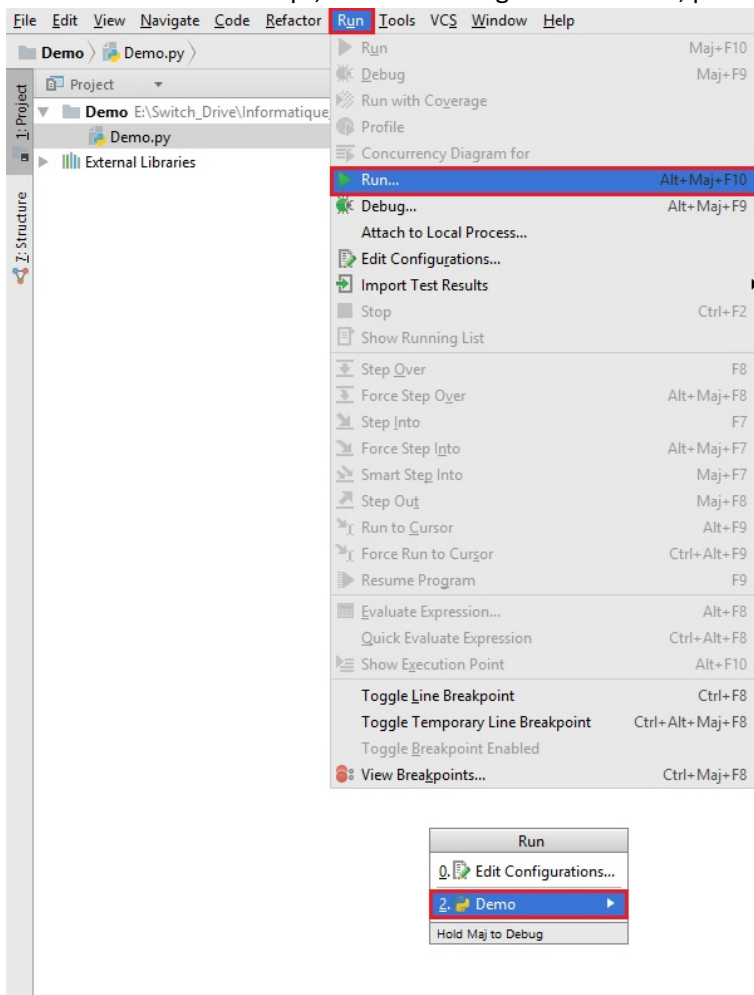


17. Pour ouvrir un projet déjà créé, allez dans l'onglet **File > Open**. De là, vous pouvez ouvrir un projet complet ou juste un script. Les dossiers contenant des projets apparaissent avec un carré noir au coin du dossier. Pour l'ouvrir, sélectionnez le dossier ou le fichier voulu et faites **Ok**.





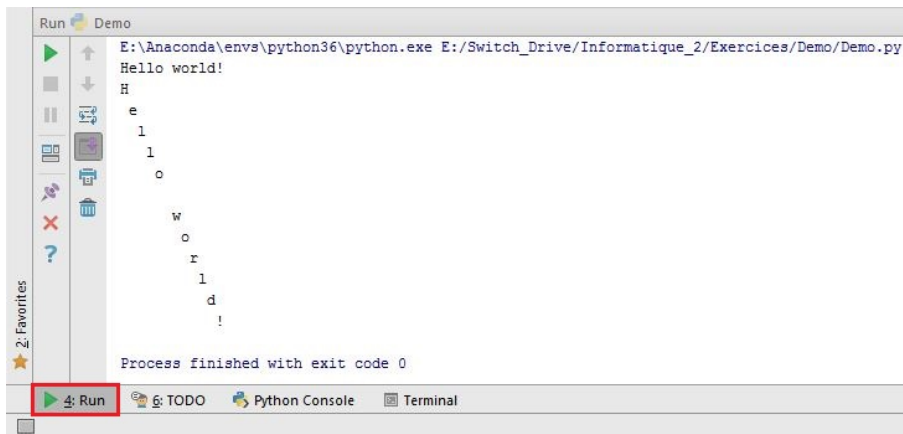
18. Pour exécuter votre script, allez dans l'onglet **Run > Run**<sup>1</sup>, puis cliquez sur le fichier à exécuter.



