# TP : Bonjour c'est le week-end ...

Le but est de créer une petite application qui affiche un message différent en fonction du nom de l’utilisateur et du moment de la journée :

* Bonjour XXX pour la tranche horaire 9h <-> 18h, les lundi, mardi, mercredi, jeudi et vendredi
* Bonsoir XXX pour la tranche horaire 18h <-> 9h, les lundi, mardi, mercredi, jeudi
* Bon week-end XXX pour la tranche horaire vendredi 18h <-> lundi 9h

Peut-être cela vous parait simple, dans ce cas, foncez et réalisez cette première application tout seul :) .  
Sinon, décortiquons un peu l’énoncé de ce TP pour éviter d’être perdu.  
  
Pour réaliser ce premier TP, vous allez avoir besoin de plusieurs choses.   
Dans un premier temps, il faut afficher le nom de l’utilisateur, c’est une chose que nous avons déjà faite en allant puiser dans les fonctionnalités du framework .NET.  
  
Vous aurez besoin également de récupérer l’heure courante pour la comparer aux tranches horaires souhaitées. Vous avez déjà vu comment récupérer la date courante. Pour pouvoir récupérer l’heure de la date courante, il vous faudra utiliser l’instruction DateTime.Now.Hour qui renvoie un entier représentant l’heure du jour.  
  
Pour comparer l’heure avec des valeurs entières il vous faudra utiliser les opérateurs de comparaisons et les instructions conditionnelles que nous avons vus précédemment.   
  
Pour traiter le cas spécial du jour de la semaine, vous aurez besoin que le framework .NET vous indique quel jour nous sommes. C’est le rôle de l’instruction DateTime.Now.DayOfWeek qui est une énumération indiquant le jour de la semaine. [Les différentes valeurs sont consultables à cette adresse](http://msdn.microsoft.com/fr-fr/library/system.dayofweek.aspx).  
  
Pour plus de clarté, nous les reprenons ici :

| **Valeur** | **Traduction** |
| --- | --- |
| Sunday | Dimanche |
| Monday | Lundi |
| Tuesday | Mardi |
| Wednesday | Mercredi |
| Thursday | Jeudi |
| Friday | Vendredi |
| Saturday | Samedi |

Il ne restera plus qu’à comparer deux valeurs d’énumération, comme on l’a vu au chapitre sur les énumérations.  
  
Voilà, vous avez tous les outils en main pour réaliser ce premier TP. N’hésitez pas à revenir sur les chapitres précédents si vous avez un doute sur la syntaxe ou sur les instructions à réaliser. On ne peut pas apprendre un langage par cœur du premier coup.  
  
À vous de jouer !

**Correction**

Vous êtes autorisés à lire cette correction uniquement si vous vous êtes arraché les cheveux sur ce TP ! Je vois qu'il vous en reste, encore un effort ! :lol:   
Quoique, si vous avez réussi avec brio le TP, vous pourrez également comparer votre travail au mien.  
  
Quoi qu’il en soit, voici la correction que je propose. Bien évidemment, il peut y en avoir plusieurs, mais elle contient les informations nécessaires pour la réalisation de ce TP.  
  
Première chose à faire : créer un projet de type console. J’ai ensuite ajouté le code suivant :  
  
  
**Code : C# -**[**Sélectionner**](http://www.siteduzero.com/tutoriel-3-545059-tp-bonjour-c-est-le-week-end.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55 | **static** **void** Main(string[] args)  {  **if** (DateTime.Now.DayOfWeek == DayOfWeek.Saturday || DateTime.Now.DayOfWeek == DayOfWeek.Sunday)  {  *// nous sommes le week-end*  AfficherBonWeekEnd();  }  **else**  {  *// nous sommes en semaine*  **if** (DateTime.Now.DayOfWeek == DayOfWeek.Monday && DateTime.Now.Hour < 9)  {  *// nous sommes le lundi matin*  AfficherBonWeekEnd();  }  **else**  {  **if** (DateTime.Now.Hour >= 9 && DateTime.Now.Hour <= 18)  {  *// nous sommes dans la journée*  AfficherBonjour();  }  **else**  {  *// nous sommes en soirée*  **if** (DateTime.Now.DayOfWeek == DayOfWeek.Friday && DateTime.Now.Hour > 18)  {  *// nous sommes le vendredi soir*  AfficherBonWeekEnd();  }  **else**  {  AfficherBonsoir();  }  }  }  }  }  **static** **void** AfficherBonWeekEnd()  {  Console.WriteLine("Bon week-end " + Environment.UserName);  }  **static** **void** AfficherBonjour()  {  Console.WriteLine("Bonjour " + Environment.UserName);  }  **static** **void** AfficherBonsoir()  {  Console.WriteLine("Bonsoir " + Environment.UserName);  } |

Détaillons un peu ce code :   
  
Au chargement du programme (méthode Main) nous faisons les comparaisons adéquates.  
Dans un premier temps, nous testons le jour de la semaine de la date courante (DateTime.Now.DayOfWeek) et nous la comparons aux valeurs représentant Samedi et Dimanche. Si c’est le cas, alors nous appelons une méthode qui affiche le message « Bon week-end » avec le nom de l’utilisateur courant que nous pouvons récupérer avecEnrivonment.UserName.  
  
