

## Installation de Python

Commençons par une bonne nouvelle : sous Ubuntu, Python est déjà pré-installé.

Pour Windows et Mac OS X, la première étape consiste à installer l'implémentation officielle *CPython*.

### EN PRATIQUE « Vieilles » versions de Python sous Mac OS X

Mac OS X est livré en standard avec Python. Malheureusement, celui-ci n'est mis à jour qu'à chaque sortie d'une nouvelle version du système, soit environ tous les deux ans. On se retrouve souvent avec une version de Python largement dépassée. Il est donc indispensable d'installer la version qui nous intéresse à côté de celle existante et qui cohabitera en très bon voisinage avec cette dernière.

### EN PRATIQUE Version de Python

La version à installer est la 2.7 et non la 3, cette dernière n'étant pas compatible avec Django. Nous recommandons d'installer la dernière version de la lignée 2.7. À l'heure où ces lignes sont écrites, il s'agit de la version 2.7.3.

Pour la télécharger, il faut se rendre dans la section [Download](#) de [www.python.org](http://www.python.org). Différents paquets sont disponibles. Nous avons choisi :

- pour Windows, le paquet [Windows x86 MSI Installer](#), comprenant un installeur destiné aux Windows 32 bits ;
- pour Mac OS X, le paquet [Mac OS X 64-bit/32-bit x86-64/i386 Installer](#). C'est en réalité une image disque [DMG](#) qu'il suffit de monter en l'ouvrant.

► [www.python.org](http://www.python.org)

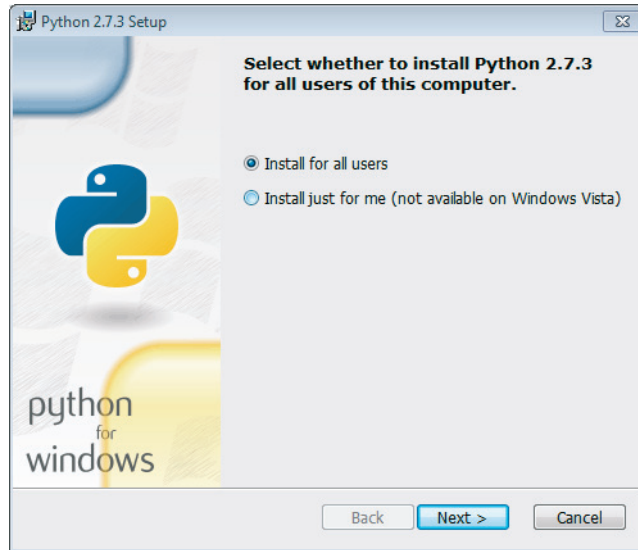
## Pour Windows

Une fois le paquet téléchargé, il suffit de l'exécuter et de suivre les étapes de l'assistant d'installation. À la première étape, l'assistant vous demande si vous désirez installer Python pour tous les utilisateurs de l'ordinateur ou juste pour l'utilisateur courant ; il est recommandé de choisir [Install for all users](#).

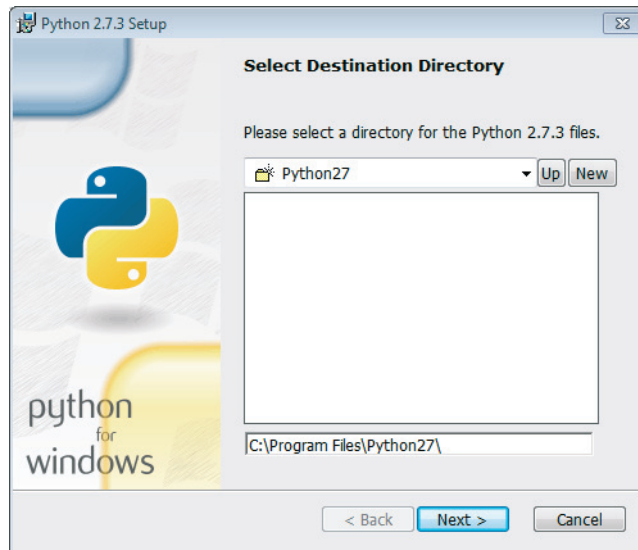
L'assistant demande ensuite de choisir un emplacement pour l'installation. Afin de respecter les standards Windows, nous recommandons d'installer Python dans [Program Files](#) et non à la racine du disque système, comme proposé par défaut par l'installeur.

**Figure A-4**

Première étape de l'assistant d'installation

**Figure A-5**

Deuxième étape de l'assistant d'installation



À la troisième étape, l'installeur demande quels composants de Python installer. Nous recommandons de tout sélectionner, afin de tout installer.

**Figure A-6**

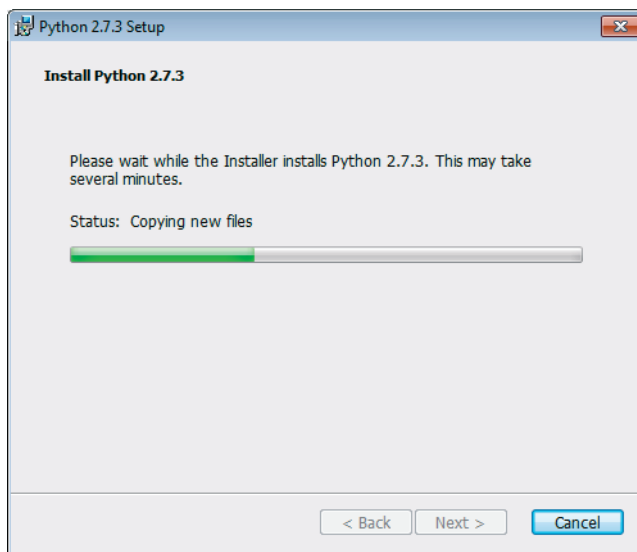
Troisième étape de l'assistant d'installation



À l'étape suivante, l'installation commence.

**Figure A-7**

Quatrième étape de l'assistant d'installation



Python demeurant un langage et un environnement somme toute très léger, l'installation est terminée après quelques minutes. Il ne reste plus qu'à cliquer sur le bouton *Finish* pour fermer l'assistant.

**Figure A-8**  
Dernière étape de l'assistant  
d'installation



## Pour Mac OS X

Une fois l'image montée, il suffit d'exécuter l'application `Python.mpkg` et de suivre les étapes de l'assistant d'installation. À la fin du processus, on obtient cette fenêtre :

**Figure A-9**  
Dernière étape de l'assistant  
d'installation



Cet installateur met Python dans le dossier `Applications` de votre Mac.

## Vérification de l'installation

Il reste à valider la bonne installation du langage en lançant par exemple la ligne de commande graphique Python :

- *Démarrer > Tous les programmes > Python > IDLE (Python GUI)* sous Windows ;
- *Applications > Python 2.7 > IDLE* sous Mac OS X.

