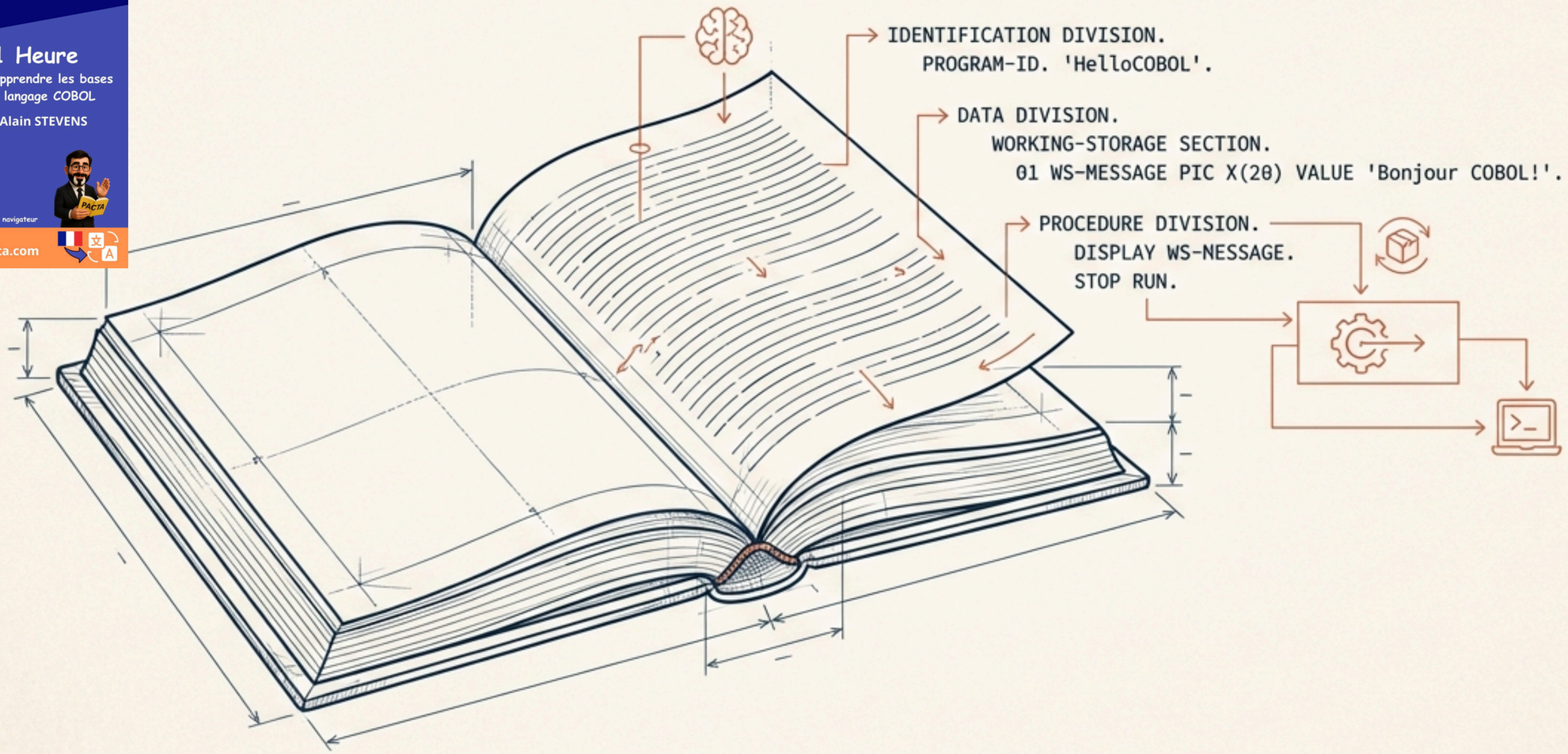
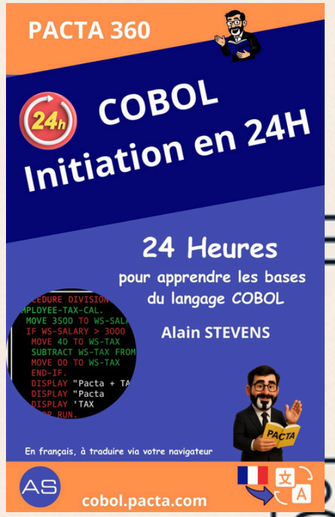
Anatomie d'un Programme COBOL

Un voyage au cœur du code, page par page.



cobol.pacta.com

Un programme COBOL se lit comme un livre.



