

Maîtriser Git : Gérez vos Versions et Collaborez Efficacement

Module 1 : Introduction à Git

Objectif: Comprendre l'utilité de Git dans la gestion de versions et le développement de projets.

- Introduction à la gestion de versions dans le cadre d'un projet de site web statique.
- Installer et configurer Git ainsi que Python, Pip, Github, Gitlab et MkDocs.
- Présenter Git et son écosystème (Github, Gitlab, Bitbucket, etc.).

Module 2 : Travailler seul avec Git

Objectif: Maîtriser les fondamentaux de Git pour un projet individuel de site web statique en Python et Markdown.

- Les commandes de base : `git init`, `git config`.
- Suivre et enregistrer des modifications : `git add`, `git status`, `git commit`.
- Explorer les versions précédentes : `git log`, `git diff`.
- Gérer des versions : `git tag`.
- Mettre en ligne : `git push`.
- Découvrir MkDocs, un peu de Python et le Markdown pour émuler le site localement et l'héberger en ligne.
- Découvrir d'autres concepts : les branches (`git branch`, `git checkout`), un YAML, un README.md, sur Github et Gitlab.
- Cloner un projet en ligne : `git clone`.
- Travailler avec des alias de commandes.

Module 3 : Travailler en équipe avec Git

Objectif: Apprendre à utiliser Git dans un environnement collaboratif.

- Découvrir les types d'équipe : propriétaire et collaborateur pour un projet à 2 ou collaborateurs égaux.
- Administrer les équipes et déterminer les permissions.
- Revoir les commandes et ajouter les mises à jour pour synchroniser son dépôt local avec le dépôt central : `git fetch`, `git pull`.
- Les bonnes pratiques pour le travail en équipe.

Module 4 : Pull Request (requête de fusion)

Objectif: Maîtriser le processus de requête de fusion de code via un Pull Request.

- Introduction à Pull Request, la contribution à un projet et le processus de requête de fusion.
- Proposer une requête de fusion (Pull Request).
- Traiter la requête sans conflit : confirmer ou rejeter la requête.
- Faire la mise à jour : git fetch, git pull.
- Stratégies de gestion des Pull Requests en équipe.

Module 5 : Gestion des conflits de versions

Objectif: Apprendre à gérer les conflits de versions et à les résoudre.

- Comprendre les conflits de versions : causes et détection.
- Traiter la requête avec conflit : procéder ou rejeter la requête.
- Résoudre manuellement des conflits : git status, git mergetool.
- Utiliser des logiciels de gestion de conflits (VSCode, Meld, etc.).
- Meilleures pratiques pour éviter les conflits : utiliser la remise git stash.
- Gérer la remise : ajouter, extraire, nettoyer, etc.

Module 6 : Synchronisation et intégration continue

Objectif: Gérer les flux de travail d'intégration en équipe.

- Différencier les dépôts origin et upstream.
- Introduction aux concepts de CI/CD, intégration continue et structure triangulaire.
- Introduction aux branches : git branch, git checkout -b.
- Introduction aux passerelles HTTP et SSH et gestion des clés de chiffrement SSH.
- Découvrir les backtracks, l'historique et d'autres logiciels conviviaux.
- Explorer le pipeline CI/CD, intégration et déploiement continus.

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 nécessaires, 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 "Identifier",  
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

Exercise #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.

Exercise #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.

Exercise #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](#)

Formation Camtasia

Module 1 formation Camtasia : Explorer et enregistrer

Ce module vous donnera une brève introduction à Camtasia et à son espace de travail. Vous allez ouvrir un projet existant et explorer les outils, le dossier multimédia, les canevas et le montage chronologique. Vous pourrez également enregistrer des actions à l'aide de Camtasia Recorder.

- Ouvrez un projet Camtasia
- Découvrez les outils Camtasia
- Explorez le dossier multimédia
- Aperçu d'un projet
- Définir les options d'enregistrement
- Sélectionnez une zone d'enregistrement
- Enregistrer une vidéo

Module 2 formation Camtasia: Ajout de médias

Vous allez créer un nouveau projet et apprendre à importer des enregistrements dans l'éditeur en tant que média.

- Importer une vidéo
- Importer des images
- Ajouter une piste
- Créer un filigrane
- Ajouter des effets de curseur

Module 3: Groupes, annotations et animation

Améliorez les médias avec des légendes et des animations. Vous apprendrez également à rendre les sauts entre les segments vidéo plus professionnels en utilisant les Transitions.