Si nous ne sommes pas le week-end, alors nous testons l’heure de la date courante avec DateTime.Now.Hour. Si nous sommes le lundi matin avant 9h, alors nous continuons à afficher « Bon week-end ». Sinon, si nous sommes dans la tranche horaire 9h – 18h alors nous pouvons appeler la méthode qui affiche Bonjour. Si non, il reste juste à vérifier que nous ne sommes pas vendredi soir, qui fait partie du week-end, sinon on peut afficher le message de « Bonsoir ».  
  
Et voilà, un bon exercice pour manipuler les conditions et les énumérations…

**Aller plus loin**

Ce TP n’était pas très compliqué, il nous a permis de vérifier que nous avions bien compris le principe des if et que nous sachions appeler des éléments du framework .NET.  
  
Nous aurions pu simplifier l’écriture de l’application en compliquant en peu les tests avec une combinaison de conditions.   
Par exemple, on pourrait avoir :  
  
**Code : C# -**[**Sélectionner**](http://www.siteduzero.com/tutoriel-3-545059-tp-bonjour-c-est-le-week-end.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | **if** (DateTime.Now.DayOfWeek == DayOfWeek.Saturday ||  DateTime.Now.DayOfWeek == DayOfWeek.Sunday ||  (DateTime.Now.DayOfWeek == DayOfWeek.Monday && DateTime.Now.Hour < 9) ||  (DateTime.Now.DayOfWeek == DayOfWeek.Friday && DateTime.Now.Hour > 18))  {  *// nous sommes le week-end*  AfficherBonWeekEnd();  }  **else**  {  *// nous sommes en semaine*  **if** (DateTime.Now.Hour >= 9 && DateTime.Now.Hour <= 18)  {  *// nous sommes dans la journée*  AfficherBonjour();  }  **else**  {  AfficherBonsoir();  }  } |

Le premier test permet de vérifier que nous sommes soit samedi, soit dimanche, soit que nous sommes lundi et que l’heure est inférieure à 9, soit que nous sommes vendredi et que l’heure est supérieure à 18.  
Nous avons, pour ce faire, combiné les tests avec l’opérateur logique OU : ||. Remarquons que les parenthèses nous permettent d’agir sur l’ordre d’évaluation des conditions. Pour que ce soit le week-end, il faut bien sûr être « vendredi et que l’heure soit supérieure à 18 » ou « lundi et que l’heure soit inférieure à 9 » ou samedi ou dimanche.  
  
On pourrait encore simplifier en évitant de solliciter à chaque fois le framework .NET pour obtenir la date courante. Il suffit de stocker la date courante dans une variable de type DateTime. Ce qui donnerait :  
  
**Code : C# -**[**Sélectionner**](http://www.siteduzero.com/tutoriel-3-545059-tp-bonjour-c-est-le-week-end.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | DateTime dateCourante = DateTime.Now;  **if** (dateCourante.DayOfWeek == DayOfWeek.Saturday ||  dateCourante.DayOfWeek == DayOfWeek.Sunday ||  (dateCourante.DayOfWeek == DayOfWeek.Monday && dateCourante.Hour < 9) ||  (dateCourante.DayOfWeek == DayOfWeek.Friday && dateCourante.Hour > 18))  {  *// nous sommes le week-end*  AfficherBonWeekEnd();  }  **else**  {  *// nous sommes en semaine*  **if** (dateCourante.Hour >= 9 && dateCourante.Hour <= 18)  {  *// nous sommes dans la journée*  AfficherBonjour();  }  **else**  {  AfficherBonsoir();  }  } |

On utilise ici le type DateTime comme le type string ou int. Il sert à gérer les dates et l’heure. Il est légèrement différent des types que nous avons vus pour l’instant, nous ne nous attarderons pas dessus. Nous aurons l’occasion de découvrir de quoi il s’agit dans la partie suivante.  
Cette optimisation n’a rien d’extraordinaire, mais cela nous évite un appel à chaque fois au framework .NET.

Voilà pour ce TP. J’espère que vous aurez réussi avec brio l’exercice.   
  
Vous avez pu remarquer que ce TP n'était pas trop difficile. Il a simplement fallu réfléchir à comment imbriquer correctement nos conditions.  
  
N'hésitez pas à pratiquer et à vous entraîner avec d'autres problèmes de votre cru. Si vous avez le moindre problème, vous pouvez relire les chapitres précédents.  
  
Vous verrez que nous aurons l'occasion d'énormément utiliser ces instructions conditionnelles dans tous les programmes que nous allons écrire.