Le Livre
→ Divisions: Les 4 grandes parties du programme.

Les Chapitres
→ Sections: Les sous-parties d'une division.

Les Paragraphes du texte
→ Paragraphes COBOL: Les blocs logiques de traitement.

Les Phrases du texte
→ Phrases COBOL: Les instructions individuelles.

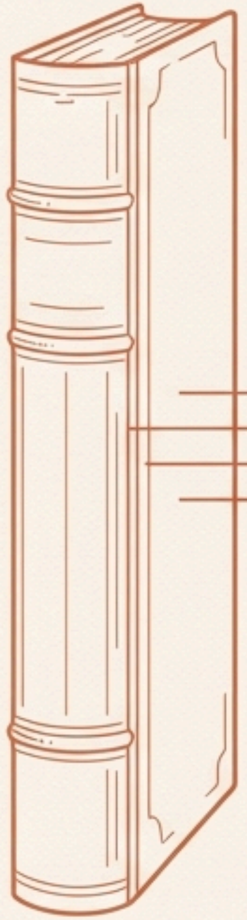
Les Fiches de personnages
→ Déclarations: La description des données.

Cette structure, bien que rigide, rend le code extrêmement lisible et maintenable. Nous allons explorer chaque niveau, du plus général au plus spécifique.

Le Squelette du Programme : Les 4 Divisions

Un programme COBOL est toujours organisé en quatre grandes divisions. Chacune a un rôle unique et précis.

cobol.pacta.com



- **1. IDENTIFICATION DIVISION.**
  Rôle : Donner un nom et une identité au programme. La ‘couverture’ du livre.

- **2. ENVIRONMENT DIVISION.**
  Rôle : Décrire l’environnement d’exécution (fichiers, etc.). Souvent minimale dans les programmes simples.

- **3. DATA DIVISION.**
  Rôle : Déclarer toutes les données : variables, constantes, structures. Les ‘personnages et lieux’.

- **4. PROCEDURE DIVISION.**
  Rôle : Contenir les instructions et la logique. C’est ici que ‘l’action se déroule’.

Chapitre 1 : L'Identité et l'Environnement

IDENTIFICATION DIVISION.

Le Concept

La carte d'identité obligatoire du programme.

L'Analogie

La couverture et la page de titre du livre.

L'Exemple

```
IDENTIFICATION DIVISION.
PROGRAM-ID. HELLO-PRACTICE.
```

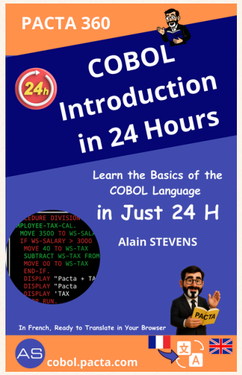
ENVIRONMENT DIVISION.

Le Concept

Le lien entre le programme et le monde extérieur (fichiers, imprimantes).

L'Analogie

Les notes de l'éditeur sur le contexte de publication.



Note : Pour un programme minimaliste, cette division peut être vide ou même omise. Nous nous concentrerons sur les deux divisions suivantes, qui sont le cœur de tout programme.

Chapitre 2 : La Présentation des Personnages (DATA DIVISION)

Le Concept

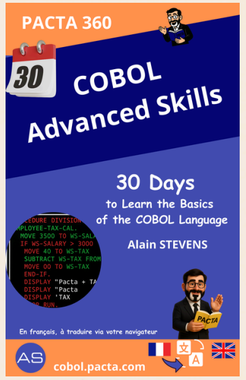
La `DATA DIVISION` est l'inventaire centralisé de toutes les données qu'un programme utilisera. Rien ne peut être utilisé dans la `PROCEDURE DIVISION` sans avoir été déclaré ici en premier.