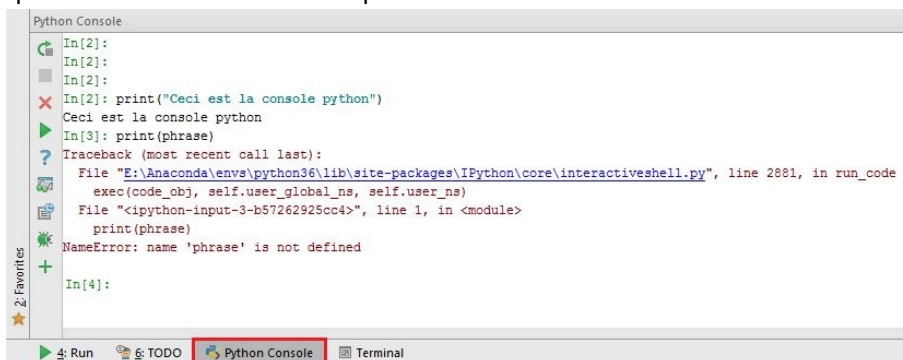
Une nouvelle zone apparaît. Il s'agit de la console. C'est ici que vous observerez les résultats de vos scripts.

19. Par défaut, l'output d'un script est renvoyé dans la console **Run**.

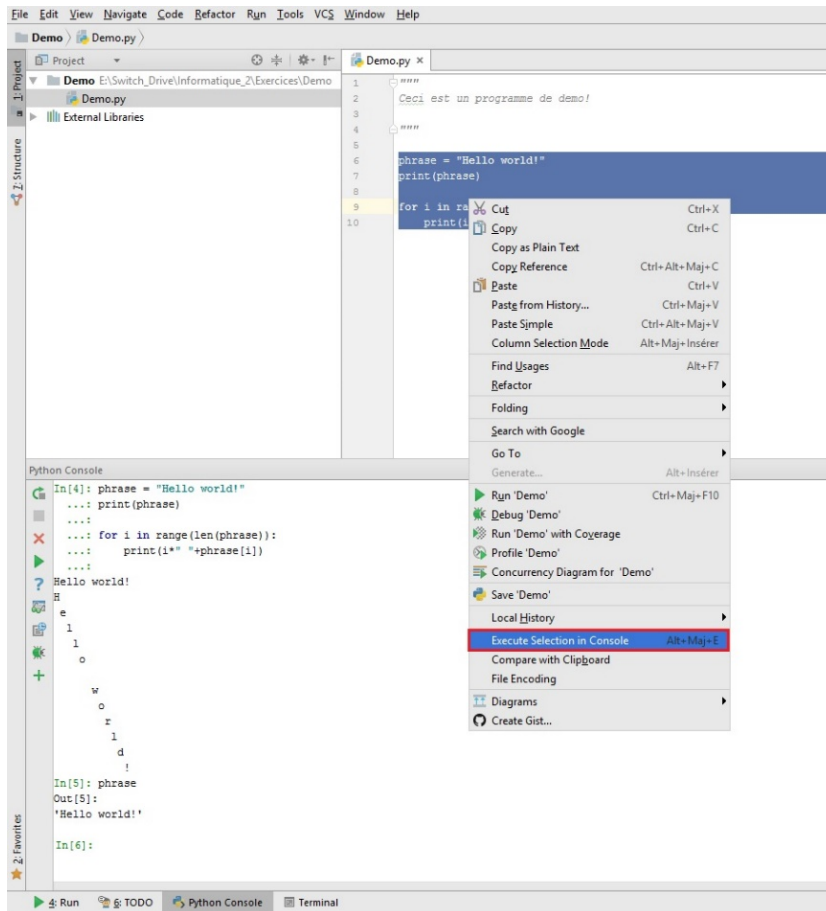
<sup>1</sup> Vous pouvez configurer les raccourcis clavier en allant dans les paramètres de Pycharm (**File > Settings**) puis dans l'onglet **Keymap**.



