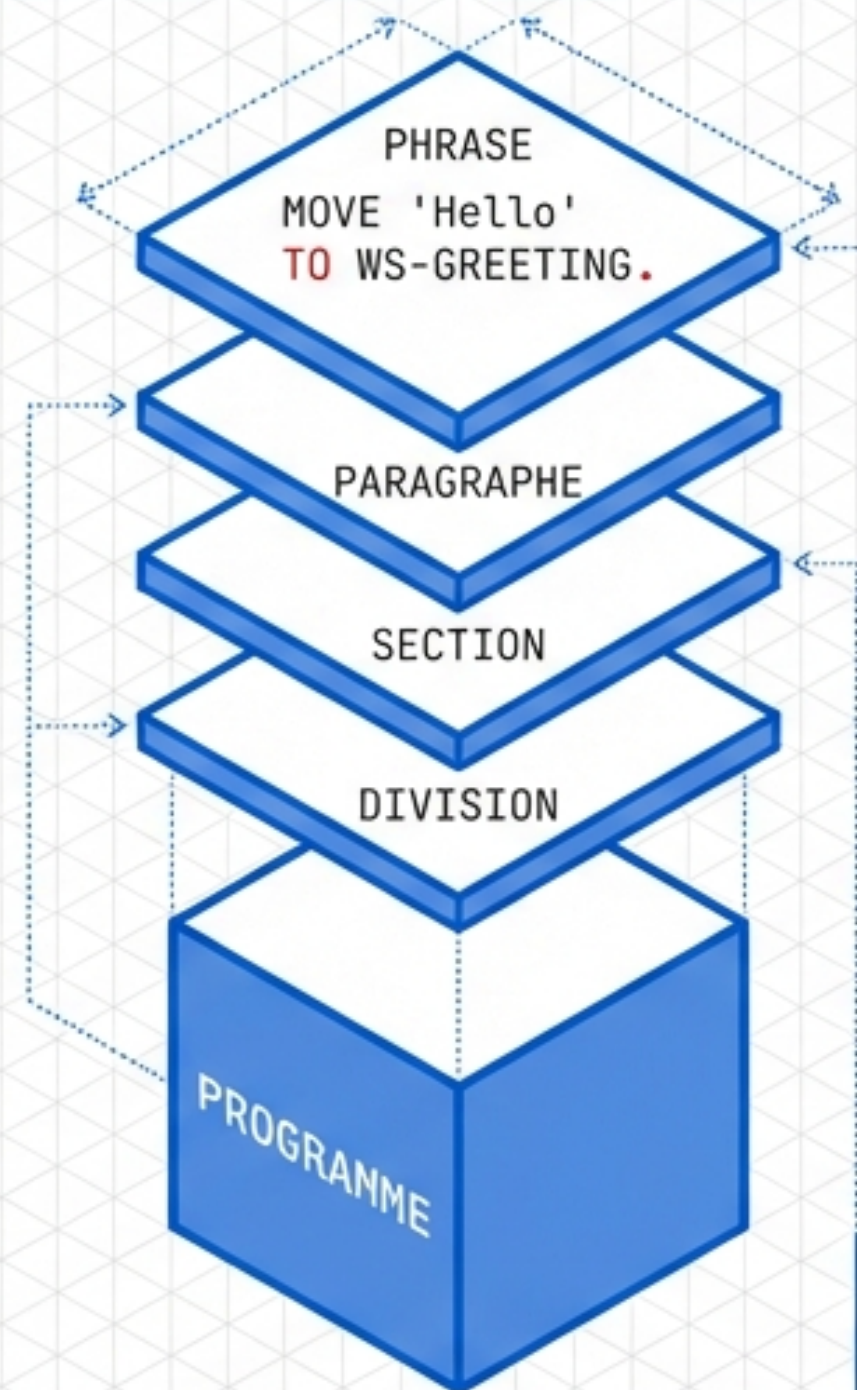


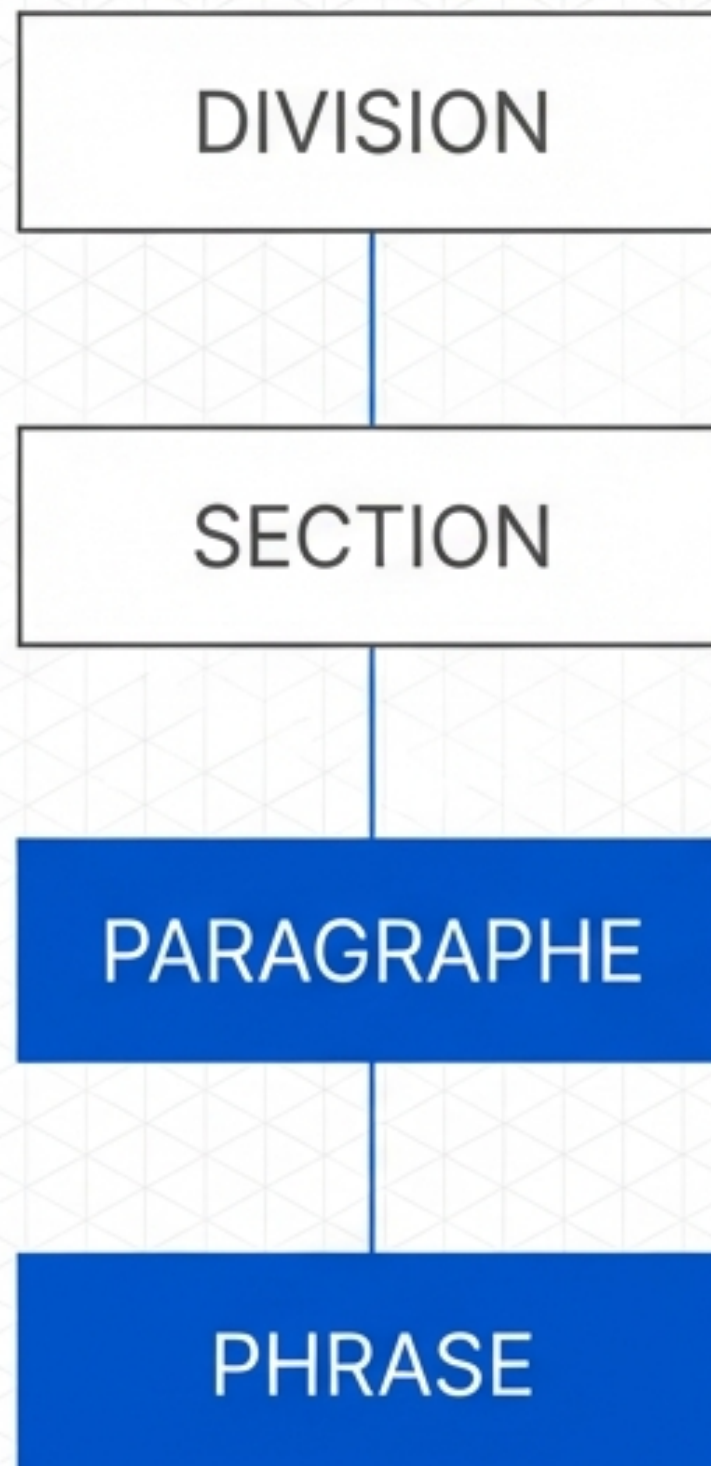
L'Anatomie Modulaire : Paragraphes et Phrases en COBOL

Structurer le code pour la lisibilité, la maintenance et l'efficacité.



Comprendre la structure fondamentale :
Divisions → Sections → Paragraphes → Phrases.

La Hiérarchie du Code COBOL



En COBOL, l'architecture est stricte. Si les Divisions et Sections forment l'ossature, les Paragraphes et Phrases en sont les briques et le mortier.

Le Paragraphe

Un bloc de code identifié par un nom unique.

La Phrase

Une séquence d'instructions se terminant par un **point**.

Ces éléments sont fondamentaux pour écrire des programmes clairs et non monolithiques.

Le Paragraphe : L'Unité Logique

Un paragraphe est une unité logique regroupant des instructions exécutables. Il agit comme un conteneur pour des opérations liées (ex: initialisation, traitement).

```
NOM-PARAGRAPHE.  
instruction 1.  
instruction 2.
```

Identification : Nom unique défini par l'utilisateur.

Règle Critique : Le nom doit être suivi immédiatement d'un point et d'un espace.

```
INITIALISATION.  
MOVE 5 TO TOTAL.
```



La Phrase : La Séquence d'Exécution

La plus petite unité exécutable. Une phrase peut contenir une ou plusieurs instructions, mais elle doit impérativement se terminer par un point.

