

eclipsetotale.com

Le site francophone consacré au projet Eclipse et à OSGi

Nos formations

Filière Eclipse

[Formation Eclipse RCP](#)

[Formation OSGi](#)

Filière WebSphere

[Formation 'Administration de WebSphere Application Server'](#)

[Formation 'Administration par script de WebSphere Application Server'](#)

Articles et tutoriaux

[Présentation de la fondation Eclipse et de ses projets](#)

[Installer Eclipse](#)

[Premiers pas avec Eclipse](#)

[Présentation d'Eclipse RCP](#)

[Tutorial : Développement de plug-ins \(1ère partie\)](#)

[Tutorial : Développement de plug-ins \(2ème partie\)](#)

[Tutorial : Développement d'applications Eclipse RCP \(1ère partie\)](#)

[Présentation de WTP](#)

[Tutorial : Servlet et JSP avec WebTools](#)

[Tutorial : EJB3 avec WebTools et JBoss](#)

[Présentation de BIRT](#)

[Présentation de TPTP](#)

[Architectures SOA et Eclipse](#)

[Europa et Eclipse 3.3](#)

[Callisto et Eclipse 3.2](#)

[Didacticiel 'développons en Java avec Eclipse' par Jean-Michel Doudoux](#)

↳ Consulter l'actualité du projet Eclipse

(publié le 10/01/2007 - Versions utilisées : Eclipse 3.2.1 - JDK 5.0)

Tutorial : premier pas avec Eclipse

La fondation Eclipse propose un environnement de développement Java qui rencontre depuis plusieurs années un vif succès.

Ce tutorial a pour objectif de donner aux nouveaux utilisateurs d'Eclipse, les bases pour bien démarrer. La première partie illustre les étapes de la prise en main d'Eclipse pour le développement d'applications Java. La deuxième partie présente certaines des fonctionnalités qui ont assuré la popularité d'Eclipse.

Sommaire

[1 Créer et tester une première application Java](#)

[1.1 Création d'un premier projet](#)

[1.2 Création d'une première classe](#)

[1.3 Exécution](#)

[1.4 Débogage](#)

[2 Trucs et astuces](#)

[2.1 Fonctionnalités importantes de l'éditeur](#)

- Complétion et import
- Complétion et modèles
- Correction rapide
- Gestion des imports
- Formatage
- Génération de blocs try/catch
- Génération de méthodes
- Scinder la zone d'édition

[2.2 Navigation](#)

[2.3 Plus de trucs et astuces](#)

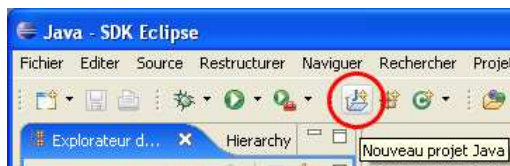
↳ Créer et tester une première application Java

La prise en main d'Eclipse est relativement simple. Ce tutorial commence après les étapes suivantes :

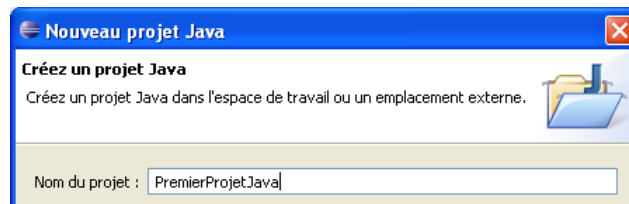
- téléchargement et installation d'un JDK, d'Eclipse et des traductions d'Eclipse (voir notre page '[Installer Eclipse](#)')
- exécution d'Eclipse et sélection d'un répertoire qui contiendra le répertoire de travail.
- fermeture de la fenêtre 'Welcome'.

