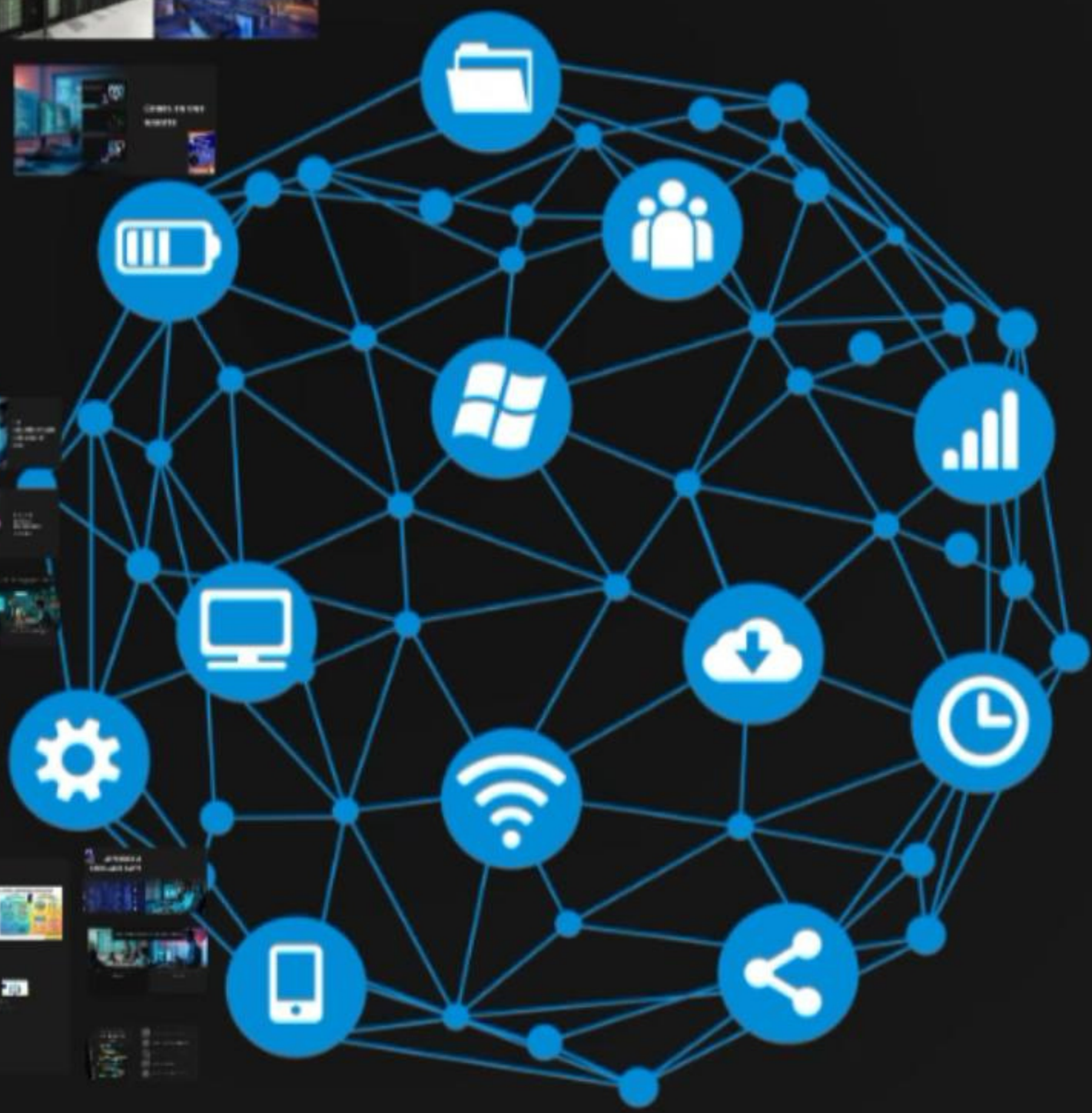




APPRENEZ LE COBOL AVEC PACTA

Le COBOL, malgré son âge, reste un acteur clé dans le monde des systèmes critiques. Sa standardisation et sa capacité à évoluer tout en étant intégré dans les infrastructures modernes démontrent que son remplacement total est peu probable, soulignant ainsi sa longévité et son importance dans les secteurs tels que la finance, la santé et les administrations publiques.





PACTA 360



Découverte de COBOL

1h

1 Heure

pour apprendre les bases du langage COBOL

Alain STEVENS

```
OPEN DEVISION  
EMPLOYEE-TAX-CAL  
MOVE 3500 TO WS-SAL  
IF WS-SALARY > 3000  
MOVE 40 TO WS-TAX  
SUBTRACT WS-TAX FROM  
MOVE WS-TAX TO WS-TAX  
END-IF  
DISPLAY "PACTA = TA  
DISPLAY "PACTA  
DISPLAY "TAX  
END-IF
```