Instruction Seule (Incomplète)

```
MOVE 0 TO TOTAL ...
```

Phrase Complète (Exécutable)

```
MOVE 0 TO TOTAL.  
ADD 1 TO INDEX.
```

Dans cet exemple, deux instructions sont regroupées en une seule phrase logique.

PACTA 360

30 **COBOL :**
Compétences
avancées

30 jours
pour apprendre les bases
du langage COBOL

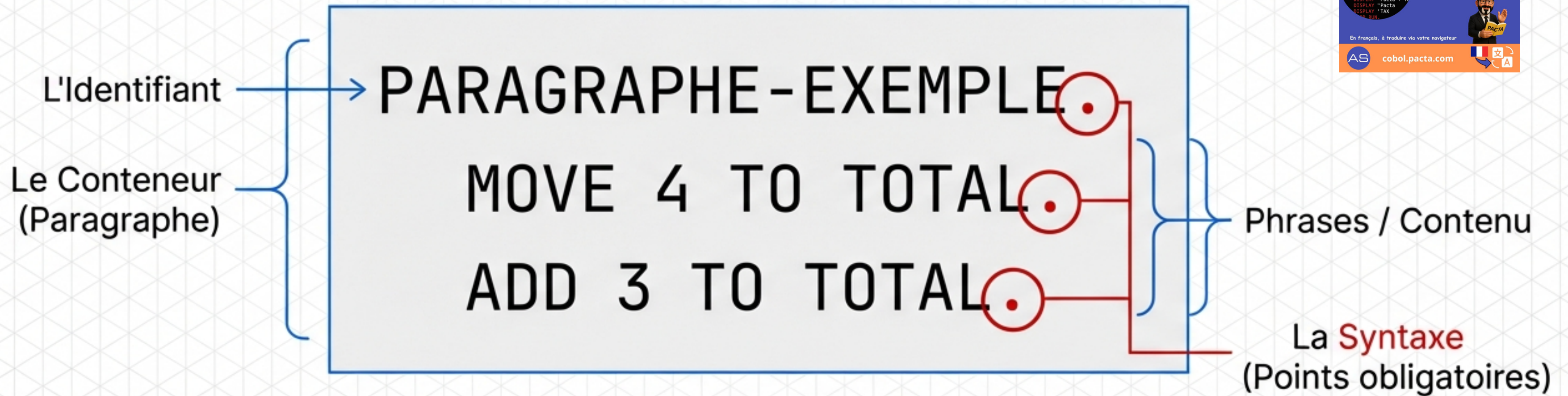
Alain STEVENS

En français, à traduire via votre navigateur

AS cobol.pacta.com



