

# Notion de variable

**Mohamed MESSABIHI**

[mohamed.messabihi@gmail.com](mailto:mohamed.messabihi@gmail.com)

Université de Tlemcen  
Département d'informatique

30 septembre 2014



# Vous avez dit variable ?

**Question :** Écrire un programme qui calcule la somme de deux entier.

- L'ordinateur doit pouvoir retenir les deux entiers avant d'effectuer l'opération d'addition.
- Il a donc besoin d'une mémoire !



# Vous avez dit variable ?

**Question :** Écrire un programme qui calcule la somme de deux entier.

- L'ordinateur doit pouvoir retenir les deux entiers avant d'effectuer l'opération d'addition.
- Il a donc besoin d'une mémoire !



# Vous avez dit variable ?

**Question :** Écrire un programme qui calcule la somme de deux entier.

- L'ordinateur doit pouvoir retenir les deux entiers avant d'effectuer l'opération d'addition.
- Il a donc besoin d'une mémoire !

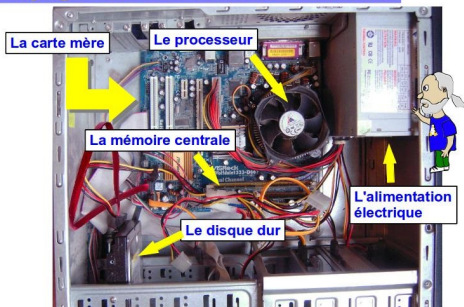


# Vous avez dit variable ?

**Question :** Écrire un programme qui calcule la somme de deux entier.

- L'ordinateur doit pouvoir retenir les deux entiers avant d'effectuer l'opération d'addition.
- Il a donc besoin d'une mémoire !

## Voyage au cœur de l'unité centrale

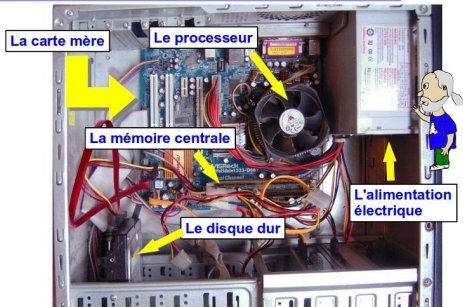


# Vous avez dit variable ?

**Question :** Écrire un programme qui calcule la somme de deux entiers.

- L'ordinateur doit pouvoir retenir les deux entiers avant d'effectuer l'opération d'addition.
- Il a donc besoin d'une mémoire !

## Voyage au cœur de l'unité centrale

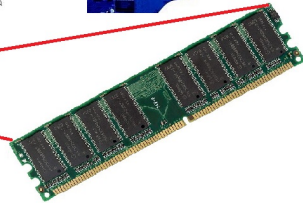
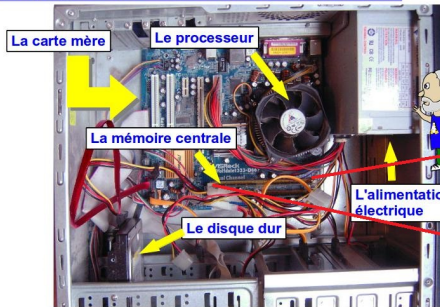


# Vous avez dit variable ?

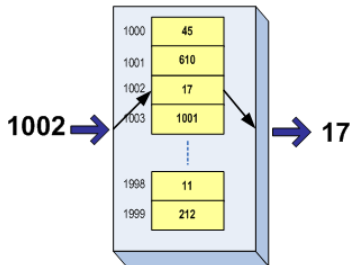
**Question :** Écrire un programme qui calcule la somme de deux entiers.

- L'ordinateur doit pouvoir retenir les deux entiers avant d'effectuer l'opération d'addition.
- Il a donc besoin d'une mémoire !

## Voyage au cœur de l'unité centrale



# Schéma du fonctionnement de la mémoire vive



1. **Les adresses** : une adresse est un nombre qui permet à l'ordinateur de se repérer dans la mémoire vive.
2. **Les valeurs** : à chaque adresse, on peut stocker une valeur (un nombre). On ne peut stocker qu'un nombre par adresse !

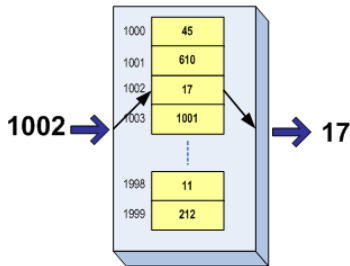
## Problème

Cette mémoire ne peut stocker que des nombres ? comment fait-on pour retenir des mots par exemple ?