En français, à traduire via votre navigateur

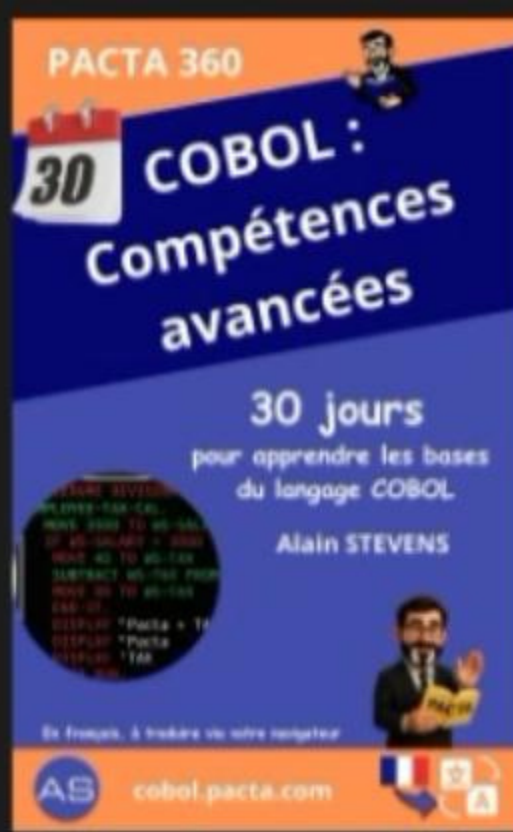


cobol.pacta.com



L'IMPORTANCE CONTINUE DE COBOL DANS LES SYSTEMES CRITIQUES

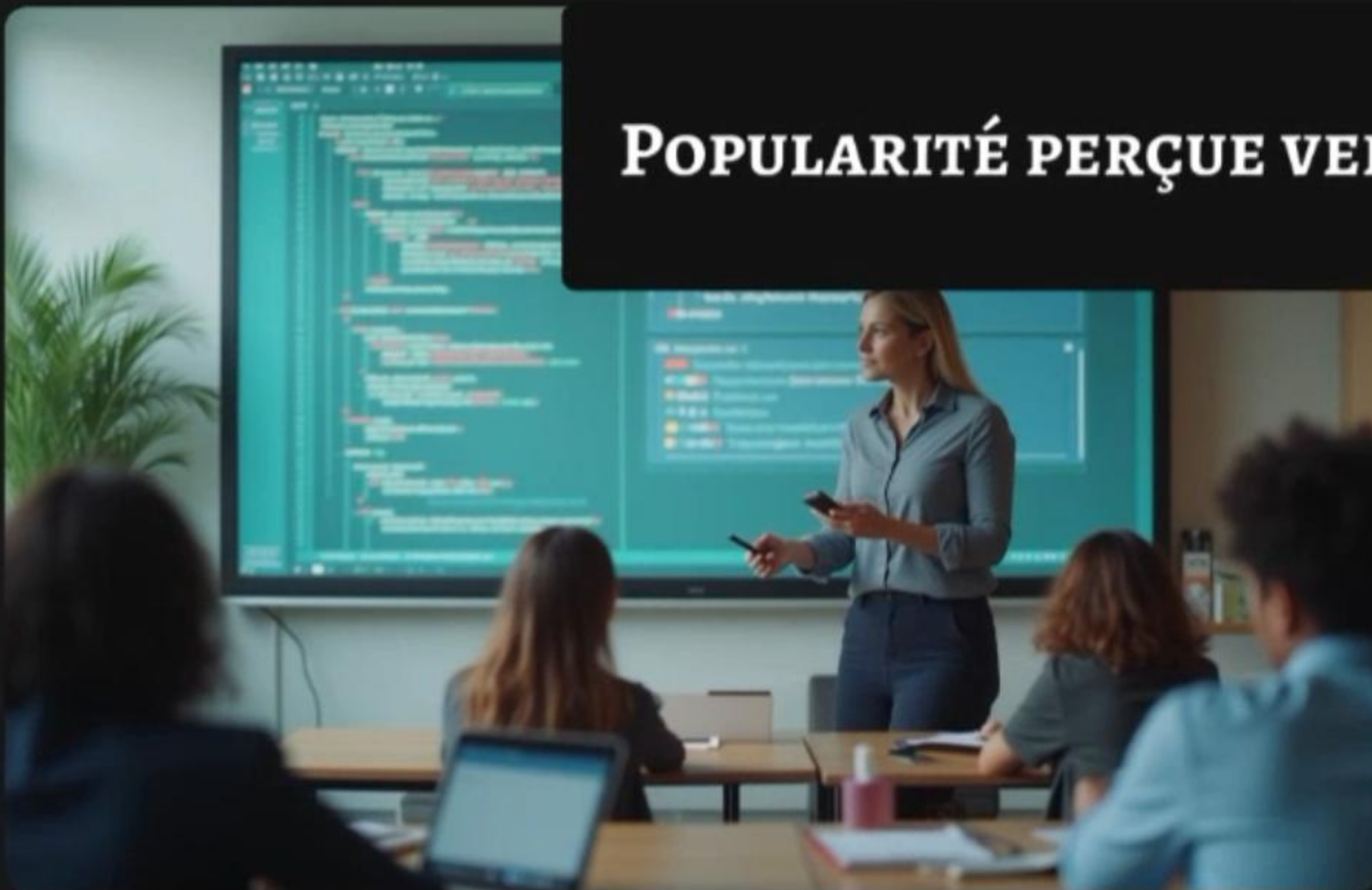
Exploration du rôle essentiel de COBOL dans les infrastructures critiques et pourquoi demeure un choix incontournable au-delà des préjugés de modernité



APPRENEZ LE

COBOL AVEC PACTA

POPULARITÉ PERÇUE VERSUS IMPORTANCE RÉELLE



INDICE DE POPULARITÉ BASÉ SUR DES TUTORIELS

L'indice de popularité des langages de programmation se base souvent sur le volume de recherches de tutoriels, ce qui reflète leur visibilité et leur enseignement. Cela peut donner une image faussée de leur utilisation réelle dans l'industrie.



IMPORTANCE CRITIQUE MALGRÉ LA FAIBLE VISIBILITÉ

En revanche, des langages comme Cobol, bien que moins en vogue dans les classements de popularité, jouent un rôle crucial dans les systèmes critiques. Leur importance ne se mesure pas par leur popularité, mais par leur intégration dans des applications transactionnelles majeures.

LE FUTUR SELON LA VIDÉO : INERTIE, COÛT, RISQUE

```
31
32
33
34
35 self.debug = debug
36 self.logger = logging.getLogger(__name__)
37 if path:
38     self.file = open(os.path.join(path, "fingerprint.log"), "a")
39     self.file.seek(0)
40     self.fingerprints.update(self.fingerprints)
41
42 @classmethod
43 def from_settings(cls, settings):
44     debug = settings.getbool("DEBUG")
45     return cls(job_dir(settings), debug)
46
47 def request_seen(self, request):
48     fp = self.request_fingerprint(request)
49     if fp in self.fingerprints:
50         return True
51     self.fingerprints.add(fp)
52     if self.file:
53         self.file.write(fp + os.linesep)
54
55 def request_fingerprint(self, request):
56     return request_fingerprint(request)
```



Discours de mort annoncée



Inertie du patrimoine applicatif



Coûts de remplacement élevés



Risque de régression



Croissance à rythme constant

REPÈRES HISTORIQUES : NAISSANCE, STANDARDISATION, MODERNISATION

Un aperçu du développement et de l'évolution du langage COBOL depuis sa création jusqu'à sa standardisation et modernisation continue.



1950



Apparition d'un besoin croissant pour un langage orienté vers la gestion, capable de décrire des processus administratifs complexes.

1959



Réunion historique au Pentagone aboutissant à la finalisation des premières spécifications de COBOL, marquant le début de son développement formel.

1960



Les années 1960 voient la structuration et la diffusion de COBOL, avec des ajustements continus et l'adoption croissante au sein des organisations.

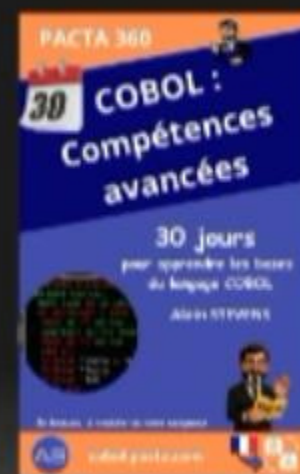
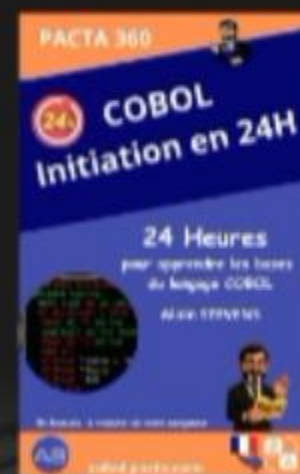
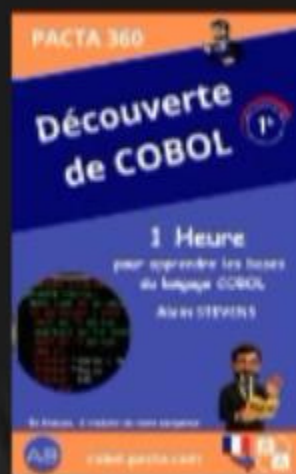
1968



Standardisation avec la création de la norme ANSI, garantissant la stabilité et l'uniformité du langage COBOL sur le long terme.

