

Formation élaborer son plan marketing

Introduction à la formation élaborer son plan marketing

Module 1

-Dimension de base et objectif de plan marketing

-Analyse interne et externe de l'environnement de votre entreprise et identification des facteurs :

- Technologiques
- Sociologiques
- Politiques
- Environnementaux
- Économiques
- Juridiques

-Élaboration de la matrice SWOT et étude de cas

-Choix stratégique selon l'étude des 3 dimensions fondamentales :

- Segmentation
- Ciblage
- Positionnement

-Le Mix stratégique : Produit/Prix/Communication/Distribution

-Le Contrôle et le suivi : Mesure des écarts

Module 2

Atelier pratique: Mettre en pratique les notions acquises lors de la première session pour élaboration d'un plan marketing concret

Formation: Maîtriser la transformation d'un document XML avec XSLT

Module 1 – Bases XML et rappel essentiel

- Structure d'un document XML
- Hiérarchie des nœuds : éléments, attributs, texte
- Introduction aux namespaces
- Bonnes pratiques de conception d'un XML exploitable
- Analyse d'un XML source réel

Module 2 – Introduction à XSLT

- Rôle et cas d'usage de XSLT
- Structure d'une feuille de style XSLT
- Notions de template, match et apply-templates
- Différences entre XSLT 1.0, 2.0 et 3.0
- Première transformation XML vers HTML

Module 3 – XPath indispensable

- Sélection des nœuds avec XPath
- Axes, chemins absolus et relatifs
- Utilisation des prédicats et filtres conditionnels
- Fonctions XPath courantes
- Exercices pratiques de navigation dans un XML

Module 4 – Templates et génération de contenu

- Création et organisation des templates XSLT
- Génération de contenu avec `xsl:value-of` et `xsl:text`
- Création dynamique d'attributs HTML ou XML
- Conditions avec `xsl:if` et `xsl:choose`
- Comparaison entre `xsl:for-each` et `xsl:apply-templates`

Module 5 – Tri, formatage et sorties

- Tri des données avec `xsl:sort`
- Formatage du texte, des nombres et des dates
- Paramétrage de la sortie avec `xsl:output`
- Gestion de l'encodage et de l'indentation
- Exercice : génération d'un rapport structuré

Module 6 – Atelier pratique : cas réel

- Analyse d'un besoin métier
- Transformation complète d'un document XML
- Production d'une sortie HTML ou XML cible
- Validation et ajustements
- Revue de code et bonnes pratiques

Module 7 – Namespaces et structures XML complexes

- Comprendre le rôle des namespaces
- Déclaration et utilisation dans XSLT
- XPath avec namespaces
- Erreurs fréquentes et méthodes de résolution
- Exercices ciblés

Module 8 – Paramètres et templates réutilisables

- Définition et utilisation de `xsl:param`
- Transmission de paramètres avec `xsl:with-param`
- Création de templates génériques
- Factorisation et réutilisabilité du code
- Organisation modulaire d'une feuille XSLT

Module 9 – Modes et rendus multiples

- Principe des modes en XSLT
- Utilisation de l'attribut `mode`
- Création de plusieurs rendus à partir d'un même XML
- Cas pratiques : vue résumé, vue détaillée, export
- Exercice guidé

Formation Kubernetes : Déploiement, Gestion et Supervision de Conteneurs

Module 1 : Introduction et concepts fondamentaux

- Comprendre la conteneurisation
- Introduction à Kubernetes : principes et architecture
- Utilisation de la CLI et de l'API Kubernetes

- Exploration des PODs : création et gestion
- Surveillance d'un cluster Kubernetes
- Analyse des ressources Kubernetes
- Création pratique de ressources Kubernetes

Module 2 : Découverte des ressources Kubernetes

- Recherche et inspection des images de conteneurs
- Gestion des déploiements : ReplicaSet et StatefulSet
- Exposition des applications avec les ressources Service
- Gestion des applications à court et long terme
- Résolution des problèmes liés aux conteneurs et aux PODs

Module 3 : Approfondissement et notions avancées

- Réserve et limitation des capacités de calcul pour les applications
- Gestion des mises à jour des applications
- Externalisation et gestion des configurations
- Persistance des données : PVC et PV
- Classes de stockage : exploration et configuration
- Introduction à Helm : gestionnaire de packages Kubernetes

Module 4 : Outils de gestion et supervision

- Installation et utilisation d'une interface web Kubernetes
- Création et mise en place d'un cluster Kubernetes
- Gestion des utilisateurs et des accès dans Kubernetes
- Supervision et monitoring des applications avec Prometheus et Grafana

Module 5 : Révision et projet pratique

- Projet récapitulatif pour consolider les connaissances
- Mise en pratique des notions apprises sur un cas réel

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 "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.