- Créer un groupe
- Ajouter une légende
- Appliquer la couleur de l'image au texte
- Ajouter des effets à une légende
- Ajouter une transition à un groupe
- Modifier les transitions
- Ajouter une animation

Module 4: Audio

L'ajout d'audio (comme les effets sonores et la narration vocale) améliore l'expérience eLearning pour vos apprenants. Vous apprendrez à ajouter de l'audio en tant que média et à enregistrer et éditer votre propre audio.

- Ajouter une musique de fond à une vidéo
- Fondu d'entrée et de sortie audio
- Enregistrer la narration vocale
- Fractionner un clip musical
- Renommer les pistes
- Silence et coupure audio

Module 5: Partage, extension et quizz

Vous apprendrez à produire et à partager vos projets en tant que vidéo autonome et à partager le contenu. Vous apprendrez à ajouter un quiz à votre projet.

- Partager un MP4
- Partager sur YouTube
- Exporter en tant que page Web
- Prolonger une image vidéo
- Ajouter un quiz et une question
- Ajouter un remplissage dans la question vierge

Module 6: Marqueurs, hotspots et légendes

Vous apprendrez à ajouter de l'interactivité à votre projet eLearning via des marqueurs, des tables des matières et des points d'accès. Ensuite, vous apprendrez à créer un eLearning accessible en ajoutant des sous-titres codés.

- Ajouter un marqueur
- Ajouter un point d'accès au montage chronologique
- Créer des sous-titres codés
- Contrôle de la synchronisation des sous-titres

Formation Java : Niveau 1

Introduction à la formation Java : Niveau 1

Mise en place de l'environnement de développement

Syntaxe de programmation Java

Structure d'un fichier .java

Variable de type primitif et opérateur

Commenter et documenter du code

Les conditions dans le code: if/else,...

Opérateurs de comparaison et de logique

Les tableaux

Les boucles

Méthode de classe

La programmation orientée objet

Objectif

Comparaison POO et langage procédural

Classe et objet

Méthodes et attributs

Rélations entre classes

Notion d'interface

Classe abstraite

Atelier pratique : *conceptions de classes et mise en relation entre classes*

Exceptions dans le langage Java

Principe d'une exception

Manipuler des exceptions

Collections

Présentation des collections

Itérer sur des collections

Atelier pratique : *Manipulation de collections d'objets*

Formation MVC ASP.NET | Cours en ligne

Introduction à la formation MVC ASP.NET

Architecture d'ASP.NET MVC

Création d'un projet ASP.NET MVC

Apports de MVC pour l'application

Prise en main de ASP MVC

Comprendre le rôle des Contrôleurs

Comprendre le rôle des Modèles

Comprendre le rôle des vues

Configuration le routage par défaut

Les redirections

Communication Contrôleur Vue dans une application ASP.NET MVC

Envoyer des données par ViewData

Envoyer des données par ViewBag

Envoyer des données par TempData

Comprendre la différence entre les méthodes de transmission de données

Définir un routage par annotation

Préfixer les contrôleurs

Définir les routes pour chaque action

Définir une route par défaut

Maîtriser Razor avec ASP MVC

Comprendre la syntaxe razor

Affichage des données dans la vues
Les structures de contrôles
Les boucles
Créer une page de disposition personnalisée
Comprendre la notion de Bundle

Formulaire et helpers dans une application ASP.NET MVC

Écrire un code de formulaire
Améliorer la syntaxe avec les helpers
Validation des champs de formulaire
Validation par code avec ModelState
Validation par annotation

Stocker en base de données une application ASP.NET MVC

Présentation d'Entity Framework
Liaison avec les données(Code First)
Rappel de la syntaxe LINQ
Gestion des relations un à plusieurs
Gestion des relations plusieurs à plusieurs

Déployer une application ASP.NET MVC

Activer le serveur IIS
Configurer le serveur IIS
Modes de déploiement
Atelier pratique: déployer son site ASP.NET sur IIS

Notions avancées(S'il reste du temps)