L'Analogie

Avant de raconter une histoire, on doit définir les personnages, leurs caractéristiques et les objets qu'ils utiliseront. C'est le rôle de la `DATA DIVISION`.

Section Clé

La section la plus importante à connaître pour commencer est la `WORKING-STORAGE SECTION`. C'est là que nous définirons nos variables principales.



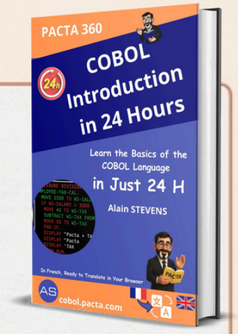
Dans la DATA DIVISION : La `WORKING-STORAGE SECTION`

Le Concept

Une `**SECTION**` est un découpage à l'intérieur d'une `**DIVISION**`. La `**WORKING-STORAGE SECTION**` est la zone mémoire principale où les variables du programme sont conservées pendant toute son exécution.

L'Analogie

Si la `**DATA DIVISION**` est la liste de tous les personnages, la `**WORKING-STORAGE SECTION**` est le répertoire des personnages principaux, ceux qui sont actifs tout au long du récit.

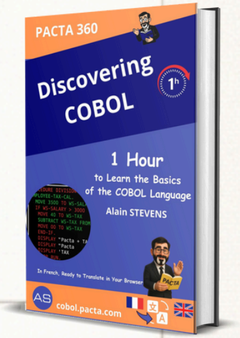



```
DATA DIVISION.
WORKING-STORAGE SECTION.
* C'est ici que nous allons déclarer nos
variables...
```

Créer une Fiche d'Identité : Les Déclarations

Une déclaration est une “fiche d’identité” pour une donnée. Elle définit son nom, son type, sa taille et sa valeur initiale.

```
01 WS-MESSAGE PIC X(20) VALUE "PACTA COBOL".
```



Niveau

Indique la hiérarchie de la donnée. `01` est le plus haut niveau.

Nom

Le nom unique de notre variable.

Type et Taille

`PIC` pour “Picture”. `X` signifie alphanumérique (texte), et `(20)` est la taille (20 caractères).

Valeur Initiale

La valeur par défaut assignée à la variable.

Chapitre 3 : Le Déroulement de l'Action
(PROCEDURE DIVISION)