Exercise #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](#)

Formateur, Analyste consultant/consultant en informatique

Description de l'organisation :

Organisme privé de formation continue en informatique, soucieux du développement des compétences et des performances des entreprises, Doussou Formation est spécialisé dans la formation des ressources informatiques, l'accompagnement des entreprises dans la maîtrise des nouvelles techniques dans le monde des nouvelles technologies et dans le développement des produits.

Doussou Formation est un assureur des prestations de développement avec les technologies de pointes et des formations.

Pour faire face aux nombreux défis et relever les challenges, Doussou Informatique cherche un Analyste en informatique pour étoffer ses ressources et accroître ses performances.

Description du poste

Il sera chargé du transfert des compétences clés à l'entreprise et d'accompagner les entreprises pour une meilleure stabilité de leurs systèmes d'information ou produit afin de produire de la valeur. À ce titre, il aura pour tâches :

- Identifier et quantifier les besoins de formations et l'élaborer le cahier des charges ;
- L'élaborer le syllabus de formation en prenant en compte les principes de pédagogies et le niveau des stagiaires ;
- Préparer le matériel pédagogique ;
- Animer des formations et évaluer la formation et les participants ;
- Participer au développement des produits ou le coaching des entreprises dans

l'évolution ou la mise en place des produits ;

- Faire des veilles technologiques ;
- Et tout autres fonction en lien avec ses compétences.

Exigences nécessaires :

Les candidats doivent détenir un diplôme en génie Informatique, être certifié ITL et avoir son certificat PMP ou avoir suivi un cours de préparation PMP.

En plus, vous devez avoir des connaissances en planification stratégique.

Posséder 7 – 10 années d'expérience dans la gestion des services

informatique, l'analyse et développement des applications ;

Posséder une parfaite maîtrise de la langue française orale et écrite;

Posséder une bonne capacité de production de rapport ;

Etre capable de produire dans un environnement Agile ;

Avoir dispense des cours informatiques dans des universitaires;

Posséder des connaissances dans les domaines des Systèmes d'informations géographique.

Avoir une grande capacité de communication et une facilité d'adaptation et d'intégration.

Etre disponible, mobile, proactif, autonome et rigoureux.

Maîtriser les technologies et outils suivantes : PHP, Java, VB.net, Python, Gestion des projetés avec PMP, Agil, Merise, UML, BPMN camuda, Système MultiAgent, MySQL, ACCESS, Oracle, SQL Server, PosgreSql, PostGis, Oracle Form, JBuilder, J2EE, ZK, JSP, JPA, Apex, Foex, JSF, Angular, nodejs, Eclips, typescript, Visual Basic, SQL, PL/SQL, – Flash, CSS3, HTML, XML, Pascal, Project 2003 et 2007, Microsoft Windows server, Linux, Exchange, Apache, Easyphp, -Glassfich, tomcat, Jboss, Fortinet, -Cisco.

Qualités personnelles recherchées : 1

Nombre d'heures par semaine : 40

Statut d'emploi : permanent

Formation Rhino 3D | Cours à **Montréal, Gatineau, Québec**

Introduction à Rhino 3D

L'interface de Rhino 3D

Menu Rhino 3D

Fenêtre de l'historique

Invite de commande

Barres d'outils

Fenêtres

Panneaux
Barre d'état
Commandes de Rhino 3D
Options de commande

Atelier pratique: Lancer la commande Cône et dessiner le cône

Objets de Rhino 3D

Points
Courbes
Surfaces ouvertes et surfaces fermées
Surfaces limitées et non limitées
Degré des courbes et des surfaces
Points de contrôle – Position
Points de contrôle -Visibilité et Ajouter
Points de contrôle – supprimer
Points de contrôle – redistribuer
Polysurfaces
Solides
Objets d'extrusion légers
Objets maillés

Sélectionner des objets

Sélectionner et désélectionner des objets
Menu de sélection
Sélectionner les objets avec une fenêtre
Autres façons de sélectionner des objets
Sélection de sous-objets

Naviguer dans les fenêtres de Rhino 3D

Titre de la fenêtre
Projection de la fenêtre
Navigation dans les fenêtres
Modes d'affichage

Modélisation précise

Le curseur de Rhino 3D
Magnétisme sur grille
Mode Ortho
Accrochages aux objets
Contraindre le mouvement du curseur
Ortho
Contrainte de distance
Contrainte d'angle
Contraintes de distance et d'angle simultanées

Repérage intelligent
Systèmes de coordonnées
Mode élévation
Atelier pratique : Dessiner une courbe 3D en utilisant le mode élévation

Créer des surfaces à partir de courbes

Surface à partir des bords
Extruder des courbes
Surface par sections
Révolution de courbes
Révolution d'une courbe le long d'un rail
Balayer une courbe le long d'un rail
Balayer le long de deux rails