PACTA COBOL

Le langage COBOL, né dans les années 1950, demeure un fondement des systèmes critiques modernes, malgré sa perception comme un artefact du passé. Alors que beaucoup le considèrent comme ancien, il fait partie intégrante d'une infrastructure technologique omniprésente, essentielle au fonctionnement de services contemporains. Cette présentation vise à explorer les affirmations sur la longévité de COBOL, ses implications techniques, et pourquoi une disparition totale de ce langage est peu probable à l'avenir.



COBOL EN UNE MINUTE

DÉVELOPPEMENT ORIENTÉ MÉTIER

Le langage COBOL a été conçu pour faciliter le développement orienté métier, permettant ainsi une description claire et précise des traitements de gestion, ce qui en fait un outil incontournable pour les entreprises.



SIGNIFICATION DE L'ACRONYME

L'acronyme COBOL signifie 'Common Business Oriented Language', soulignant ainsi son objectif principal de servir les besoins des métiers.



POSITIONNEMENT HAUT NIVEAU

COBOL se positionne comme un langage haut niveau, favorisant l'écriture de règles de traitement qui sont facilement lisibles et compréhensibles, contrairement à d'autres langages plus orientés vers les détails machine.



PACTA 360

30 COBOL : Compétences avancées

30 jours pour apprendre les bases du langage COBOL

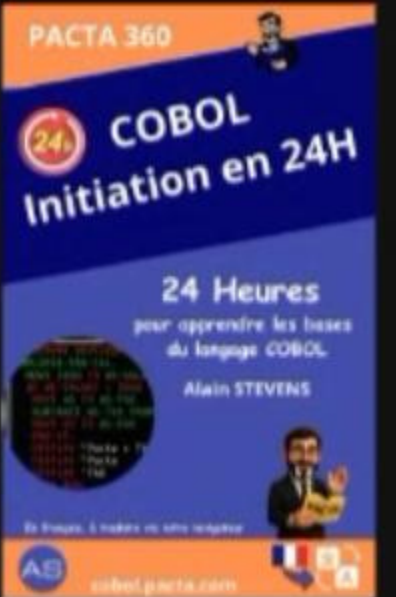
Alain STEVENS

AS cobol.pacta.com

En français, à l'usage de votre navigateur

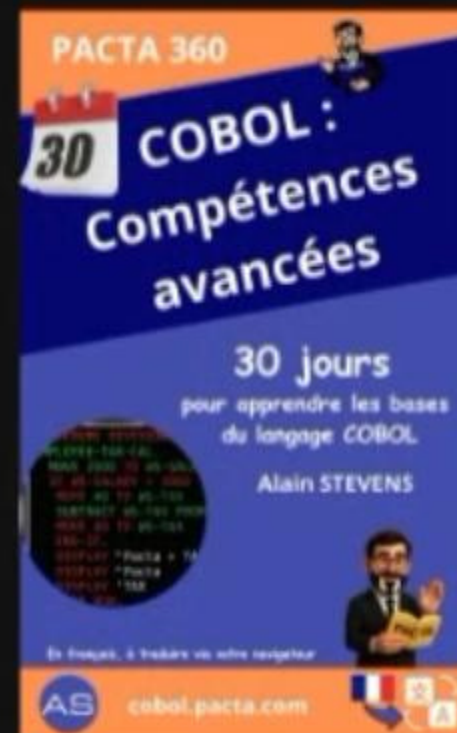
SIGNIFICATION DE L'ACRONYME

L'acronyme COBOL signifie
'Common Business Oriented
Language', soulignant ainsi son
objectif principal de servir les
besoins des métiers.



DÉVELOPPEMENT ORIENTÉ MÉTIER

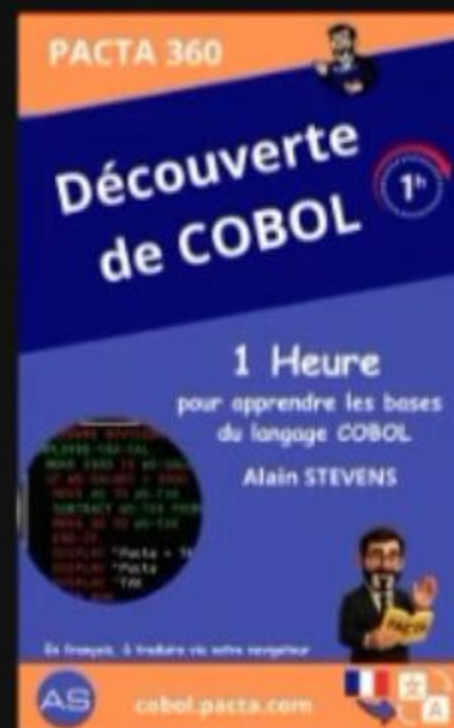
Le langage COBOL a été conçu pour faciliter le développement orienté métier, permettant ainsi une description claire et précise des traitements de gestion, ce qui en fait un allié incontournable pour les entreprises.



POSITIONNEMENT

HAUT NIVEAU

COBOL se positionne comme un langage haut niveau, favorisant l'écriture de règles de traitement qui sont facilement lisibles et compréhensibles, contrairement à d'autres langages plus orientés vers les détails machine.



UN LANGAGE STANDARDISÉ ET ROBUSTE



Le langage Cobol est conçu pour être standardisé, ce qui lui confère une stabilité et une longévité exceptionnelles. Sa portabilité permet aux programmes d'être exécutés sur différentes plateformes sans nécessiter de réécriture majeure, favorisant ainsi leur adoption sur le long terme.

