

Bienvenue dans la formation COBOL

Apprenez les bases du COBOL et développez vos compétences en programmation legacy avec une approche moderne.

Info : Ce module vous permet d'accéder rapidement aux contenus essentiels de la formation COBOL. Le lien de la formation peut changer, pensez à vous connecter systématiquement sur votre lien personnel à chaque fois, celui qui vous a été transmis par mail !

Les mises à jour récentes

PACTA 360

30

COBOL :
Compétences
avancées

30 jours

pour apprendre les bases
du langage COBOL

Alain STEVENS

En français, à traduire via votre navigateur

AS

cobol.pacta.com

COBOL

Bienvenue dans la formation COBOL

Vous êtes ici pour apprendre les bases de COBOL de manière progressive, avec des exemples concrets et une méthode d'acquisition de la compétence régulière. Prenez le temps de suivre l'ordre des modules et de tester chaque exercice.

 **Accès réservé.** Le contenu pédagogique est protégé contre la diffusion externe (copie, republication, partage non autorisé). Les connexions et actions de navigation peuvent être enregistrées à des fins de sécurité, de prévention des abus et de conformité.

- Démarrer le premier module
- Niveau 2
- Accéder au sommaire

Progression	Pratique	Support
Suivez les étapes dans l'ordre et validez	Chaque section se termine par une mise	Notez vos questions et revenez e

cobol.pacta.com

COBOL en 1H

Initiation rapide

01 - Introduction

02 - Origine

03 - Caractéristiques

04 - Structure

05 - Exemples

06 - Cobol de nos jours

07 - Apprendre le cobol

08 - Questions / réponses

Index Alfabétique

Cobol 24H

Cobol 30J

⚡ Bienvenue dans l'espace cobol en 1 h

Accès à un module d'initiation rapide

Introduction aux premières commandes

Deux parcours courts pour entrer dans le langage sans vous disperser : une démonstration guidée, puis les premières commandes à pratiquer.

Parcours débutant

Initiation rapide

Une mise en route immédiate, en mode démonstration : vous voyez le résultat, puis vous comprenez le code pas à pas. Idéal pour prendre confiance avant d'approfondir.

Accéder à l'initiation rapide

Conseillé sur mobile : vous suivez, vous testez, vous recommencez.

Découvrir les premières commandes

Un parcours clair des instructions essentielles : affichage, saisie, conditions, manipulation de texte. Chaque commande est expliquée simplement, avec un exemple minimal et une erreur fréquente à éviter.

Explorer les premières commandes

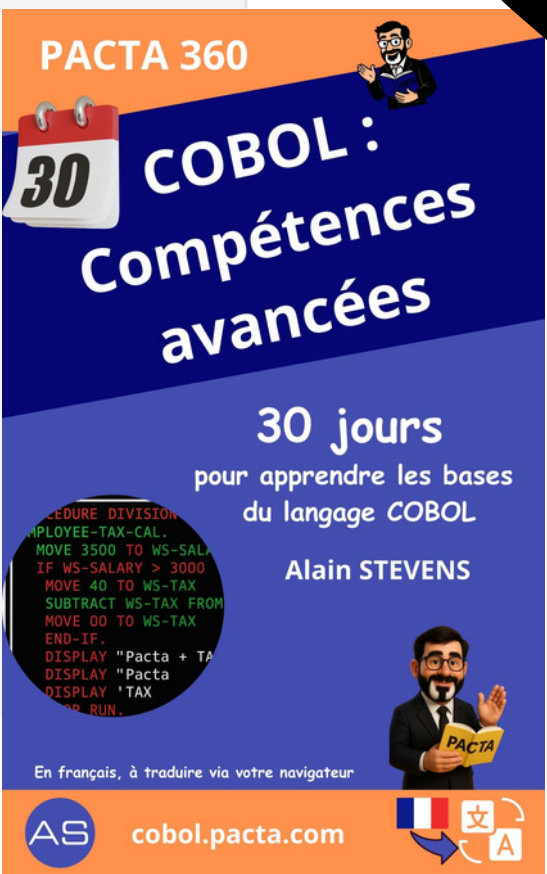
Objectif : pratiquer vite, comprendre mieux, mémoriser durablement.