Anatomie d'un Bloc de Code



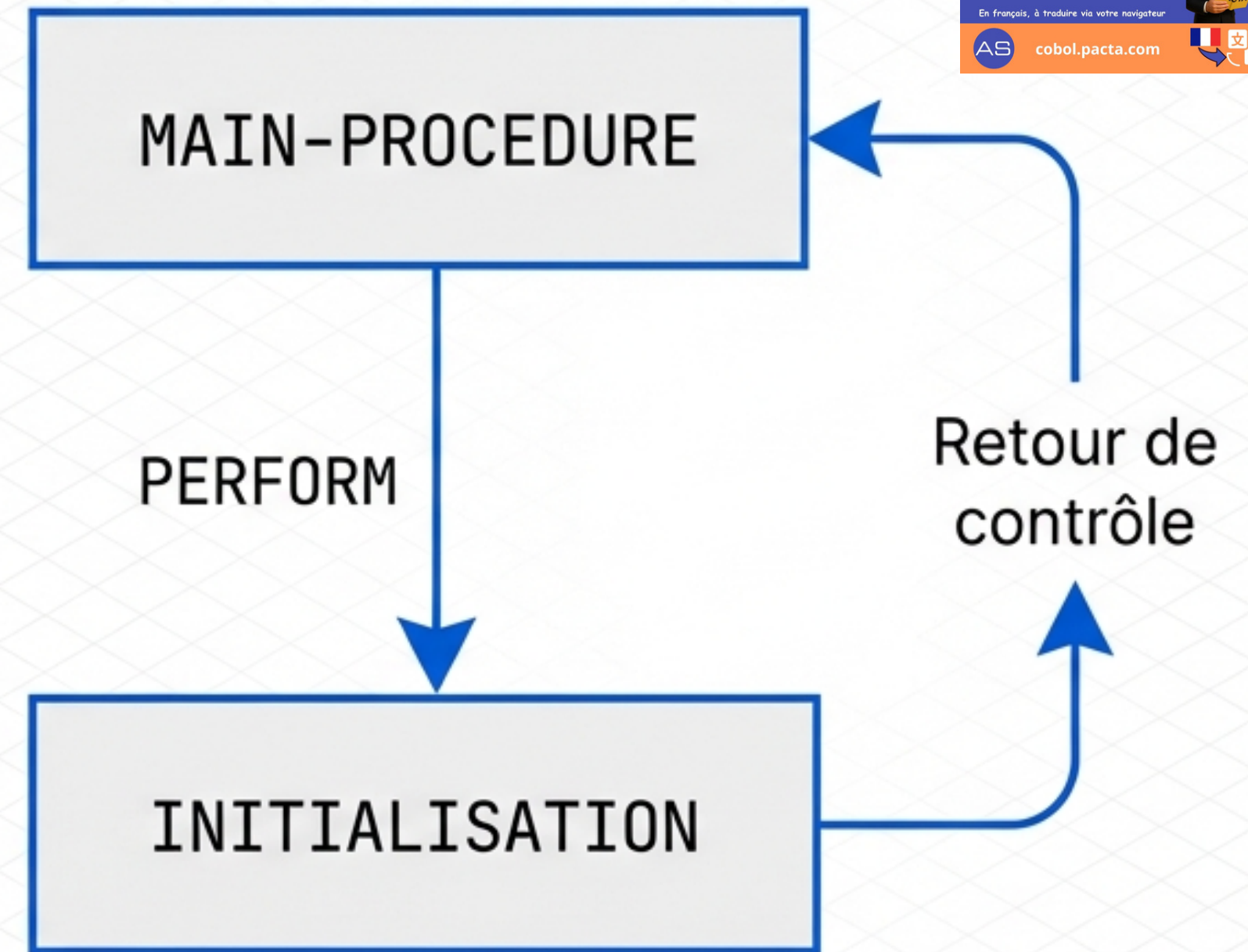
La structure garantit que le compilateur sait exactement où commence et finit chaque bloc logique.

Orchestrer le Flux avec PERFORM

Les paragraphes ne sont pas seulement du stockage ; ils sont des sous-programmes **appelables**. Cela permet la **réutilisation du code** (DRY - Don't Repeat Yourself).

Commande : PERFORM

Syntaxe : PERFORM NOM-PARAGRAPHE.