UNE SYNTAXE PROCHE DE L'ANGLAIS ET UNE LISIBILITÉ REVENDIQUÉE



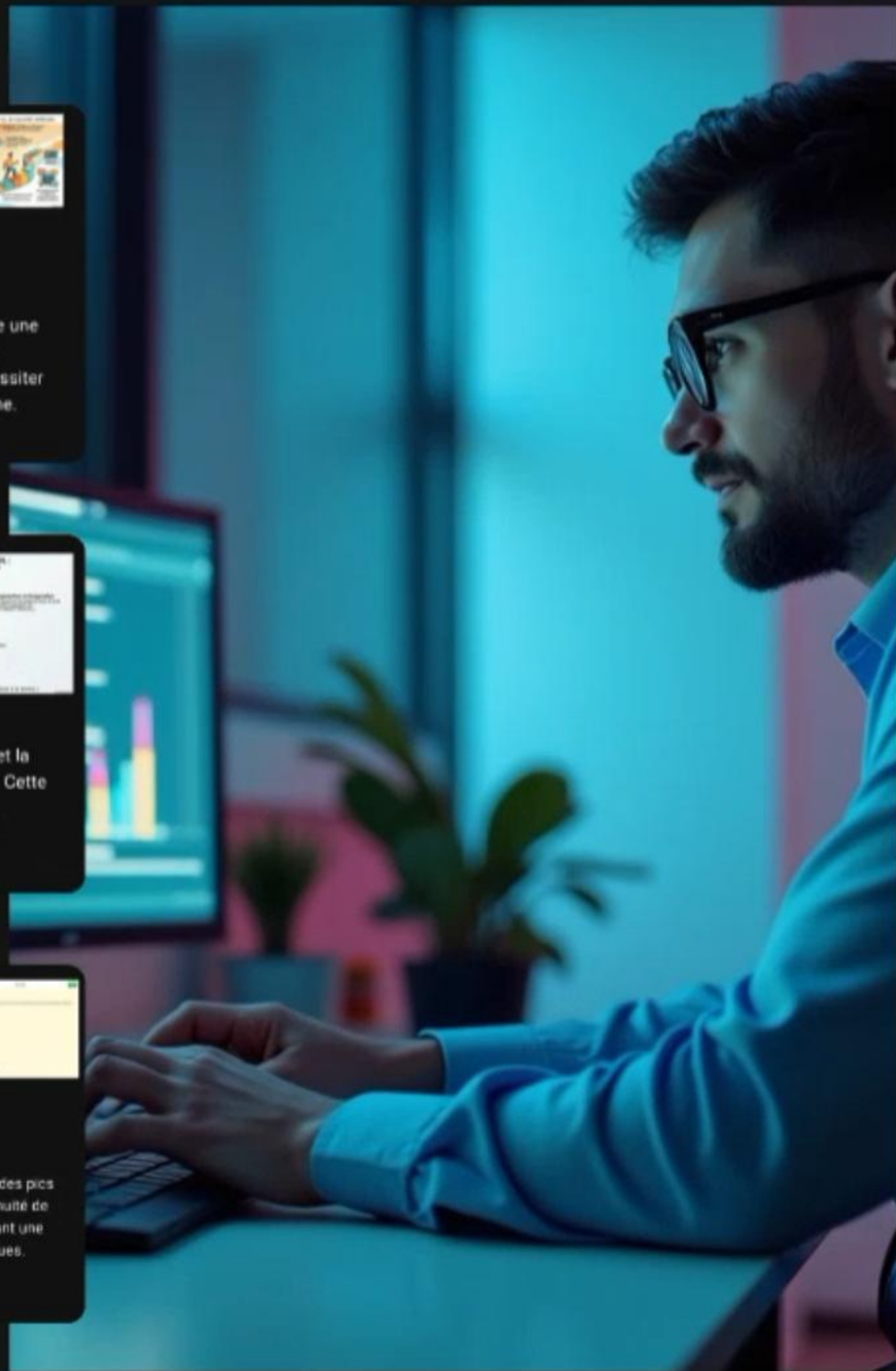
La syntaxe de Cobol, proche de l'anglais, facilite la compréhension et la lisibilité du code, ce qui est essentiel pour les équipes nombreuses. Cette lisibilité permet un audit efficace et une maintenance simplifiée, essentielle dans les systèmes critiques où la clarté est primordiale.

SCALABILITÉ, FIABILITÉ ET INDÉPENDANCE DE PLATEFORME



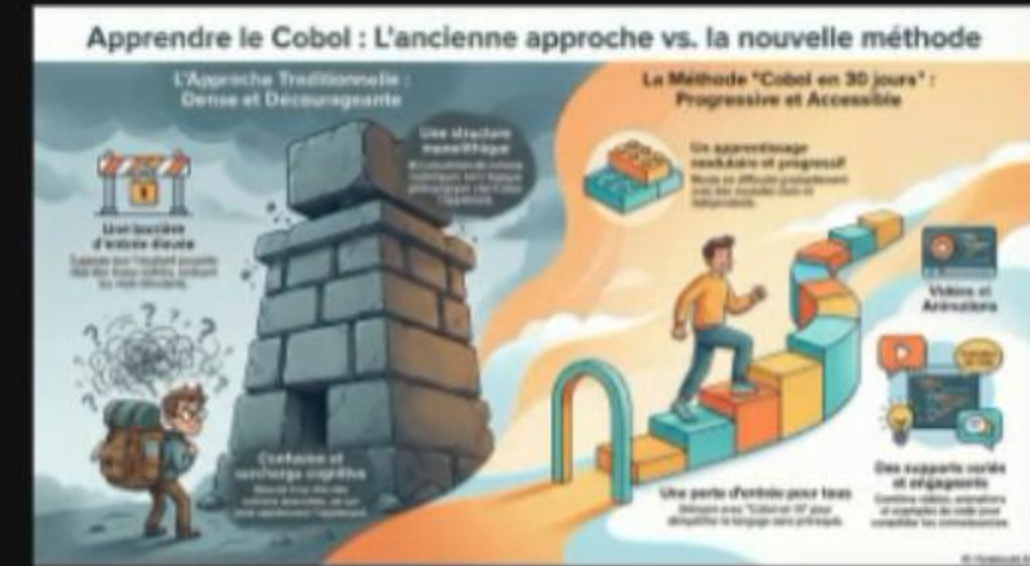
Cobol offre une scalabilité impressionnante, permettant aux systèmes de gérer des pics de charge importants sans perte de performance. Sa fiabilité garantit une continuité de service, ce qui est essentiel pour les applications transactionnelles, tout en offrant une facilité des migrations et les évolutions techniques.

LES CARACTÉRISTIQUES CLÉS MISES EN AVANT



UN LANGUAGE STANDARDISÉ ET ROBUSTE

Le langage Cobol est conçu pour être standardisé, ce qui lui confère une stabilité et une longévité exceptionnelles. Sa portabilité permet aux programmes d'être exécutés sur différentes plateformes sans nécessiter de réécriture majeure, favorisant ainsi leur adoption sur le long terme.



UNE SYNTAXE PROCHE DE L'ANGLAIS ET UNE LISIBILITÉ REVENDIQUÉE

La syntaxe de Cobol, proche de l'anglais, facilite la compréhension et la lisibilité du code, ce qui est essentiel pour les équipes nombreuses. Cette lisibilité permet un audit efficace et une maintenance simplifiée, essentielle dans les systèmes critiques où la clarté est primordiale.

La PROCEDURE DIVISION en COBOL : Le Moteur de Votre Programme

Le Cœur du Programme
Contient toutes les instructions, les calculs
et la logique d'exécution du programme.

Le Point d'Entrée de l'Exécution
Sauf exception, l'exécution commence
toujours au premier paragraphe de
cette division.



Organisation en Paragraphes
La logique est structurée en blocs de code
appelés paragraphes (ex :
TRAITEMENT-PRINCIPAL).

Terminer l'Exécution

Deux instructions principales permettent de clore le programme.

Commande	Usage Principal
STOP RUN	Termine un programme principal et rend la main au système.
GOBACK	Termine un sous-programme et retourne au programme appelant.