# Schéma du fonctionnement de la mémoire vive



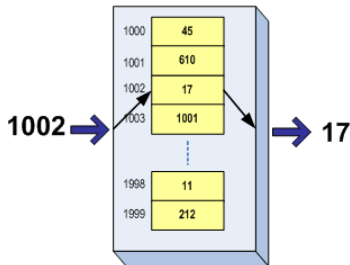
1. **Les adresses** : une adresse est un nombre qui permet à l'ordinateur de se repérer dans la mémoire vive.
2. **Les valeurs** : à chaque adresse, on peut stocker une valeur (un nombre). On ne peut stocker qu'un nombre par adresse !

## Problème

Cette mémoire ne peut stocker que des nombres ? comment fait-on pour retenir des mots par exemple ?



# Schéma du fonctionnement de la mémoire vive



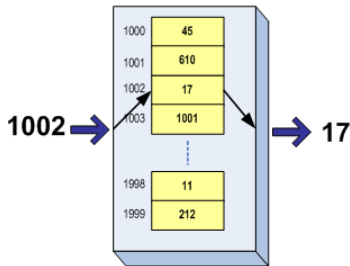
1. **Les adresses** : une adresse est un nombre qui permet à l'ordinateur de se repérer dans la mémoire vive.
2. **Les valeurs** : à chaque adresse, on peut stocker une valeur (un nombre). On ne peut stocker qu'un nombre par adresse !

## Problème

Cette mémoire ne peut stocker que des nombres ? comment fait-on pour retenir des mots par exemple ?



# Schéma du fonctionnement de la mémoire vive



1. **Les adresses** : une adresse est un nombre qui permet à l'ordinateur de se repérer dans la mémoire vive.
2. **Les valeurs** : à chaque adresse, on peut stocker une valeur (un nombre). On ne peut stocker qu'un nombre par adresse !

## Problème

Cette mémoire ne peut stocker que des nombres ? comment fait-on pour retenir des mots par exemple ?

# Notion de variable

- Une variable est une information temporaire qu'on stocke dans la mémoire.
- on l'appelle « variable » car c'est une valeur qui peut changer pendant le déroulement du programme.
- En langage C, une variable est constituée de deux choses :
  1. **un nom** : c'est ce qui permet de la reconnaître. En programmation, on n'aura pas à retenir l'adresse mémoire, on va juste indiquer des noms de variables. C'est le compilateur qui fera la conversion entre le nom et l'adresse.
  2. **une valeur** : c'est le nombre qu'elle stocke, par exemple 7 ;



# Notion de variable

- Une variable est une information temporaire qu'on stocke dans la mémoire.
- on l'appelle « variable » car c'est une valeur qui peut changer pendant le déroulement du programme.
- En langage C, une variable est constituée de deux choses :
  1. **un nom** : c'est ce qui permet de la reconnaître. En programmation, on n'aura pas à retenir l'adresse mémoire, on va juste indiquer des noms de variables. C'est le compilateur qui fera la conversion entre le nom et l'adresse.
  2. **une valeur** : c'est le nombre qu'elle stocke, par exemple 7 ;



# Notion de variable

- Une variable est une information temporaire qu'on stocke dans la mémoire.
- on l'appelle « variable » car c'est une valeur qui peut changer pendant le déroulement du programme.
- En langage C, une variable est constituée de deux choses :
  1. **un nom** : c'est ce qui permet de la reconnaître. En programmation, on n'aura pas à retenir l'adresse mémoire, on va juste indiquer des noms de variables. C'est le compilateur qui fera la conversion entre le nom et l'adresse.
  2. **une valeur** : c'est le nombre qu'elle stocke, par exemple 7 ;



# Notion de variable

- Une variable est une information temporaire qu'on stocke dans la mémoire.
- on l'appelle « variable » car c'est une valeur qui peut changer pendant le déroulement du programme.
- En langage C, une variable est constituée de deux choses :
  1. **un nom** : c'est ce qui permet de la reconnaître. En programmation, on n'aura pas à retenir l'adresse mémoire, on va juste indiquer des noms de variables. C'est le compilateur qui fera la conversion entre le nom et l'adresse.
  2. **une valeur** : c'est le nombre qu'elle stocke, par exemple 7 ;



# Notion de variable

- Une variable est une information temporaire qu'on stocke dans la mémoire.
- on l'appelle « variable » car c'est une valeur qui peut changer pendant le déroulement du programme.
- En langage C, une variable est constituée de deux choses :
  1. **un nom** : c'est ce qui permet de la reconnaître. En programmation, on n'aura pas à retenir l'adresse mémoire, on va juste indiquer des noms de variables. C'est le compilateur qui fera la conversion entre le nom et l'adresse.
  2. **une valeur** : c'est le nombre qu'elle stocke, par exemple 7 ;





# Nommer une variables

Dans la plus part des langage de programmation, il y a quelques contraintes et conventions à respecter pour appeler des variables :

- il ne peut y avoir que des minuscules, majuscules et des chiffres (abcABC012) ;
- votre nom de variable doit commencer par une lettre ;
- les espaces sont interdits. À la place, on peut utiliser le caractère « underscore » \_ C'est le seul caractère différent des lettres et chiffres autorisé ;
- vous n'avez pas le droit d'utiliser des accents (èâê etc.).

