Formation Cobol

Découverte du langage

Le COBOL (Common Business-Oriented Language) est un langage structuré principalement utilisé dans les applications commerciales et administratives. Je vais vous présenter les commandes et concepts de base pour débuter, avec des explications simples. Notez que le COBOL est très verbeux et suit une structure rigide avec des divisions spécifiques.

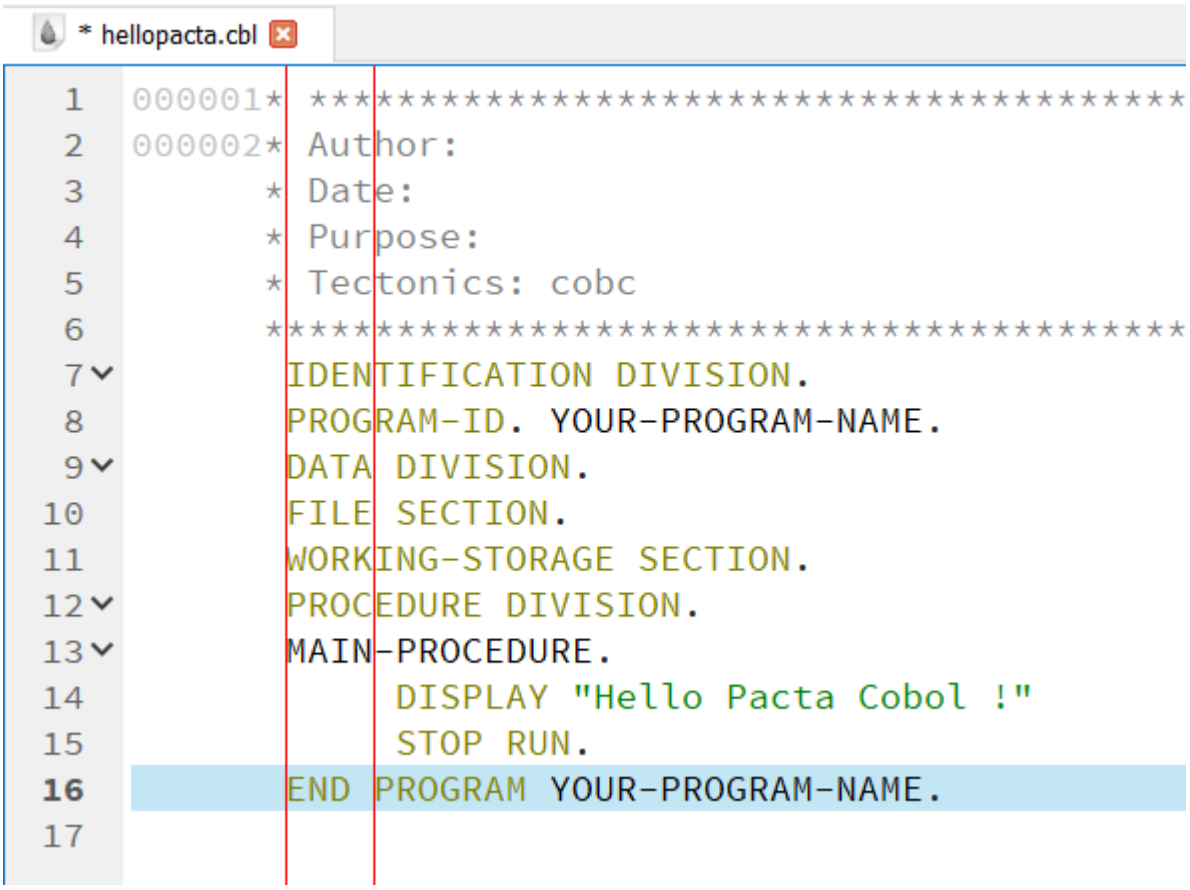
Vous retrouverez d'autres extrait de notre formation Cobol en libre accès sur notre site https://cobol.pacta.com/

Voici un guide pour débutant, qui pourra faire des essais sur les compilateurs gratuits disponibles sur le Web. Ces exemples sont envoyés par des participants sur nos forums, en cas d'erreur, envoyez-nous un mail pour apporter des rectifications.

