1 Environnement intégré de développement

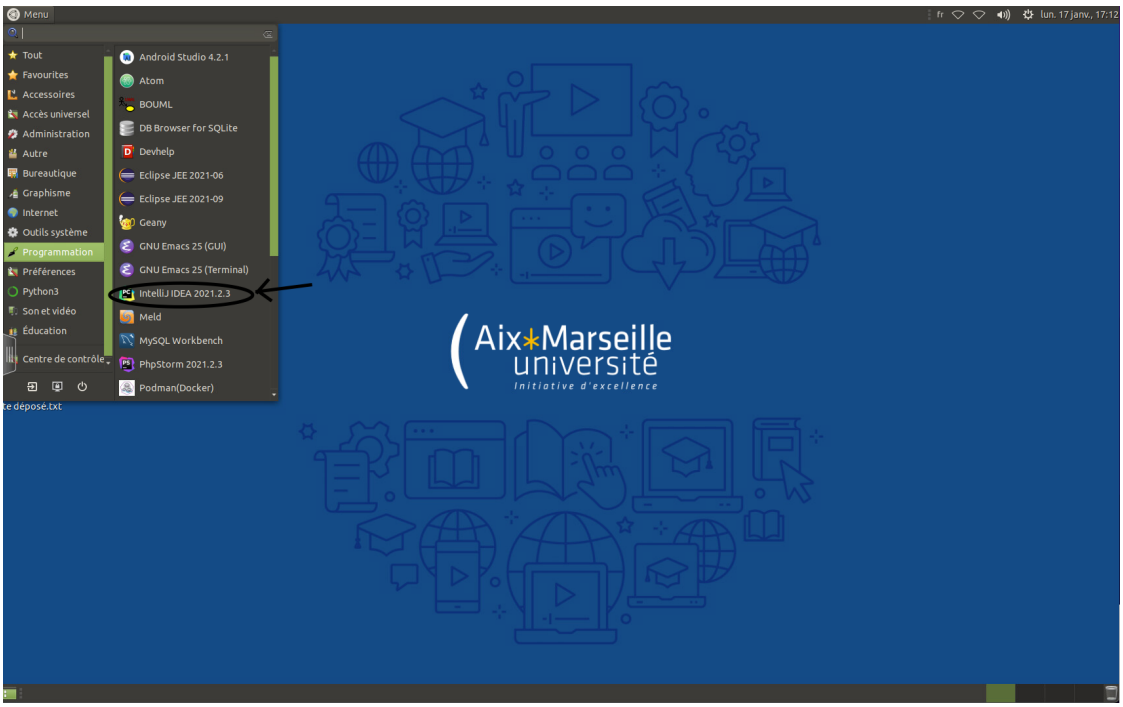
Si vous préférez travailler en local plutôt que via le site repl.it, vous pouvez utiliser n'importe quel logiciel permettant l'édition de fichier pour programmer. Cependant, il est plus pratique d'utiliser un logiciel dédié à la programmation : un *environnement intégré de développement* (IDE). Vos enseignants utilisent IntelliJ, un outil professionnel, complexe mais puissant. Si vous vous sentez débrouillard, nous vous invitons à l'essayer aussi, en l'installant et en suivant les instructions ci-dessous. Vous pouvez aussi utiliser un des autres IDE conseillés ci-dessous

IDE conseillés :

— IntelliJ IDEA de JetBrains
— Eclipse
— Visual studio code

1.1 Lancement de l'IDE

1. Si vous êtes sur les machines de l'université, il vous faut choisir l'image Luminux-22 en VDI.
2. Pour lancer le logiciel sur les machines de l'université, il vous faut aller dans le Menu (en haut à gauche de l'écran) et cliquer sur 'IntelliJ IDEA 2021.2.3' dans la partie Programmation du menu.



