Introduction à Scala

1. Installation des outils

1.1. Installation du JDK

1.1.1. Linux (Ubuntu, Debian)

On utilise apt-get :

```
sudo  apt-get install openjdk-8-jdk
```

1.1.2. MacOs

Sur Mac une version de JDK est déjà installée ou bien elle s'installe automatiquement.
Pour vérifier ouvrer le terminal et taper :

```
java -version
```

Si le JDK n'est pas installé, l'OS vous proposera d'en installer une version. Vérifier que c'est bien java 8.

1.1.3. Windows

- Télécharger l'installer depuis le site d'Oracle
- Lancer l'installer
- Ajouter le répertoire *bin* sur le PATH dans vos variables d'environnement comme il est indiqué ici

1.2. Installation du REPL Scala

Read-Eval-Print-Loop : shell Scala.
Pour Windows : télécharger ici le msi. Double-cliquez ou ouvrez le fichier .msi et sélectionnez Exécuter. L'assistant d'installation apparaît, cliquez sur Suivant et terminez le processus d'installation. Scala installera également la variable d'environnement Path pour que vous puissiez l'exécuter depuis n'importe où.
Pour Max et Linux : utilisez brew ou apt-get.

1.3. Installation de sbt

Vous trouverez ici les instructions pour vos OS respectifs.

Note : pour MacOS vous pouvez utiliser *brew*, pour Linux *apt-get*

1.4. Installation d'Intelli

1.4.1. Télécharger Intellij IDEA Community Edition

Intellij IDEA est un IDE open source qui permet de faire de développer en Scala, Java et autres langages qui utilisent une JVM. Vous pouvez télécharger une version depuis le site officiel.

1.4.2. Installer le Plugin Scala

Première chose à faire pour développer en Scala sur Intellij !

```
Configure → Plugins → Browse JetBrains Plugins
```

A chaque nouveau plugin installé il vous sera proposé de redémarrer l'application.

1.4.3. Mettre en place le JDK

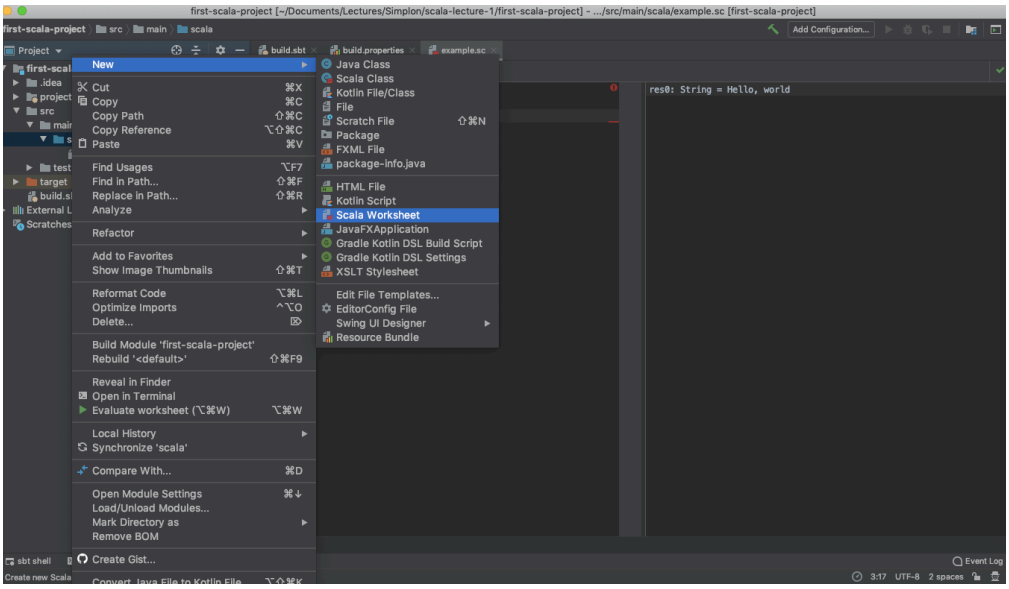
Dans l'écran d'accueil, allez dans Configurer → Projet par défaut → Structure du projet et ajoutez le JDK. Le JDK sera automatiquement détecté, sinon trouvez le chemin où il est stocké et sélectionnez-le dans le sélecteur de fichiers. Des instructions plus détaillées sont disponibles ici.