🗣️ L'essentiel à retenir

« C'est le "main" du programme COBOL : là où le traitement commence, s'exécute et se termine. »

40 / November 2024

SCALABILITÉ, FIABILITÉ ET INDÉPENDANCE DE PLATEFORME

Cobol offre une scalabilité impressionnante, permettant aux systèmes de gérer des pics de charge importants sans perte de performance. Sa fiabilité garantit une continuité de service, ce qui est essentiel pour les applications transactionnelles, tout en offrant une indépendance de plateforme qui facilite les migrations et les évolutions techniques.



USAGE INDIRECT MASSIF

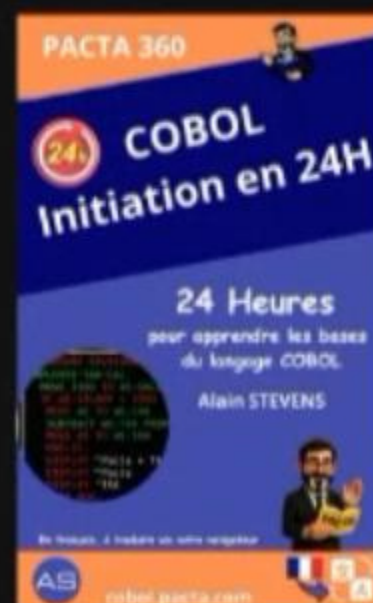
L'usage indirect de COROL est omniprésent dans notre quotidien, notamment à travers des processus comme les voyages, les retraites d'argent, les rendez-vous médicaux et les commandes en ligne. Chaque interaction quotidienne repose souvent sur des systèmes gérés par ce langage.



**SITUATION
ACTUELLE :
OMNIPRÉSENCE
INVISIBLE**

USAGE INDIRECT MASSIF

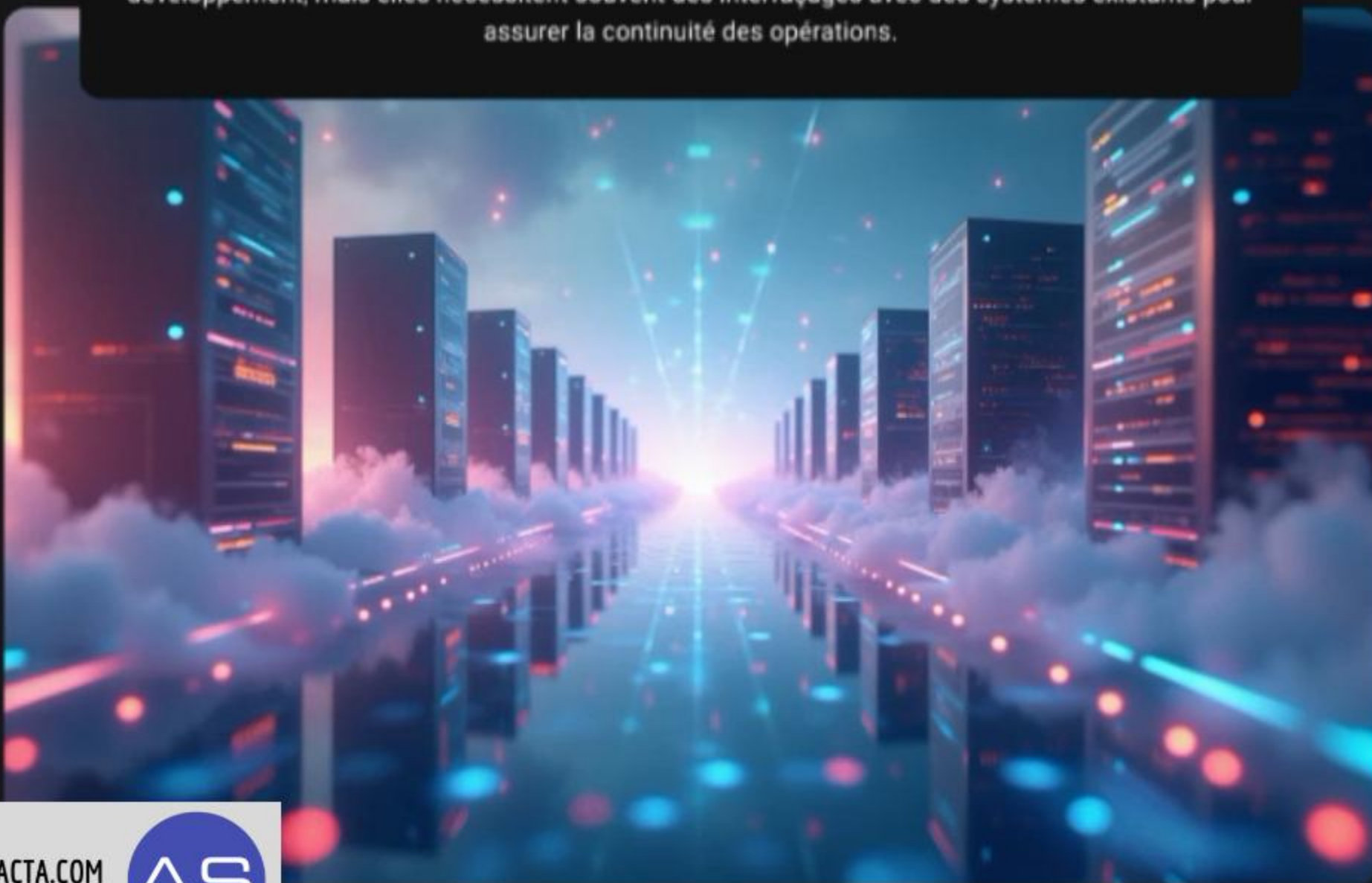
L'usage indirect de COBOL est omniprésent dans notre quotidien, notamment à travers des processus comme les voyages, les retraits d'argent, les rendez-vous médicaux et les commandes en ligne. Chaque interaction quotidienne repose souvent sur des systèmes gérés par ce langage.



UN "ACTIF CACHÉ" : DOMINATION DANS L'INFORMATIQUE D'ENTREPRISE

TECHNOLOGIES RÉCENTES

La multiplication des technologies récentes, telles que les bibliothèques modernes et les plateformes innovantes, transforme le paysage technologique, mais n'élimine pas l'utilisation des langages éprouvés comme Cobol. Ces nouvelles solutions offrent flexibilité et rapidité de développement, mais elles nécessitent souvent des interfaçages avec des systèmes existants pour assurer la continuité des opérations.



APPLICATIONS CRITIQUES

En revanche, les applications critiques qui gèrent des composantes essentielles comme l'argent, les droits ou des services vitaux reposent souvent sur des infrastructures solides et éprouvées, où Cobol joue un rôle clé. Ces systèmes hérités, bien qu'anciens, sont essentiels pour la sécurité et la fiabilité des opérations, illustrant ainsi la notion d'un 'actif caché' dans l'informatique d'entreprise.