Tests unitaires
Les filtres

Conclusion de la formation MVC ASP.NET

Formation C#(sharp) perfectionnement | Niveau avancé

Introduction à la formation C#(sharp) perfectionnement

Les Standards en C#

Casts

Les initialiseurs d'objets

Les délégués

Les méthodes d'extension

Expression Lambda

Type anonyme

LINQ

LINQ To Object

LINQ To Entity

LINQ To XML

Base de données

Entity Framework

Mapping relationnel objet

Pattern DAO

Approche DataBase First

Approche Code First

Formation Cobol

Introduction à la formation Cobol

Présentation de mainframe et cobol

Historique

Environnement mainframe

Autres environnements

Installer IDE NetExpress

Vos premiers pas

Structure générale

Les divisions

Les sections

Les commentaires

Les paragraphes

Les étiquettes

Les phrases

Les variables

- Les plages
- Les procédures
- Perform: les débranchements
- L'aide au débogage

Les fondamentaux

- Les variables
- Noms de variable/noms réservés
- Les types de variables
- Les values
- Les opérateurs
- Les commentaires
- Les entrées/sorties
- if ... else
- Choix multiples
- Les boucles

Les fichiers

- Declaration
- ouverture/fermeture
- lecture/ecriture
- Impression
- Mecanisme

Les tableaux

- Déclarations
- Affectation
- Opérations
- Indice
- Indexe(Set, Search)

Les chaînes de caractères

- Déclaration et caractéristiques
- Manipulations avancées
- Quelques fonctions

Les sous programmes

- Appel de sous programme: CALL
- Le passage des paramètres
- RETURN-CODE

Formation C++ | En ligne | Montréal



Introduction à la formation C++

Historique du langage C++
Différences C / C++
Compiler sous Windows
Portabilité du code

Les fondamentaux

Les variables
Les tableaux
Les types de variables
Les références et pointeurs
Gestion de la mémoire
Organisation du code

Programmation orientée objet

Classe et objet
Constructeur et destructeur
Encapsulation
Héritage
Polymorphisme
Surcharge d'opérateurs
Opérateurs de conversion
La gestion d'exceptions
Relations entre objets
Modèles de fonctions
Modèles de classes

La bibliothèque Standard Template Library

Présentation de la STL
Flux C++ (entrées-sorties)
Les chaînes dans la STL
Conteneurs dynamiques

Conclusion de la formation C++

Québec

Formations professionnelles à Québec

À Québec, Doussou Formation conçoit des parcours orientés résultats : Excel, Power BI, Google Workspace, InDesign et efficacité professionnelle. Nos experts adaptent les contenus à vos cas d'usage pour un impact immédiat en entreprise.

[Demander un devis](#)

Nos formations les plus demandées à Québec

- Excel (Fondamentaux, Perfectionnement, Tableaux de bord)
- Power BI (Desktop, Service, Dataflows)
- Google Workspace (Drive, Docs, Sheets, Slides)
- InDesign (mise en page professionnelle)
- Efficacité professionnelle (gestion du temps & priorités)

Modalités et formats

- Présentiel (centre de Québec) ou en ligne
- Intra-entreprise sur mesure ou inter-entreprises
- Supports et fichiers d'exercices inclus

Adresse & contact

Doussou Formation – Québec
131-686 Grande Allée Est

Québec, QC G1R 2K5

Téléphone : [+1 \(438\) 869-8543](tel:+14388698543)

Montréal

Formations professionnelles à Montréal

Doussou Formation accompagne les entreprises et les professionnels à Montréal avec des parcours concrets en bureautique, data & IA, design et efficacité professionnelle. En présentiel ou à distance, nous adaptons chaque programme à vos besoins et objectifs.

[Demander un devis](#)

Nos formations les plus demandées à Montréal

- Excel (Fondamentaux, Perfectionnement, Tableaux de bord)
- Power BI (Desktop, Service, Dataflows)
- Google Workspace (Drive, Docs, Sheets, Slides)
- InDesign (mise en page professionnelle)
- Efficacité professionnelle (gestion du temps & priorités)

Modalités et formats

- Présentiel (centre de Montréal) ou en ligne
- Intra-entreprise sur mesure ou inter-entreprises
- Supports et fichiers d'exercices inclus

Adresse & contact

Doussou Formation – Montréal
1155 Metcalfe Street, Suite 1500
Montréal, QC H3B 2V6

Téléphone : [+1 \(438\) 402-2459](tel:+14384022459)