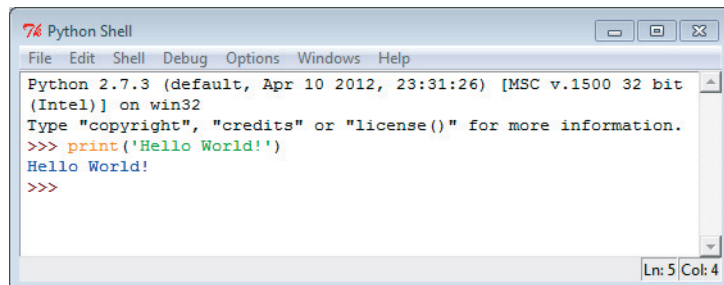
Ensuite, il suffit de saisir un peu de code, par exemple le sempiternel « hello world », bien connu des programmeurs :

### EXEMPLE A.1 Petit programme de test

```
print('Hello World !')
```

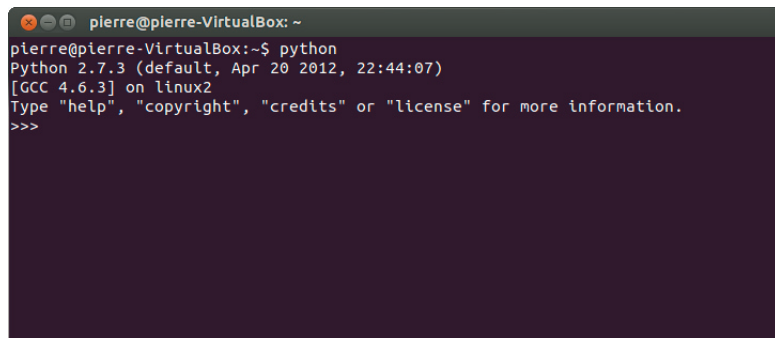
Ce code devrait afficher le texte « Hello World ! » dans la console, comme illustré sur la figure suivante. Vous pourriez refaire l'exercice avec « Bonjour Monde ! », mais sachez que cela fait tout de suite beaucoup moins pro.

**Figure A-10**  
Test de l'installation



Sous Ubuntu, il suffit de lancer un terminal (via le *Tableau de bord*, on lance l'application *Terminal*) et de taper la commande `python`. Apparaît alors le numéro de version du langage :

**Figure A-11**  
Version de Python  
include dans Ubuntu



## Installation de Django

Django, écrit en Python, n'est pas livré en standard avec ce dernier. Il s'agit d'un framework tiers à installer manuellement.

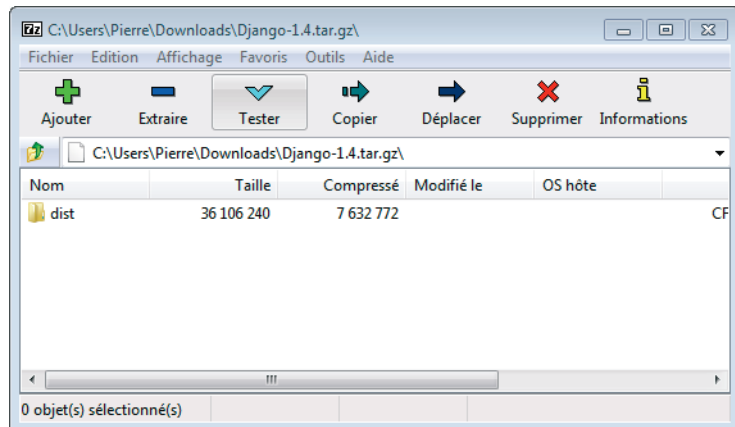
Pour Windows et Mac OS X, le framework est disponible en téléchargement sur le site du projet Django. Il faut se rendre dans la section [Download](#) et prendre la dernière version officielle. À l'heure où ces lignes sont écrites, il s'agit de la version 1.4. Le fichier à télécharger se présente sous la forme d'un fichier [tar.gz](#), en d'autres mots une archive TAR qui a été compressée en format GZIP.

► [www.djangoproject.com](http://www.djangoproject.com)

### Pour Windows

Il faut utiliser une application tierce qui va ouvrir, décompresser et extraire les fichiers d'installation de Django que renferme le fichier [tar.gz](#). L'application que nous avons utilisée est 7-Zip, disponible gratuitement. D'autres décompresseurs et extracteurs peuvent bien entendu être utilisés. Ouvrez le fichier [.gz](#) à l'aide de l'application 7-Zip. Un seul dossier se présente dans l'explorateur de l'archive : [dist](#).

**Figure A-12**  
Contenu de l'archive TAR



Double-cliquez sur le dossier [dist](#) pour en afficher le contenu. Un seul fichier s'y trouve : [Django-1.4.tar](#). À nouveau, double-cliquez sur ce fichier TAR pour en afficher le contenu. Dans cette archive, se trouve un seul dossier, nommé [Django-1.4](#) ; cliquez sur le bouton [Extraire](#) de 7-Zip. Un emplacement destiné à accueillir le dossier extrait vous est demandé ; choisissez celui qui vous convient le mieux.

L'archive extraite, nous pouvons lancer l'installateur qui s'y trouve. Cet installateur étant écrit en Python, on ne peut pas le lancer comme un exécutable Windows traditionnel en double-cliquant dessus. Il va falloir le lancer en ligne de commande en utilisant Python.

Il faut d'abord lancer une ligne de commande Windows, avec les droits d'administrateur. La ligne de commande Windows se trouve dans [Menu démarrer>Tous les programmes>Accessoires>Invite de commande](#). Pour la lancer avec les droits d'administrateur, il faut cliquer droit sur son raccourci et choisir [Exécuter en tant qu'administrateur](#).

Une fois l'invite de commande lancée, il faut se rendre, à l'aide de la commande `cd`, dans le dossier d'installation de Django que l'on vient d'extraire. Par exemple, ayant extrait ce dossier dans `C:\Users\Pierre\Downloads\`, nous avons tapé la ligne de commande suivante :

#### EXEMPLE A.2 Positionnement dans le dossier d'installation

```
cd Downloads\Django-1.4
```

Nous n'avons pas spécifié le chemin absolu dans la commande, car en ouvrant la console, celle-ci nous avait déjà positionnés dans le dossier `c:\Users\Pierre`.

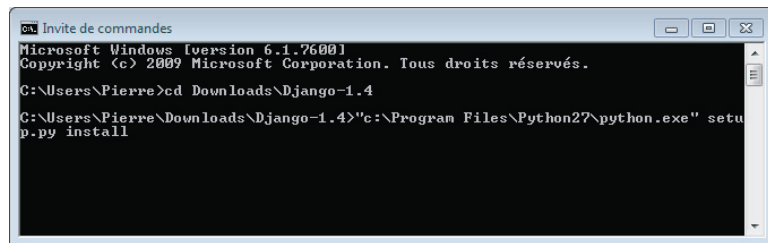
Pour lancer l'installation de Django, il faut entrer la commande suivante :

#### SYNTAXE. Lancement de l'installation

```
"c:\Program Files\Python27\python.exe" setup.py install
```

La figure suivante illustre l'utilisation de ces deux commandes.

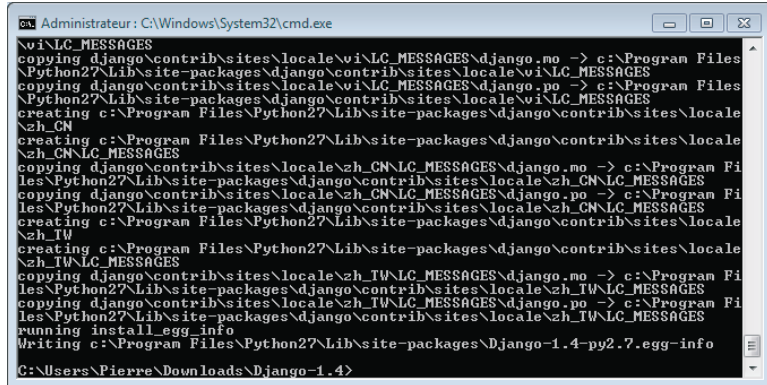
**Figure A-13**  
Lancement de l'installation de Django



Si aucune erreur ne survient, c'est que l'installation s'est bien déroulée.