Le Concept

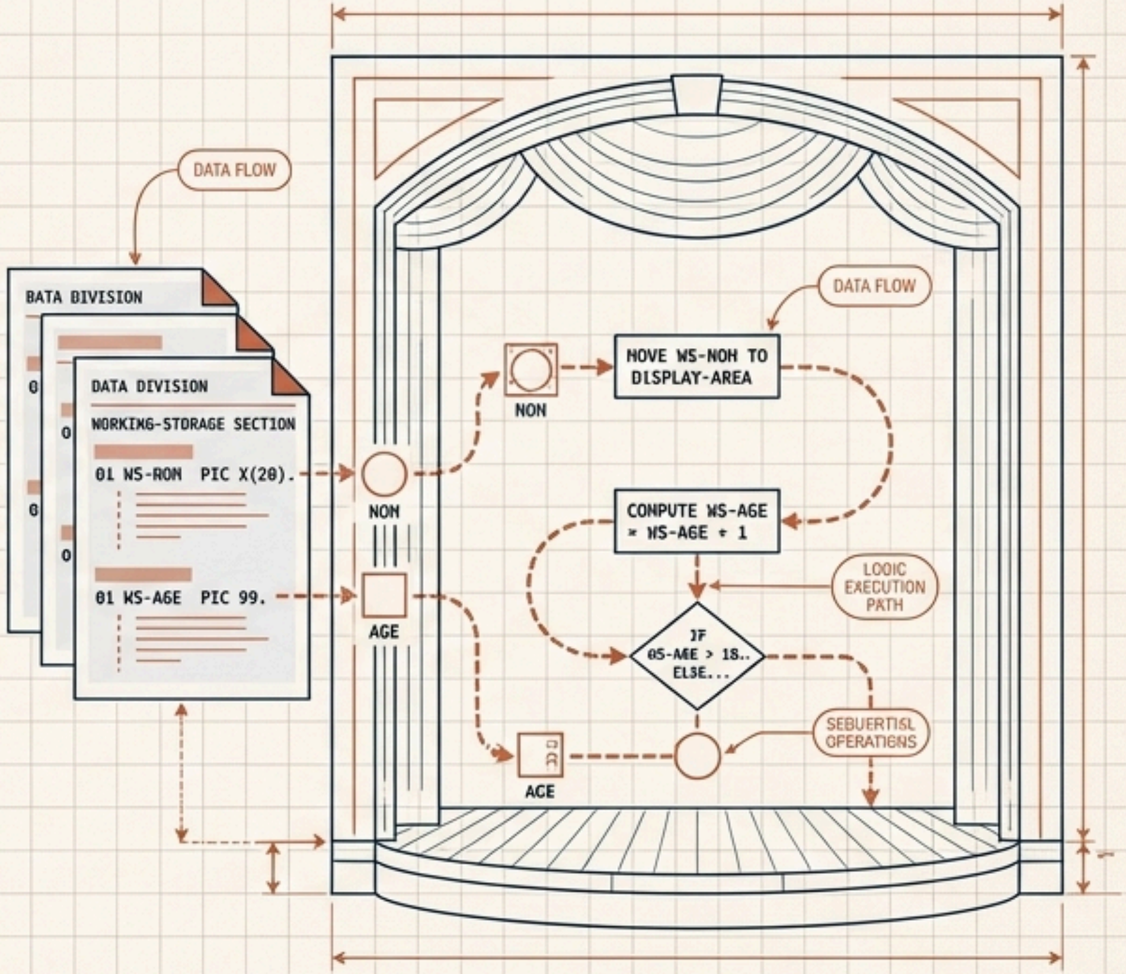
C'est le cœur du programme, là où les instructions sont exécutées séquentiellement. Toute la logique de traitement, les calculs, les affichages et les décisions se trouvent ici.

L'Analogie

Le corps du récit. Après avoir présenté les personnages (`DATA DIVISION`), c'est ici que l'histoire se déroule, que les actions sont entreprises et que l'intrigue avance.

Structure Interne

Cette division est structurée en utilisant des `Paragraphes` et des `Phrases`.



Organiser l'Action : Les Paragraphes

Le Concept

Un paragraphe est un bloc de code logique avec un nom, se terminant par un point. Il permet de découper le programme en étapes claires et de le rendre plus lisible.

L'Analogie

Les sous-parties d'un chapitre, ou les scènes d'un acte. Ils regroupent des actions liées sous une seule étiquette (ex: `INITIALISER-DONNEES.`, `TRAITER-FICHIER.`, `AFFICHER-RESULTATS.`).

L'Exemple

```
PROCEDURE DIVISION.
MAIN-PARA.
    * Les instructions (phrases)
de ce paragraphe viennent ici…
```

Ici, `MAIN-PARA.` est le nom du paragraphe. C'est un point d'entrée logique pour notre traitement.

Les Briques Élémentaires : Les Phrases COBOL

Le Concept

Une phrase COBOL est une instruction ou un ensemble d'instructions qui commence par un mot-clé (verbe d'action) et se termine **impérativement par un point (`.`)**.

L'Analogie

La phrase dans un texte. Le point final est non-négociable ; il signale la fin d'une commande complète.

Règle d'Or du Débutant

Pour commencer, prenez l'habitude de mettre un point à la fin de chaque instruction. C'est la manière la plus simple d'éviter les erreurs.