3. La première fois, il vous posera quelques questions, si vous ne savez pas quoi répondre laissez le choix par défaut.
4. Une fois ouvert, il vous faut créer un nouveau projet :
    1. Sélectionner `Java` en haut de la liste, et cliquer `next`,
    2. Dans l'écran suivant, cliquer sur `next`

5. Donner comme nom `TP1` à votre projet, et le placer dans un répertoire approprié (par exemple un répertoire qui vous servira exclusivement pour le cours de programmation 1). Cliquer sur `next` pour terminer la création du projet.

À partir de ces quelques informations, IntelliJ a créé les bases de votre futur programme. Il affiche maintenant sa vue standard. Normalement, sur la gauche de la fenêtre apparaît un onglet nommé `Project`, vous montrant les fichiers constituant votre projet (si ce n'est pas le cas, utiliser le raccourci clavier `Alt + 1` pour le faire apparaître).

Dans le répertoire de votre projet, IntelliJ a créé des sous-répertoires et des fichiers de configurations (que vous pouvez ignorer). Le répertoire `src` est important : **c'est uniquement dans ce répertoire que vous allez écrire votre programme.** Ainsi, IntelliJ saura que les fichiers de ce répertoire constitue votre programme, et que les fichiers ailleurs ne font pas partie de votre programme.

5. Avant d'aller plus loin, nous allons faire une petite modification dans les préférences d'IntelliJ qui nous fera gagner beaucoup de temps˜ : nous allons faire en sorte de rendre les erreurs du programme plus visibles.

> Cliquer dans le menu `File`, `Setting`, puis chercher dans la liste de gauche la catégorie `Editor` puis `Color Scheme` puis `General`. Dans la liste de la partie droite, chercher `Errors and warnings` puis `Error`. Activer `background` en mettant un rouge suffisamment visible. De même pour `Warning`, activer `background` avec par exemple du jaune-orange. Vous pourrez bien sûr modifier plus de préférences, mais cela suffit pour le moment. Fermer la fenêtre de préférences (`OK`).

6. Un programme Java est organisé en classe, chaque classe étant définie dans un fichier.

> Dans l'onglet de projet, cliquer droit sur le répertoire `src`, et choisir `new` puis `Java class`. Entrer le nom `ShoppingApp` et valider.

Vous venez de créer une classe, et c'est ainsi que vous créerez la plupart de vos classes. IntelliJ affiche maintenant le contenu du fichier `ShoppingApp.java` contenant votre classe. Vous pouvez constater que pour l'instant votre classe ne contient rien˜ : seulement une déclaration suivi d'un *bloc* délimité par les accolades. Tout ce que vous ajouterez à la classe doit être mis entre ces deux accolades.

7. Pour avoir un programme exécutable, le projet doit avoir un *point d'entrée*, qui contient les instructions qui seront évaluées. En Java, les points d'entrée sont toujours déclarés ainsi :

```
public static void main(String[] args) {
  // nous mettrons les instructions du programme dans ce bloc
}
```

> Ajouter dans la classe `ShoppingApp` cette déclaration.

8. Java, comme Python, est un langage impératif basé sur des instructions, exécutées une après l'autre. À l'intérieur du bloc de `main`, nous pouvons donc ajouter des instructions. Les différentes instructions possibles sont données en annexe en fin de sujet.

Ajouter les instructions nécessaires pour afficher le texte `Hello World!` en console.

9. IntelliJ a repéré que vous avez ajouté un point d'entrée à votre programme. Il est donc en mesure de *compiler* votre programme puis de l'exécuter. Pour cela,

Cliquer sur la flèche verte dans la marge de votre programme au niveau de la déclaration de `main`. Analyser le résultat.

Si vous n'avez pas fait d'erreur, une nouvelle vue apparaît en bas de la fenêtre avec le résultat du programme. En première ligne, IntelliJ a écrit la commande qui lui a permis de lancer le programme (vous pouvez l'ignorer), puis on trouve la sortie console du programme, et enfin un message indiquant que le programme s'est terminé avec un code de sortie qui devrait être 0.

Si vous avez fait une erreur, la nouvelle vue contiendra du texte en rouge pour vous expliquer ce qui ne va pas. Essayez de comprendre ce qu'il vous dit et relisez les instructions.