**Figure A-14**

Installation terminée de Django



```
Administrateur : C:\Windows\System32\cmd.exe
vi\LC_MESSAGES
copying django\contrib\sites\locale\vi\LC_MESSAGES\django.mo -> c:\Program Files
\Python27\Lib\site-packages\django\contrib\sites\locale\vi\LC_MESSAGES
copying django\contrib\sites\locale\vi\LC_MESSAGES\django.po -> c:\Program Files
\Python27\Lib\site-packages\django\contrib\sites\locale\vi\LC_MESSAGES
creating c:\Program Files\Python27\Lib\site-packages\django\contrib\sites\locale
\zh_CN
creating c:\Program Files\Python27\Lib\site-packages\django\contrib\sites\locale
\zh_CN\LC_MESSAGES
copying django\contrib\sites\locale\zh_CN\LC_MESSAGES\django.mo -> c:\Program Fi
les\Python27\Lib\site-packages\django\contrib\sites\locale\zh_CN\LC_MESSAGES
copying django\contrib\sites\locale\zh_CN\LC_MESSAGES\django.po -> c:\Program Fi
les\Python27\Lib\site-packages\django\contrib\sites\locale\zh_CN\LC_MESSAGES
creating c:\Program Files\Python27\Lib\site-packages\django\contrib\sites\locale
\zh_TW
creating c:\Program Files\Python27\Lib\site-packages\django\contrib\sites\locale
\zh_TW\LC_MESSAGES
copying django\contrib\sites\locale\zh_TW\LC_MESSAGES\django.mo -> c:\Program Fi
les\Python27\Lib\site-packages\django\contrib\sites\locale\zh_TW\LC_MESSAGES
copying django\contrib\sites\locale\zh_TW\LC_MESSAGES\django.po -> c:\Program Fi
les\Python27\Lib\site-packages\django\contrib\sites\locale\zh_TW\LC_MESSAGES
running install_egg_info
Writing c:\Program Files\Python27\Lib\site-packages\Django-1.4-py2.7.egg-info
G:\Users\Pierre\Downloads\Django-1.4>
```

## Pour Mac OS X

Pour l'archive sous Mac, il faut utiliser la ligne de commande. Ouvrez d'abord un *Terminal* (dans *Applications > Utilitaires*). Rendez-vous ensuite dans le dossier où a été téléchargée l'archive, à l'aide de la commande `cd` :

### EXEMPLE A.3 Positionnement dans le dossier de téléchargement

```
cd Downloads
```

Puis lancez la décompression de l'archive et rendez-vous dans le dossier créé :

### SYNTAXE. Décompression de l'archive

```
tar xzvf Django-1.4.tar.gz
cd Django-1.4
```

Ensuite, lancez l'installation de Django à l'aide de l'interpréteur Python que nous venons d'installer (et non de celui fourni de base dans Mac OS X). Il se trouve dans *Library > Frameworks*. Utilisez la commande `sudo` car l'installation doit se faire en tant qu'administrateur.

### SYNTAXE. Installation de Django

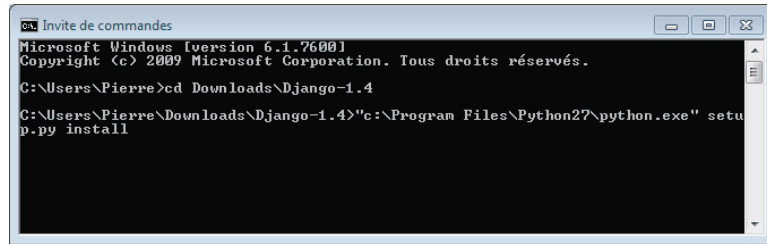
```
sudo /Library/Frameworks/Python.framework/Versions/2.7/bin/python
setup.py install
```

Attention, il va falloir entrer le mot de passe administrateur.

La figure suivante illustre l'utilisation de cette commande.



**Figure A-15**  
Lancement de l'installation  
de Django

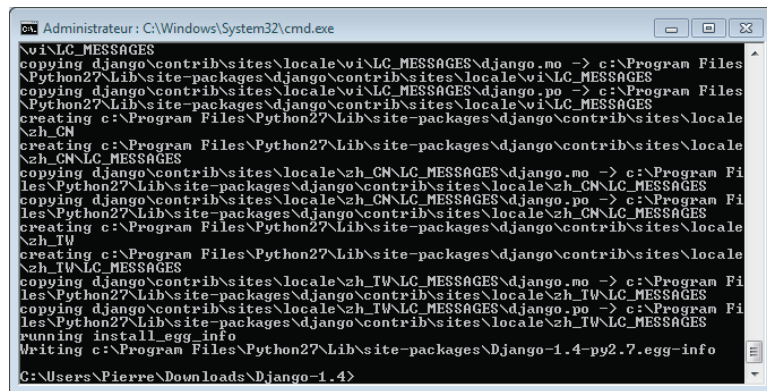


```
Microsoft Windows [version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Tous droits réservés.

C:\Users\Pierre>cd Downloads\Django-1.4
C:\Users\Pierre\Downloads\Django-1.4>"c:\Program Files\Python27\python.exe" setup.py install
```

Si aucune erreur ne survient, l'installation s'est bien déroulée.

**Figure A-16**  
Installation terminée de Django



```
Administrateur : C:\Windows\System32\cmd.exe

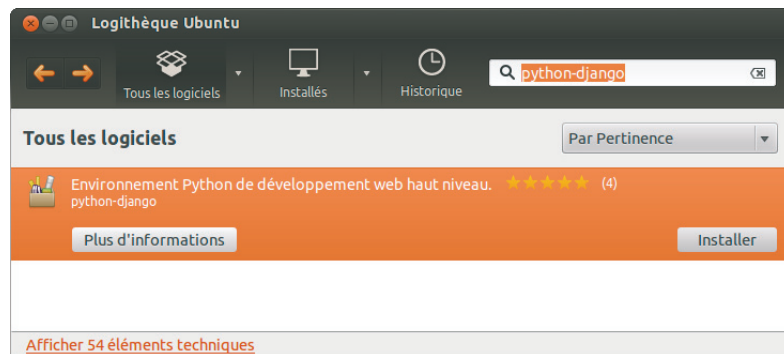
\n\LC_MESSAGES
copying django\contrib\sites\locale\vi\LC_MESSAGES\django.mo -> c:\Program Files\Python27\Lib\site-packages\django\contrib\sites\locale\vi\LC_MESSAGES
copying django\contrib\sites\locale\vi\LC_MESSAGES\django.po -> c:\Program Files\Python27\Lib\site-packages\django\contrib\sites\locale\vi\LC_MESSAGES
creating c:\Program Files\Python27\Lib\site-packages\django\contrib\sites\locale\zh_CN
creating c:\Program Files\Python27\Lib\site-packages\django\contrib\sites\locale\zh_CN\LC_MESSAGES
copying django\contrib\sites\locale\zh_CN\LC_MESSAGES\django.mo -> c:\Program Files\Python27\Lib\site-packages\django\contrib\sites\locale\zh_CN\LC_MESSAGES
copying django\contrib\sites\locale\zh_CN\LC_MESSAGES\django.po -> c:\Program Files\Python27\Lib\site-packages\django\contrib\sites\locale\zh_CN\LC_MESSAGES
creating c:\Program Files\Python27\Lib\site-packages\django\contrib\sites\locale\zh_TW
creating c:\Program Files\Python27\Lib\site-packages\django\contrib\sites\locale\zh_TW\LC_MESSAGES
copying django\contrib\sites\locale\zh_TW\LC_MESSAGES\django.mo -> c:\Program Files\Python27\Lib\site-packages\django\contrib\sites\locale\zh_TW\LC_MESSAGES
copying django\contrib\sites\locale\zh_TW\LC_MESSAGES\django.po -> c:\Program Files\Python27\Lib\site-packages\django\contrib\sites\locale\zh_TW\LC_MESSAGES
running install_egg_info
Writing c:\Program Files\Python27\Lib\site-packages\Django-1.4-py2.7.egg-info

C:\Users\Pierre\Downloads\Django-1.4>
```

## Pour Ubuntu

Django est disponible dans la [Logithèque](#) d'Ubuntu. Il suffit de chercher le logiciel `python-django` et, celui-ci trouvé, de cliquer sur [Installer](#).