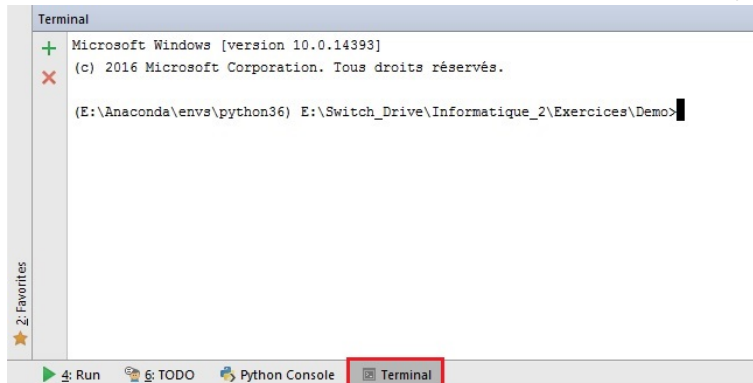
20. Vous pouvez aussi utiliser la console python pour effectuer des tests indépendamment de ce que vous mettez dans vos scripts.



Comme vous pouvez le voir, même après l'exécution du script, les variables du script n'existent pas dans la console python. Il est toutefois possible d'exécuter une partie de votre script dans la console. Pour cela, sélectionnez les lignes à exécuter, faites un clic droit et appuyez sur **Execute Selection in Console**. Faire cela produit le même effet que si vous aviez tapé toutes les instructions sélectionnées dans la console.

