

Initialisation : = ou () ?

J.-C. Chappelier, V. Lepetit, J. Sam

version 1.0 de septembre 2013

On rencontre souvent dans la littérature et dans des exemples de code l'utilisation du signe égal pour l'initialisation, comme par exemple

```
double x = 3.14;
```

alors que dans le cours nous avons systématiquement opté pour une initialisation avec des parenthèses :

```
double x(3.14);
```

Y a-t-il une différence et, si oui, laquelle ? Et pourquoi un tel choix pour le cours ?

Si vous utilisez des types « de base », tels que ceux présentés au début de ce cours (`int`, `double`, `bool`, `char`, etc.), il n'y a *aucune* différence.

Par contre lorsque vous utiliserez des « objets »¹, comme par exemple dans ce cours les `vector`, `string` ou `array`, voire plus tard vos propres objets, alors il y a une subtile différence : le premier (avec le signe égal) est ce qu'on appelle une « initialisation par copie » alors que le second (avec les parenthèses) est une « initialisation directe ».

Expliquer les différences entre ces deux initialisations est un peu compliqué, et sort du contexte de notre cours, mais il y a deux raisons pour lesquelles nous préférons l'initialisation avec les parenthèses dans notre cours.

La première raison est simplement la similitude avec l'initialisation directe des objets. Vous n'aurez ainsi qu'une seule syntaxe à retenir.

La seconde raison est que nous voulons clairement séparer, surtout dans un cours d'initiation, les deux concepts que sont initialisation et affectation. Il nous semble en effet plus clair d'utiliser deux notations différentes : des parenthèses pour l'initialisation et le signe égal pour l'affectation.

C'est pour ces deux raisons que nous vous conseillons dans ce cours de préférer l'initialisation avec parenthèses, comme par exemple :

```
double x(3.14);
```

Notez enfin que depuis C++11 une troisième syntaxe, qui n'est pas présentée dans ce cours, est possible :

1. Les objets au sens de la programmation ne sont pas abordés dans ce cours d'initiation en tant que tels, mais le seront dans notre cours « Introduction à la Programmation Orientée Objet » qui débutera en Février. Par contre, vous allez déjà en utiliser sans le savoir : les types `vector`, `string`, et `array` sont des objets.

```
double x{3.14};
```

Elle est encore que très peu utilisée, mais tendra peut-être à se généraliser. Pour ma part, je n'y adhère pas du tout en raison de la confusion entre d'une part une notation d'initialisation et d'autre part une notation pour une « *valeur* d'un ensemble », comme par exemple

```
v = { 1, 2, 3 };
```

qui est l'*affectation* d'une *valeur* à, par exemple, un `vector`, telle que présentée plus tard dans ce cours.