BANQUE ET SERVICES FINANCIERS : LE ROYAUME DES TRANSACTIONS

Le secteur bancaire est au cœur des transactions économiques. Les systèmes COBOL gèrent des opérations critiques telles que les paiements, les retraits et la gestion de comptes avec une fiabilité inégalée, garantissant ainsi la sécurité et la stabilité des flux financiers.



TRANSPORT AÉRIEN : RÉSERVATION MODERNE, INFRASTRUCTURE HÉRITÉE

Dans le transport aérien, les systèmes de réservation modernes reposent souvent sur des infrastructures héritées. COBOL continue de jouer un rôle clé dans la gestion des réservations, assurant une interopérabilité fluide entre les différents systèmes de traitement.



SANTÉ : DOSSIERS ET FLUX ASSURANTIELS

Dans le secteur de la santé, les données des patients et les traitements d'assurance reposent sur des systèmes fiables et sécurisés. COBOL assure la continuité des soins en facilitant l'accès aux informations critiques et en simplifiant les flux de traitement des données.

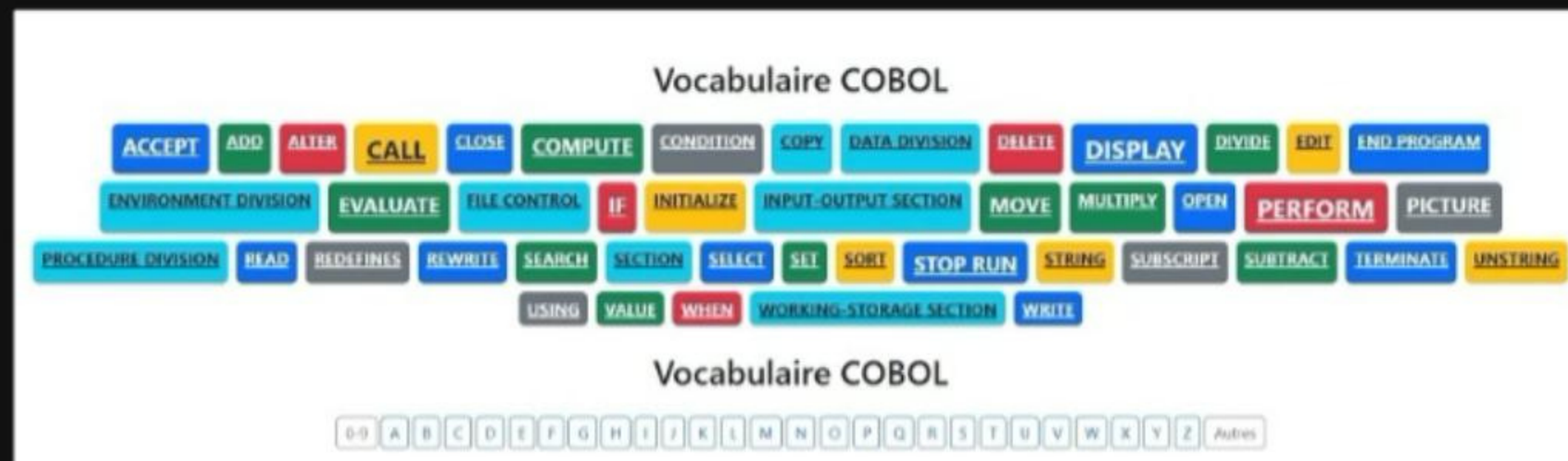


ADMINISTRATIONS : SYSTÈMES HISTORIQUES ET MISSIONS PUBLIQUES

Les administrations s'appuient sur des systèmes historiques pour gérer les missions publiques. Les applications COBOL traitent des données sensibles telles que les impôts, les demandes de permis et d'autres services essentiels, garantissant ainsi la stabilité et la continuité des services publics.



SECTEURS CITÉS : OÙ CE SOCLE EST UTILISÉ



TRANSPORT AÉRIEN : RÉSERVATION MODERNE, INFRASTRUCTURE HÉRITÉE

Dans le transport aérien, les systèmes de réservation modernes reposent souvent sur des infrastructures héritées. COBOL continue de jouer un rôle clé dans la gestion des réservations, assurant une interconnexion fluide entre les différents systèmes de traitement.



BANQUE ET SERVICES FINANCIERS : LE ROYAUME DES TRANSACTIONS

Le secteur bancaire est au cœur des transactions économiques. Les systèmes COBOL gèrent des opérations critiques telles que les paiements, les retraits et la gestion de comptes avec une fiabilité inégalée, garantissant ainsi la sécurité et la traçabilité des flux financiers.

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 !

COBOL Bienvenue dans la formation COBOL

Vous êtes ici pour apprendre les bases de COBOL de manière progressive, avec des exemples concrets et une montée en 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 Suivez les étapes dans l'ordre et validez les exercices.	Pratique Chaque section se termine par une mise en application.	Support Récupérez vos questions et revenez sur les points clés.
--	---	---

En continuant, vous acceptez les conditions d'utilisation du site et les règles de confidentialité liées à la plateforme. Le contenu est ajouté de façon progressive tout au long de votre parcours.

[Haut](#)



SANTÉ : DOSSIERS ET FLUX ASSURANTIELS

Dans le secteur de la santé, les dossiers patients et les traitements d'assurance reposent sur des systèmes fiables et sécurisés. COBOL assure la continuité des soins en facilitant l'accès aux informations critiques et en simplifiant les flux de traitement des données.

cobol.pacta.com

Mémoire centrale

VILLE → Reims

DEPARTEMENT → Marne

PROFESSION → Consultant

AGE 60



ADMINISTRATIONS : SYSTÈMES HISTORIQUES ET MISSIONS PUBLIQUES

Les administrations s'appuient sur des systèmes historiques pour gérer les missions publiques. Les applications COBOL traitent des données sensibles telles que les impôts, les demandes de permis et d'autres services essentiels, garantissant ainsi la stabilité et la continuité des services publics.

```
1  *****  
2  * Author:  
3  * Date:  
4  * Purpose:  
5  * Technics: cobc  
6  *****  
7  IDENTIFICATION DIVISION.  
8  PROGRAM-ID. ExempleSimple.  
9  DATA DIVISION.  
10 FILE SECTION.  
11 WORKING-STORAGE SECTION.  
12 PROCEDURE DIVISION.  
13 MAIN-PROCEDURE.  
14     DISPLAY "Programme COBOL simple."  
15     STOP RUN.  
16 END PROGRAM ExempleSimple.  
17
```

Navigation

- IDENTIFICATION DIVISION
- DATA DIVISION
 - FILE SECTION
 - WORKING-STORAGE SECTION
- PROCEDURE DIVISION
 - MAIN-PROCEDURE

Log

Compiler Issues Output

D:\cobol\code-pour-cours\bin\07-5-exemplesimple.exe
Programme COBOL simple.
Process finished with exit code 0

REPÈRES HISTORIQUES : NAISSANCE, STANDARDISATION, MODERNISATION

Un aperçu du développement et de l'évolution du langage COBOL depuis sa création jusqu'à sa standardisation et modernisation continue.



1950



Apparition d'un besoin croissant pour un langage orienté vers la gestion, capable de décrire des processus administratifs complexes.