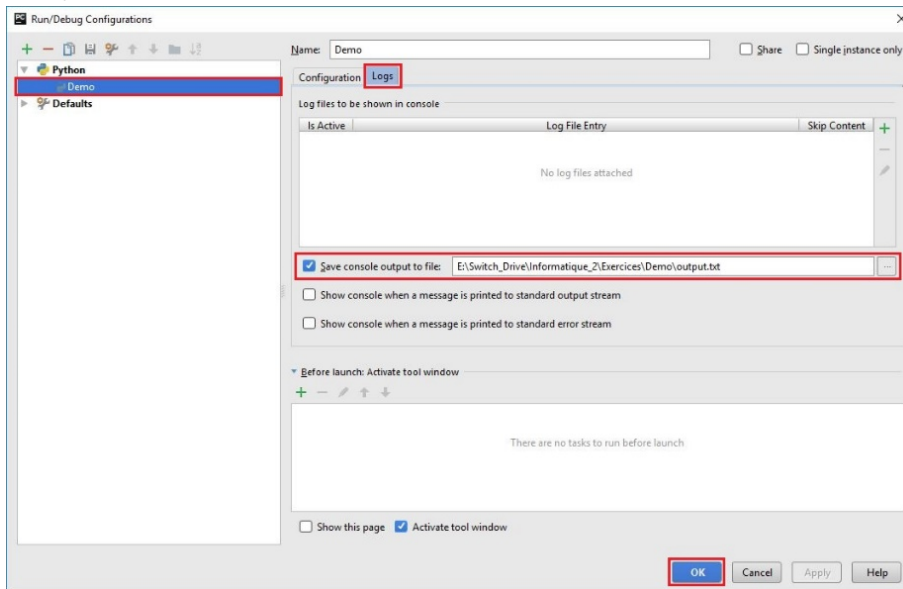


21. Vous avez aussi accès à un terminal de commande de votre système d'exploitation.

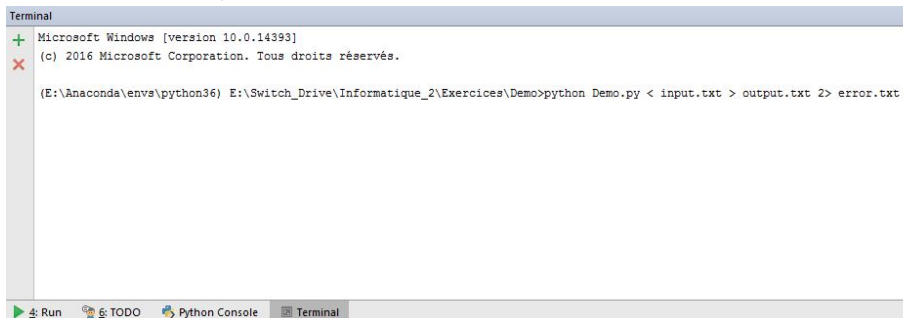


## Redirection de l'input et de l'output standard

22. Nous avons vu que normalement, le résultat d'un script (son output) est renvoyé dans la console **Run**. Il vous est aussi possible de rediriger l'output dans un fichier. Pour cela, allez dans l'onglet **Run > Edit Configurations...** Allez ensuite dans l'onglet **Logs** et cochez la case **Save console output to file** en choisissant le nom du fichier dans lequel vous voulez enregistrer l'output.



23. Il vous est aussi possible de faire ces redirections en utilisant le terminal de commande.



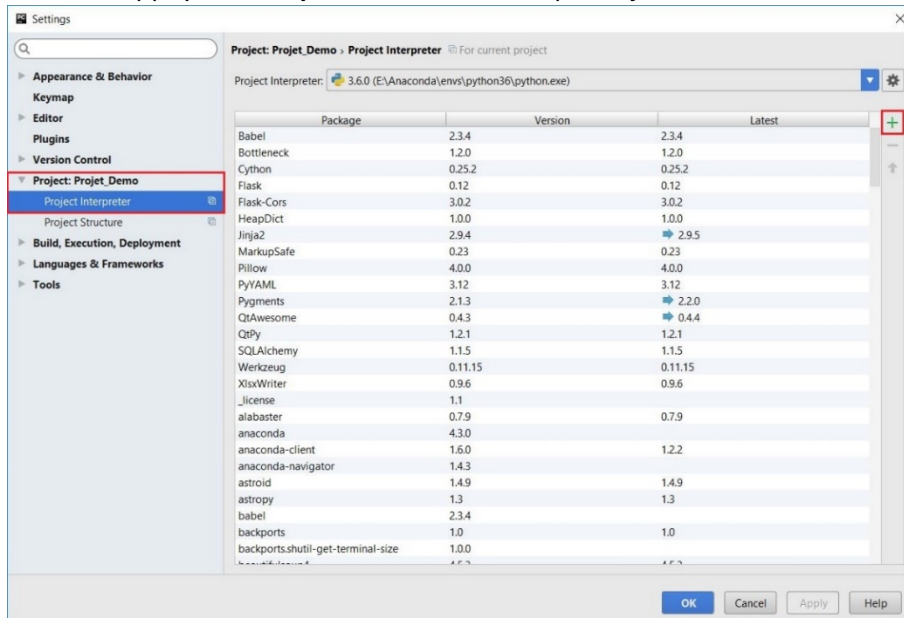
Cette commande permet de dire à la machine d'exécuter *Demo.py* avec le programme *Python*, de transmettre en input standard le fichier *input.txt* (< **input.txt**), de rediriger l'output sur le fichier *output.txt* (> **output.txt**) et enfin de rediriger les erreurs dans le fichier *error.txt* (**2> error.txt**).

