

Variables et premiers calculs

```
Terminal
Arkiv Redigera Visa Terminal Flikar Hjälp
[0:0] [azatoth@azabox linux]$ tar --version
tar (GNU tar) 1.16
Copyright (C) 2006 Free Software Foundation, Inc.
This is free software. You may redistribute copies of it under the terms of
the GNU General Public License <http://www.gnu.org/licenses/gpl.html>.
There is NO WARRANTY, to the extent permitted by law.

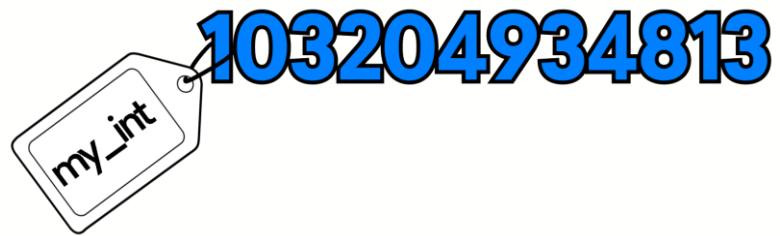
Written by John Gilmore and Jay Fenlason.
[0:0] [azatoth@azabox linux]$ tar -cf kernel.tar kernel/
[0:0] [azatoth@azabox linux]$ tar -zcf kernel.tar.gz kernel/
[0:0] [azatoth@azabox linux]$ tar -jcf kernel.tar.bz2 kernel/
[0:0] [azatoth@azabox linux]$ du -sh kernel
1,9M   kernel
[0:0] [azatoth@azabox linux]$ ll -h kernel.tar*
-rw-r--r-- 1 azatoth azatoth 1,7M 2007-03-20 18:43 kernel.tar
-rw-r--r-- 1 azatoth azatoth 344K 2007-03-20 18:43 kernel.tar.bz2
-rw-r--r-- 1 azatoth azatoth 432K 2007-03-20 18:43 kernel.tar.gz
[0:0] [azatoth@azabox linux]$
```



A diagram showing a variable name and its value. A white tag with a black border and a small circle at the top left corner contains the text "variable name". A line connects the tag to the word "value", which is written in a large, bold, blue font with a black outline.

variable name

value



A diagram showing a specific variable name and its value. A white tag with a black border and a small circle at the top left corner contains the text "my_int". A line connects the tag to the number "103204934813", which is written in a large, bold, blue font with a black outline.

my_int

103204934813

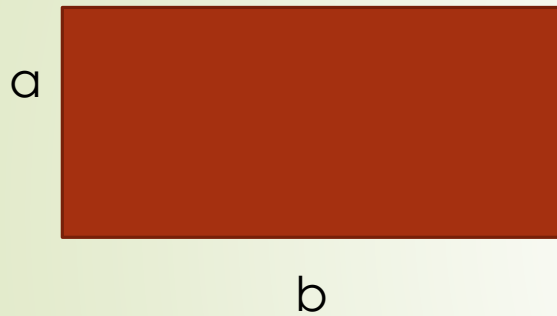


A decorative graphic consisting of several thin, curved lines in shades of grey and brown that converge at a point. A thick, solid red arrow points to the right, positioned horizontally across the middle of the lines.

**Les
variables**

Données dans un programme

Un programme est constitué d'instructions (actions) qui agissent sur des données.



$$\text{Perimetre} = 2 \times a + 2 \times b$$

```
int a; int b; int perimetre;
```

Une **variable** dans un programme permet de stocker la valeur d'une donnée: comme le mot l'indique, cette valeur peut varier (d'une exécution à l'autre ou au cours d'une même exécution).

Une variable est identifiée par un nom.

Au niveau de l'ordinateur, elle est stockée dans un emplacement de la mémoire centrale.


Une variable est typée (est ce un nombre entier, un nombre à virgule, un caractère ?)

Elle doit être déclarée avant son utilisation : on doit indiquer au programme son nom et son type. On peut ensuite consulter sa valeur ou modifier sa valeur.

Les noms des variables en C

- Constitués de lettres et de chiffres et _
- Premier caractère → lettre
- Casse respectée, pas d'accent ni d'espace

Pourcent	
1exemple	
Pourcent_124	
PouRcent_124	
Un exemple	

Choisir des
 noms de
variables
ayant un sens

Les types des variables numériques

int	Entier	int nb;
short	Entier	short nb;
long	Entier	long mtype;
float	Flottant simple précision	float nbf;

Les types caractère

char	caractère	8 bits	char c;
------	-----------	--------	---------

- Une chaîne de caractères est une suite de caractères
- char chaine [100]

Les déclarations de constantes

- Le mot clé `const`
- Initialisation obligatoire
- Par convention, nom en majuscules

```
const float PI = 3.1416;
```

Les déclarations de variables

- Toutes les variables doivent être déclarées

```
int a, b, c;  
char lettre;  
short i = 10;  
float j = 5.2;  
lettre = 't';
```

Il est possible
d'initialiser des
variables au
moment de leur
déclaration

=

Affectation
Mettre une valeur dans
une variable

```
int a = 10;  
int b = 6;
```

Les opérateurs arithmétiques

+	addition	<code>int a = 10; int b = 6; b = a + b;</code>
-	soustraction	<code>b = a - b;</code>
*	multiplication	<code>b = a * b;</code>
/	division	<code>b = a / b;</code>
%	reste de division	<code>b = a % b;</code>

Un exemple

```
main() {  
    int a = 10;  
    int b = 4;  
    int c;  
    const int VAL = 2;  
    c = a + b + VAL;  
    c = a - b + VAL;  
    c = a * b + VAL;  
    c = a / b + VAL;  
    c = a % b + VAL;  
}
```

- Les opérateurs + et - ont même priorité
- Les opérateurs *, / et % sont plus prioritaires

Des exercices

- Écrivez un programme `age.c` qui demande l'âge de l'utilisateur, puis qui l'affiche
- Ecrivez un programme qui demande une somme en francs, la convertit en euros et affiche le résultat
- Soient deux variables de type entier A et B. Vous écrivez un programme qui réalise les opérations suivantes :
 - Demandez à l'utilisateur une valeur et l'affecter à A ;
 - Demandez à l'utilisateur une valeur et l'affecter à B ;
 - Inversez les deux valeurs (A reçoit la valeur contenue dans B et B reçoit la valeur contenue dans A).