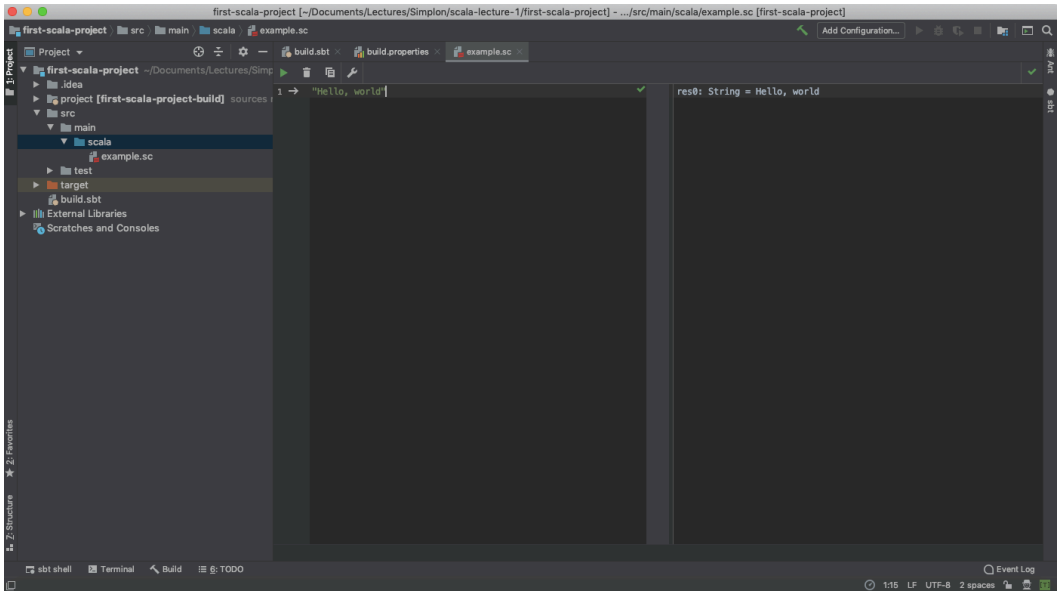
1.4.4. Créer un projet

La façon la plus simple de créer un projet est d'utiliser l'Assistant de projet. Pour l'utiliser, cliquez sur Créer un nouveau projet sur l'écran d'accueil, puis sélectionnez Scala, et enfin SBT Project.
Cliquez sur Suivant pour spécifier le nom et l'emplacement du projet. Une fois ces informations saisies, IntelliJ IDEA créera un projet vide contenant un fichier build.sbt.
Cela peut prendre du temps la première fois, sbt doit télécharger toutes les dépendances nécessaires.
Donner à ce projet le nom et l'emplacement que vous voulez!

Note : La première fois que nous créons ou importons un projet SBT, Intellij peut ne pas afficher la structure du projet tout de suite. Cela se produit parce qu'Intellij décide de retarder cette tâche jusqu'à ce qu'il ait exécuté avec succès sbt et téléchargé toutes les dépendances requises. Une fois cela fait, le dossier `src' habituel et ses sous-dossiers seront créés.