## Exemple

Quels sont les noms de variables corrects : notelInfo, mail, nom, bac2014, numero\_de\_telephone, Prix-TTC, prénom, main.



# Nommer une variables

Dans la plus part des langage de programmation, il y a quelques contraintes et conventions à respecter pour appeler des variables :

- il ne peut y avoir que des minuscules, majuscules et des chiffres (abcABC012) ;
- votre nom de variable doit commencer par une lettre ;
- les espaces sont interdits. À la place, on peut utiliser le caractère « underscore » \_ C'est le seul caractère différent des lettres et chiffres autorisé ;
- vous n'avez pas le droit d'utiliser des accents (èâê etc.).

## Exemple

Quels sont les noms de variables corrects : notelInfo, mail, nom, bac2014, numero\_de\_telephone, Prix-TTC, prénom, main.



# Nommer une variables

Dans la plus part des langage de programmation, il y a quelques contraintes et conventions à respecter pour appeler des variables :

- il ne peut y avoir que des minuscules, majuscules et des chiffres (abcABC012) ;
- votre nom de variable doit commencer par une lettre ;
- les espaces sont interdits. À la place, on peut utiliser le caractère « underscore » \_ C'est le seul caractère différent des lettres et chiffres autorisé ;
- vous n'avez pas le droit d'utiliser des accents (èâê etc.).

## Exemple

Quels sont les noms de variables corrects : notelInfo, mail, nom, bac2014, numero\_de\_telephone, Prix-TTC, prénom, main.



# Nommer une variables

Dans la plus part des langage de programmation, il y a quelques contraintes et conventions à respecter pour appeler des variables :

- il ne peut y avoir que des minuscules, majuscules et des chiffres (abcABC012) ;
- votre nom de variable doit commencer par une lettre ;
- les espaces sont interdits. À la place, on peut utiliser le caractère « underscore » \_ C'est le seul caractère différent des lettres et chiffres autorisé ;
- vous n'avez pas le droit d'utiliser des accents (èâê etc.).

## Exemple

Quels sont les noms de variables corrects : notelInfo, mail, nom, bac2014, numero\_de\_telephone, Prix-TTC, prénom, main.



# Nommer une variables

Dans la plus part des langage de programmation, il y a quelques contraintes et conventions à respecter pour appeler des variables :

- il ne peut y avoir que des minuscules, majuscules et des chiffres (abcABC012) ;
- votre nom de variable doit commencer par une lettre ;
- les espaces sont interdits. À la place, on peut utiliser le caractère « underscore » \_ C'est le seul caractère différent des lettres et chiffres autorisé ;
- vous n'avez pas le droit d'utiliser des accents (èâê etc.).

## Exemple

Quels sont les noms de variables corrects : notelInfo, mail, nom, bac2014, numero\_de\_telephone, Prix-TTC, prénom, main.



# Nommer une variables

Dans la plus part des langage de programmation, il y a quelques contraintes et conventions à respecter pour appeler des variables :

- il ne peut y avoir que des minuscules, majuscules et des chiffres (abcABC012) ;
- votre nom de variable doit commencer par une lettre ;
- les espaces sont interdits. À la place, on peut utiliser le caractère « underscore » \_ C'est le seul caractère différent des lettres et chiffres autorisé ;
- vous n'avez pas le droit d'utiliser des accents (èâê etc.).

## Exemple

Quels sont les noms de variables corrects : `notelInfo`, `mail`, `nom`, `bac2014`, `numero_de_telephone`, `Prix-TTC`, `prénom`, `main`.



# Nommer une variables

Dans la plus part des langage de programmation, il y a quelques contraintes et conventions à respecter pour appeler des variables :

- il ne peut y avoir que des minuscules, majuscules et des chiffres (abcABC012) ;
- votre nom de variable doit commencer par une lettre ;
- les espaces sont interdits. À la place, on peut utiliser le caractère « underscore » \_ C'est le seul caractère différent des lettres et chiffres autorisé ;
- vous n'avez pas le droit d'utiliser des accents (èâê etc.).

## Exemple

Quels sont les noms de variables corrects : `notelInfo`, `mail`, `nom`, `bac2014`, `numero_de_telephone`, `Prix-TTC`, `prénom`, `main`.



