

Cours Apl 01 : Introduction : Pourquoi le langage APL ?

S'il est un langage autour duquel la polémique règne depuis longtemps, c'est bien l'APL. Il est régulièrement qualifié de langage exotique utilisant des caractères cabalistiques, de langage non structuré, un peu vieillot.

Il est vrai que la majorité de ses instructions est exprimée en 1 caractère. Ceci dit, dans les autres langages on trouve aussi des fonctions exprimées en 1 caractère : + - * /
Vous les reconnaissez ? En Apl, on les écrit : + - x ÷
Est-ce moins parlant ?

Cette originalité est due au fait qu'APL s'est inspiré de la notation mathématique et que dans ce domaine, on utilise plus de signes que de mots...

La vraie puissance d'APL réside dans sa capacité à exprimer simplement une idée simple. On programme en APL comme on pense, pas comme pense l'ordinateur. Il est d'ailleurs assez amusant de constater que les personnes pour qui les premiers pas en APL sont les plus difficiles sont souvent des ... informaticiens !
APL permet de programmer très rapidement, surtout si on manipule des tableaux (ou matrices) de nombres, de caractères, ou les 2 mélangés.
Pour résumer, on pourrait dire que c'est un "super tableur", qui peut fonctionner avec des vecteurs (1 dimension), des matrices (2 dimensions), des cubes (3 dimensions), des hypercubes (n dimensions).

De part l'esprit de sa programmation, Apl est particulièrement indiqué pour tous les domaines qui nécessitent beaucoup de manipulations de chiffres et où les programmes sont amenés à évoluer fréquemment : science, finance.

Voici un premier petit exemple pratique :

Soit une matrice MAT de 3 colonnes et n lignes contenant des commandes de clients pour un produit quelconque.

Les colonnes contiennent un nom de client, une quantité et un prix unitaire.

On peut connaître instantanément le montant de chaque commande par l'expression suivante (quantité x prix unitaire) :

```
MAT[;2] x MAT[;3]
```

On peut également coller cette nouvelle colonne à celles déjà existantes ainsi :

```
MAT ← MAT, MAT[;2] x MAT[;3]
```

Si on veut connaître le chiffre d'affaires total on peut l'obtenir simplement en demandant la somme de notre nouvelle 4ème colonne :

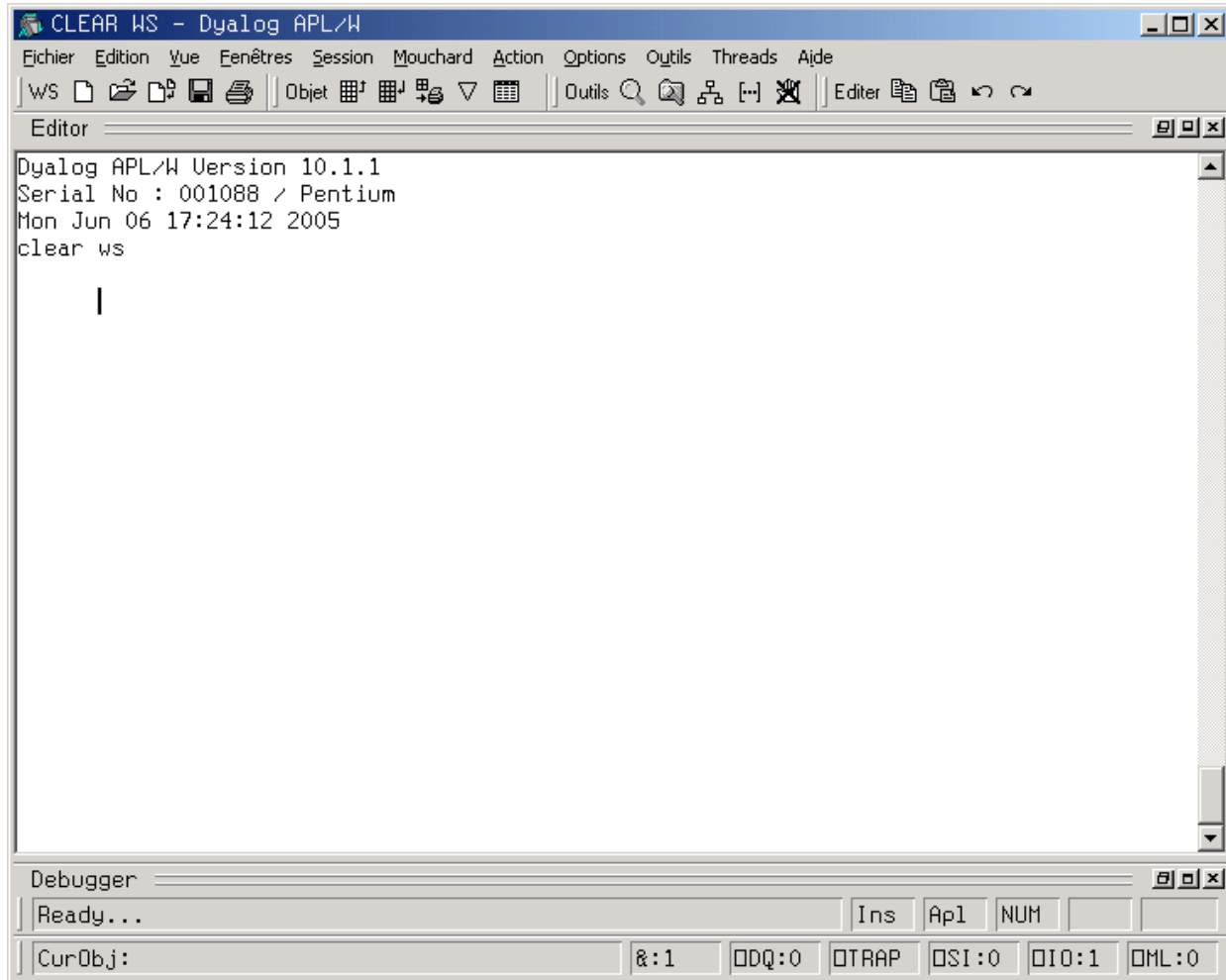
```
+ / MAT[;4]
```

Et pour conclure, pour augmenter les tarifs et le chiffre d'affaires de 10% il suffirait d'écrire :

```
MAT[;3 4] ← MAT[;3 4] x 1.1
```

L'environnement de travail et de développement

Quand vous lancez Dyalog APL, la fenêtre suivante s'ouvre :



- La grande zone blanche est appelée Session. C'est à travers elle que vous dialoguez avec APL.
- Clear ws indique que vous êtes dans un espace de travail vierge, équivalent à un nouveau document dans un autre programme.
- La zone " Editor " permet de visualiser et de modifier les contenus des variables et les textes de fonctions que vous aurez écrites.
- La zone " Debugger " permet de dérouler les programmes pas à pas en voyant le texte de chaque ligne de code exécutée.
- Dans cet exemple, le débogueur est en bas et l'éditeur en haut, mais vous pouvez soit les coller à gauche, à droite, en haut ou en bas, ou bien les laisser flotter, en faisant du " drag and drop ".
- Lorsque vous cessez de travailler vous pouvez bien entendu sauver votre travail soit via en écrivant) SAVE dans la session, soit en cliquant sur la disquette de la barre d'outils.