📌 Objectif de ce module

- Découvrir la logique du cobol
- Comprendre la structure d'un programme
- Poser des bases solides et claires

🔗 Intégration dans la formation

- Première étape du parcours en 30 jours
- Point d'entrée recommandé pour débiter
- Référence utile tout au long de la formation



cobol.pacta.com

▶ Première approche cobol – pas à pas (Initiation rapide)

Cette rubrique propose une **première approche directe du cobol**, conçue pour permettre une **prise en main rapide** du langage, même sans connaissance préalable en informatique.

L'objectif n'est pas d'entrer dans la théorie ou les détails techniques, mais de **découvrir concrètement comment fonctionne le cobol**, étape par étape, à travers des exemples simples et commentés.

- ⚡ À quoi sert cette partie
- Comprendre rapidement la logique du cobol
 - Voir à quoi ressemble un programme réel
 - Lever les blocages avant d'aller plus loin

- 📌 Comment l'utiliser
- Suivre les étapes dans l'ordre proposé
 - Observer les exemples sans pression
 - Revenir sur cette section à tout moment

📄 Cette partie est idéale pour démarrer. Elle sert de **point d'entrée recommandé** avant les modules plus détaillés de la formation cobol en 30 jours.

Derniers articles :

- 📄 Les divisions 📄 Structure d'un programme 📄 Display 📄 Clause Picture A 📄 Clause Picture 9 📄 Clause Picture 9 Variantes 📄 Clause Picture X 📄 SUBTRACT 📄 Hierarchie du code 📄 ACCEPT
- ➕ ADD A B GIVING C 📄 COMPUTE ⚡ IF 🔄 PERFORM ⚡ MOVE 📄 READ 📄 EVALUATE ▶ ADD avec animation 📄 ADD avec animation pas a pas

📄 Premier programme Hello Pacta

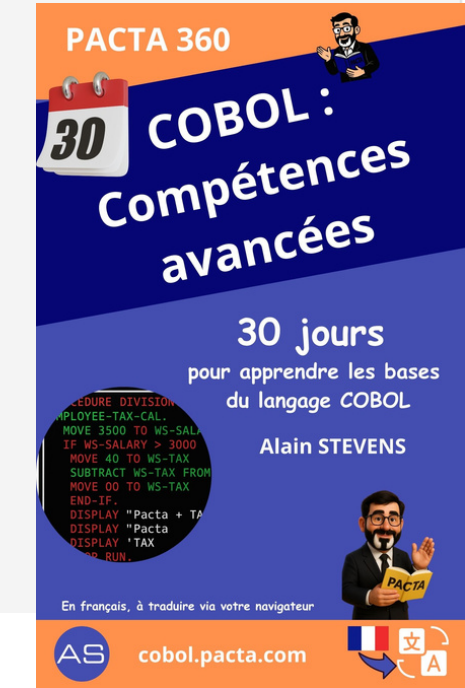
Pourquoi choisir le mode pas à pas ?

Apprendre le COBOL peut sembler intimidant au départ, surtout lorsque l'on découvre pour la première fois la structure d'un programme, les notions de mémoire ou encore les instructions conditionnelles. C'est pour cette raison que nous avons conçu le **mode pas à pas**.

Ce mode d'apprentissage progressif a été pensé pour vous permettre de :

- 🧠 **Mieux mémoriser** les instructions COBOL grâce à l'exécution guidée, ligne par ligne ;
- 👁 **Comprendre ce qui se passe réellement** dans la mémoire de l'ordinateur à chaque étape ;

De nouvelles
rubriques



COURS D'INTRODUCTION AU COBOL

Les 4 divisions principales d'un programme COBOL

Un programme COBOL est organisé en quatre grandes divisions qui doivent apparaître dans un ordre précis : IDENTIFICATION, ENVIRONMENT, DATA et PROCEDURE.

Structure du programme

Niveau débutant

Exemples de code



DIVISION 1

IDENTIFICATION DIVISION

Première division du programme, elle fournit l'identité et les informations descriptives du programme COBOL.



DIVISION 2

ENVIRONMENT DIVISION

Décrit l'environnement d'exécution : machine, système, fichiers physiques et supports d'entrée/sortie.



DIVISION 3

DATA DIVISION

Contient la description des fichiers et des variables utilisées par le programme (FILE, WORKING-STORAGE, etc.).



DIVISION 4

PROCEDURE DIVISION

Regroupe les instructions exécutées : affichages, lectures de fichiers, calculs, tests, boucles.