Flux d'Exécution : Le Fil Rouge

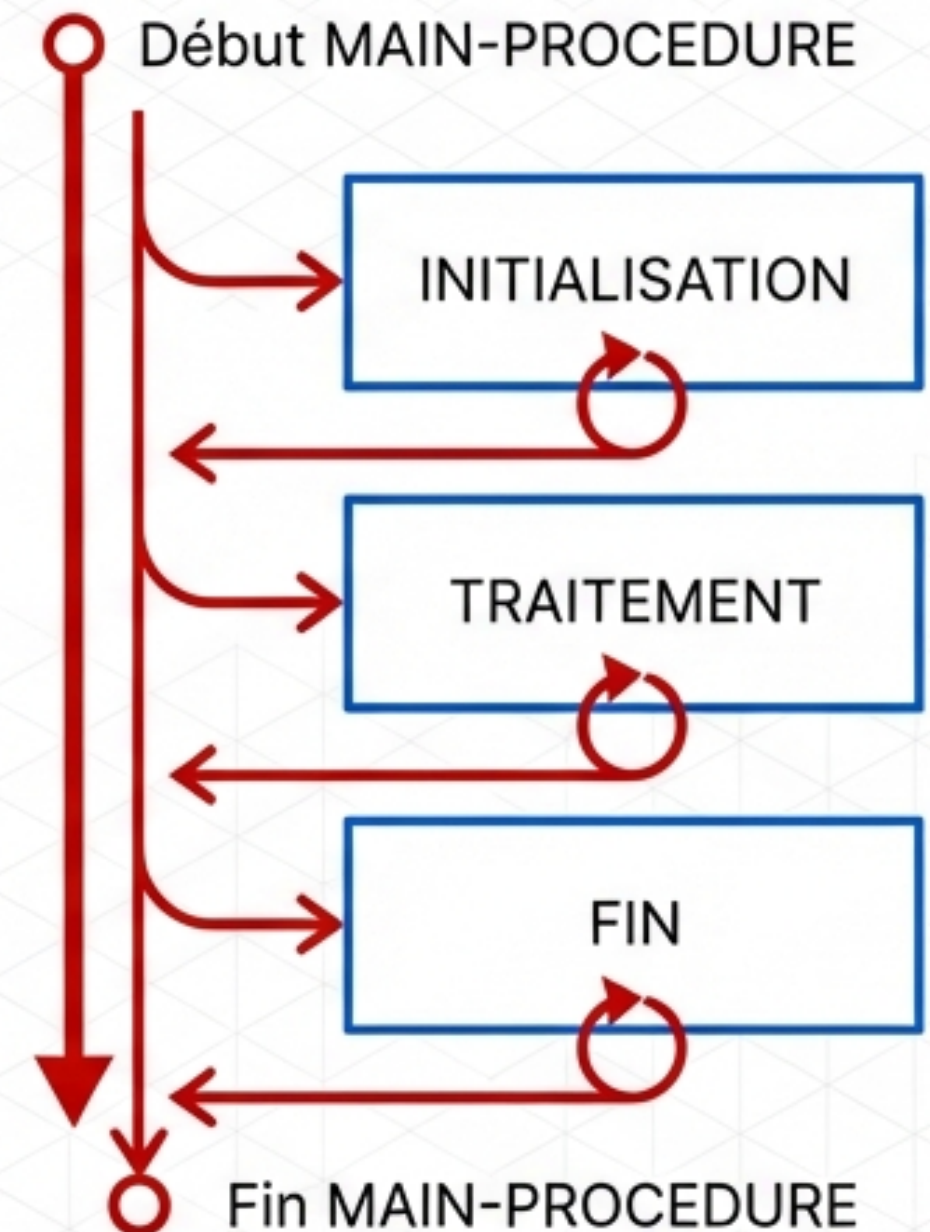
MAIN-PROCEDURE.

PERFORM INITIALISATION.

PERFORM TRAITEMENT.

PERFORM FIN.

Minimap



Le paragraphe MAIN-PROCEDURE agit comme un chef d'orchestre.

Exemple Complet : Mise en Place

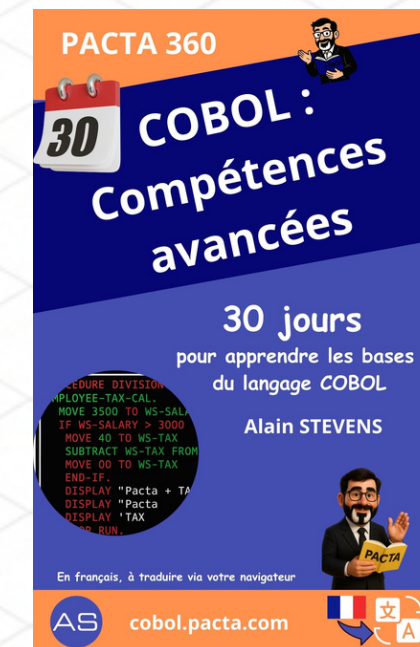
Programme LES-PARAGRAPHES

Avant de manipuler les données, nous définissons les **variables** dans la **WORKING-STORAGE SECTION**.

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. LES-PARAGRAPHES.
```

```
DATA DIVISION.  
WORKING-STORAGE SECTION.  
01  TOTAL          PIC 9(4) VALUE 0.  
01  INDEX          PIC 9(4) VALUE 1.
```

Variables : TOTAL (0)
et INDEX (1)



Exemple Complet : La Procédure Principale

L'entrée du programme. C'est ici que la logique globale est définie.