## Installer des modules

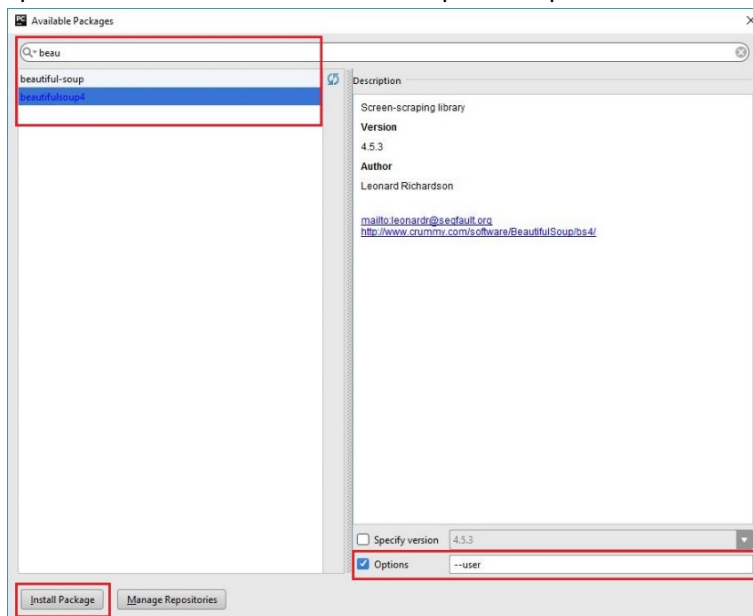
PyCharm permet d'installer et de gérer des modules python à l'aide du module pip<sup>2</sup>. Pour cela, ce dernier doit être installé sur la machine.

24. Dans PyCharm allez dans l'onglet **File > Settings**.

25. Allez ensuite dans l'onglet **Project > Project interpreter**. Ici s'affiche la liste des modules déjà installés. Appuyez sur le **plus** en haut à droite pour ajouter un module.



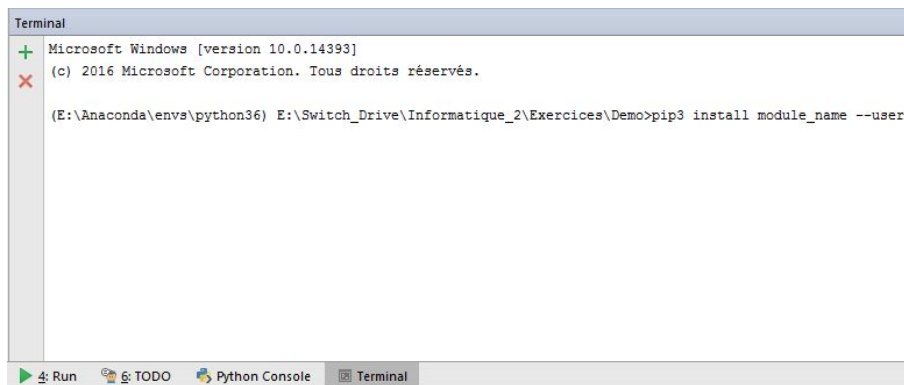
26. Dans cette fenêtre, vous pouvez rechercher un module et l'installer. Vous pouvez aussi spécifier la version du module ainsi que des options si nécessaire.



Si vous travaillez sur un ordinateur de la salle informatique, vous devez indiquer l'option **--user** afin d'installer le module seulement pour l'utilisateur connecté.

<sup>2</sup> Une documentation des commandes disponibles avec pip est accessible à l'adresse <https://pip.pypa.io/en/stable/>

27. Vous pouvez aussi installer les modules en utilisant le terminal de l'ordinateur.



```
Terminal
+ Microsoft Windows [version 10.0.14393]
(c) 2016 Microsoft Corporation. Tous droits réservés.

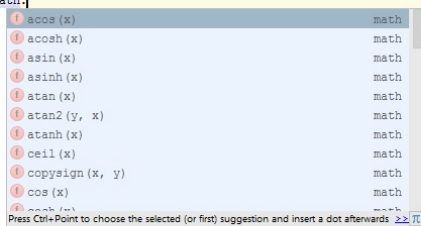
(E:\Anaconda\envs\python36) E:\Switch_Drive\Informatique_2\Exercices\Demo>pip3 install module_name --user
```

The image shows a terminal window with a title bar labeled 'Terminal'. The window contains the following text: a green plus sign followed by 'Microsoft Windows [version 10.0.14393]', a red minus sign followed by '(c) 2016 Microsoft Corporation. Tous droits réservés.', and a command prompt line '(E:\Anaconda\envs\python36) E:\Switch\_Drive\Informatique\_2\Exercices\Demo>pip3 install module\_name --user'. At the bottom of the window, there is a taskbar with icons for 'Run', 'TODO', 'Python Console', and 'Terminal'.