↳ Création d'un projet Java

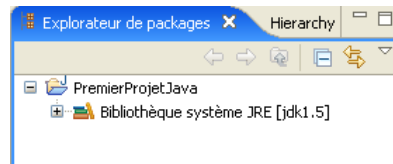
La première étape pour développer une application Java avec Eclipse est de créer un 'Projet Java'. L'ouverture de l'assistant se fait en utilisant le bouton correspondant dans la barre de boutons :



La seule information nécessaire pour l'assistant est le nom du projet. Une fois le nom saisi, le bouton **'Terminer'** est activé, cliquer sur ce bouton pour demander la création du projet (L'assistant est composé d'une deuxième page qui permet de configurer de façon précise le chemin de compilation du projet. Les paramètres offerts par cette deuxième page peuvent être fait ultérieurement dans les propriétés du projet).

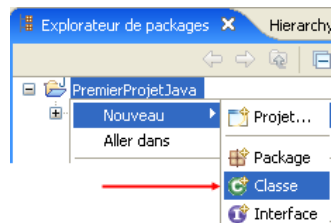


Après la création, le projet apparaît dans la vue 'Explorateur de projets' :



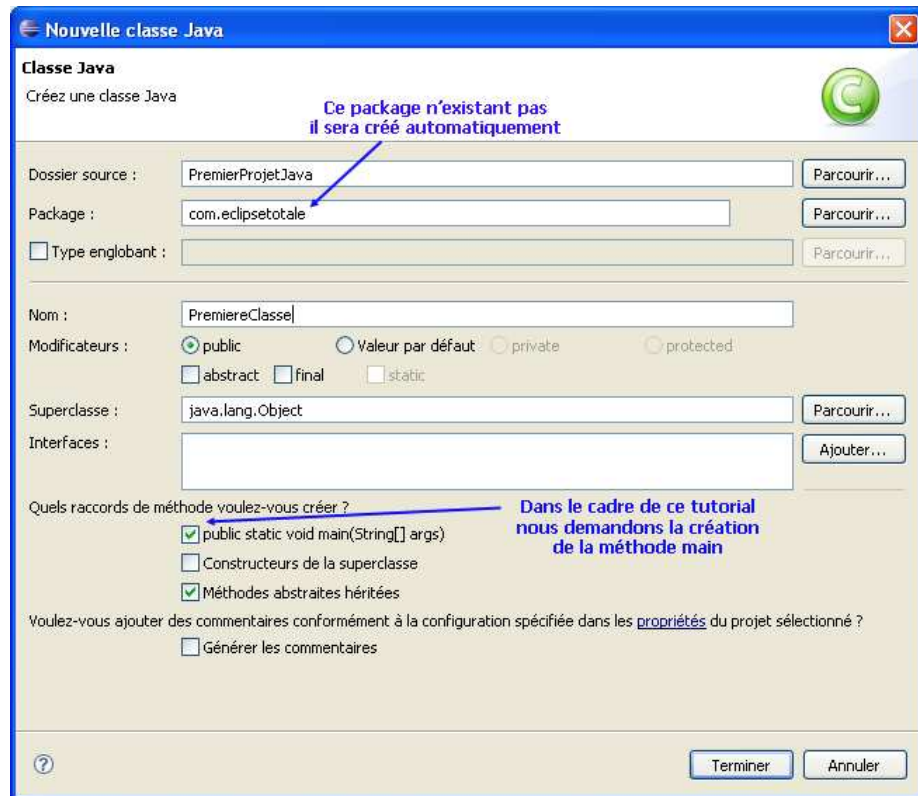
➔ **Création d'une classe**

Pour ouvrir l'assistant de création de classes Java: faire afficher le menu contextuel du projet Java précédemment créé et sélectionner l'option 'Nouveau->Classe'.