Une vue d'ensemble



COURS D'INTRODUCTION AU COBOL

Les 4 divisions principales d'un programme COBOL

Un programme COBOL est organisé en quatre grandes divisions qui doivent apparaître dans un ordre précis : IDENTIFICATION, ENVIRONMENT, DATA et PROCEDURE.

Structure du programme

Niveau débutant

Exemples de code



DIVISION 1

IDENTIFICATION DIVISION

Première division du programme, elle fournit l'identité et les informations descriptives du programme COBOL.

IDENTIFICATION DIVISION : carte d'identité du programme

Cette division indique au compilateur comment s'appelle le programme et qui l'a écrit. Le paragraphe **PROGRAM-ID** est obligatoire ; d'autres paragraphes comme **AUTHOR** ou **DATE-WRITTEN** servent de documentation.

Pour un débutant, tu peux te dire : « je commence toujours mon programme COBOL en précisant son nom ici. »

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. HELLO-PROG.  
AUTHOR. Pacta.  
DATE-WRITTEN. 2026-01-20.
```



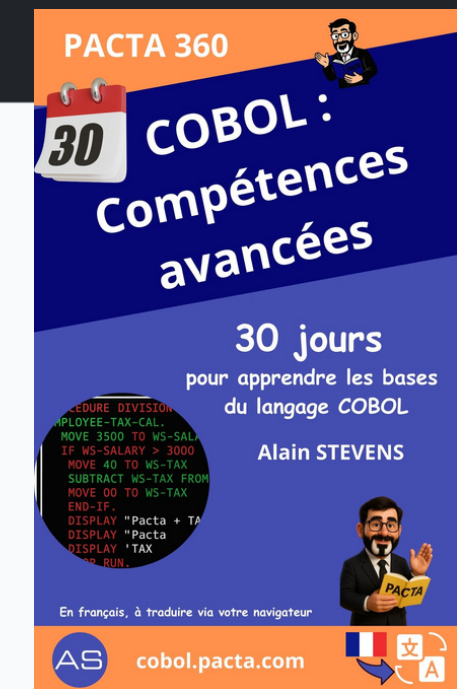
DIVISION 2

ENVIRONMENT DIVISION

Décrit l'environnement d'exécution : machine, système, fichiers physiques et supports d'entrée/sortie.



DIVISION 3



Des informations complémentaires

cobol.pacta.com

COBOL Tutorial

COBOL - Home

Mode demo

Index alphabétique

01 - Introduction

02 - Origine

03 - Caractéristiques

04 - Structure

05 - Exemples

06 - Cobol de nos jours

07 - Apprendre le Cobol

08. Questions / Réponses

PACTA 360

30

COBOL :
Compétences
avancées

30 jours

pour apprendre les bases
du langage COBOL

Alain STEVENS

PROCEDURE DIVISION.
EMPLOYEE-TAX-CALC.
MOVE 3500 TO WS-SAL.
IF WS-SALARY > 3000
MOVE 40 TO WS-TAX
SUBTRACT WS-TAX FROM
MOVE 00 TO WS-TAX
END-IF.
DISPLAY "Pacta + TA
DISPLAY "Pacta
DISPLAY "TAX
STOP RUN.

En français, à traduire via votre navigateur

AScobol.pacta.com

[Accueil 1h](#) - [Accueil démo](#) - [Accueil 24H](#)

Structure de base d'un programme COBOL

Cette leçon présente la structure minimale d'un programme COBOL, en affichant simplement un message : `Bonjour Pacta Cobol !`. C'est l'occasion de comprendre le rôle de chaque division du programme.

Un programme COBOL est structuré en plusieurs divisions, chacune ayant un rôle bien précis. La **IDENTIFICATION DIVISION** sert à nommer le programme et à en définir l'identité. La **ENVIRONMENT DIVISION** permet de spécifier l'environnement dans lequel le programme s'exécute, bien que dans les programmes simples elle soit souvent réduite ou vide. La **DATA DIVISION** contient les sections dédiées aux données utilisées, comme la **WORKING-STORAGE SECTION** où sont définies les variables. Enfin, la **PROCEDURE DIVISION** contient les instructions à exécuter, comme `DISPLAY` pour afficher un message ou `STOP RUN` pour terminer le programme. Ces divisions structurent le code de façon lisible et ordonnée, même pour les programmes les plus simples.

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. BONJOUR-PROGRAMME.  
ENVIRONMENT DIVISION.  
CONFIGURATION SECTION.  
DATA DIVISION.  
WORKING-STORAGE SECTION.  
PROCEDURE DIVISION.  
    DISPLAY "Bonjour Pacta Cobol !"  
    STOP RUN.
```