**Figure A-17**  
Django dans la Logithèque  
Ubuntu



## Vérification de l'installation

Sous Windows et Mac OS X, on peut tester si l'installation s'est exécutée correctement en lançant une console Python et en tapant le code suivant :

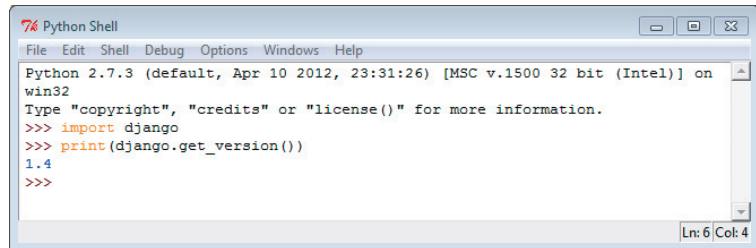
**SYNTAXE.** Vérification de la bonne installation de Django sous Windows et Mac OS X

```
>>> import django
>>> print(django.get_version())
1.4
```

Le code doit afficher la version de Django, soit « 1.4 », comme illustré à la figure suivante.

**Figure A-18**

Vérification de la bonne installation de Django sous Windows et Mac OS X



Sous Ubuntu, on peut tester la bonne installation de Django en lançant un terminal et en tapant le code suivant :

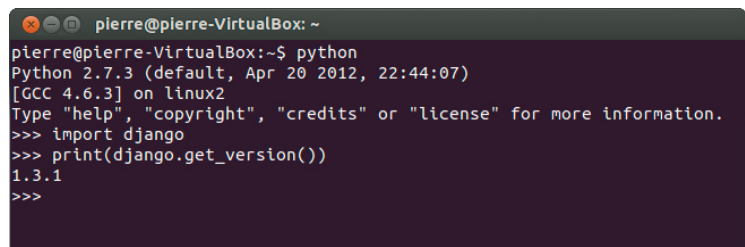
**SYNTAXE.** Vérification de la bonne installation de Django sous Ubuntu

```
python
>>> import django
>>> print(django.get_version())
1.3.1
```

Le code doit afficher la version de Django, soit « 1.3.1 », comme illustré à la figure suivante.

**Figure A-19**

Vérification de la bonne installation de Django sous Ubuntu



## Installation de Java

Si Java n'est pas déjà installé sur votre ordinateur (ce qui est peu probable), vous devrez y remédier.

### Pour Windows et Mac OS X

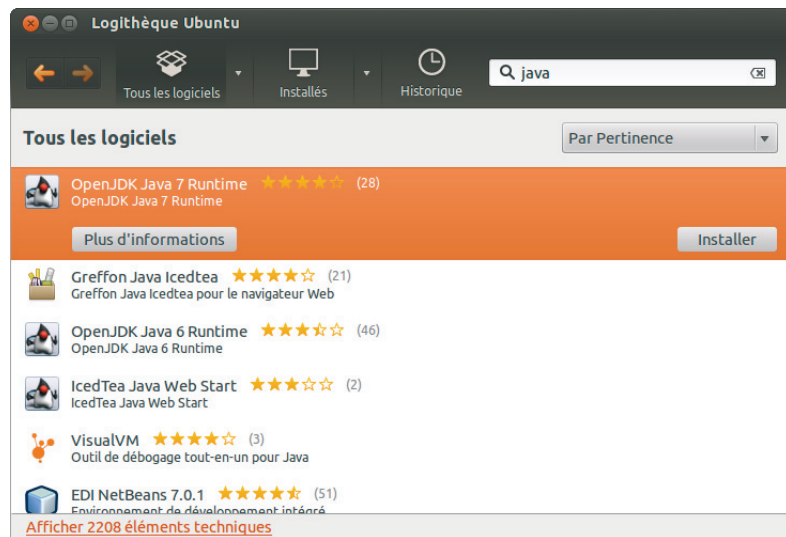
Sur la page d'accueil du site officiel de Java, un lien mis en évidence vous permet de télécharger l'installateur. Choisissez la version de Java correspondant à votre système (32 ou 64 bits). L'installation ne nécessite pas de paramétrage particulier.

► [www.java.com](http://www.java.com)

### Pour Ubuntu

Rendez-vous à nouveau dans la *Logithèque*. Il suffit d'y chercher « Java », de sélectionner *OpenJDK Java 7 Runtime* et de cliquer sur *Installer*.

**Figure A-20**  
Installation de Java  
avec Ubuntu



## Installation d'Eclipse

Nous allons maintenant installer Eclipse, notre environnement de développement intégré.

Le téléchargement d'Eclipse se fait via le site officiel du projet pour Windows et Mac OS X. On retrouve sur cette page plusieurs versions de l'outil, chacune ciblant un langage de programmation ou un usage bien précis. Elles ne sont en fait que la version de base de l'outil sur laquelle ont été pré-installés les plug-ins ad hoc.

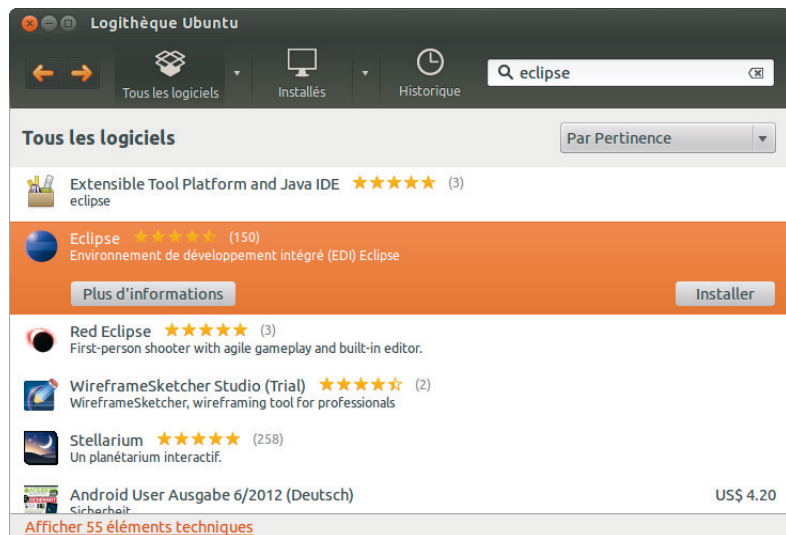
► <http://www.eclipse.org/downloads/>

Comme il n'en existe pas pour Python et Django, nous allons simplement télécharger la version de base, « Eclipse Classic » pour Windows ou Mac (version 3.7.2 au moment où nous écrivons ces lignes), et nous installerons par la suite les plug-ins qui nous intéressent.

