





- `double` : réel en double précision (2 mots machines), et sa variante `long double` (3 ou 4 mots machine),
- `void` qui spécifie un ensemble vide de valeurs.

Les *consts* caractères s'écrivent entre quotes simples :

```
'a' 'G' '3' '*' '['
```

Certains caractères de contrôle s'écrivent par des séquences prédéfinies ou par leur code octal ou hexadécimal, comme par exemple :

```
\n \t \r \135 \' \x0FF
```

Les const(s)es entières peuvent s'écrire en notations décimale, hexadécimale (précédées de `0x`)<sup>3</sup>

### **3.3 L'affectation**

= affectation

Il faut noter que le signe =

### 3.7 Expressions conditionnelles

`expr1 ? expr2 : expr3`

est évaluée de la manière suivante :

## 4 Structures d'un programme C++

Contrairement à Java, toutes les fonctions ne sont pas incluses dans une classe en C++. En ce sens, C++ hérite de son prédécesseur une structure modulaire, et on peut très bien concevoir un programme C++ composé d'un grand nombre de *modules*

## 4.2 Structures conditionnelles

La *condition* s'exprime de la manière suivante :

```
if (<expression>) <instruction-1>  
[else <instruction-2>
```





*Les variables*



*Remarque* : on peut écrire cela de manière encore plus efficace en profitant du fait qu'on utilise `p` pour l'incrémenter en même temps. Par ailleurs, une seule boucle suffit, et il est inutile d'utiliser des compteurs :

```
int tab[200][200];
long int moyenne=0;
register int* p = tab;
register int* stop = p + 200 * 200;
for ( ; p < stop ; ) /*on ne fait plus p++ ici*/
```

Mais comme vous risquez d'être parfois confrontés à des chaînes de caractères «à l'ancienne» (c'est-à-dire à la mode C), sachez que ce sont des tableaux de caractères terminés par le caractère nul (de





} ;  
Cette (amitié )