Exemples d'instructions

```
DISPLAY WS-MESSAGE.

MOVE "Hello" TO WS-GREETING.

STOP RUN.
```

cobol.pacta.com

La Synthèse : Notre Premier Programme COBOL

DIVISION

SECTION

DÉCLARATION

PARAGRAPHE

PHRASE

```
IDENTIFICATION DIVISION.          *> Division 1
PROGRAM-ID. HELLO-PRACTICE.

DATA DIVISION.                    *> Division 3
WORKING-STORAGE SECTION.          *> Section
    01 WS-MESSAGE PIC X(20)       *> Déclaration
       VALUE "PACTA COBOL".

PROCEDURE DIVISION.               *> Division 4
MAIN-PARA.                        *> Paragraphe
    DISPLAY WS-MESSAGE.           *> Phrase
    STOP RUN.                     *> Phrase
```

cobol.pacta.com

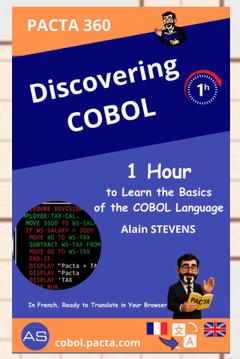
Lecture Ligne par Ligne (1/2) : La Préparation

```
IDENTIFICATION DIVISION.
PROGRAM-ID. HELLO-PRACTICE.
```

```
IDENTIFICATION DIVISION.
PROGRAM-ID. HELLO-PRACTICE.
```

La `***DIVISION***' n°1, 'Qui suis-je ?'.
On y déclare le nom officiel du programme.

```
MAIN-PARA.
DISPLAY  WS-MESSAGE. VALUE "PACTA COBOL".
STOP RUN.
```




```
IDENTIFICATION DIVISION.
PROGRAM-ID. HELLO-PRACTICE.
```

```
DATA DIVISION.
WORKING-STORAGE SECTION.
  01 WS-MESSAGE PIC X(20) VALUE "PACTA COBOL".
```

La `***DIVISION***' n°3 où l'on déclare les données.
On y ouvre la `***SECTION***' pour nos variables.
On y effectue la `***DÉCLARATION***' complète de notre variable `***WS-MESSAGE***' : un texte de 20 caractères avec une valeur par défaut.

```
MAIN-PARA.
 DISPLAY WS-MESSAGE.
 STOP RUN.
```

Lecture Ligne par Ligne (2/2) : L'Action

```
PROCEDURE DIVISION.
    MAIN-PARA.
```

DIVISION

PARAGRAPHE

La ***DIVISION*** n°4, où le code s'exécute. On définit le ***PARAGRAPHE*** `***MAIN-PARA.***' comme notre bloc de traitement principal.

```
DISPLAY WS-MESSAGE.
```

PHRASE

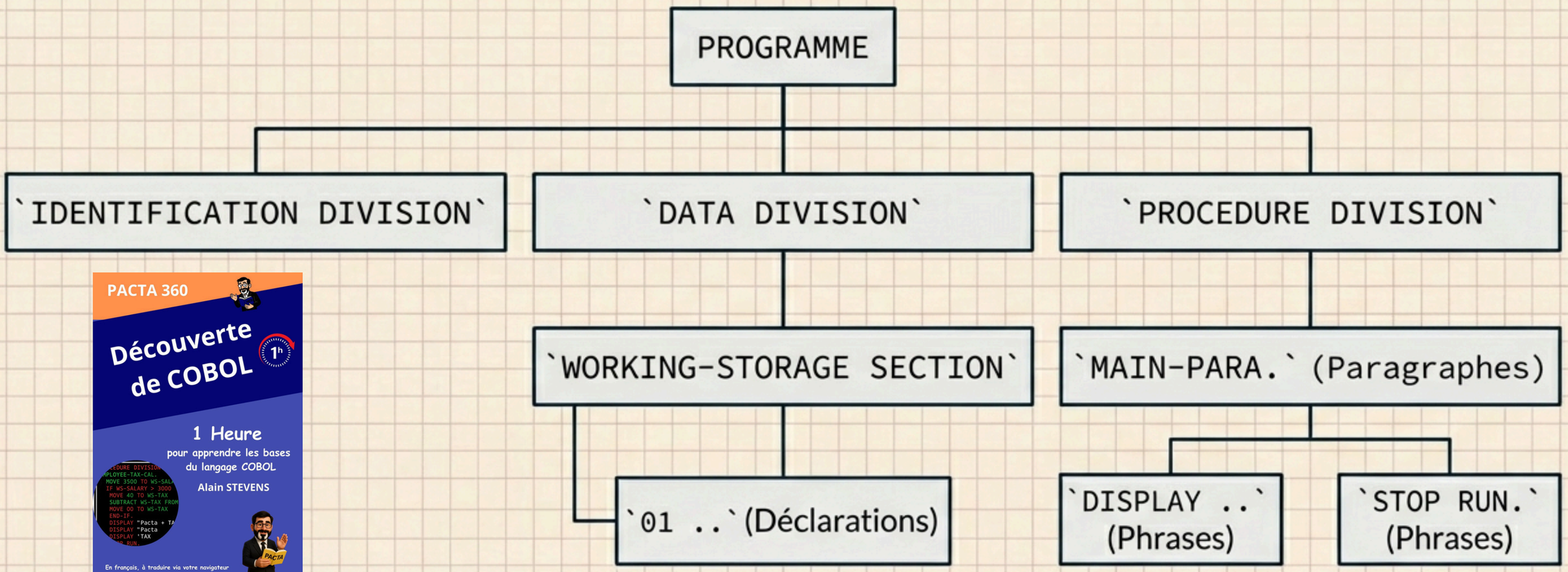
Une ***PHRASE*** COBOL. L'instruction `***DISPLAY*** affiche le contenu de la variable ***WS-MESSAGE*** à l'écran.

