

== MEMENTO SUR LES POINTEURS EN QUATRE POINTS ==

1. INTRODUCTION

la MEMoire d'une machine peut être vue comme un grand tableau où chaque case est désignée par une ADRESSE (ie un POINTEUR)

2. LE TRIPTYQUE : ÉTOILE, NEW et ÉTOILE (bis !) EXEMPLE MINI :

```
int *ptr = 0;
//
// Déclaration d'un pointeur : ptr est une variable de type <<int *>> (pointeur
sur un entier en anglais "integer pointer")
// ptr est initialisé au pointeur nul qui signifie "adresse invalide" (on peut
aussi écrire <<nullptr>>)
```

```
ptr = new int;
//
// allocation MEM dynamique (ie pendant l'exécution) new alloue une nouvelle
case MEM suffisamment grande pour contenir un entier.
// Maintenant, ptr contient l'adresse d'une case qui contient un entier.
```

```
*ptr = 12;
//
// L'étoile sert aussi à "déréférencer" un pointeur ie désigner "ce qui est
pointé par"
```

3. NE PAS CONFONDRE !

```
cout << ptr;
//
// Affiche la valeur de "ptr" ie une adresse MEM souvent écrite en base 16 (ça
ne nous dit pas grand chose...)
```

```
cout << *ptr;
//
// Affiche ce qui est à l'adresse "ptr" (ie ce qui est pointé par "ptr")
```

4. NE PAS CONFONDRE ! (bis)

Dans une déclaration, l'étoile sert à déclarer un pointeur
Dans une expression, l'étoile sert à "dé-référencer" un pointeur