# Nommer une variables

Dans la plus part des langage de programmation, il y a quelques contraintes et conventions à respecter pour appeler des variables :

- il ne peut y avoir que des minuscules, majuscules et des chiffres (abcABC012) ;
- votre nom de variable doit commencer par une lettre ;
- les espaces sont interdits. À la place, on peut utiliser le caractère « underscore » \_ C'est le seul caractère différent des lettres et chiffres autorisé ;
- vous n'avez pas le droit d'utiliser des accents (èâê etc.).

## Exemple

Quels sont les noms de variables corrects : `notelInfo`, `mail`, `nom`, `bac2014`, `numero_de_telephone`, `Prix-TTC`, `prénom`, `main`.





# Nommer une variables

Dans la plus part des langage de programmation, il y a quelques contraintes et conventions à respecter pour appeler des variables :

- il ne peut y avoir que des minuscules, majuscules et des chiffres (abcABC012) ;
- votre nom de variable doit commencer par une lettre ;
- les espaces sont interdits. À la place, on peut utiliser le caractère « underscore » \_ C'est le seul caractère différent des lettres et chiffres autorisé ;
- vous n'avez pas le droit d'utiliser des accents (èâê etc.).

## Exemple

Quels sont les noms de variables corrects : `notelInfo`, `mail`, `nom`, `bac2014`, `numero_de_telephone`, `Prix-TTC`, `prénom`, `main`.



# Nommer une variables

Dans la plus part des langage de programmation, il y a quelques contraintes et conventions à respecter pour appeler des variables :

- il ne peut y avoir que des minuscules, majuscules et des chiffres (abcABC012) ;
- votre nom de variable doit commencer par une lettre ;
- les espaces sont interdits. À la place, on peut utiliser le caractère « underscore » \_ C'est le seul caractère différent des lettres et chiffres autorisé ;
- vous n'avez pas le droit d'utiliser des accents (èâê etc.).

## Exemple

Quels sont les noms de variables corrects : notelInfo, mail, nom, bac2014, numero\_de\_telephone, Prix-TTC, prénom, main.



# Nommer une variables

Dans la plus part des langage de programmation, il y a quelques contraintes et conventions à respecter pour appeler des variables :

- il ne peut y avoir que des minuscules, majuscules et des chiffres (abcABC012) ;
- votre nom de variable doit commencer par une lettre ;
- les espaces sont interdits. À la place, on peut utiliser le caractère « underscore » \_ C'est le seul caractère différent des lettres et chiffres autorisé ;
- vous n'avez pas le droit d'utiliser des accents (èâê etc.).

## Exemple

Quels sont les noms de variables corrects : notelInfo, mail, nom, bac2014, numero\_de\_telephone, Prix-TTC, prénom, main.



# Nommer une variables

Dans la plus part des langage de programmation, il y a quelques contraintes et conventions à respecter pour appeler des variables :

- il ne peut y avoir que des minuscules, majuscules et des chiffres (abcABC012) ;
- votre nom de variable doit commencer par une lettre ;
- les espaces sont interdits. À la place, on peut utiliser le caractère « underscore » \_ C'est le seul caractère différent des lettres et chiffres autorisé ;
- vous n'avez pas le droit d'utiliser des accents (èâê etc.).

## Exemple

Quels sont les noms de variables corrects : notelInfo, mail, nom, bac2014, numero\_de\_telephone, Prix-TTC, prénom, main.



# Nommer une variables

Dans la plus part des langage de programmation, il y a quelques contraintes et conventions à respecter pour appeler des variables :

- il ne peut y avoir que des minuscules, majuscules et des chiffres (abcABC012) ;
- votre nom de variable doit commencer par une lettre ;
- les espaces sont interdits. À la place, on peut utiliser le caractère « underscore » \_ C'est le seul caractère différent des lettres et chiffres autorisé ;
- vous n'avez pas le droit d'utiliser des accents (èâê etc.).

## Exemple

Quels sont les noms de variables corrects : notelInfo, mail, nom, bac2014, numero\_de\_telephone, Prix-TTC, prénom, main.



# Nommer une variables

Dans la plus part des langage de programmation, il y a quelques contraintes et conventions à respecter pour appeler des variables :

- il ne peut y avoir que des minuscules, majuscules et des chiffres (abcABC012) ;
- votre nom de variable doit commencer par une lettre ;
- les espaces sont interdits. À la place, on peut utiliser le caractère « underscore » \_ C'est le seul caractère différent des lettres et chiffres autorisé ;
- vous n'avez pas le droit d'utiliser des accents (èâê etc.).

## Exemple

Quels sont les noms de variables corrects : notelInfo, mail, nom, bac2014, numero\_de\_telephone, Prix-TTC, prénom, main.