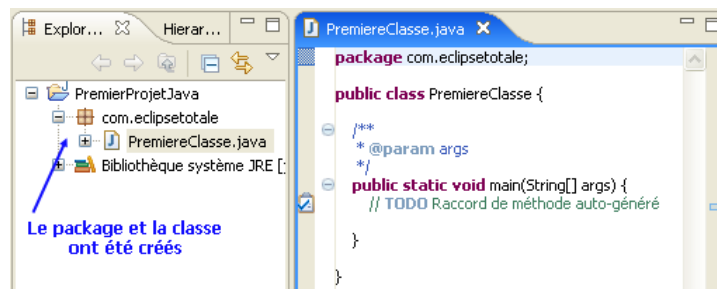


Les informations importantes attendues par cet assistant sont le nom de la classe et le nom de son package.

Entrer un nom de package (le package sera créé automatiquement), saisir le nom de la classe et cocher la case indiquant à l'assistant de générer une méthode main.

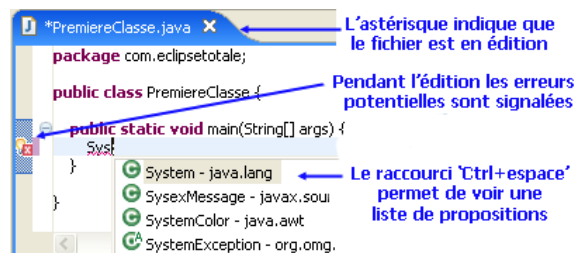


Après avoir cliqué sur le bouton 'Terminer', la classe est créée et ouverte en édition :

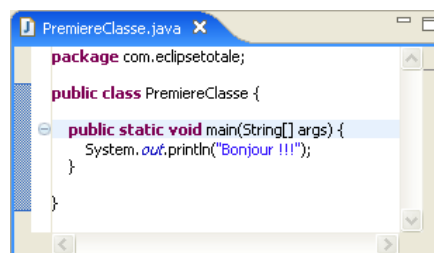


Dans l'éditeur, compléter la méthode 'main'. Une fois le code saisi, demander l'enregistrement (Ctrl+S ou menu Fichier), la classe est enregistrée **et** compilée.


Pendant la saisie :

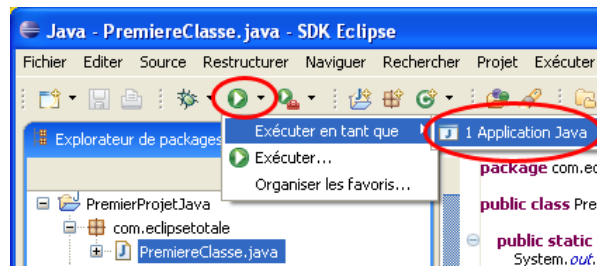


Après l'enregistrement :

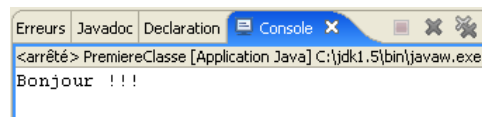


➔ Exécution

Pour demander l'exécution de la classe : déplier le menu associé au bouton  et sélectionner l'option 'Exécuter en tant que...' -> 'Application Java' :

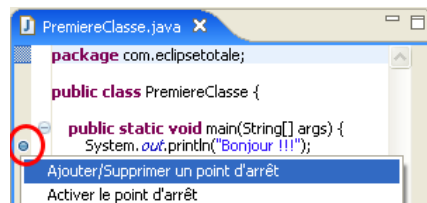


La vue 'Console' affiche le résultat :



