

# Complément sur les expressions conditionnelles

Ivan Noyer



- Zestedesavoir

# Crédits

- [Zestedesavoir](#)
- « Langage C » (Claude Delannoy) Eyrolles.

# Choix en cascade

Considérons les choix en cascades suivants, dans lesquels les instructions concernées peuvent être des instructions simples (c.a.d suivies d'un point-virgule) ou des blocs :

```
1 if (x > 1) instruction_1
2 else if (x != 0) instruction_2
3 else if (x < 1) instruction_3
4 else instruction_4
```

On peut trouver cette présentation un peu lourde (pas moi). Dans ce cas, le branchement multiple `switch` a ses avantages.

# Branchement multiple

```
1 switch (x)
2   {case constante-1:
3     sequence_d_instructions_1
4     break;
5   case constante-2:
6     sequence_d_instructions_2
7     break;
8     ...
9   case constante-n:
10    sequence_d_instructions_n
11    break;
12  default:
13    sequence_d_instructions_par_defaut
14    break;
15  }
```

Dans ce qui précède `x` est une *expression* de type entier (ou caractère) mais pas un flottant.

# Observations

- Parenthèses autour de `x`.
- `case yyy:`, `case yyy :`, `case yyy :` sont valides. Il faut que `yyy` soit une expression CONSTANTE (par exemple `12` ou une variable déclarée constante).
- Il est conseillé de mettre `break` à la fin d'une liste d'instructions.
- Le cas par défaut n'est pas obligatoire.

# Exemple

```
1 char note;
2
3 printf("Quelle note as-tu obtenue ? "); scanf("%c", &note);
4
5 switch(note){
6     /* Si la note est comprise entre E et C */
7     case 'E': case 'D': case 'C':
8         printf("No comment.\n");
9         break;
10    /* Si la note est comprise entre B et A */
11    case 'B':
12    case 'A':
13        printf("C'est correct voire mieux\n");
14        break;
15
16    default:
17        printf("Euh... note improbable...\n");
18        break;
19 }
```



# Opérateur ternaire

L'*opérateur conditionnel* ou *opérateur ternaire* est un opérateur particulier dont le résultat dépend de la réalisation d'une condition. Son deuxième nom lui vient du fait qu'il est le seul opérateur du langage C à nécessiter 3 opérands : une condition et deux expressions.

```
1 (condition) ? expression si vrai : expression si faux
```

# Exemple

```
1 int heure;  
2  
3 scanf("%d", &heure);  
4  
5 (heure > 8 && heure < 20) ?  
6 printf(" Il fait jour.\n") : printf(" Il fait nuit.\n");  
7
```