## Trucs et astuces

28. **Auto-complétion** PyCharm propose un système d'auto-complétion qui permet de faciliter l'utilisation de nombreux modules.

```
1 """
2 Ceci est un programme de demo!
3
4 """
5 import math
6
7 phrase = "Hello world!"
8 print(phrase)
9
10 for i in range(len(phrase)):
11     print(i* " "+phrase[i])
12
13 a = math.
```

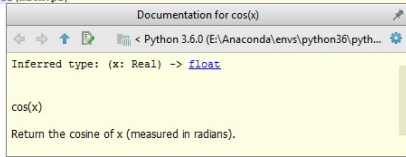


The screenshot shows a code editor with a Python script. The cursor is at the end of the line 'a = math.'. A dropdown menu is open, displaying a list of functions from the 'math' module, including 'acos(x)', 'acosh(x)', 'asin(x)', 'asinh(x)', 'atan(x)', 'atan2(y, x)', 'atanh(x)', 'ceil(x)', 'copysign(x, y)', 'cos(x)', and 'cosh(x)'. The 'math' module name is listed to the right of each function name. At the bottom of the dropdown, there is a small instruction: 'Press Ctrl+Point to choose the selected (or first) suggestion and insert a dot afterwards >>>|'.

Lorsque vous écrivez, le logiciel vous propose les différentes possibilités selon ce que vous écrivez. Les différentes variables et fonctions contenues dans un module vous sont aussi proposées. Il vous est donc possible de choisir une des propositions et celle-ci sera écrite automatiquement.

29. **Documentation** PyCham propose également d'avoir accès à de la documentation concernant une variable ou une fonction. Pour cela, placez votre curseur sur un élément et allez dans l'onglet **View > Quick Documentation**<sup>3</sup>

```
1 """
2 Ceci est un programme de demo!
3
4 """
5 import math
6
7 phrase = "Hello world!"
8 print(phrase)
9
10 for i in range(len(phrase)):
11     print(i* " "+phrase[i])
12
13 a = math.cos(math.pi)
```



The screenshot shows the same code editor as before, but now the cursor is on the line 'a = math.cos(math.pi)'. A 'Documentation for cos(x)' window is open, showing the inferred type '(x: Real) -> float' and the description 'Return the cosine of x (measured in radians)'. The window also shows the Python version 'Python 3.6.0 (E:\Anaconda\envs\python36\python...)' and some navigation icons.

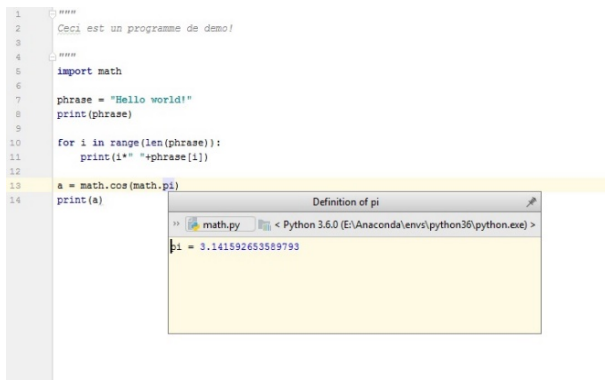
30. **Définition de fonction** Vous pouvez accéder à la définition d'une fonction ou d'une variable. Pour cela, placez votre curseur sur un élément et allez dans **View > Quick Definition**.