1.4.5. Créer un worksheet Scala





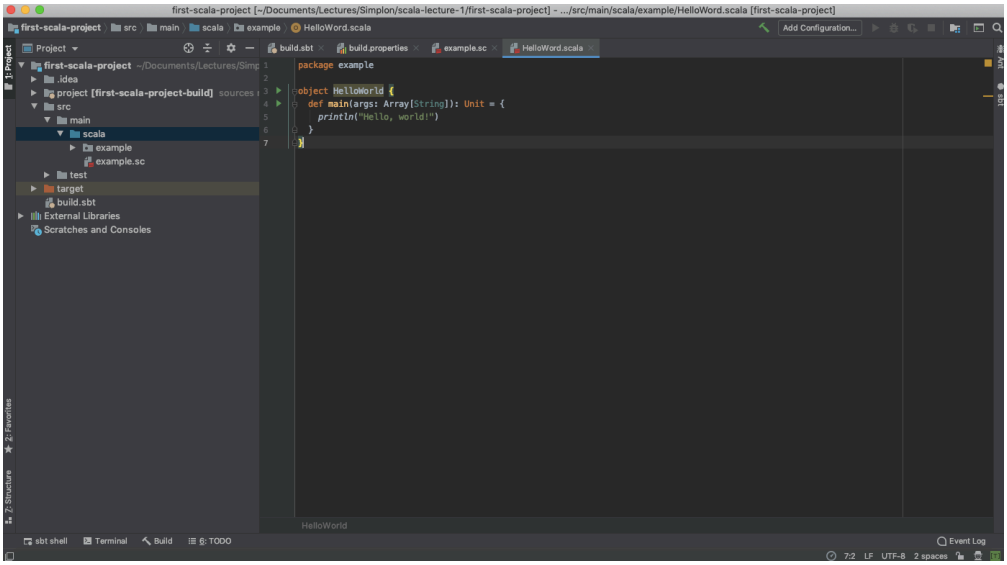
Utilisez simplement l'action Créer à partir du menu contextuel ou d'un package Scala (tout ce qui se trouve dans le dossier `scala').

Pour évaluer les feuilles de calcul, utilisez l'icône correspondante dans la barre d'outils (le bouton de lecture) ou appuyez sur Alt+Ctrl+W (Alt+Cmd+W sous OS X).

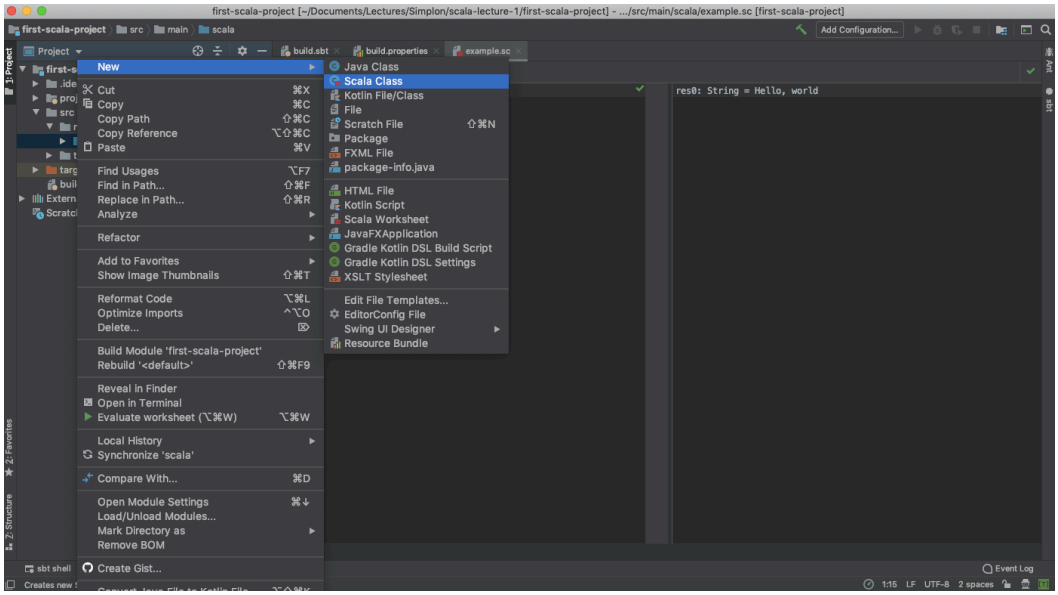
1.4.6. Importer un projet Sbt dans Intellij

Pour ouvrir un projet SBT dans IntelliJ IDEA, allez à l'écran d'accueil, cliquez sur Importer un projet et sélectionnez le fichier de compilation SBT que vous souhaitez ouvrir. IntelliJ IDEA créera alors un nouveau projet et y importera le fichier sélectionné.

1.4.7.Créer un classe Scala



Une fois que vous êtes prêt, exécutez votre application en appuyant sur Ctrl+Maj+F10, ou en utilisant le menu contextuel de l'éditeur.



2. Exercice

Explorer les notions de valeur, variables, les différents types, l'interpolation des chaines de caractère, les tuple.