IDENTIFICATION DIVISION : Indique l'identité du programme.
PROGRAM-ID : Nom unique du programme COBOL.
ENVIRONMENT DIVISION : Spécifie l'environnement d'exécution (facultative ici).
CONFIGURATION SECTION : Partie de l'ENVIRONMENT DIVISION (souvent vide dans un exemple simple).
DATA DIVISION : Contient les déclarations de données.
WORKING-STORAGE SECTION : Zone mémoire pour stocker des données.
PROCEDURE DIVISION : Partie où les instructions s'exécutent.
DISPLAY : Affiche le texte à l'écran.
STOP RUN : Termine le programme COBOL.

[▶ Lancer](#) [|| Pause](#) [← Suivant](#) [↺ Rejouer](#)

Téléchargez le code source : [🔍 Voir dans un nouvel onglet](#) | [📄 Télécharger le fichier](#)

Des explications détaillées
avec des animations claires

cobol.pacta.com

Leçon sur les lignes du code

Ici, vous apprenez à "lire" un programme. On commence par l'identité du programme, puis on décrit les données, puis on exécute des actions. Chaque mot-clé est un repère : si vous suivez l'ordre, vous comprenez déjà la logique générale.

des infographies

Fonction DISPLAY en COBOL: Affichage à l'écran

Code COBOL

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. DEMO-DISPLAY.  
  
DATA DIVISION.  
WORKING-STORAGE SECTION.  
01 WS-NOM-FORMATION PIC X(30) VALUE "Micro-learning COBOL".  
  
PROCEDURE DIVISION.  
  DISPLAY "Bonjour le monde COBOL!".  
  DISPLAY "Formation: " WS-NOM-FORMATION.  
  STOP RUN.
```



Terminal (Sortie)



Envoie
vers l'écran



Affiche la valeur
de la variable

cobol.pacta.com

Ce schéma résume l'idée : la **DATA DIVISION** décrit les variables (par exemple le nom d'une formation), puis la **PROCEDURE DIVISION** utilise **DISPLAY** pour afficher un texte et la valeur de la variable. Le but n'est pas d'apprendre par cœur, mais de reconnaître le rôle de chaque bloc.

cobol.pacta.com



Animation : une ligne de code toutes les 5 secondes

Un mode pas à pas avec une simulation de console

```
IDENTIFICATION DIVISION.  
  
PROGRAM-ID. DEMO-DISPLAY.  
  
DATA DIVISION.  
  
WORKING-STORAGE SECTION.  
  
01 WS-NOM-FORMATION PIC X(30) VALUE "Micro-learning COBOL".  
  
PROCEDURE DIVISION.  
  
    DISPLAY "Bonjour le monde COBOL!".  
  
    DISPLAY "Formation: " WS-NOM-FORMATION.  
  
    STOP RUN.
```

Sortie console

```
Bonjour le monde COBOL!  
Formation: Micro-learning COBOL
```

Points clés à retenir

cobol.pacta.com

une variable, ou les deux. Le texte entre guillemets est affiché tel quel.

dans la **DATA DIVISION**, souvent en **WORKING-STORAGE**, sinon l'affichage devient incohérent ou impossible à compiler.

PACTA 360

30 COBOL :
Compétences
avancées

30 jours
pour apprendre les bases
du langage COBOL

Alain STEVENS

En français, à traduire via votre navigateur

AS cobol.pacta.com