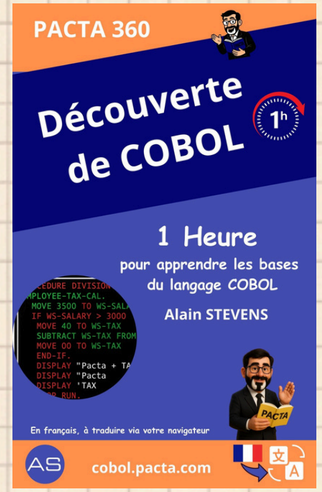
```
STOP RUN.
```

PHRASE

La dernière ***PHRASE*** COBOL. L'instruction `***STOP RUN***' marque la fin définitive de l'exécution du programme.

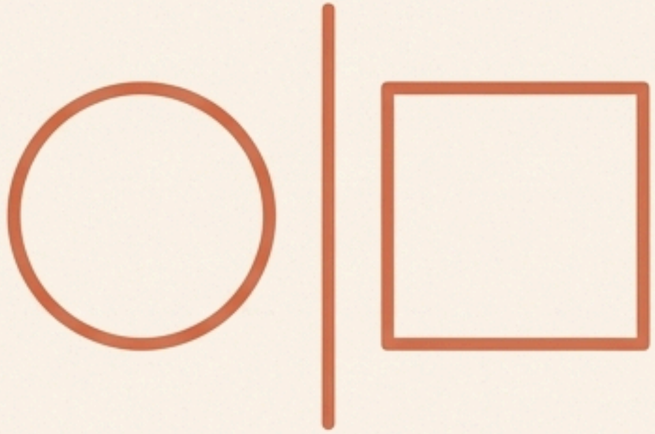
La Hiérarchie en un Coup d'Œil





Chaque élément s'emboîte dans le précédent, créant une structure prévisible et ordonnée, de la division **la plus large** à la **phrase la plus spécifique**.

Les Principes Fondamentaux à Retenir



1. La Structure est Reine.

La hiérarchie `Division > Section > Paragraphe > Phrase` n'est pas une suggestion, c'est la loi. La respecter garantit un code lisible.

2. Les Données sont Séparées de l'Action.

On déclare TOUT dans la `DATA DIVISION` avant de l'utiliser dans la `PROCEDURE DIVISION`. Cette séparation stricte est une force du langage.

3. Le Point est la Loi.

Chaque phrase se termine par un **point**. Oublier un point est l'une des erreurs les plus courantes et les plus simples à éviter. C'est la ponctuation qui donne son rythme au programme.