```
PROCEDURE DIVISION.
```

```
MAIN-PROCEDURE.
```

```
    PERFORM INITIALISATION.
```

```
    PERFORM TRAITEMENT.
```

```
    PERFORM FIN.
```

```
    STOP RUN.
```



Clarté de lecture : On comprend l'intention (Initialiser, Traiter, Finir) sans voir les détails techniques.

Exemple Complet : Les Sous-Routines

INITIALISATION.

* Initialisation des variables

```
MOVE 0 TO TOTAL.
```

```
MOVE 1 TO INDEX.
```

TRAITEMENT.

* Traitement des données

```
ADD 10 TO TOTAL.
```

```
ADD 1 TO INDEX.
```

FIN.

* Fin du programme

```
DISPLAY 'TOTAL = ' TOTAL.
```

```
DISPLAY 'INDEX = ' INDEX.
```



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

The banner features a blue background with a white calendar icon showing '30'. It includes a circular inset with COBOL code snippets. At the bottom, there are logos for 'AS' and 'cobol.pacta.com', along with a small illustration of a man holding a book labeled 'PACTA'.

Chaque paragraphe est autonome et effectue une tâche précise.

Bonnes Pratiques : Clarté et Concision

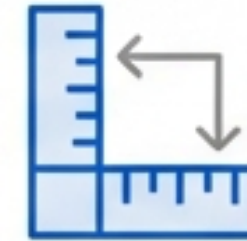


1. Nommage des Paragraphes

Utilisez des noms descriptifs.

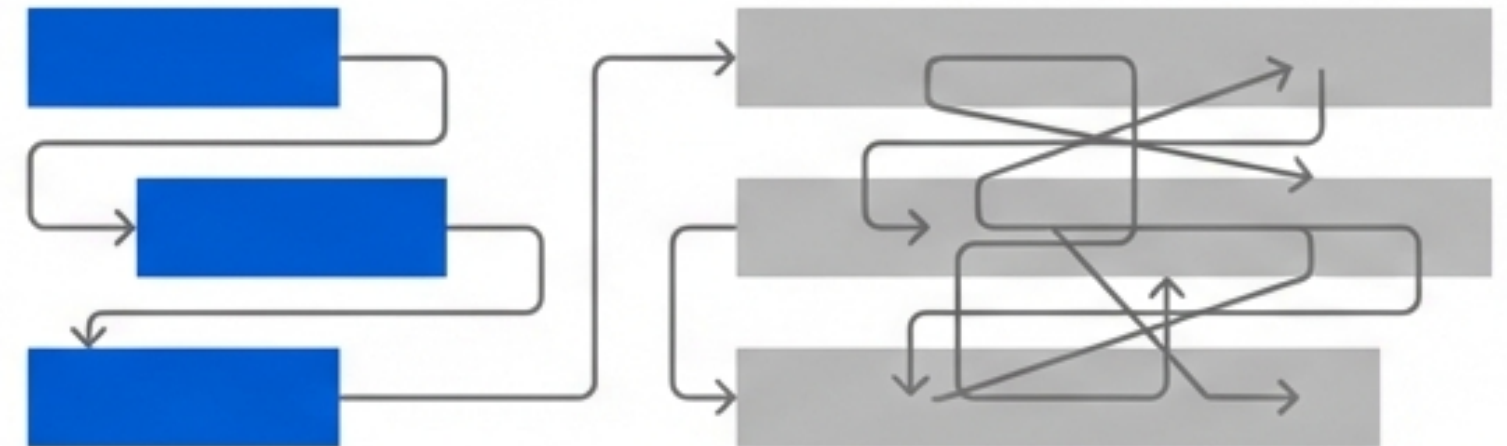
"CALCUL-SALAIRE" (Good)

"PARA-1" (Bad)



2. Utilisation de Phrases Courtes

Préférez des phrases simples. Une logique complexe sur une seule ligne devient illisible.



Court = Lisible

Long = Complexe

Bonnes Pratiques : Documentation

Le code explique 'comment', les commentaires expliquent 'pourquoi'.
Ajoutez des commentaires pour expliquer le rôle des paragraphes.

INITIALISATION.

* Initialisation des variables

MOVE 0 TO TOTAL.

...

L'astérisque en colonne 7
indique un commentaire.



Conclusion : Vers un Code Professionnel

- ✓ **Structure Logique** : Diviser les problèmes complexes en étapes gérables.
 - ✓ **Modularité** : Utiliser PERFORM pour éviter la duplication.
 - ✓ **Maintenance** : Nommer et commenter pour la pérennité du code.
-

"Comprendre l'utilisation et l'organisation des paragraphes et phrases est essentiel pour écrire des programmes clairs, lisibles et efficaces."