Une fois téléchargé, Eclipse se présente sous la forme d'une archive (ZIP pour Windows et tar.gz pour Mac), qu'il suffit de décompresser comme nous l'avons fait pour Django. Elle contient un dossier reprenant tous les fichiers de l'application. Aucun installateur n'est donc à lancer ; il suffit de déplacer le dossier que l'on a décompressé de l'archive à l'emplacement de son choix. Pour notre part, nous avons choisi de déplacer le dossier dans `C:\Program Files` sur Windows et `Applications` pour Mac.

Pour Ubuntu, nous allons faire usage une fois encore de la *Logithèque*. Il suffit de chercher *Eclipse*, de le sélectionner et de cliquer sur *Installer*.

**Figure A-21**  
Installation d'Eclipse  
sous Ubuntu



Nous pouvons maintenant lancer Eclipse :

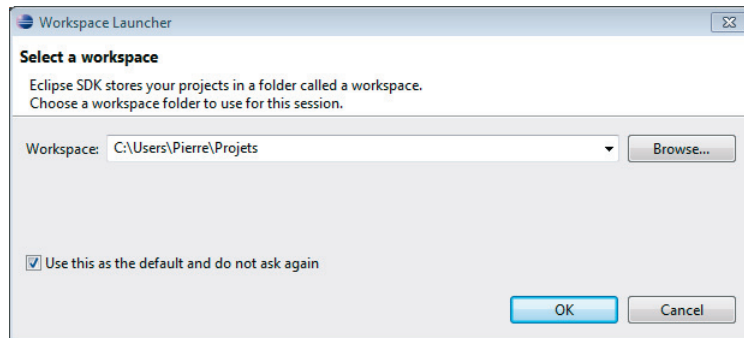
- Sous Windows, double-cliquez sur `eclipse.exe`.
- Sous Mac, si Java n'est pas installé à ce stade, *Mise à jour de logiciels* permet de le faire en affichant un dialogue le proposant.
- Sous Ubuntu, démarrez Eclipse à l'aide du tableau de bord.

Au démarrage, Eclipse nous demande de choisir un *workspace* (espace de travail) et nous propose, par défaut, d'utiliser `C:\Users\XXXXX\workspace` pour Windows et Ubuntu ou `/Users/XXXXX/Documents/workspace` pour Mac. Un workspace regroupe différents projets, spécifie leur emplacement sur le disque et enregistre un certain nombre de préférences propres à Eclipse. Définir plusieurs *workspaces* permet donc d'avoir des environnements de travail différents (paramétrés différemment) contenant des projets différents. Tout au long de ce livre, nous n'utilisons qu'un seul workspace.

Vous pouvez très bien utiliser le workspace proposé par défaut. Néanmoins, nous avons préféré créer et utiliser un nom plus convivial : `C:\Users\Pierre\Projets` pour Windows et Ubuntu ou `/Users/Pierre/Documents/Projets` pour Mac. Pour créer un nouveau workspace, il suffit d'entrer son nom dans la boîte de dialogue de choix de workspace.

Vous pouvez également cocher la case *Use this as the default and do not ask again* afin qu'au prochain démarrage, Eclipse ne vous demande plus quel workspace utiliser.

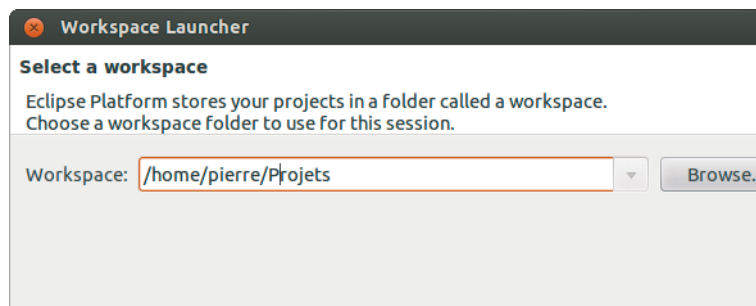
**Figure A-22**  
Création et utilisation  
du workspace  
`C:\Users\Pierre\Projets`  
(Windows)



**Figure A-23**

Création et utilisation  
du workspace

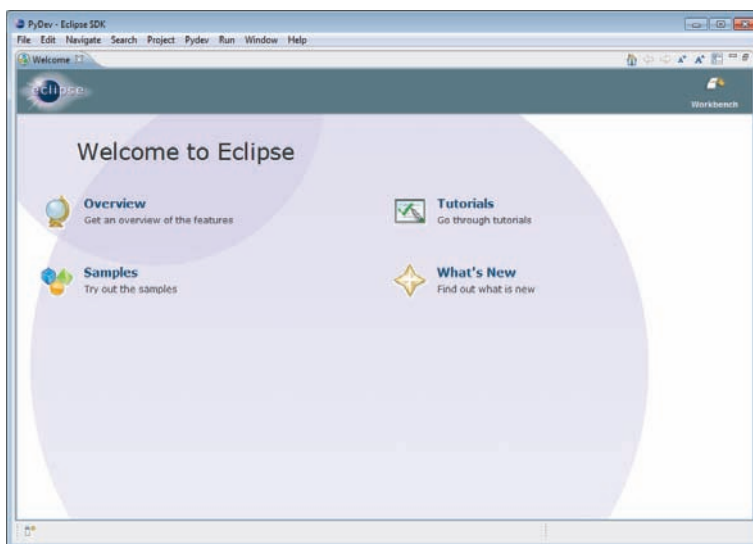
C:\Users\Pierre\Projets  
(Ubuntu)



Il reste à cliquer sur **OK**. Eclipse s'ouvre alors et affiche sa page d'accueil.

**Figure A-24**

Écran d'accueil d'Eclipse



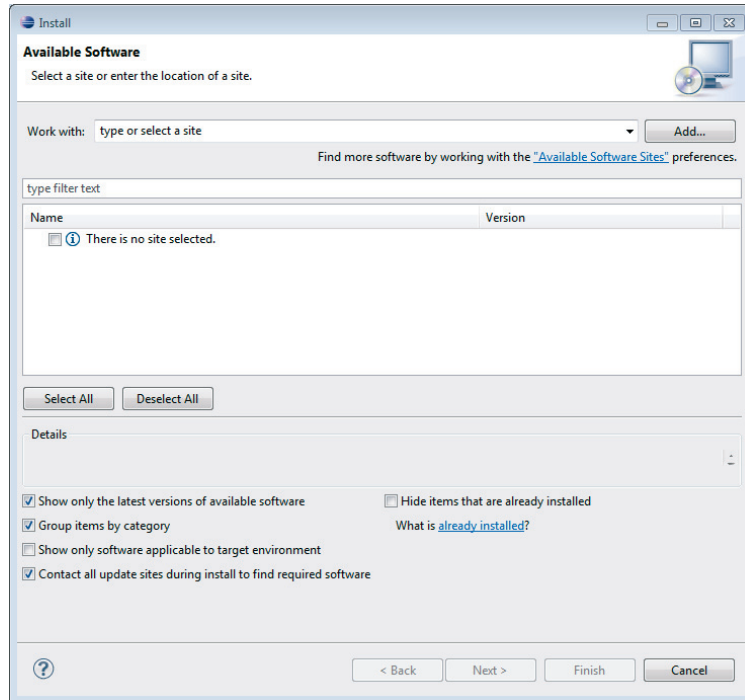
Et voilà ! Eclipse est installé, Nous n'avons plus à nous occuper que du plug-in PyDev.

## Installation du plug-in Eclipse PyDev

Avant de pouvoir commencer à développer en Python et Django dans Eclipse, il nous faut installer le plug-in PyDev.