Structure de base d'un programme COBOL

Un programme COBOL est organisé en **divisions**, qui sont obligatoires et doivent apparaître dans cet ordre :

1. **IDENTIFICATION DIVISION** : Identifie le programme.
2. **ENVIRONMENT DIVISION** : Décrit l'environnement (facultatif pour un débutant).
3. **DATA DIVISION** : Définit les variables.
4. **PROCEDURE DIVISION** : Contient la logique du programme.



Exemple simple avec commandes de base

Voici un programme COBOL minimaliste qui affiche "Bonjour, monde !" :

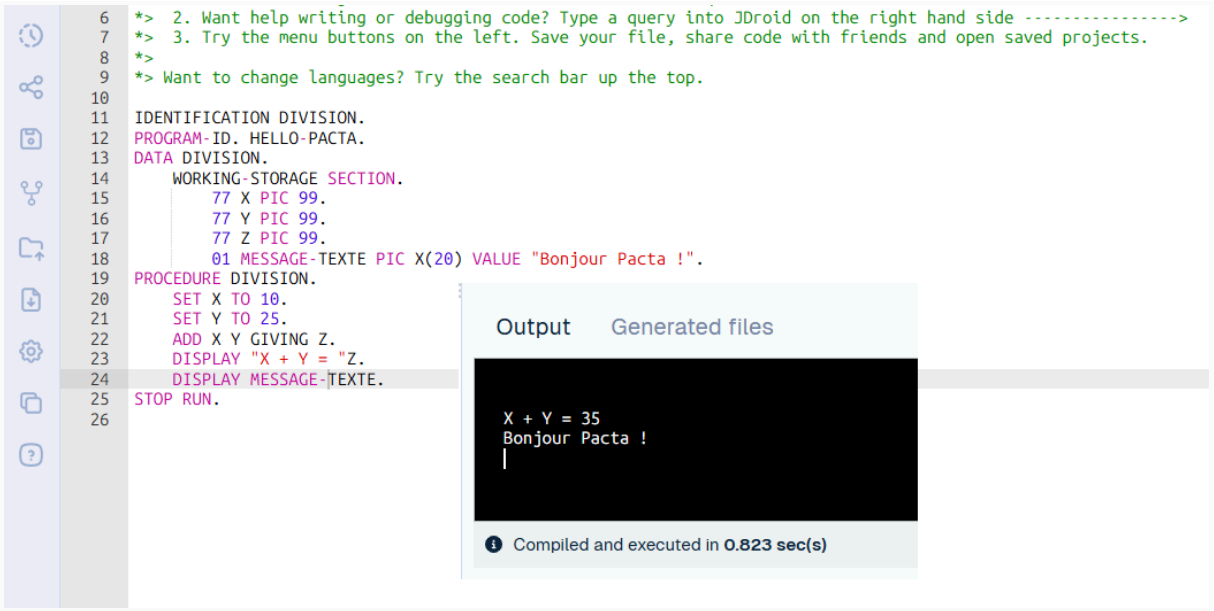
```
IDENTIFICATION DIVISION.
PROGRAM-ID. BonjourMonde.

DATA DIVISION.
WORKING-STORAGE SECTION.
01  MESSAGE-TEXTE    PIC X(20) VALUE "Bonjour, monde !".

PROCEDURE DIVISION.
```

```
    DISPLAY MESSAGE-TEXTE.
    STOP RUN.
```

Voici un exemple sur un éditeur en ligne .