# Quelques bonnes pratiques pour nommer vos variables

- Chaque programmeur a sa propre façon de nommer des variables ;
- Commencer tous les noms de variables par une lettre minuscule ;
- s'il y a plusieurs mots dans le nom de variable, mettez une lettre majuscule au début de chaque nouveau mot. Par exemple :  
PrixHorsTaxe ou NombreDePointGagnes
- Faites en sorte de donner des noms clairs à vos variables. On aurait pu abrégé NombreDePointGagnes, en l'écrivant par exemple ndpg. C'est peut-être plus court, mais c'est beaucoup moins clair pour vous quand vous relisez votre code. N'ayez donc pas peur de donner des noms un peu plus longs pour que ça reste compréhensible.



# Quelques bonnes pratiques pour nommer vos variables

- Chaque programmeur a sa propre façon de nommer des variables ;
- Commencer tous les noms de variables par une lettre minuscule ;
- s'il y a plusieurs mots dans le nom de variable, mettez une lettre majuscule au début de chaque nouveau mot. Par exemple :  
PrixHorsTaxe ou NombreDePointGagnes
- Faites en sorte de donner des noms clairs à vos variables. On aurait pu abréger NombreDePointGagnes, en l'écrivant par exemple ndpg. C'est peut-être plus court, mais c'est beaucoup moins clair pour vous quand vous relisez votre code. N'ayez donc pas peur de donner des noms un peu plus longs pour que ça reste compréhensible.





# Quelques bonnes pratiques pour nommer vos variables

- Chaque programmeur a sa propre façon de nommer des variables ;
- Commencer tous les noms de variables par une lettre minuscule ;
- s'il y a plusieurs mots dans le nom de variable, mettez une lettre majuscule au début de chaque nouveau mot. Par exemple :  
PrixHorsTaxe ou NombreDePointGagnes
- Faites en sorte de donner des noms clairs à vos variables. On aurait pu abréger NombreDePointGagnes, en l'écrivant par exemple ndpg. C'est peut-être plus court, mais c'est beaucoup moins clair pour vous quand vous relisez votre code. N'ayez donc pas peur de donner des noms un peu plus longs pour que ça reste compréhensible.



# Quelques bonnes pratiques pour nommer vos variables

- Chaque programmeur a sa propre façon de nommer des variables ;
- Commencer tous les noms de variables par une lettre minuscule ;
- s'il y a plusieurs mots dans le nom de variable, mettez une lettre majuscule au début de chaque nouveau mot. Par exemple :  
PrixHorsTaxe ou NombreDePointGagnes
- Faites en sorte de donner des noms clairs à vos variables. On aurait pu abréger NombreDePointGagnes, en l'écrivant par exemple ndpg. C'est peut-être plus court, mais c'est beaucoup moins clair pour vous quand vous relisez votre code. N'ayez donc pas peur de donner des noms un peu plus longs pour que ça reste compréhensible.



# Les types de variables

- Il existe plusieurs types de nombres : 327, 47.10, -38, -68597.00007654
- Lorsque vous créez une variable, vous devez indiquer son type,
- Voici les principaux types de variables existant en langage C :

1. pour un nombre **entier**, on utilisera le plus souvent **int** ;
2. pour un nombre **flottant** (ou réel ou décimal), on utilisera généralement **double**.



# Les types de variables

- Il existe plusieurs types de nombres : 327, 47.10, -38, -68597.00007654
- Lorsque vous créez une variable, vous devez indiquer son type,
- Voici les principaux types de variables existant en langage C :

1. pour un nombre **entier**, on utilisera le plus souvent **int** ;
2. pour un nombre **flottant** (ou réel ou décimal), on utilisera généralement **double**.



# Les types de variables

- Il existe plusieurs types de nombres : 327, 47.10, -38, -68597.00007654
- Lorsque vous créez une variable, vous devez indiquer son type,
- Voici les principaux types de variables existant en langage C :

1. pour un nombre **entier**, on utilisera le plus souvent **int** ;
2. pour un nombre **flottant** (ou réel ou décimal), on utilisera généralement **double**.



# Les types de variables

- Il existe plusieurs types de nombres : 327, 47.10, -38, -68597.00007654
- Lorsque vous créez une variable, vous devez indiquer son type,
- Voici les principaux types de variables existant en langage C :

Nom du type	Minimum	Maximum
signed char	-127	127
int	-32 767	32 767
long	-2 147 483 647	2 147 483 647
float	$3.4 \cdot 10^{-38}$	$3.4 \cdot 10^{38}$
double	$1.7 \cdot 10^{-308}$	$1.7 \cdot 10^{308}$

1. pour un nombre **entier**, on utilisera le plus souvent **int** ;
2. pour un nombre **flottant** (ou réel ou décimal), on utilisera généralement **double**.



