



Quels sont les premiers pas avec Cobol?

Ce module vise à vous apprendre les bases du langage de programmation **Cobol**. Nous y verrons la structure de base, les divisions, les mots clefs et les commentaires. Voici un aperçu du module:

1. La structure d'un programme
2. Les divisions
3. Les sections
4. La structure d'une ligne de code
5. Les commentaires

La structure d'un programme en Cobol

Introduction

Un programme en **Cobol** s'écrit, principalement, dans un seul fichier. Lorsque vous programmerez de très grands programmes, ils seront invariablement dans plusieurs fichiers mais pour ce cours, tout sera dans un seul fichier.

Exercice #1 – Lancer un programme en Cobol

1. Ouvrez le document `exercices/module1/e1.cbl`
2. Appuyez sur CTRL+SHIFT+B pour lancer la compilation.

Vous devriez voir quelque chose comme:

Votre premier programme

Si tout à bien fonctionné: félicitations! Vous avez lancé votre premier programme en **Cobol**. Rappelez-vous des touches CTRL+SHIFT+B. Ces touches permettent d'envoyer le fichier actuellement ouvert au compilateur **Cobol** et d'ensuite lancer le programme.

Si vous faites des modifications au programme, n'oubliez pas de le sauvegarder, utilisez CTRL+S pour limiter l'usage des menus.

Les divisions

Un programme en **Cobol** doit suivre une structure précise que l'on appelle des divisions. Dans le dernier exemple, vous aurez remarqué 2 divisions:

- IDENTIFICATION DIVISION
- PROCEDURE DIVISION

Ces deux divisions sont vitales pour un programme en **Cobol**.

Il existe d'autres divisions qui seront utilisées dans le cadre de ce cours, voici un résumé de chacune d'elles:

Division	Description
IDENTIFICATION	Permet l'identification du programme, de son auteur et quelques autres informations de type méta-données
ENVIRONNEMENT	Permet de définir diverses informations sur l'environnement et sa configuration
LES DONNÉES	Permet de définir les données utilisées dans le programme comme les fichiers, les variables, etc
PROCÉDURE	Permet de définir les instructions, le code et les actions du programme

La division IDENTIFICATION

Cette division permet d'entrer des informations pour identifier le programme. Pour l'instant, notez que seulement l'instruction PROGRAM-ID est nécessaire. Elle s'accompagne d'un identificateur alphanumérique. Remarquez les points .? Ils sont absolument nécessaire, nous appelons ceci des phrases en **Cobol**. La structure d'une instruction PROGRAM-ID doit impérativement être comme suit:

```
PROGRAM-ID. IDENTIFIANT.
```

L'identifiant est alphanumérique donc seulement des lettres, des chiffres et des tirets. Les deux mots doivent être suivis d'un point ..

Nous verrons, dans un module ultérieur, les autres informations que nous pouvons mettre dans cette section.

Exercice #2 – Modifier l'identification du programme

1. Ouvrez le document exercices/module1/e2.cbl,
2. Modifiez l'identifiant du programme pour tester l'effet d'un:
3. Nom invalide
4. Nom valide

Lorsque vous exécutez un programme avec une erreur comme dans le cas #1, vous devriez voir quelque chose comme:

```
exercices/module1/e2.cbl:2: Error: syntax error, unexpected "Identifieur",
expecting COMMON or RECURSIVE or "INITIAL"
The terminal process terminated with exit code: 1
```

Lorsque vous exécutez un programme correctement, seulement le résultat du programme devrait s'afficher:

Votre premier programme

Si vous avez de la difficulté, demandez de l'aide à votre formateur.

La division PROCEDURE

Cette division est le coeur de votre programme. Elle contient toutes les instructions qui permet au programme de faire quelque chose. Nous verrons dans les modules suivants les instructions que vous pouvez entrer dans cette section. Pour l'instant, nous n'utilisons que l'instruction DISPLAY:

```
DISPLAY "Texte à afficher"
```

Lorsque vous utilisez l'instruction DISPLAY, vous devez la faire suivre d'un texte ou d'une variable. En fait, tout ce que vous mettrez par la suite sera affiché à l'écran. Nous verrons comment fonctionnent les variables dans le prochain module.

Exercice #3 – Afficher du texte

1. Ouvrez le document exercices/module1/e3.cbl,
2. Modifiez le texte qui est affiché par le programme
3. Affichez des lignes de textes supplémentaires

Demandez de l'aide au formateur si vous avez besoin d'aide.

Les sections

Les sections sont de simples sous-divisions des divisions. Par contre, comme nous n'avons ni abordé la division DATA ni la division ENVIRONMENT, nous allons donc sauter par dessus ce concept pour l'instant. Sachez tout simplement que ces divisions ont des sections spécifiques.