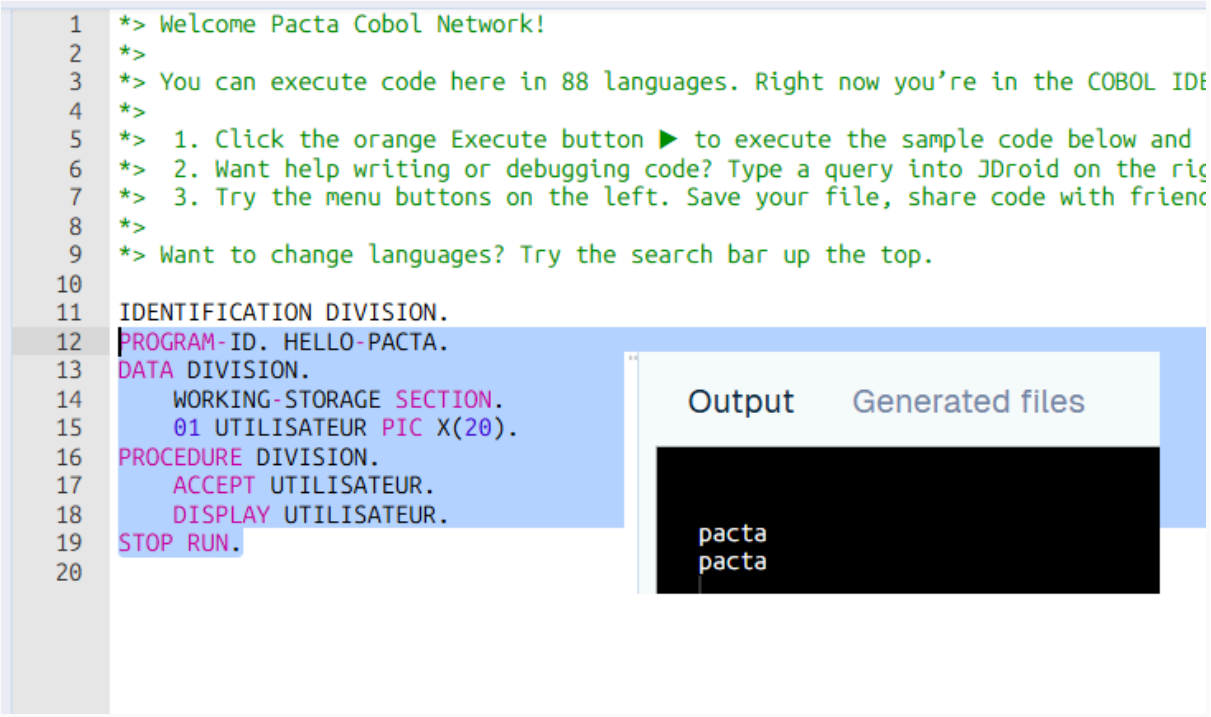
Explications des commandes de base

1. **IDENTIFICATION DIVISION** :
   - `PROGRAM-ID.` : Nom du programme (obligatoire). Exemple : `PROGRAM-ID. BonjourMonde.`
2. **DATA DIVISION** :
   - Utilisée pour déclarer des variables.
   - `WORKING-STORAGE SECTION.` : Section pour définir les variables temporaires.
   - `01` : Niveau de définition d'une variable (01 est le plus haut niveau pour une variable individuelle).
   - `PIC X(20)` : Type de données (ici, une chaîne de caractères de 20 caractères max).
   - `VALUE` : Initialise la variable avec une valeur.
3. **PROCEDURE DIVISION** :
   - Contient les instructions exécutables.
   - `DISPLAY` : Affiche une variable ou un texte à l'écran. Exemple : `DISPLAY "Hello".`
   - `STOP RUN` : Termine le programme (obligatoire pour arrêter l'exécution).

Autres commandes essentielles pour débutants

- **ACCEPT** : Lit une entrée de l'utilisateur.