Les plug-ins s'installent via le menu *Help > Install New Software...* La fenêtre suivante apparaît :

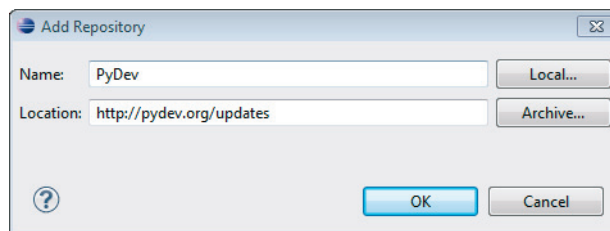
**Figure A-25**  
Écran d'installation de plug-ins



Tout d'abord, nous devons renseigner le *site* sur lequel se trouve le plug-in PyDev. Cliquez sur le bouton **Add...** et, dans la fenêtre qui apparaît, inscrivez **PyDev** et <http://pydev.org/updates>.

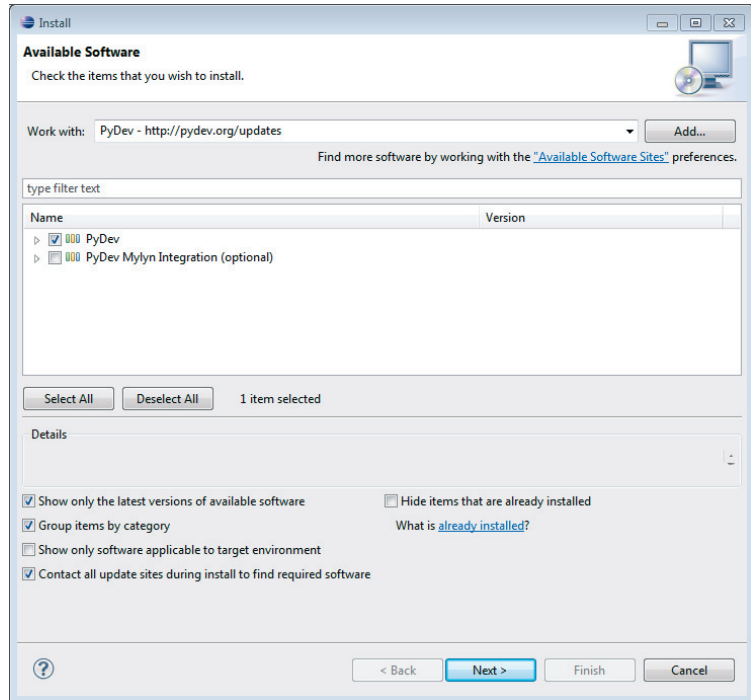
► [pydev.org/updates](http://pydev.org/updates)

**Figure A-26**  
Ajout du site pour  
le plug-in PyDev



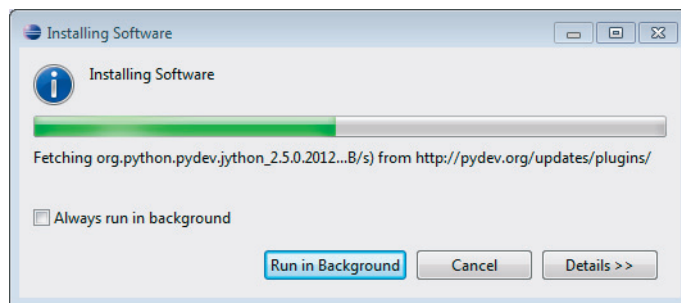
Cliquez sur **OK**. Choisissez ensuite **PyDev** dans la liste des plug-ins disponibles à installer.

**Figure A-27**  
Choix de PyDev



On clique sur *Next* deux fois, on accepte la licence, on clique sur *Finish* et l'installation débute.

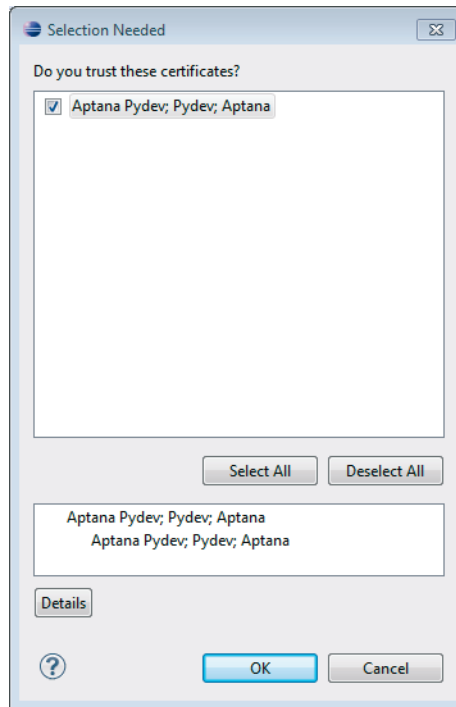
**Figure A-28**  
Installation de PyDev



Au cours de l'installation, Eclipse vous demandera de valider un certificat. Il suffit de s'assurer qu'ils sont tous sélectionnés et de cliquer sur *OK*.



**Figure A-29**  
Installation de PyDev



Il est possible qu'à ce stade, on vous demande de redémarrer votre ordinateur, pour que les plug-ins s'installent vraiment. Vous pouvez vous contenter de réexécuter Eclipse.

Il reste maintenant à configurer PyDev pour lui indiquer où se trouve le compilateur Python et ses bibliothèques (que nous avons installées au tout début de ce chapitre).

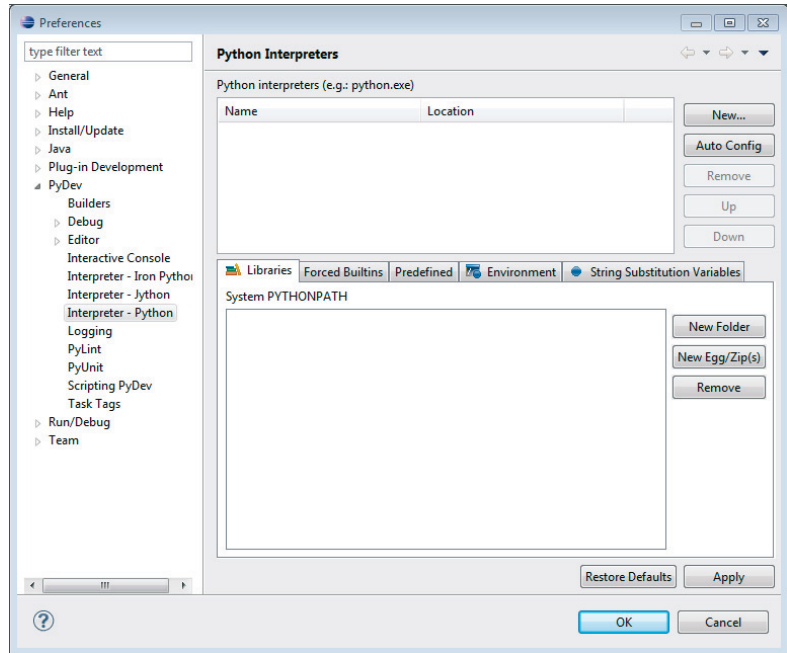
Pour ce faire, il faut aller dans le menu *Window > Preference* (ou *Eclipse > Preference*). Dans l'arborescence de paramètres, rendez-vous à la section *PyDev > Interpreter – Python*.

Cliquez sur le bouton *Auto Config*. Dans la fenêtre qui apparaît, choisissez tous les éléments du dossier d'installation de Python. Cliquez sur *OK* pour valider.