➔ Débogage

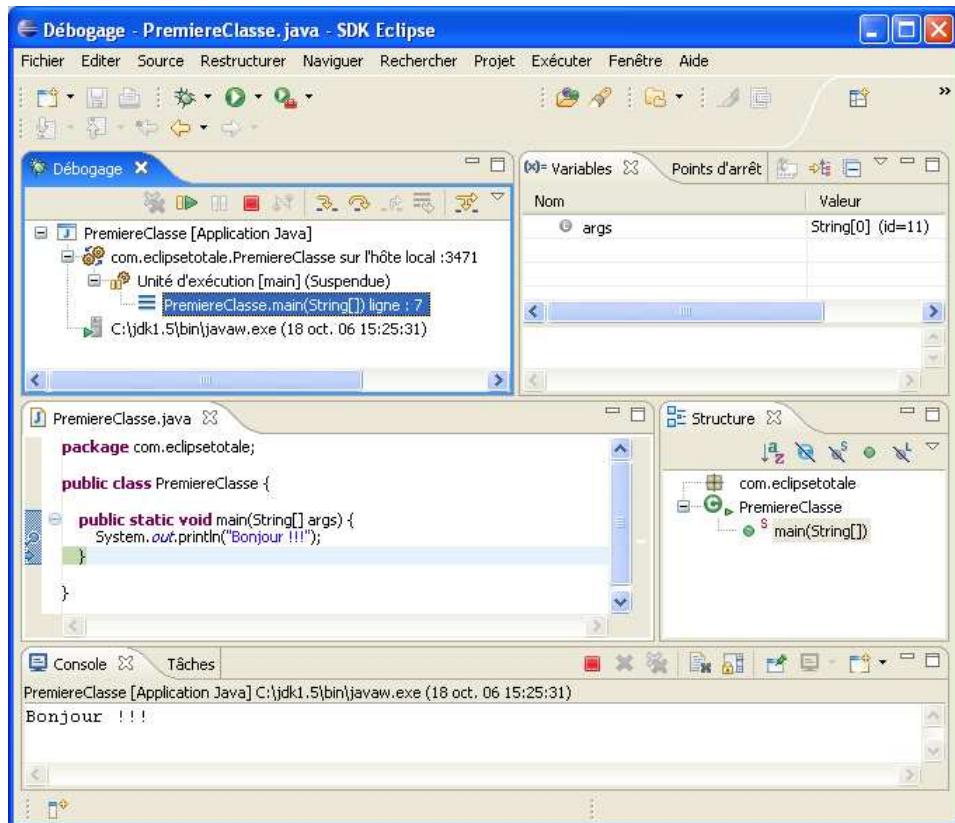
Placer un point d'arrêt en double-cliquant dans la marge ou en utilisant le menu contextuel dans la marge :






Lancer l'application en mode 'Débogage' :



Lorsque que le point d'arrêt est rencontré, Eclipse propose d'afficher le débogueur :



Les boutons   permettent d'avancer en pas à pas, le premier en rentrant dans le code de la méthode appelée, le second en passant à l'instruction suivante. Utiliser le bouton  pour reprendre l'exécution normale.

➤ Trucs et astuces

➤ Fonctionnalités importantes de l'éditeur

La richesse de l'éditeur Java est une des bases du succès d'Eclipse. Dans la première partie de ce tutorial deux caractéristiques de l'éditeur ont été signalées :

- la complétion : le raccourci **Ctrl+espace** affiche une liste de propositions.
- la marquage des erreurs : le code est analysé au fur et à mesure de la saisie, tout comme dans un traitement de texte le code non valide est souligné en rouge.

L'éditeur propose de nombreuses autres fonctionnalités, en voici quelques unes :

➤ Complétion et import

Lorsque la complétion est utilisée pour un nom de classe, Eclipse ajoute automatiquement la déclaration de l'import si ce dernier est absent.

➤ Complétion et notion de 'modèles'

En plus des noms de types, de méthodes ou de variables la liste des choix proposés peut correspondre à des modèles. Ces modèles sont des blocs de code prédéfinis, leur liste est consultable dans la section **Préférences->Java->Editeur->Modèles** (Il est possible de créer ces propres modèles).