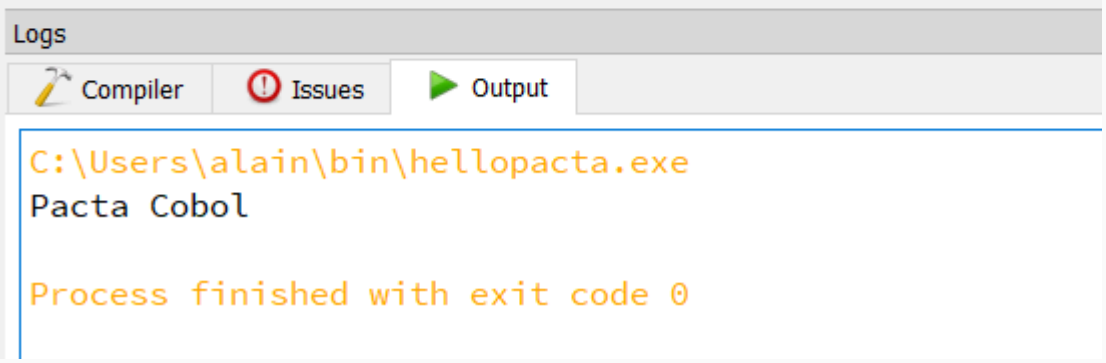
```
01  ENTREE-UTILISATEUR  PIC X(10).
ACCEPT ENTREE-UTILISATEUR.
DISPLAY "Vous avez entré : " ENTREE-UTILISATEUR.
```



- **MOVE** : Affecte une valeur à une variable.

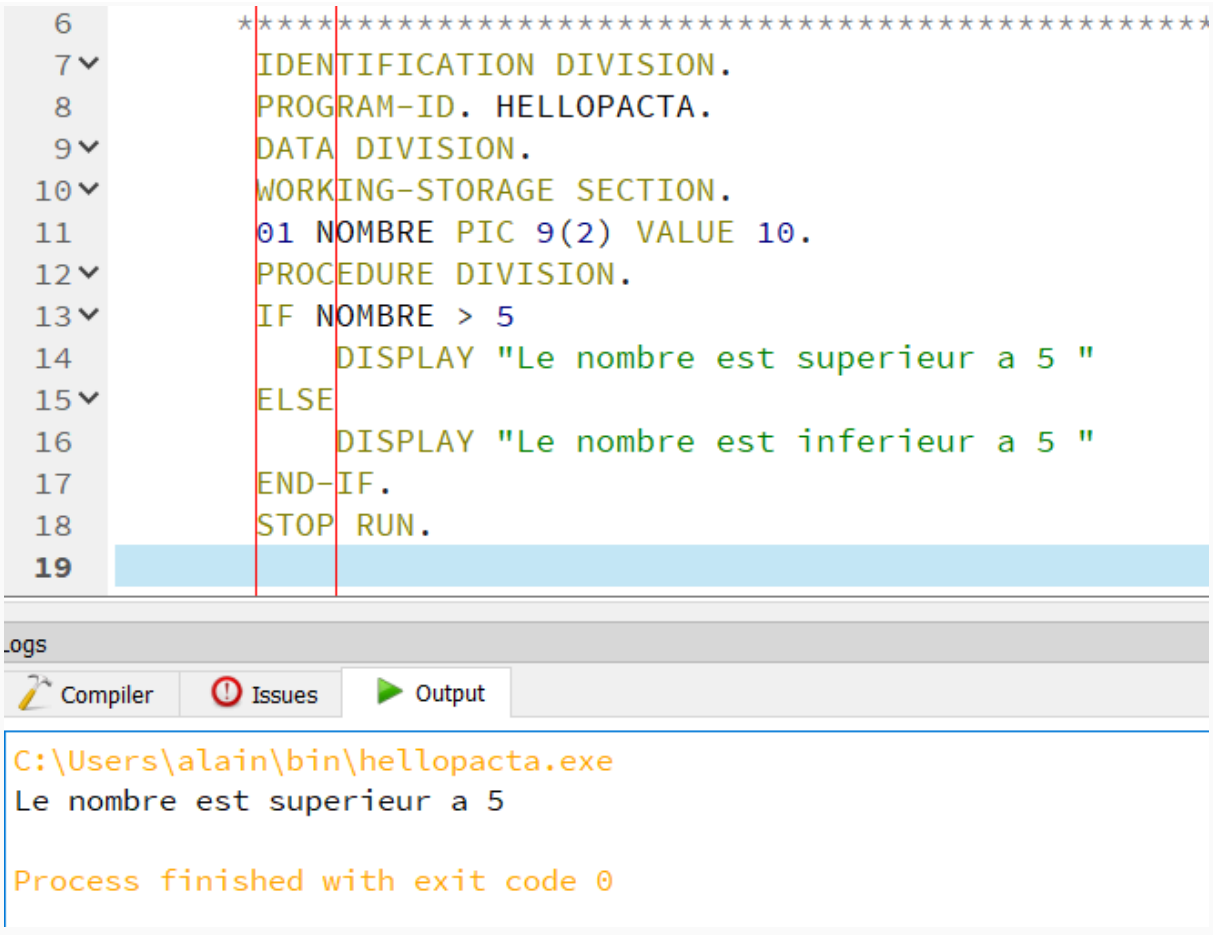
```
01  MA-VARIABLE  PIC X(10).
MOVE "Test" TO MA-VARIABLE.
DISPLAY MA-VARIABLE.
```

```
7        IDENTIFICATION DIVISION.
8        PROGRAM-ID. HELLOPACTA.
9        DATA DIVISION.
10       WORKING-STORAGE SECTION.
11       01 VARIABLE PIC X(15).
12       PROCEDURE DIVISION.
13       MOVE "Pacta Cobol" to VARIABLE
14       DISPLAY VARIABLE.
15       STOP RUN.
16
```