<sup>3</sup> Vous pouvez aussi utiliser les raccourcis clavier associés.

```

1 """
2 Ceci est un programme de demo!
3
4 """
5 import math
6
7 phrase = "Hello world!"
8 print(phrase)
9
10 for i in range(len(phrase)):
11     print(i*" "+phrase[i])
12
13 a = math.cos(math.pi)
14 print(a)

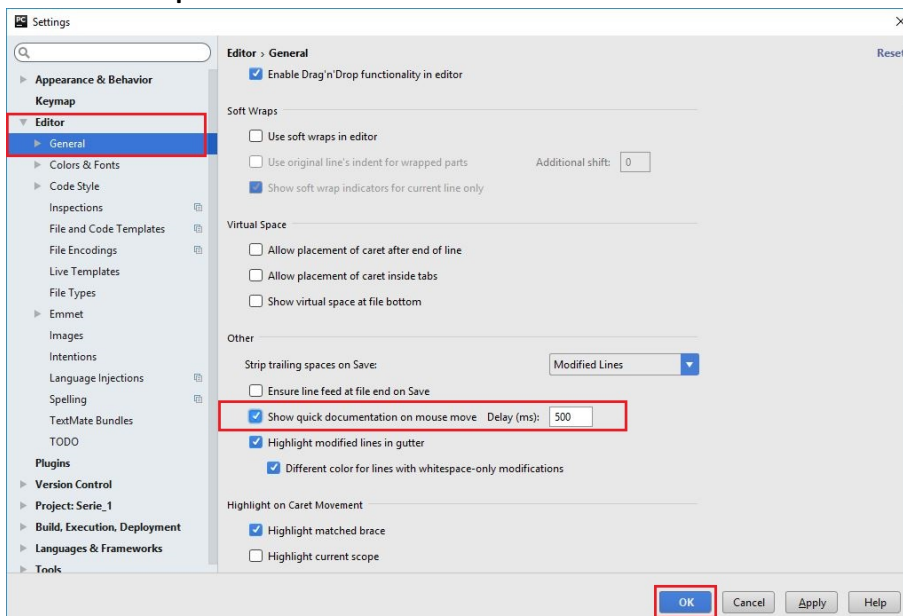
```



The image shows a code editor window with a Python script. A tooltip titled "Definition of pi" is displayed over the variable `math.pi` on line 13. The tooltip shows the definition: `math.pi = 3.141592653589793`.

Vous pouvez obtenir un accès complet au fichier où l'élément est défini. Pour cela, utilisez le raccourcis **CTRL + Clic gauche** sur Windows (ou **CMD + Clic gauche** sur MAC) en cliquant sur l'élément souhaité. Le fichier contenant la définition de l'élément s'ouvrira dans l'éditeur de texte. Cela est très utile si vous voulez retrouver et changer la valeur d'une variable ou la définition d'une fonction.

Vous pouvez faire en sorte de voir la définition d'un élément en passant votre souris sur celui-ci. Pour cela, allez dans l'onglet **File > Settings**, puis dans l'onglet **Editor > General**. Ici, cochez la case **Show quick documentation on mouse over**.



31. **Refactoring** Pycharm permet d'utiliser des refactors, qui permettent de modifier facilement votre code. Pour cela, faites un clic droit sur un élément et affichez les **Refactors**. Selon l'élément sur lequel vous cliquez, différents refactors seront disponibles. Sur une variable, vous pouvez par exemple la renommer : cela la changera dans tous votre script.



```
1 """
2 Ceci est un programme de demo!
3
4 """
5 import math
6
7 phrase = "Hello world!"
8 print(phrase)
9
10 for i in range(len(phrase)):
11     print(i*" "+phrase)
12
13 a = math.cos(math.pi)
14 print(a)
```

The image shows a code editor window with a Python script. The script contains a docstring, an import statement, a variable assignment, a print statement, and a loop. A yellow highlight is under the line `for i in range(len(phrase)):`. A context menu is open over this line, listing various actions such as Copy Reference, Paste, Refactor, and Run 'Demo'. The Refactor submenu is also visible, showing options like Rename, Change Signature, Move, Copy, and Extract.