La structure d'une ligne de code

Dans tous les langages de programmation, les lignes de code doivent respecter une certaine syntaxe et structure. **Cobol** n'est pas différent, en fait, il est légèrement plus complexe et restrictif que d'autres langages de programmation.

Les lignes de code en **Cobol** doivent respecter une limite de 80 caractères mais **attention!** Vous ne pouvez pas écrire ce que vous voulez où vous le voulez: il y a une structure à respecter.

Il est important de comprendre que plusieurs versions de **Cobol** existent de nos jours. Dans la version **OpenCobol** que vous utilisez dans ce cours, vous pouvez dépasser la plage du 73e et même du 80e caractère mais ce n'est pas toujours le cas pour toutes les versions de **Cobol**.

Structure à respecter

Voici un tableau qui illustre bien les zones/plages à respecter dans un programme **Cobol**:

Plage	Description
1-6	Numéro de la ligne, optionel, peut être même omis complètement
7	Caractère spécial pour l'analyse du code
8-11	Zone de base pour les mots clés réservés comme les divisions et les sections
12-73	Zone étendue: vous pouvez commencer votre code à cet endroit pour lui donner une impression d'indentation
74-80	Zone custom: vous pouvez mettre ce que vous voulez

Exercice #4 – Démonstration des plages

1. Ouvrez le document `exercices/module1/e4.cbl`,
2. Observez les lignes de démarcation pour les zones que **VS.Code** vous affiche
3. Cet exercice ne peut pas être lancé, il n'est pas un programme valide

OpenCobol et le respect de la structure

Dans **OpenCobol**, il n'est pas nécessaire de suivre la structure proposée. Votre code fonctionnera pareil même si vous utilisez les zone 1-6 et 7 pour écrire du code. De plus, vous pouvez dépasser la zone 73 et 80.

Exercice #5 – Dépasser les marges

1. Ouvrez le document `exercices/module1/e5.cbl`,
2. Observez les lignes de démarcation pour les zones que **VS.Code** vous affiche
3. Lancez le programme

Remarquez que la chaîne de texte peut très bien dépasser les 80 caractères.

Exercice #6 – Ne pas respecter les 7 premiers caractères

1. Ouvrez le document `exercices/module1/e6.cbl`,
2. Observez les lignes de démarcation pour les zones que **VS.Code** vous affiche
3. Lancez le programme

Remarquez que le code ne suit pas la structure proposée. Vous avez du code dans la zone 1-7 et l'instruction `DISPLAY` n'est pas dans la zone 12-73. Tout fonctionne parfaitement. Évitez, par contre, cette approche si vous ne savez pas quelle version de **Cobol** vous utiliserez.

Les caractères spéciaux de la zone 7

Dans la zone 7, vous pouvez ajouter 1 caractère spécial qui permet de changer la façon dont la ligne de code sera analysée. Nous verrons ici deux caractères spéciaux mais il faut savoir qu'il en existe d'autres:

Caractère	Description
*	Permet de déclarer le code qui suit comme un commentaire
-	Permet de continuer le code de la ligne précédente

Nous allons maintenant tester les commentaires!

Les commentaires

Tout programmeur qui se respecte se doit de mettre des commentaires dans son code. Commenter du code est un art. Ce n'est pas aussi simple que taper du texte à chaque 2 lignes, mais au début, plus de commentaires seront mieux de moins.

Écrire des commentaires

Pour ajouter un commentaire dans votre code, vous devez commencer une ligne de code avec un astérisque * dans la zone 7 et ensuite taper ce bon vous semble. Vous pouvez faire autant de commentaires que vous le souhaitez. Les commentaires ne feront pas partis du code source final compilé. Si vous deviez renverser la compilation du programme **Cobol**, vous ne retrouveriez plus ces commentaires.

Attention, **OpenCobol** ne supporte pas les commentaires avec seulement un astérisque *, vous devez utiliser *> et vous pouvez mettre le début d'un commentaire où bon vous semble; que ce soit dans la plage 1-7, après une ligne de code normale ou sur une nouvelle ligne avec ou sans espacement. Vous pouvez mettre un commentaire n'importe où tant qu'il commence par *>.

Exercice #7 – Mettre des commentaires

1. Ouvrez le document `exercices/module1/e7.cbl`,
2. Ajouter un commentaire avant l'instruction `DISPLAY`
3. Lancez le programme

Si le code est bien exécuté, alors vous avez fait votre commentaire correctement.

Conclusion

Bravo, la première étape est terminée!

Vous avez maintenant une base pour écrire des programmes en **Cobol**, passons maintenant au module #2 qui ira plus amplement dans les fonctions du langage.

Pour aller plus loin sur Cobol, nous vous recommandons [la formation Initiation Cobol](#)

Nous contacter:
Doussou Formation
Email: info@doussou-formation.com
<http://doussou-formation.com>