- **IF** : Condition simple.

```
01  NOMBRE  PIC 9(2) VALUE 10.
IF NOMBRE > 5
    DISPLAY "Nombre est supérieur à 5"
ELSE
    DISPLAY "Nombre est inférieur ou égal à 5"
END-IF.
```



- **PERFORM** : Boucle simple.

```
PERFORM 3 TIMES
    DISPLAY "Répétition"
END-PERFORM.
```

Types de données de base (PIC)

- `PIC X(n)` : Chaîne de caractères de longueur `n`.
- `PIC 9(n)` : Nombre entier de `n` chiffres.
- `PIC 9(n)V9(m)` : Nombre décimal avec `n` chiffres avant la virgule et `m` après.

```cobol
IDENTIFICATION DIVISION.
PROGRAM-ID. HELLO-WORLD.
DATA DIVISION.
WORKING-STORAGE SECTION.
01 TEXTE PIC X(15) VALUE "Pacta Cobol".
PROCEDURE DIVISION.
DISPLAY TEXTE.
STOP RUN.
```

```
<command-line>: warning: "_FORTIFY_SOURCE" redefined
<command-line>: note: this is the location of the previous definition
Pacta Cobol
```

Conseils pour débuter

- Le COBOL est sensible à l'indentation et aux points (`.`) à la fin des instructions ou sections.
- Utilisez un éditeur comme OpenCOBOL (maintenant GnuCOBOL) pour tester tes programmes.
- Commencez par des programmes simples (affichage, saisie, calculs de base).

Exercices

Essayez d'écrire un programme qui :

1. Demande à l'utilisateur son nom avec `ACCEPT`.
2. Stocke le nom dans une variable.
3. Affiche "Bonjour, [nom] !" avec `DISPLAY`.

Vous trouverez sur les sites partenaires du Pacta Cobol Network des extraits en libre accès des formations payantes sur le langage Cobol.