```
PROCEDURE DIVISION  
EMPLOYEE-TAX-CAL.  
MOVE 3500 TO WS-SAL.  
IF WS-SALARY > 3000  
MOVE 40 TO WS-TAX  
SUBTRACT WS-TAX FROM  
MOVE 00 TO WS-TAX  
END-IF.  
DISPLAY "Pacta + TA  
DISPLAY "Pacta  
DISPLAY "TAX  
STOP RUN.
```



Aa Clause PICTURE A

Variables alphanumériques en COBOL : PIC A

Le symbole **A** dans la clause PICTURE sert à déclarer des zones qui ne contiennent que des lettres et des espaces, par exemple pour des prénoms ou codes alphanumériques simples.

① A = alphabetic

Longueur avec A(n)

Lancer l'animation

Réinitialiser

Survole les blocs pour voir les explications.

SIMULATION D'UNE VARIABLE ALPHABÉTIQUE

A₁ A₂ A₃ A₄ A₅

Position 1 Position 2 Position 3 Position 4 Position 5

Exemple : **PACTA** est valide, **PAC7A** est refusé (chiffre interdit).

VISUALISATION (VALEURS TESTÉES)

PAC-NAME : _____
Test 1 : _____
Test 2 : _____

PACTA 360

**30 COBOL :
Compétences
avancées**

30 jours
pour apprendre les bases
du langage COBOL

Alain STEVENS

En français, à traduire via votre navigateur

AS cobol.pacta.com

cobol.pacta.com

**Des animations pour
comprendre les
bases du cobol**

Variables alphanumériques en COBOL : PIC A

Le symbole **A** dans la clause PICTURE sert à déclarer des zones qui ne contiennent que des lettres et des espaces, par exemple pour des prénoms ou codes alphanumériques simples.

A = alphabetic

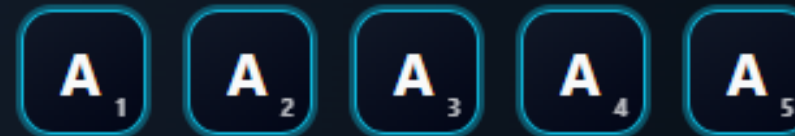
Longueur avec A(n)

► Lancer l'animation

↺ Réinitialiser

Survole les blocs pour voir les explications.

SIMULATION D'UNE VARIABLE ALPHABÉTIQUE



Position 1 Position 2 Position 3 Position 4 Position 5

Exemple : **PACTA** est valide, **PAC7A** est refusé (chiffre interdit).

VISUALISATION (VALEURS TESTÉES)

PAC-NAME : PACTA

Test 1 : PACTA (OK - alphabétique)

Test 2 : PAC7A (OK si PIC X, refusé si PIC A)

Comprendre PIC A pas à pas

ÉTAPE 1 – TYPE ALPHABETIC

①

Un type alphabétique stocke uniquement des lettres de A à Z (majuscule ou minuscule) et l'espace, sans chiffres ni signes de ponctuation.

On l'utilise pour des prénoms, des codes alphanumériques simples ou des champs texte qui ne doivent contenir aucun chiffre.

ÉTAPE 2 – LONGUEUR AVEC A(N)

②

Chaque lettre A dans la PICTURE représente une position, donc PIC A(5) réserve 5 caractères alphanumériques en mémoire.

son abrégée A(5) pour exprimer

cobol.pacta.com

EXEMPLE 1 – DÉCLARATION SIMPLE

* Variable alphabétique 5 caractères

01 WS-NAME PIC A(5).

01 WS-HELLO PIC A(5) VALUE "COBOL".

EXEMPLE 2 – PROGRAMME MINI

* Accepte seulement des lettres pour le prénom

IDENTIFICATION DIVISION.

PROGRAM-ID. PIC-A-DEMO.

DATA DIVISION.

WORKING-STORAGE SECTION.

01 WS-FIRSTNAME PIC A(10) VALUE "ALICE".

PROCEDURE DIVISION.

Des explications
claires

PACTA 360

**30 COBOL :
Compétences
avancées**

30 jours
pour apprendre les bases
du langage COBOL

Alain STEVENS

En français, à traduire via votre navigateur

AS cobol.pacta.com



Hiérarchisation des données

Niveaux 01, 03, 05... et indentation des lignes COBOL

La hiérarchie des niveaux COBOL organise les variables comme un arbre : le niveau **01** définit un groupe racine, les niveaux **03, 05, 07** décrivent ses sous-éléments, avec une indentation stricte.

Niveaux 01 / 03 / 05 / 07 Colonnes 8 et 12

Lancer l'animation Réinitialiser

Regarde comment les niveaux s'imbriquent et se décalent sur la ligne.

01	EMPLOYEE-RECORD.	col. 8
03	EMPLOYEE-NOM.	col. 12
05	EMPLOYEE-TITRE PIC X(3).	détail
05	EMPLOYEE-INITIALES PIC X(4).	déta
05	EMPLOYEE-SURNOM PIC X(30).	détail
05	EMPLOYEE-SEXE PIC X.	même niveau 03
05	EMPLOYEE-ADRESSE.	groupe
07	EMPLOYEE-LIGNE-ADRESSE PIC X(30) OCCUR...	
05	EMPLOYEE-CODE-POSTAL PIC X(8).	détail
05	EMPLOYEE-SALAIRE PIC 9(5)V99.	numérique

Des micro-contenus pour un meilleur apprentissage

Hiérarchie des niveaux COBOL

ÉTAPE 1 – NIVEAUX HIÉRARCHIQUES ①

Le niveau 01 définit un groupe principal : ici, EMPLOYEE-RECORD est l'enregistrement complet d'un employé.

Les niveaux 03, 05, 07 décrivent les sous-parties de cet enregistrement

cobol.pacta.com

②

Traditionnellement, le niveau 01 commence à la colonne 8, les

EXEMPLE – DÉCLARATION EMPLOYEE-RECORD