1959



Réunion historique au Pentagone aboutissant à la finalisation des premières spécifications de COBOL, marquant le début de son développement formel.

1960



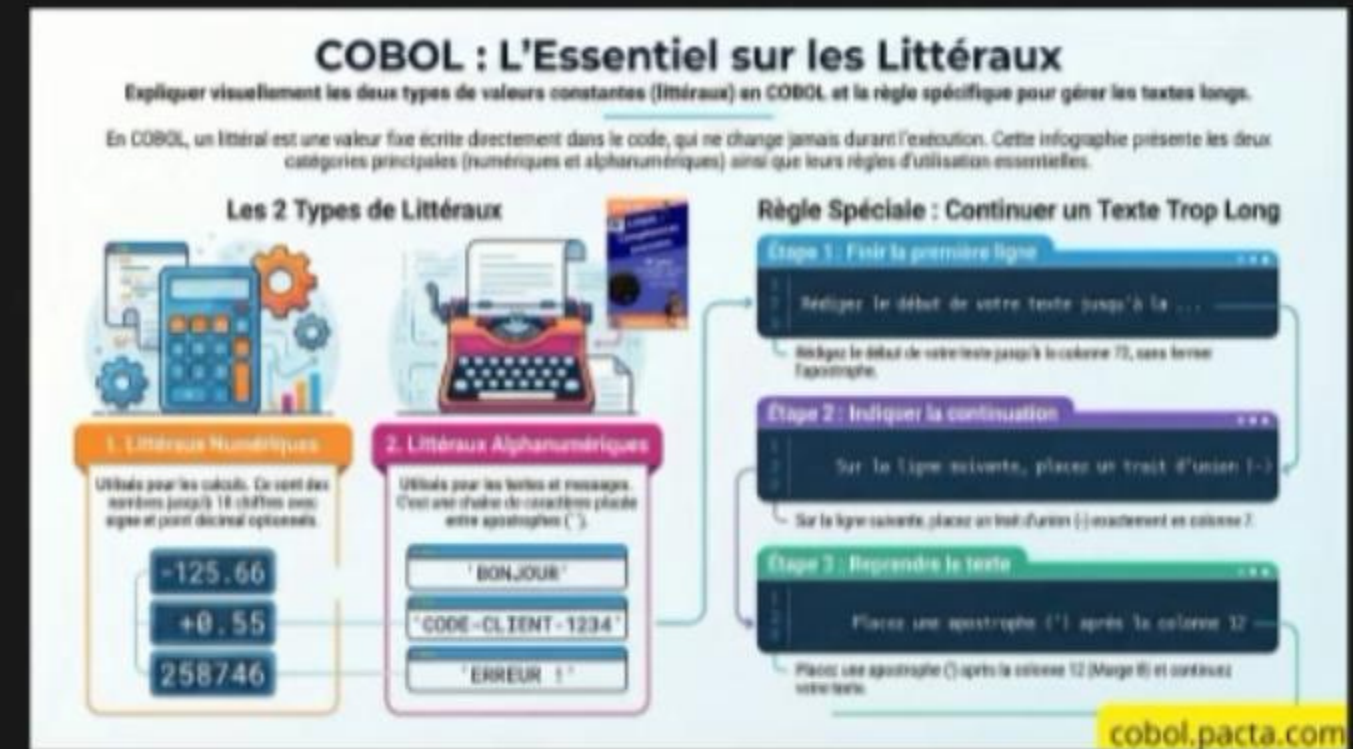
Les années 1960 voient la structuration et la diffusion de COBOL, avec des ajustements continus et l'adoption croissante au sein des organisations.

1968



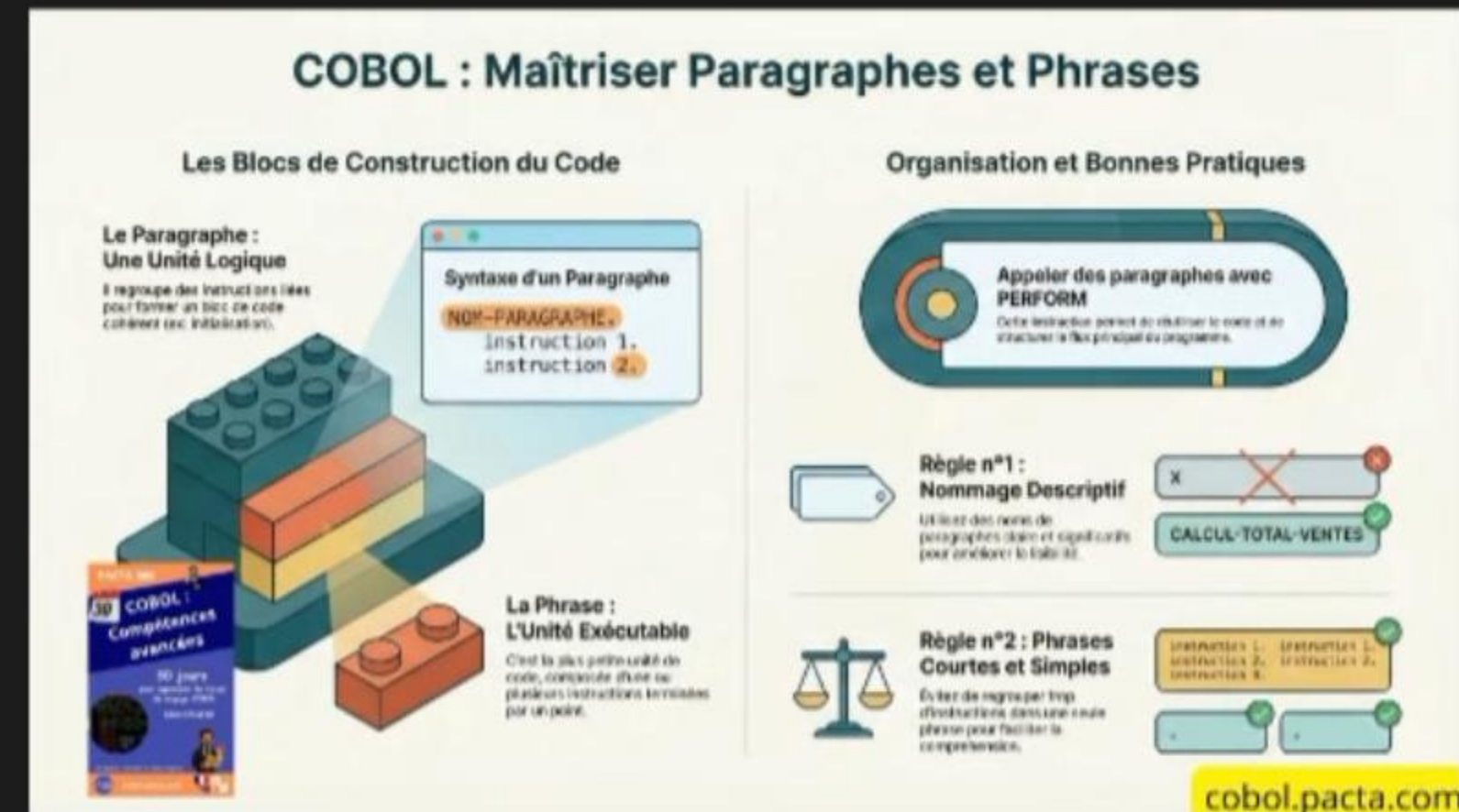
Standardisation avec la création de la norme ANSI, garantissant la stabilité et l'uniformité du langage COBOL sur le long terme.