- Convertir 22.5 de degré Celsius à Fahrenheit (en utilisant la formule (x * 9/5) + 32)
- Convertir une nouvelle fois en sauvegardant chaque étape de la conversion en valeurs séparées. Selon vous, quel sera le type de chaque valeur ?
- Modifiez la formule de Centigrade à Fahrenheit pour obtenir un nombre entier au lieu d'un nombre à virgule flottante.
- En utilisant la valeur d'entrée 2.7255, générez la chaîne "Vous devez 2,73 $." Est-ce faisable avec l'interpolation de chaînes de caractères ?
- Existe-t-il une façon plus simple d'écrire ce qui suit :

val flag : Boolean = false
val result : Boolean = (flag === false)

- Convertir un nombre en un caractère, un string, un double, puis revenez à un Int. Pensez-vous que le montant initial sera conservé ?

Explorer les notions de conditions (if…else), de pattern matching, des boucles.

La commande :load est une fonction Scala REPL et ne fait pas partie du langage Scala. Les commandes Scala REPL se distinguent de la syntaxe Scala classique par leur préfixe « : ».
Vous pouvez aussi écrire sur un Worksheet dans votre IDE.
1. Un chaine en entrée, écrivez un matching pattern qui retournera la même chaine de caractère si elle n'est pas vide, ou bien la chaîne "n/a" si elle est vide.
2. Étant donné un double, écrivez une expression pour retourner "plus grand" s'il est supérieur à zéro, "même" s'il est égal à zéro, et "moins" s'il est inférieur à zéro. Pouvez-vous écrire ça avec les blocs conditionnels if…else blocs ? Qu'en est-il des matching pattern ?
3. Ecrire une expression pour convertir une des valeurs d'entrée cyan, magenta, jaune en leurs équivalents hexadécimaux à six caractères sous forme de chaîne. Que pouvez-vous faire pour gérer les conditions d'erreur ?
4. Imprimez les chiffres de 1 à 100, chaque ligne contenant un groupe de cinq chiffres. Par exemple :

    1, 2, 3, 4, 5,
    6, 7, 8, 9, 10
    ....

5. Ecrivez une expression pour imprimer les nombres de 1 à 100, sauf pour les multiples de 3, imprimer "type" et pour les multiples de 5, imprimer "safe". Pour les multiples de 3 et 5, écrivez "typesafe".

Explorer les notions de fonctions, fonctions récursives, fonctions typées, méthodes
1. Ecrire une fonction qui calcule la surface d'un cercle en fonction de son rayon.
2. Fournir une autre forme de la fonction dans l'exercice 1 qui prend le rayon comme une chaîne de caractères. Que se passe-t-il si votre fonction est invoquée avec une chaîne vide ?
3. Ecrire une fonction récursive qui imprime les valeurs de 5 à 50 par cinq, sans utiliser la fonction for ou while dans les boucles. Pouvez-vous le rendre récursive ? Avec l'optimisateur de récursivité scala?
4. Ecrire une fonction qui prend une valeur en millisecondes et retourne une chaîne décrivant la valeur en jours, heures, minutes et secondes. Quel est le type optimal pour la valeur d'entrée ?
5. Ecrire une fonction qui calcule la première valeur augmentée à l'exposant de la seconde valeur. Essayez d'écrire ceci d'abord en utilisant math.pow, puis avec votre propre calcul. L'avez-vous implémenté avec des variables ? Existe-t-il une solution qui n'utilise que des données immuables ? Avez-vous choisi un type numérique suffisamment grand pour vos besoins ?
6. Ecrire une fonction qui calcule la différence entre une paire de points 2D et retourne le résultat sous forme de point. Conseil : ce serait une bonne utilisation des tuples.

7. Ecrire une fonction qui prend un n-uplet de taille 2 et le renvoie avec la valeur Int (si elle existe) dans la première position.
8. Ecrire une fonction qui prend un n-uplet de dimension 3 et retourne un n-uplet de dimension 6, avec chaque paramètre original suivi de sa représentation String. Par exemple, en invoquant la fonction avec (true, 22.25, "yes") doit retourner (true, "true", "true", 22.5, "22.5", "yes", "yes", « yes").

Prise en main de sbt.
Suivre le tuto sur la documentation officielle de sbt.

Installer Intellij.
Créer un projet puis faire votre première application SPARK!