```
* Hiérarchie des niveaux pour un enregistrement employé
* 01 en haut de l'arbre, 03 / 05 / 07 pour les sous-parties
01 EMPLOYEE-RECORD.
    03 EMPLOYEE-NOM.
        05 EMPLOYEE-TITRE          PIC X(3).
        05 EMPLOYEE-INITIALES      PIC X(4).
        05 EMPLOYEE-SURNOM         PIC X(30).
    05 EMPLOYEE-SEXE               PIC X.
    05 EMPLOYEE-ADRESSE.
```

PACTA 360

30 COBOL :
Compétences
avancées

30 jours
pour apprendre les bases
du langage COBOL

Alain STEVENS

En français, à traduire via votre navigateur

AS cobol.pacta.com

PACTA 360

30

COBOL :
Compétences
avancées

30 jours

pour apprendre les bases
du langage COBOL

Alain STEVENS



En français, à traduire via votre navigateur

AS

cobol.pacta.com



Commandes cobol à pratiquer en priorité

Cette page rassemble 20 instructions de base à manipuler avant de poursuivre : affichage, saisie, conditions, boucles, calculs, traitement de chaînes et premières opérations sur fichiers. Pour chaque commande, un bouton vous renvoie vers l'index afin de retrouver des exemples et des leçons associées.

Des exemples de code

Vue colonnes

Vue tableau

Liste des commandes

DISPLAY

Affiche un texte, une variable, ou les deux. Très utile pour vérifier rapidement ce que fait un programme.

Voir les exemples

ACCEPT

Récupère une saisie (ou une valeur système selon l'environnement). Sert à rendre un programme interactif.

Voir les exemples

MOVE

Copie une valeur dans une variable. Indispensable pour initialiser, préparer un calcul ou construire un message.

Voir les exemples

IF

Teste une condition. Permet de déclencher une action seulement si une règle est respectée.

Voir les exemples

ELSE

Complète un test : indique quoi faire lorsque la condition n'est pas vraie.

Voir les exemples

END-IF

Ferme clairement un bloc conditionnel. Améliore la lisibilité et évite les erreurs d'imbrication.

Voir les exemples

WHEN

Permet de sélectionner une action à effectuer (switch/case"). Très utile pour les menus.

Voir les exemples

PERFORM

Exécute un paragraphe/une section, ou répète un traitement. Sert à structurer un programme en étapes.

Voir les exemples

END-PERFORM

Ferme un bloc de répétition/traitement écrit en forme "inline". Utile pour garder un code bien délimité.

Voir les exemples

cobol.pacta.com

Commande	Résumé	Action
DISPLAY	Affichage de texte et/ou variables, utile pour tester et comprendre l'exécution.	Voir les exemples
ACCEPT	Saisie utilisateur ou valeur système selon l'environnement.	Voir les exemples
MOVE	Copie et préparation des données dans les variables.	Voir les exemples
IF	Test conditionnel pour appliquer une règle.	Voir les exemples
ELSE	Cas alternatif lorsque la condition n'est pas vraie.	Voir les exemples
END-IF	Fermeture d'un traitement "inline" en IF.	Voir les exemples
EVALUATE	Gestion de données selon des conditions.	Voir les exemples
PERFORM	Appel de paragraphes/sections.	Voir les exemples
END-PERFORM	Fermeture d'un traitement "inline" en PERFORM.	Voir les exemples
ADD	Addition (compteurs, totaux).	Voir les exemples
SUBTRACT	Soustraction (différences, décréments).	Voir les exemples
MULTIPLY	Multiplication (quantités, coefficients).	Voir les exemples
DIVIDE	Calcul du quotient et reste possibles.	Voir les exemples
COMPUTE	Calcul d'expressions plus complètes.	Voir les exemples

Une navigation pratique
et conviviale

cobol.pacta.com

PACTA 360

30

COBOL :
Compétences
avancées

30 jours
pour apprendre les bases
du langage COBOL

Alain STEVENS

