

Cours n° : 3 Le If



Objectifs : Les instructions conditionnelles : Le if
 Difficultés : Aucune

Pour ce nouveau cours, nous allons aborder les instructions conditionnelles et plus particulièrement le if.

Les blocs d'instructions

Comme nous allons le voir plus loin dans ce chapitre, il peut-être utile, selon les situations, de placer plusieurs instructions à un endroit précis. Or certaines constructions (instructions), n'accepte qu'une seule sous-instruction (comme, par exemple, pour le if). Pour palier à ce problème, une solution agréable à était utilisée (en fait, héritée de C) : on définit un bloc d'instruction comme étant une instruction.

Au point de vue de la syntaxe, un bloc d'instructions commence par une accolade ouvrante et se termine par une accolade fermante. Entre ces deux caractères, nous y trouvons les sous-instructions, qui sont séparées les unes des autres par des points-virgules. Attention : la dernière instruction d'un bloc doit se terminer, elle aussi, par un point-virgule. Le bloc peut contenir zéro (bloc vide), une (équivalent à une instruction unique) ou plusieurs sous-instructions.

Fichier "tp31.cs"
<pre style="font-family: monospace; font-size: 0.9em;">using System; public class tp31 { static void Main(string[] args) { { // Un nouveau bloc, lui même dans un bloc Console.WriteLine("Un"); Console.WriteLine("Deux"); Console.WriteLine("Trois"); } Console.WriteLine("Exit"); } }</pre>

L'instruction conditionnelle if

Syntaxiquement parlant, l'instruction conditionnelle s'introduit avec le mot clé **if** (attention, majuscules et minuscules sont différentes en C# : IF ou If ou bien encore iF sont des identificateurs différents). Il faut faire suivre cet identificateur d'un test (une expression booléenne) parenthésée, puis d'une instruction. Cette dernière sera exécutée uniquement si la valeur du test est vraie (**true**). Si ce n'est pas le cas, il est possible de rajouter le mot clé **else** suivi d'une instruction qui sera alors lancée. Le tp suivant vous donne la syntaxe complète



```

Fichier "tp32.C#"
using System;

public class tp33
{
    static void Main(string[] args)
    {
        int i ; // on initialise i comme un entier
        Console.WriteLine("saisissez un nombre");
        i=System.Convert.ToInt32(System.Console.ReadLine());
        if (i != 1) // si i est différent de 1 alors
        {
            Console.WriteLine("Vous avez saisi :"+i);
            Environment.Exit(-1); //fin du programme
        }
        else // sinon
        {
            Console.WriteLine("bravo");
        }
    }
}

```

Notez bien que la partie associée à un résultat faux, est facultative. Autre détail important, même si votre test se réduit en la valeur d'une variable déclarée comme booléenne, il est obligatoire de la parenthéser. Rien de supplémentaire est à ajouter sur cette instruction.

Pour vous faire réfléchir exécutez ce programme et analysez le comportement de Console.WriteLine.

```

Fichier "tp33.C#"
using System;

public class tp33
{
    static void Main(string[] args)
    {
        int i ;
        Console.WriteLine("saisissez un nombre");
        i=System.Convert.ToInt32(System.Console.ReadLine());
        Console.WriteLine(i!=1 ? "Vous avez saisi :"+i: "bravo");
    }
}

```

Vous avez remarqué ? Vous pouvez revenir sur le tp 2 pour voir la caractéristique du "?" (appelé opérateur ternaire). C'est élégant non ? Maintenant vous devenez un optimisateur !



Exercices applicatifs

Exercice 1 :

Concevez le programme correspondant à l'algorithme suivant :

```

algo. RESULTAT
début
    écrire(clavier) " Veuillez saisir la note"
    lire(clavier) la note
    si note >= 10
        écrire(écran) "Vous êtes admis(e)"
        si note < 12
            écrire(écran) "Votre mention est assez bien".
        sinon
            si note < 14
                écrire(écran) "Votre mention est bien".
            sinon
                si note < 16
                    écrire(écran) "Votre mention est très bien".
                finsi
            finsi
        finsi
    sinon
        si note >= 8
            écrire(écran) "Vous êtes sur liste d'attente"
        sinon
            écrire (écran) "Vous êtes refusé(e)"
        finsi
    finsi
fin
  
```

Exercice 2 :

L'utilisateur donne deux valeurs entières distinctes. Ecrire un algorithme qui calcule dans une variable la valeur du plus grand de ces deux nombres puis l'affiche.

Note : on peut écrire deux variantes :

- une avec **si...alors...sinon...**
- une avec **si...alors...**

Modifier l'algorithme pour calculer le maximum de 4 nombres distincts saisis par l'utilisateur.

Une fois les 3 algorithmes réalisés, programmez les en C#.