➤ Correction rapide

Le raccourci **Ctrl+1** affiche le 'QuickFix'. Ce 'correcteur rapide' propose une liste d'actions possibles pour résoudre une erreur (généralement une ampoule apparaît aussi dans la marge). A noter que le correcteur peut être utilisé même s'il n'y a pas d'erreur, il propose alors diverses actions comme le renommage d'une variable :

```
public void ditBonjour(String sOutil) {
    System.out.println("Bonjour");
}

```

Sélection puis 'Ctrl+1'
Liste de propositions

- Renommer dans le fichier (accès direct par...)
- Modifier le modificateur en final
- Affecter un paramètre au nouveau champ

➤ Gestion des imports

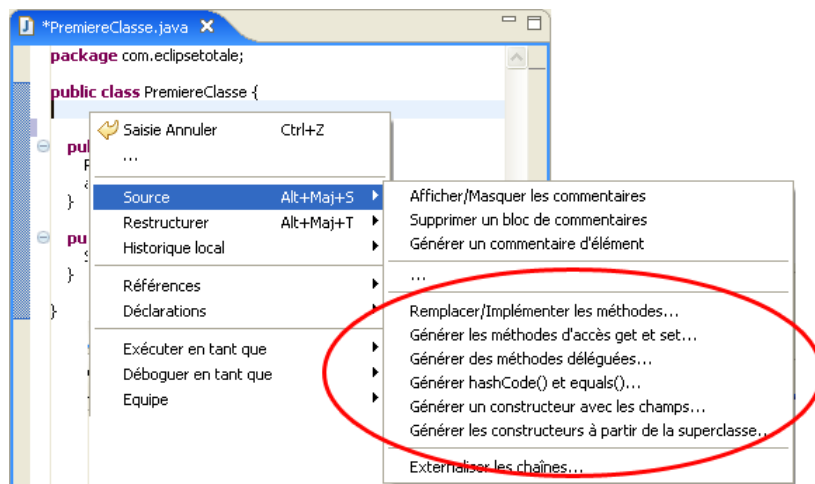
Le raccourci **Ctrl+Shift+O** (aussi accessible par le menu contextuel 'Source->Organiser les importations') gère la liste des imports : les imports manquants sont ajoutés, les inutiles supprimés et la liste est classée. Ces actions se font conformément aux paramètres définis dans **Préférences->Java->Style de code->Organiser les importations**.

➤ Formatage

Le raccourci **Ctrl+Shift+F** déclenche le formatage soit de tout le fichier soit des lignes sélectionnées. Les options de formatage sont configurables dans **Préférences->Java->Style de Code->Formater**.

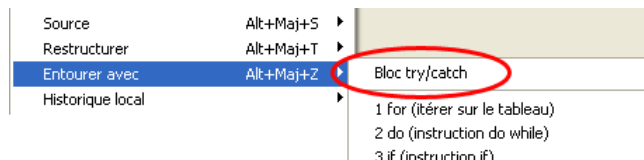
➤ Génération de méthodes

Dans l'éditeur, la section 'Source' du menu contextuel permet de générer plusieurs types de méthode, par exemple les accesseurs (méthodes get et set) ou encore des constructeurs.



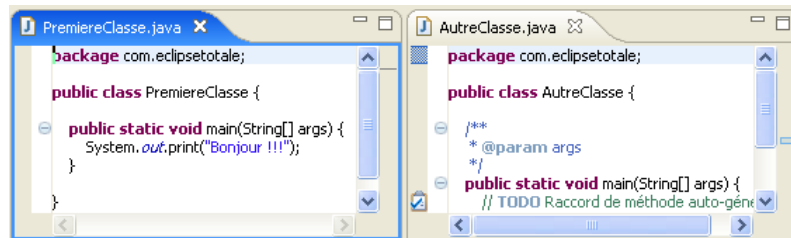
➤ Génération de blocs try/catch

Dans le menu contextuel de l'éditeur, la section 'Entourer avec' permet d'encadrer le bloc de code sélectionné par différents types d'instructions (try/catch, boucle, condition...). Dans le cas du 'try/catch', un bloc catch est généré pour chaque exception possible.