# Les types de variables

- Il existe plusieurs types de nombres : 327, 47.10, -38, -68597.00007654
- Lorsque vous créez une variable, vous devez indiquer son type,
- Voici les principaux types de variables existant en langage C :

Nom du type	Minimum	Maximum
signed char	-127	127
int	-32 767	32 767
long	-2 147 483 647	2 147 483 647
float	$3.4 \cdot 10^{-38}$	$3.4 \cdot 10^{38}$
double	$1.7 \cdot 10^{-308}$	$1.7 \cdot 10^{308}$

1. pour un nombre **entier**, on utilisera le plus souvent **int** ;
2. pour un nombre **flottant** (ou réel ou décimal), on utilisera généralement **double**.



# Les types de variables

- Il existe plusieurs types de nombres : 327, 47.10, -38, -68597.00007654
- Lorsque vous créez une variable, vous devez indiquer son type,
- Voici les principaux types de variables existant en langage C :

Nom du type	Minimum	Maximum
signed char	-127	127
int	-32 767	32 767
long	-2 147 483 647	2 147 483 647
float	$3.4 \cdot 10^{-38}$	$3.4 \cdot 10^{38}$
double	$1.7 \cdot 10^{-308}$	$1.7 \cdot 10^{308}$

1. pour un nombre **entier**, on utilisera le plus souvent **int** ;
2. pour un nombre **flottant** (ou réel ou décimal), on utilisera généralement **double**.





# Déclarer une variable

Vous devez faire les déclarations de variables au début des fonctions. Il suffit juste :

1. d'indiquer le type de la variable que l'on veut créer ;
2. d'insérer un espace ;
3. d'indiquer le nom que vous voulez donner à la variable ;
4. et enfin, de ne pas oublier le point-virgule.

## Exemple :

```
int main()  
{  
    int age ;  
    double salaire ;  
    unsigned int somme , nombreEtudiant , coefficient ;  
  
    return 0 ;  
}
```

# Déclarer une variable

Vous devez faire les déclarations de variables au début des fonctions. Il suffit juste :

1. d'indiquer le type de la variable que l'on veut créer ;
2. d'insérer un espace ;
3. d'indiquer le nom que vous voulez donner à la variable ;
4. et enfin, de ne pas oublier le point-virgule.

## Exemple :

```
int main()  
{  
    int age ;  
    double salaire ;  
    unsigned int somme , nombreEtudiant , coefficient ;  
  
    return 0 ;  
}
```

# Déclarer une variable

Vous devez faire les déclarations de variables au début des fonctions. Il suffit juste :

1. d'indiquer le type de la variable que l'on veut créer ;
2. d'insérer un espace ;
3. d'indiquer le nom que vous voulez donner à la variable ;
4. et enfin, de ne pas oublier le point-virgule.

## Exemple :

```
int main()  
{  
    int age ;  
    double salaire ;  
    unsigned int somme , nombreEtudiant , coefficient ;  
  
    return 0 ;  
}
```

# Déclarer une variable

Vous devez faire les déclarations de variables au début des fonctions. Il suffit juste :

1. d'indiquer le type de la variable que l'on veut créer ;
2. d'insérer un espace ;
3. d'indiquer le nom que vous voulez donner à la variable ;
4. et enfin, de ne pas oublier le point-virgule.

## Exemple :

```
int main()  
{  
    int age ;  
    double salaire ;  
    unsigned int somme , nombreEtudiant , coefficient ;  
  
    return 0 ;  
}
```

# Déclarer une variable

Vous devez faire les déclarations de variables au début des fonctions. Il suffit juste :

1. d'indiquer le type de la variable que l'on veut créer ;
2. d'insérer un espace ;
3. d'indiquer le nom que vous voulez donner à la variable ;
4. et enfin, de ne pas oublier le point-virgule.

## Exemple :

```
int main()  
{  
    int age ;  
    double salaire ;  
    unsigned int somme , nombreEtudiant , coefficient ;  
  
    return 0 ;  
}
```

# Déclarer une variable

Vous devez faire les déclarations de variables au début des fonctions. Il suffit juste :

1. d'indiquer le type de la variable que l'on veut créer ;
2. d'insérer un espace ;
3. d'indiquer le nom que vous voulez donner à la variable ;
4. et enfin, de ne pas oublier le point-virgule.

## Exemple :

```
int main()  
{  
    int age ;  
    double salaire ;  
    unsigned int somme , nombreEtudiant , coefficient ;  
  
    return 0;  
}
```

# Affecter une valeur à une variable

Indiquez simplement le nom de la variable, puis un signe égal (=), et enfin la valeur que vous voulez y mettre.

