

## [Série 2] Les variables

### Nouvelle notion : déclaration d'une variable

Les variables :

- sont des « boîtes » dans lesquelles on pourra stocker des valeurs numériques ou du texte.
- doivent être déclarées avant d'être utilisées.
- doivent être déclarées en tout début de fonction<sup>1</sup>.

```
int main()
{
    // Déclaration des variables
    ...
    ...

    // Code
    ...
    ...

    return 0;
}
```

### Nouvelle notion : le type "entier"

Le mot-clé pour déclarer un nombre entier est « int<sup>2</sup> ».

```
int nombre; // On déclare une variable appelée "nombre".
           // Cette variable "nombre" pourra contenir un nombre entier
```

---

1 En début du « main() », juste après l'accolade ouvrante « { »

2 « int » est l'abréviation de « integer ».

## Nouvelle notion : initialisation d'une variable

Avant de pouvoir être utilisée pour un affichage, une variable doit contenir une valeur. L'initialisation d'une variable consiste à fixer une valeur de départ à cette variable. Il est possible d'effectuer cette initialisation au moment de la déclaration de la variable.

```
int main()
{
    // Déclaration des variables
    int nombre = 50; // On déclare "nombre" et on lui fixe la valeur "50"

    // Code
    ...
    ...

    return 0;
}
```

## Nouvelle notion : affichage d'une variable de type « entier »

Pour afficher une variable, on utilisera la fonction "printf()". L'emplacement (à l'intérieur des guillemets) où doit s'afficher la variable entière sera représenté par «%d ».

```
int main()
{
    // Déclaration des variables
    int nombre = 50; // On déclare "nombre" et on lui fixe la valeur "50"

    // Code
    printf("Le nombre vaut :%d\n", nombre);
    // Permet d'afficher « Le nombre vaut 50 » à l'écran

    return 0;
}
```

## [Programme 1] PR02-01

Déclarer un nombre entier appelé « age<sup>3</sup> ». Cette variable sera initialisée avec la valeur de votre âge. Il faut afficher ce nombre de la manière suivante :

J'ai xx ans

où « xx » sera remplacé par votre âge, grâce à la variable « age ».

---

3 Pas d'accent circonflexe dans une variable !

## Nouvelle notion : les commentaires

Un commentaire dans un programme :

- représente du texte utile pour le programmeur, quand il relira son programme
- n'est pas pris en compte par le compilateur
- doit être précédé par « // » ou encadré par « /\* ... \*/ »
- peut être placé sur toute une ligne ou à droite d'une instruction

```
int main()
{
    /* Ceci est un commentaire*/

    /* Ceci est un commentaire
    qui tient
    sur plusieurs lignes */

    int nombre = 50; // Ceci est un exemple de commentaire, à droite d'une
                    // instruction. Remarquons qu'il faut placer à chaque
                    // fois « // » devant chaque ligne de commentaire

    ...
    ...

    return 0;
}
```

## [Programme 2] PR02-02

Améliorez le programme précédent en plaçant 3 commentaires *de découpe* aux endroits appropriés :

- 1<sup>er</sup> commentaire : « Déclaration des variables »
- 2<sup>ème</sup> commentaire : « Code »
- 3<sup>ème</sup> commentaire : « Fin du programme » (à placer juste avant « return 0 »)

A partir de maintenant, il est demandé de placer ces 3 commentaires de découpe dans chaque futur programme.

## Nouvelle notion : le type "réel"

Mot-clé pour déclarer un nombre réel : « double<sup>4</sup> ».

Particularité: en C, pour séparer la partie entière de la partie décimale d'un nombre réel, on utilise le point et non pas la virgule.