```
PROCEDURE DIVISION  
EMPLOYEE-TAX-CALC.  
MOVE 3500 TO WS-SAL  
IF WS-SALARY > 3000  
MOVE 40 TO WS-TAX  
SUBTRACT WS-TAX FROM  
MOVE 00 TO WS-TAX  
END-IF.  
DISPLAY "Pacta + TA  
DISPLAY "Pacta  
DISPLAY "TAX  
GO RUN
```

En français, à traduire via votre navigateur

AS cobol.pacta.com

Astuce : le bloc de code est en <pre>, donc les espaces sont conservés.

Exemple minimal **COBOL** Format fixe

Télécharger le code

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. EXO-DISPLAY-01.  
  
DATA DIVISION.  
WORKING-STORAGE SECTION.  
01 WS-PRENOM          PIC X(20) VALUE "ALAIN".  
01 WS-MESSAGE          PIC X(60) VALUE SPACES.  
  
PROCEDURE DIVISION.  
    MOVE "Bonjour COBOL : " TO WS-MESSAGE.  
    DISPLAY WS-MESSAGE.  
    DISPLAY "Prénom : " WS-PRENOM.  
    STOP RUN.
```

Ce que fait le programme

- 1) Il place un texte dans WS-MESSAGE avec MOVE.
- 2) Il affiche ce texte avec DISPLAY.
- 3) Il affiche ensuite une étiquette et la valeur de WS-PRENOM.
- 4) Il s'arrête avec STOP RUN.

À retenir : DISPLAY peut afficher un texte entre guillemets, une variable, ou une combinaison d'éléments à la suite. Une variable affichée doit être déclarée dans la section WORKING-STORAGE.

Votre micro-exercice

Faites ces trois modifications (courtes, mais essentielles) :

1. Changez la valeur de WS-PRENOM avec votre prénom.
2. Ajoutez un second DISPLAY qui affiche : "Fin de test."
3. Ajoutez une variable WS-VILLE et affichez-la avec un libellé.

Erreur fréquente à éviter

Afficher une variable non déclarée : écrire DISPLAY WS-VILLE sans avoir créé WS-VILLE dans WORKING-STORAGE.

Autre piège : démarrer une instruction trop à gauche. Dans un format fixe, gardez les instructions en zone B (ici, on les place après 11 espaces).

Conseil : utilisez des libellés clairs (ex : "Prénom : ") pour

**Des exemples de code
Cobol simples à
comprendre**

cobol.pacta.com

l'index

Retour accueil



Structure en trois divisions

Les blocs apparaissent d'abord en pile, puis passent en superposition "cascade" avec décalage. Le fond devient quasi opaque au final pour éviter toute transparence du texte et tout mélange de lettres.

Fondu + glissement, puis cascade sans transparence

● Identification

Présenter le programme

Cette division sert à donner une identité au programme : nom, auteur, dates, informations de documentation. Elle agit comme une fiche descriptive.

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. DEMO-STRUCTURE.
```

● Environment

Cette division indique le contexte d'exécution : éléments de certains fichiers, et aux conventions dépendantes de

```
ENVIRONMENT DIVISION.  
CONFIGURATION SECTION.
```

Contrôles

Utilisez les boutons pour guider l'apparition des divisions. Astuce : **Entrée** déclenche « Étape suivante », et **Échap** réinitialise.

Démarrer l'animation

Début

Étape suivante

Progressif

Animation automatique

Lecture

Réinitialiser

Retour

État : Étape 2/4 : apparition de la division Environment.

des animations
interactives