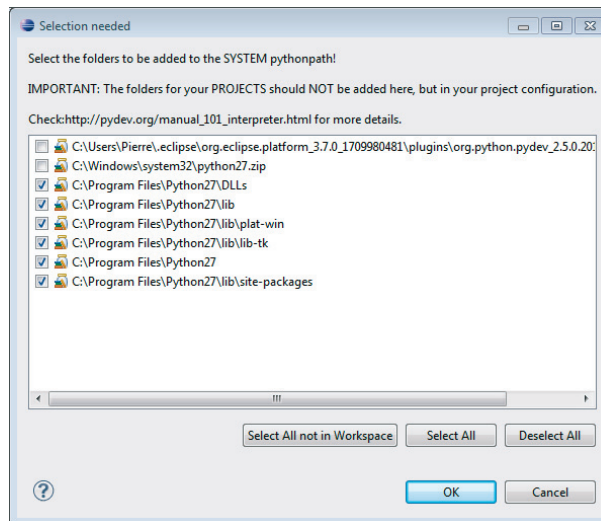
Si *Auto Config* ne marche pas, il suffit alors de faire *New* et de spécifier comme emplacement l'endroit où se trouve l'exécutable de l'interpréteur Python que l'on désire utiliser. Par exemple, pour Mac OS X, il s'agira de `/Library/Frameworks/Python.framework/Versions/2.7/bin/python`.

La configuration est maintenant terminée, cliquez sur *OK* pour fermer la fenêtre de préférences.

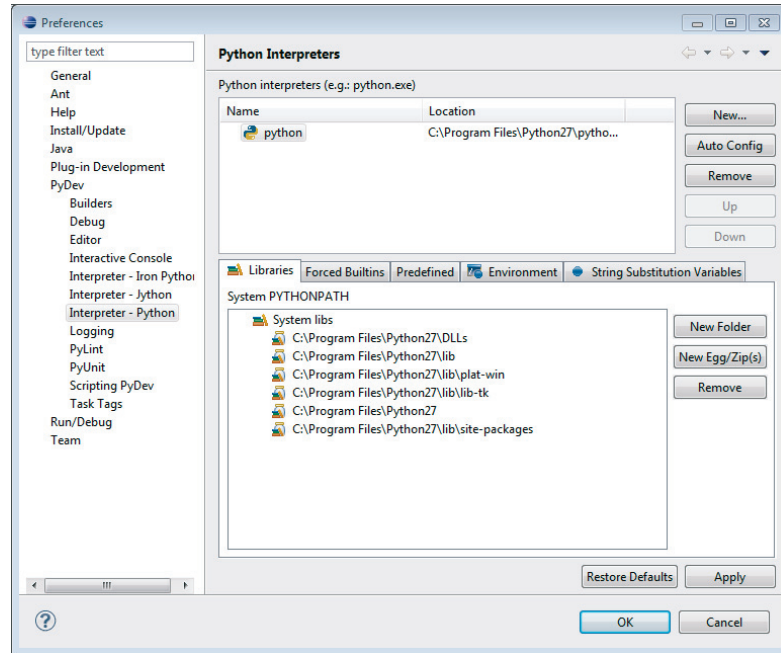
**Figure A-30**  
Configuration de PyDev



**Figure A-31**  
Configuration de PyDev



**Figure A-32**  
Configuration de PyDev



## Installation de Web Developer Tools pour Eclipse

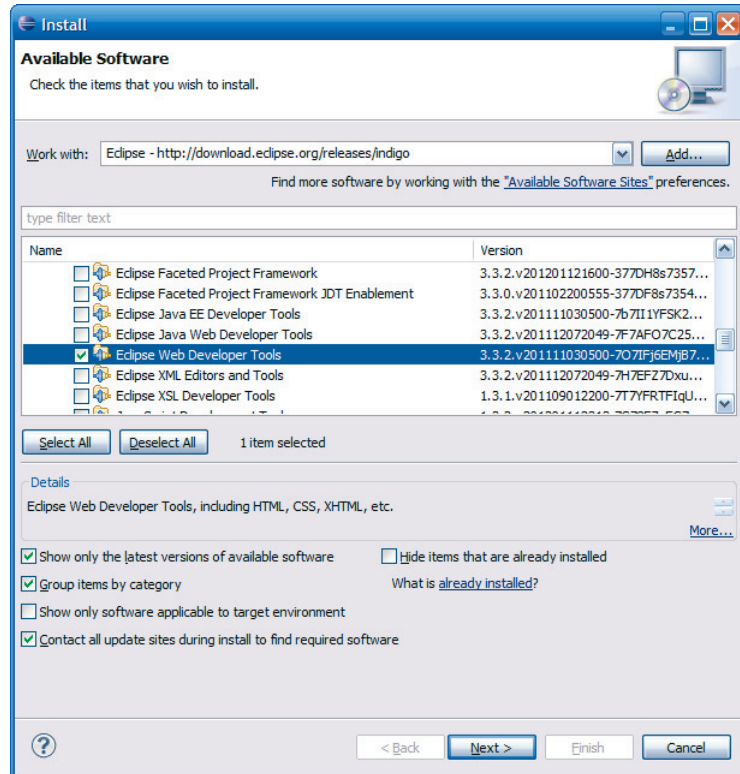
Ce paquet supplémentaire dans Eclipse nous sert dans le chapitre 8 pour éditer directement nos HTML. Il suffit de procéder de la même manière que pour PyDev.

Rendez-vous dans le menu *Help > Install New Software* et cliquez sur le bouton *Add*. Comme nom de site, indiquez ce que vous voulez, par exemple *Eclipse*. Comme emplacement (*location*), mettez <http://download.eclipse.org/releases/indigo> (du moins, si vous travaillez avec la version Indigo d'Eclipse). Si vous utilisez une autre version, il faut remplacer *indigo* dans l'adresse par le nom de celle-ci (regardez bien l'écran de démarrage d'Eclipse, il y est indiqué).

Le site ajouté, sélectionnez-le et patientez pendant que la liste des paquets disponibles se met à jour. Rendez-vous ensuite dans *Web, XML, Java EE and OSGi Enterprise Development* et sélectionnez le paquet *Eclipse Web Developer Tools*.

Suivez les instructions de l'assistant d'installation et répondez aux questions qu'il vous pose. Après un redémarrage d'Eclipse, vous voilà en mesure d'éditer vos HTML directement dans Eclipse.

**Figure A-33**  
Installation du plugin « Web Developer Tools »



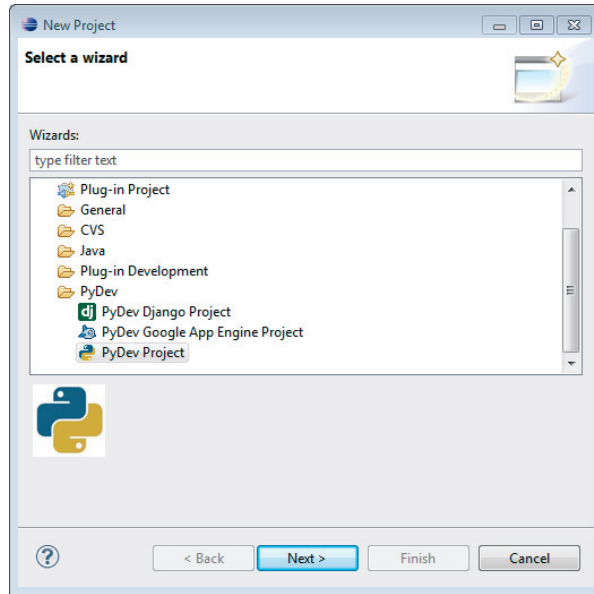
## Premier projet de test

Les différentes installations terminées, nous pouvons réaliser un premier petit projet de test. Cela permet de valider que tout fonctionne à la perfection. Ce petit projet n'a d'autre but que de réaliser un programme Python affichant toujours notre sempiternel texte « Hello World ! » à la console. Pour créer un nouveau projet, cliquez sur le menu **File > New > Project...** puis choisissez **PyDev Project**.