➤ Scinder la zone d'édition

Par défaut, Eclipse ouvre un nouvel onglet dans la zone d'édition pour chaque fichier. Il est d'une part possible de limiter le nombre d'onglets à partir de **'Préférences->Général->Editeurs->Nombre d'éditeurs ouverts avant fermeture'**, d'autre part la zone d'édition peut être scindée : sélectionner un onglet et le déplacer pour scinder la zone d'édition :



➤ Navigation

L'éditeur propose plusieurs raccourcis utiles pour naviguer de classe en classe et pour visualiser des informations sur un nom de classe ou de méthode sélectionné.

➤ Ouvrir le code de l'élément sélectionné

Le raccourci **F3** affiche le code de l'élément actuellement sélectionné dans l'éditeur (en ouvrant un autre éditeur si nécessaire).

Il est aussi possible d'utiliser la touche **Ctrl** seule : si la touche **Ctrl** est enfoncée l'éditeur affiche un lien hypertexte quand un nom de classe ou de méthode est survolé.

```
public static void main(String[] args) {
    System.out.println("Bonjour !!!");
}
```

Affichage d'un lien hypertexte

➤ Afficher la hiérarchie d'une classe

Le raccourci **F4** affiche la classe dont le nom est sélectionné dans la vue 'Hiérarchie'.

Le raccourci **Ctrl+T** affiche la hiérarchie en surimpression :

```
public static void main(String[] args) {
    // TODO Raccord de méthode auto-généré
}
```

Sélection d'un nom de classe.
Ctrl+T affiche la hiérarchie

Hiérarchie de types de 'java.lang.String':

- Object - java.lang
- String - java.lang

➤ Plus de trucs et astuces

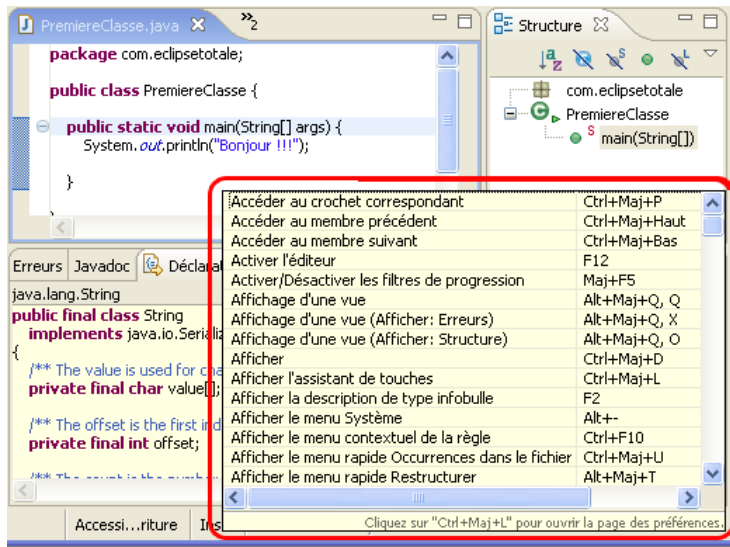
L'environnement Eclipse propose de nombreuses fonctionnalités, ce tutorial n'a pour ambition que de vous donner les bases pour débiter avec l'outillage Java d'Eclipse.

Pour aller plus loin et notamment découvrir plus de trucs et astuces voici deux premières pistes :

- consultez la section 'Tips and tricks' de l'aide d'Eclipse accessible par le menu '**Aide->Conseils et astuces...**'

- prenez le temps de parcourir l'ensemble des pages de préférences d'Eclipse pour voir les paramètres disponibles.

Les trucs et astuces les plus intéressants sont généralement accessibles rapidement via un raccourci clavier. Si vous ne souhaitez en retenir qu'un seul, **Ctrl+Shift+L** peut faire l'affaire car ce dernier affiche en surimpression la liste de tous les raccourcis clavier :



(c) EclipseTotale - contact(arobase)eclipse Totale.com