## Exemple :

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int nombreEtudiants;
    nombreEtudiants = 240;

    return 0;
}
```



# Initialiser une variable

- Quand on déclare une variable, quelle valeur a-t-elle au départ ?
- Initialiser une variable ça consiste à combiner la déclaration et l'affectation de cette variable dans la même instruction
- L'avantage, c'est que vous êtes sûrs que cette variable contient une valeur correcte, et pas du n'importe quoi.

## Exemple :

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int nombreEtudiants = 240;

    nombreEtudiants = 260
    nombreEtudiants = 255
    nombreEtudiants = 258

    return 0;
}
```





# Initialiser une variable

- Quand on déclare une variable, quelle valeur a-t-elle au départ ?
- Initialiser une variable ça consiste à combiner la déclaration et l'affectation de cette variable dans la même instruction
- L'avantage, c'est que vous êtes sûrs que cette variable contient une valeur correcte, et pas du n'importe quoi.

## Exemple :

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int nombreEtudiants = 240;

    nombreEtudiants = 260
    nombreEtudiants = 255
    nombreEtudiants = 258

    return 0;
}
```



# Initialiser une variable

- Quand on déclare une variable, quelle valeur a-t-elle au départ ?
- Initialiser une variable ça consiste à combiner la déclaration et l'affectation de cette variable dans la même instruction
- L'avantage, c'est que vous êtes sûrs que cette variable contient une valeur correcte, et pas du n'importe quoi.

## Exemple :

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int nombreEtudiants = 240;

    nombreEtudiants = 260
    nombreEtudiants = 255
    nombreEtudiants = 258

    return 0;
}
```



# Initialiser une variable

- Quand on déclare une variable, quelle valeur a-t-elle au départ ?
- Initialiser une variable ça consiste à combiner la déclaration et l'affectation de cette variable dans la même instruction
- L'avantage, c'est que vous êtes sûrs que cette variable contient une valeur correcte, et pas du n'importe quoi.

## Exemple :

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int nombreEtudiants = 240;

    nombreEtudiants = 260
    nombreEtudiants = 255
    nombreEtudiants = 258

    return 0;
}
```



# Afficher une variable

On utilise **printf** de la même manière que pour afficher un texte, sauf que l'on rajoute un symbole spécial à l'endroit où l'on veut afficher la valeur de la variable. Par exemple :

## Exemple :

```
int main()
{
    int nombreEtudiants = 240;
    printf("Il y a %d etudiants inscrits.",
        nombreEtudiants);
    ...
}
```

La lettre après le % permet d'indiquer ce que l'on doit afficher. 'd' signifie que l'on souhaite afficher un **int**.



# Afficher une variable

On utilise **printf** de la même manière que pour afficher un texte, sauf que l'on rajoute un symbole spécial à l'endroit où l'on veut afficher la valeur de la variable. Par exemple :

## Exemple :

```
int main()
{
    int nombreEtudiants = 240;
    printf("Il y a %d etudiants inscrits.",
        nombreEtudiants);
    ...
}
```

La lettre après le % permet d'indiquer ce que l'on doit afficher. '**d**' signifie que l'on souhaite afficher un **int**.



# Afficher une variable

On utilise **printf** de la même manière que pour afficher un texte, sauf que l'on rajoute un symbole spécial à l'endroit où l'on veut afficher la valeur de la variable. Par exemple :

## Exemple :

```
int main()
{
    int nombreEtudiants = 240;
    printf("Il y a %d etudiants inscrits.",
        nombreEtudiants);
    ...
}
```

La lettre après le % permet d'indiquer ce que l'on doit afficher. '**d**' signifie que l'on souhaite afficher un **int**.

Format	Type attendu
"%d"	int
"%ld"	long
"%f"	float
"%lf"	double



# Afficher plusieurs variables avec un seul printf

- Il est possible d'afficher la valeur de plusieurs variables dans un seul printf.
- Il vous suffit pour cela d'indiquer des % d ou des % f là où vous voulez, puis d'indiquer les variables correspondantes dans le même ordre, séparées par des virgules.

## Exemple :

```
int main()
{
    int nombreEtudiants = 240;
    double moyenneBac = 14.5 ;

    printf("Il y a %d etudiants inscrits avec une
           moyenne de %f en Bac", nombreEtudiants,
           moyenneBac);

    return 0;
}
```



# Afficher plusieurs variables avec un seul printf

- Il est possible d'afficher la valeur de plusieurs variables dans un seul printf.
- Il vous suffit pour cela d'indiquer des % d ou des % f là où vous voulez, puis d'indiquer les variables correspondantes dans le même ordre, séparées par des virgules.

## Exemple :