L'emplacement (à l'intérieur des guillemets) où doit s'afficher la variable réelle sera représenté par «%lf».

```
double nombre_reel = 123.456;  
printf("Le nombre vaut :%lf\n", nombre);
```

## [Programme 3] PR02-03

Déclarer 3 nombres réels :

- le premier est nommé « prix\_euro95 » et vaut « 1,61 »
- le deuxième est nommé « prix\_euro98 » et vaut « 1,683 »
- le dernier est nommé « prix\_diesel » et vaut « 1,678 »

Il faut afficher ces 3 prix de la manière suivante :

```
EURO95 : 1.610000 euros  
EURO98 : 1.683000 euros  
DIESEL : 1.678000 euros
```

### Constatations

Par défaut, lors de l'affichage d'un nombre réel, on trouve 6 chiffres après la virgule.

---

4 Dans les anciens compilateurs, il fallait utiliser le mot-clé « float ».

## Nouvelle notion : le type "caractère"

Mot-clé pour déclarer un caractère: « char ».

Particularité: en « C » : un caractère (1 seul!) sera délimité par des apostrophes.

L'emplacement (à l'intérieur des guillemets) où doit s'afficher la variable de type « caractère » sera représenté par « %c ».

```
char caract = '$';          // La variable « caract » contient le caractère « $ »
printf("Le caractere vaut : %c\n", caract);
```

### [Programme 4] PR02-04

Déclarer 3 caractères appelés « lettre », « car1 » et « car2 ». L'une des variables sera initialisée avec la valeur « B », une autre variable sera initialisée avec la valeur « { » et la dernière variable sera initialisée avec la valeur « } ». Il faut afficher le contenu des 3 variables exactement de la manière suivante:

```
Variable "lettre" : B
Variable "car1"   : {
Variable "car2"   : }
Affichage des 3 variables : {B}
```

## Nouvelle notion : le type "chaîne de caractères"

Pour déclarer une chaîne de caractères, il faut:

- utiliser le mot-clé « char »
- choisir un identifiant
- indiquer, entre crochets (« [...] »), au minimum le nombre de caractères +1 de la chaîne<sup>5</sup>

Particularité: une chaîne de caractères sera délimitée par des guillemets.

L'emplacement (à l'intérieur des guillemets) où doit s'afficher la variable de type « chaîne » sera représenté par « %s ».

```
char nom_client[7] = "Dupont";          // La variable « nom_client » peut contenir 6
                                         // caractères significatifs (7 - 1)
                                         // Elle contient le texte « Dupont »
printf("Le client s'appelle %s\n", nom_client);
```

### [Programme 5] PR02-05

Déclarer une chaîne de caractères appelée « mot ». Cette variable sera initialisée avec le mot « programmation ». Il faut réafficher ce mot.

---

<sup>5</sup> La raison du « +1 » sera détaillée plus tard, dans le chapitre sur les chaînes de caractères

## [Programme 6] PR02-06

« Dessinez » un formulaire à l'intérieur duquel on affiche différentes informations d'un client. Ces informations doivent être stockées dans 5 variables qui doivent représenter:

- le nom du client
- le prénom du client
- le jour de naissance
- le mois de naissance
- l'année de naissance

Pour chaque variable, choisissez le type le plus approprié.

```
*****  
  
Nom           : Dupont  
Prenom        : Jean  
Date de naissance : 25/3/1999  
  
*****
```

## [Programme 7] PR02-07

Afficher les informations de 3 clients. Les informations de chaque client sont stockées dans des variables différentes. Il faut donc créer 15 variables au total.

```
*****  
  
Nom           : Dupont  
Prenom        : Jean  
Date de naissance : 25/3/1999  
  
*****  
  
Nom           : Vandentruc  
Prenom        : Marie  
Date de naissance : 30/6/1985  
  
*****  
  
Nom           : Dupond  
Prenom        : Pierre  
Date de naissance : 12/5/2005  
  
*****
```

### Contraintes de programmation

Les noms des variables pour le 1<sup>er</sup> client posséderont le suffixe « 1 » (par exemple, « nom1 », « mois1 », etc.). Les variables pour le 2<sup>ème</sup> client posséderont le suffixe « 2 » (par exemple, « jour2 », « mois2 », etc.). Même remarque pour les variables du 3<sup>ème</sup> client.