Modification des courbes et des surfaces

Joindre
Limite et division
Limiter
Diviser
Annuler Limite

Transformations – Déplacer, copier, faire tourner et échelle

Déplacer
Déplacer des objets en les faisant glisser
Déplacer des objets en utilisant le mode élévation
Copier des objets
Faire tourner une vue
Échelle
Symétrie

Organisation – Calques, groupes et blocs

Calques
Liste déroulante des calques
Panneau des calques
Groupes
Bloc

Annotation : Cotes et texte

Cotes
Texte d'annotation
Lignes de repère
Points
Hachures

Suppression des lignes cachées

Analyse des courbes et des surfaces

Mesurer la distance, l'angle et le rayon

Direction

Analyse de surface visuelle

Analyser la courbure

Placage d'environnement

Rayures

Analyse de l'angle de dépouille

Analyse des bords

Diagnostic

Rendu Rhino 3D

Lumières

Matériaux

Environnement

Plan au sol

Aperçu du rendu

Hugues S.

homme/Scientifique de données

Formation Langage R, RStudio et Tidyverse

Introduction à la formation langage R

Introduction au langage R

Introduction

Qu'est-ce que R?

Découvrir l'aspect analytique

Le côté informatique décisionnelle de R/RStudio

Les domaines d'utilisation de R
Découvrir RStudio
Reporting, Tidyverse

Prise en main de l'environnement RStudio

Installer RStudio
Configurer RStudio
Utiliser la console
Installer un package
Utilisation des aides
Découvrir le Markdown

Les bases du langage R

Les variables
Les fonctions
Les calculs
Évaluer des conditions et contrôles de flux
Boucles
Créer des fonctions sur mesure
Calculer le temps d'exécution
Structure de données: matrice
Structure de données: ts
Structure de données: Date et time
Structure de données: facteur
Environnements de travail

Manipuler les structures de données

Les vecteurs
Les listes
Les matrices
Les dataFrames

Faire le parallèle entre R et un tableur

Adapter en R/RStudio les opérations de tableur.
Adapter en R/RStudio les fonctions de tableur.
Importer et exporter des données (tableur versus R/RStudio)
Exploiter le meilleur des deux logiciels.

Manipulation de données avec R

Importation de données
Exportation de données
Sélection et regroupements de données

Etude de cas: consolider ses bases du langage R et du logiciel RStudio

Utilisation de tidyverse pour plus d'efficacité

Les modules du tidyverse: dplyr, ggplot2, tibble, readr, tidyr, purrr

La préparation des données

Importer (et exporter) des jeux de données.

Comprendre les données (Explorer et visualiser)

Traiter les valeurs manquantes

Traiter les données aberrantes.

Modéliser: sélectionner, filtrer, trier, modifier, ajouter, supprimer.

Manipuler les données catégoriques.

Manipuler les séries chronologiques.

Produire des rapports PDF, HTML, etc.

Visualiser les données avec R

Réalisation des graphes avec le logiciel R

Personnaliser des graphes avec le logiciel R

Tracer des cartes avec le logiciel R

Autre Formations en Lien

[Formation en Analyse Statistique et Préparation des Données avec R](#)

Formation en science des données : Numpy, Pandas, Matplotlib

Module 1: Python pour le calcul scientifique: Numpy

Installer la distribution Anaconda

Introduction à Numpy

Créer des objets ndarray

Sélection de données

Ajouter, modifier, supprimer des éléments

Utiliser les fonctions numpy

Entrée / sortie

Module 2: Manipulation des données avec Pandas

Les objets Séries

Les objets DataFrames
Sélection des données
Fonctions d'agrégation
Fusionner, Joindre, Remodelage
Utiliser des fonctions lambda
Faire un croisé dynamique(Pivot Table)
Manipuler des données excel(csv) et json

Module 3: Visualiser les données avec Matplotlib

Affichage de courbes 2D
Affichage de nuages de points
Affichage d'histogramme
<!--

API Web

Requête vers une API
Récupérer la réponse
Traiter la réponse
Application: API Twitter, analyser et visualiser dans le temps
-->

Formation Geoconcept

Introduction à la formation Geoconcept

Principe de GEOCONCEPT

Généralités sur les Systèmes d'information géographique
Les primitives géométriques
Les Objets (Linéaires, surfaciques)
Les données sémantiques
Structure d'une table
Les types de données

Navigation et position

Navigation et position
Localisation automatique
Modifier la visibilité des objets
Mode de Sélection, Modifier des objets
Requêtes avec les données sémantiques

Requêtes simples
Requêtes paramétrées
Requêtes multicritères

Aller plus loin

Calage d'un Raster
Les étiquettes
Analyses thématiques
Données externes -Importer – exportation
Exploitation des vues>>