```
int main()
{
    int nombreEtudiants = 240;
    double moyenneBac = 14.5 ;

    printf("Il y a %d etudiants inscrits avec une
           moyenne de %f en Bac", nombreEtudiants ,
           moyenneBac);

    return 0;
}
```



# Saisir la valeur d'une variable

- On va utiliser une autre fonction toute prête : **scanf**.
- Cette fonction ressemble beaucoup à printf. Vous devez mettre un format pour indiquer ce que l'utilisateur doit entrer (un int, un float, ...)
- Puis vous devez indiquer le nom de la variable qui va recevoir le nombre.

## Exemple :

```
int main()
{
    int age = 0;
    scanf("%d", &age);

    return 0;
}
```

- On doit mettre le %d entre guillemets.
- Par ailleurs, il faut mettre le symbole & devant le nom de la variable qui va recevoir la valeur.



# Saisir la valeur d'une variable

- On va utiliser une autre fonction toute prête : **scanf**.
- Cette fonction ressemble beaucoup à printf. Vous devez mettre un format pour indiquer ce que l'utilisateur doit entrer (un int, un float, ...)
- Puis vous devez indiquer le nom de la variable qui va recevoir le nombre.

## Exemple :

```
int main()
{
    int age = 0;
    scanf("%d", &age);

    return 0;
}
```

- On doit mettre le %d entre guillemets.
- Par ailleurs, il faut mettre le symbole & devant le nom de la variable qui va recevoir la valeur.



# Saisir la valeur d'une variable

- On va utiliser une autre fonction toute prête : **scanf**.
- Cette fonction ressemble beaucoup à printf. Vous devez mettre un format pour indiquer ce que l'utilisateur doit entrer (un int, un float, ...)
- Puis vous devez indiquer le nom de la variable qui va recevoir le nombre.

## Exemple :

```
int main()
{
    int age = 0;
    scanf("%d", &age);

    return 0;
}
```

- On doit mettre le %d entre guillemets.
- Par ailleurs, il faut mettre le symbole & devant le nom de la variable qui va recevoir la valeur.



# Saisir la valeur d'une variable

- On va utiliser une autre fonction toute prête : **scanf**.
- Cette fonction ressemble beaucoup à printf. Vous devez mettre un format pour indiquer ce que l'utilisateur doit entrer (un int, un float, ...)
- Puis vous devez indiquer le nom de la variable qui va recevoir le nombre.

## Exemple :

```
int main()
{
    int age = 0;
    scanf("%d", &age);

    return 0;
}
```

- On doit mettre le %d entre guillemets.
- Par ailleurs, il faut mettre le symbole & devant le nom de la variable qui va recevoir la valeur.



# Saisir la valeur d'une variable

- On va utiliser une autre fonction toute prête : **scanf**.
- Cette fonction ressemble beaucoup à printf. Vous devez mettre un format pour indiquer ce que l'utilisateur doit entrer (un int, un float, ...)
- Puis vous devez indiquer le nom de la variable qui va recevoir le nombre.

## Exemple :

```
int main()
{
    int age = 0;
    scanf("%d", &age);

    return 0;
}
```

- On doit mettre le %d entre guillemets.
- Par ailleurs, il faut mettre le symbole & devant le nom de la variable qui va recevoir la valeur.

# Saisir la valeur d'une variable

- On va utiliser une autre fonction toute prête : **scanf**.
- Cette fonction ressemble beaucoup à printf. Vous devez mettre un format pour indiquer ce que l'utilisateur doit entrer (un int, un float, ...)
- Puis vous devez indiquer le nom de la variable qui va recevoir le nombre.

## Exemple :

```
int main()
{
    int age = 0;
    scanf("%d", &age);

    return 0;
}
```

- On doit mettre le %d entre guillemets.
- Par ailleurs, il faut mettre le symbole & devant le nom de la variable qui va recevoir la valeur.

# Exemple pour conclure

Un petit programme simple qui demande l'âge de l'utilisateur et qui le lui affiche ensuite :

## Exemple :

```
int main()
{
    int age = 0; // On initialise la variable a 0

    printf("Quel age avez-vous ? ");
    scanf("%d", &age); // On demande d'entrer l'age avec scanf
    printf("Ah ! Vous avez donc %d ans !\n\n", age);

    return 0;
}
```



# Faire des calculs

## Exemple :

```
int main()
{
    int resultat = 0, nombre1 = 0, nombre2 = 0;

    // On demande les nombres 1 et 2 a l'utilisateur :
    printf("Entrez le nombre 1 : ");
    scanf("%d", &nombre1);
    printf("Entrez le nombre 2 : ");
    scanf("%d", &nombre2);

    // On fait le calcul :
    resultat = nombre1 + nombre2;

    // Et on affiche l'addition a l'ecran :
    printf ("%d + %d = %d\n", nombre1, nombre2, resultat);

    return 0;
}
```