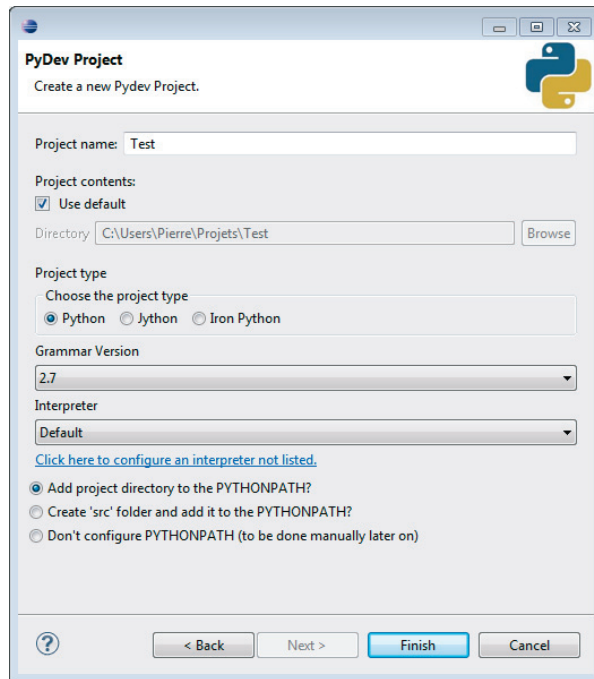
Après avoir cliqué sur **Next**, un assistant de création de projet se lance. À la première étape, il suffit de donner un nom au projet.

Cliquez alors sur **Finish**. Notre projet est créé et apparaît dans l'écran principal d'Eclipse.

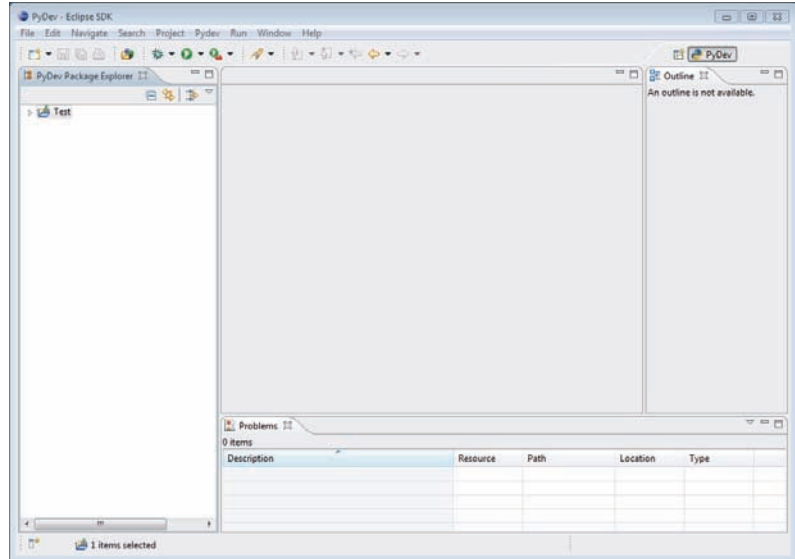
**Figure A-34**  
Création d'un projet de test



**Figure A-35**  
Création d'un projet de test

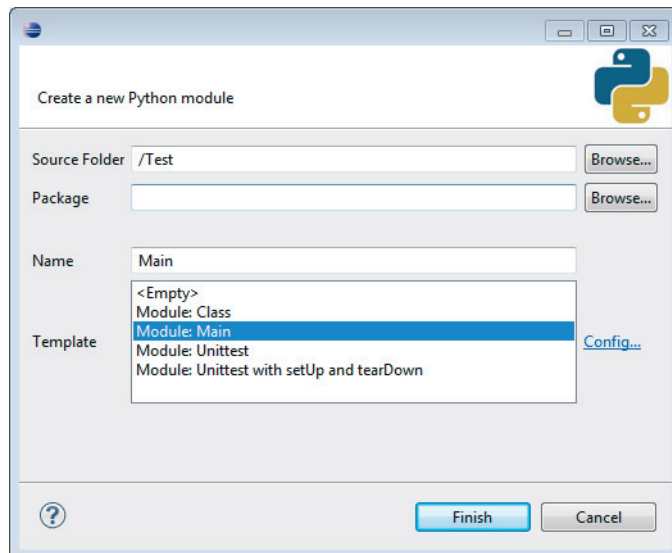


**Figure A-36**  
Création d'un projet de test



Il reste à ajouter à notre projet un fichier `py` qui va contenir nos sources. Cliquez-droit sur le projet `Test` et choisissez `New > PyDev Module`. Une fenêtre apparaît dans laquelle on précise un nom pour le projet (`Main`) et un *template* de base ; dans notre cas, nous allons choisir le template *Module: Main*, car nous allons écrire notre code au niveau du point d'entrée du programme.

**Figure A-37**  
Création d'un projet de test



Cliquez sur *Finish*, le fichier est alors créé. Supprimez-en tout le contenu, en particulier les commentaires ajoutés par défaut par Eclipse, car pour peu qu'ils contiennent des accents, Python plantera. Nous verrons comment gérer correctement les accents dans le chapitre 8. Cela nous permettra de garder le code proposé en standard dans ces fichiers.

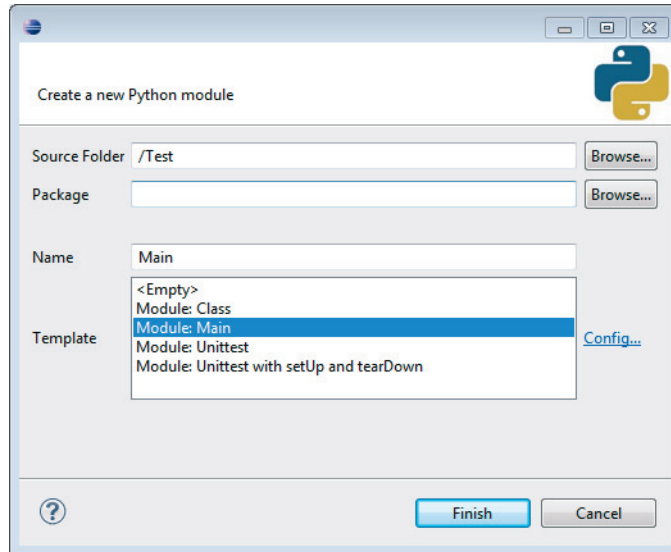
Il reste à insérer le code suivant dans le fichier fraîchement créé :

#### EXEMPLE A.4 HelloWorld ! en Python

```
if __name__ == '__main__':  
    print('Hello World !')
```

On peut maintenant exécuter le programme en utilisant l'icône *Run* représentée par un triangle blanc sur un rond vert. Une fois exécuté, le programme affiche « HelloWorld ! » à la console.

**Figure A-38**  
Création d'un projet de test



Et voilà ! Nous avons créé un programme Python en utilisant Eclipse et son plug-in PyDev. Nous sommes prêts pour des projets plus complexes ayant recours au framework Django.

## Ai-je bien compris ?

- Pourquoi faut-il installer Java alors que l'on va programmer en Python ?
- Eclipse est-il obligatoire pour développer un site en Django ?
- À quoi sert le plug-in Eclipse « PyDev » ?