1950



Apparition d'un besoin croissant pour un langage orienté vers la gestion, capable de décrire des processus administratifs complexes.

1959



Réunion historique au Pentagone aboutissant à la finalisation des premières spécifications de COBOL, marquant le début de son développement formel.

1960

Réussir votre Formation COBOL en 30 Jours

Conseils clés et bonnes pratiques pour organiser votre apprentissage et maximiser votre succès dans une formation COBOL en ligne intensive.

Votre Stratégie d'Apprentissage



La régularité est la clé

Consacrez 45 à 60 minutes par jour plutôt que de longues sessions le week-end.



Suivez le parcours étape par étape

Ne sautez pas de modules, car la structure du COBOL est très séquentielle.



Multipliez les modes d'apprentissage

Utilisez les vidéos, les textes et les quiz pour renforcer votre mémorisation.



Accès, Support & Règles

Protégez votre accès principal

Conservez précieusement le lien vers votre espace privé reçu par e-mail.



Le contenu est confidentiel

Ne partagez jamais les supports de cours ou vos accès à des tiers.



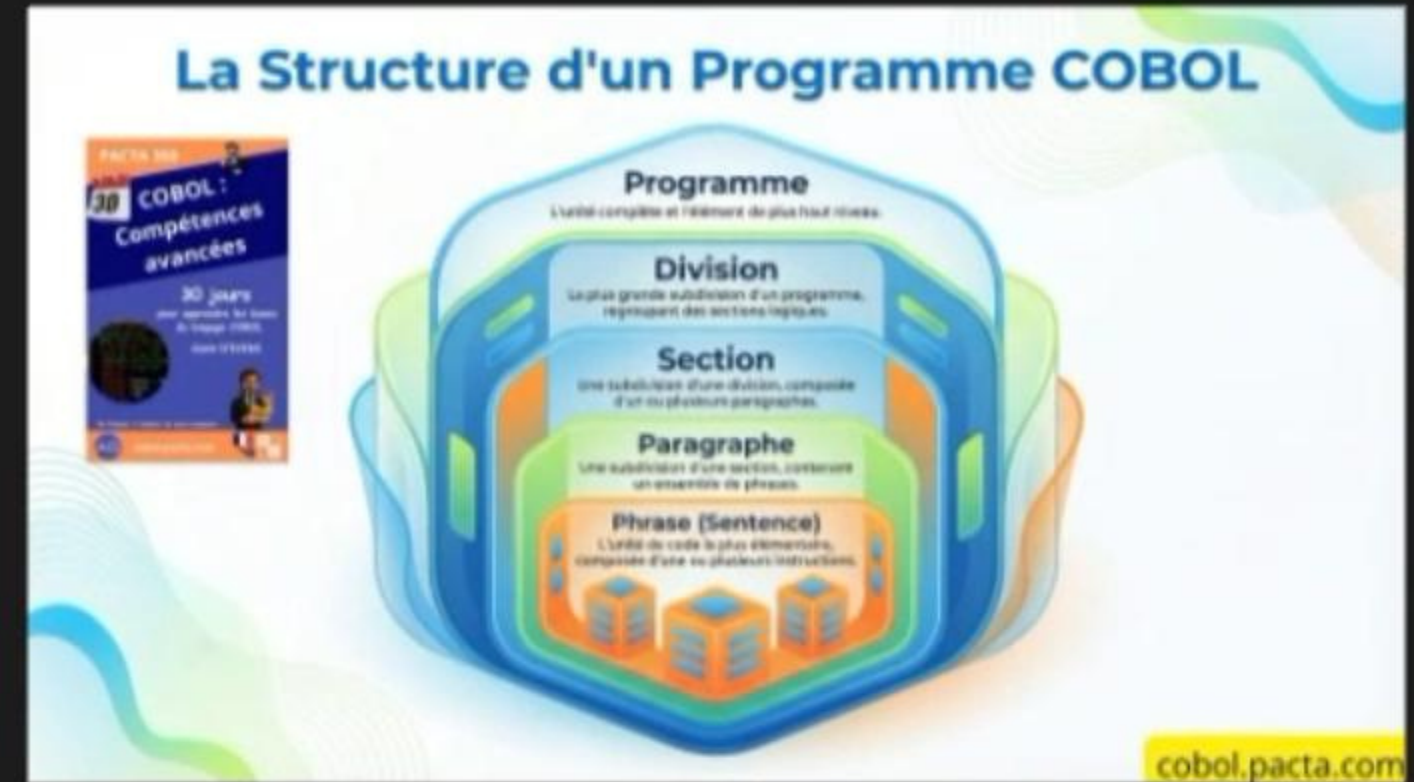
Un support en ligne, pas un cours particulier

Posez vos questions : les plus pertinentes seront sélectionnées pour une réponse groupée.

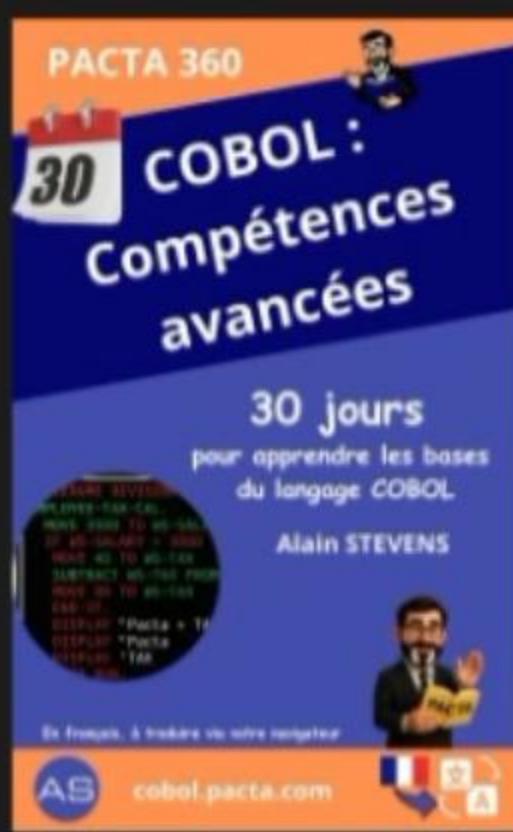
cobol.pacta.com

Les années 1960 voient la structuration et la diffusion de COBOL, avec des ajustements continus et l'adoption croissante au sein des organisations.

1968



Standardisation avec la création de la norme ANSI, garantissant la stabilité et l'uniformité du langage COBOL sur le long terme.



APPRENEZ LE

COBOL AVEC PACTA