

1. Présentation

Un objet DateTime peut contenir une date comprise entre le premier janvier de l'an un à midi et le 31 décembre 9999. il comprend à la fois une date et une heure avec accès à la milliseconde.

Tous les opérateurs (==, !=, <, <=, >, >=, +, -, +=, -=) peuvent être utilisés entre les dates mais on ne peut pas additionner une date et un entier. Il faut dans ce cas utiliser la méthode AddDays(n).

Attention lors de la comparaison de deux objets DateTime, la comparaison s'effectue sur l'ensemble des éléments : année, mois, jour, heure, minute, seconde et milliseconde.

2. Les membres de la classe

Constructeur	Rôle
DateTime(annee, mois, jour)	Initialise une nouvelle instance de la classe avec l'année, le mois et le jour spécifiés.
DateTime(a,m,j,h,mn,s)	Initialise une nouvelle instance de la classe avec l'année, le mois, le jour, les heures, minutes et secondes spécifiés.

Méthode (uneDate.methode)	Type	Rôle
AddDays(double n)	DateTime	Retourne la date dans n jours (pas de modification de l'objet)
AddMonths(double n)	DateTime	Date dans n mois
AddYears(double n)	DateTime	Date dans n années
AddHours(double n)	DateTime	Date dans n heures
AddMinutes (double n)	DateTime	Date dans n minutes
AddSeconds (double n)	DateTime	Date dans n secondes
ToString(string f)	string	Met en format une date
ToShortDateString()	string	Date au format court : jj/mm/aaaa
ToLongDateString()	string	Date au format long : jjjj jj mmm aaaa
ToShortTimeString()	string	Heure au format hh:mm
ToLongTimeString	string	Heure au format hh:mm:ss
CompareTo(DateTime d)	int	Compare la valeur de cette instance à une valeur DateTime spécifiée et retourne un entier qui indique si cette instance est antérieure, identique ou ultérieure à la valeur DateTime spécifiée.

Méthode statique (DateTime.)	Type	Rôle
DaysInMonth(int annee, int mois)	int	nombre de jour du mois
Parse(string s)	DateTime	Retourne une date correspondant à la chaîne s (jj/mm/aaaa)
IsLeapYear(int annee)	bool	vrai si l'année est bissextile
Compare(DateTime, DateTime)	int	Compare deux instances de DateTime et retourne un entier qui indique si la première instance est antérieure, identique ou ultérieure à la deuxième instance.

Propriété (uneDate.Propriété)	Type	Rôle
Date	DateTime	date (sans l'heure) permet de comparer deux dates
Day	Int	Jour de la date
DayOfWeek	DayOfWeek	Jour de la semaine DayOfWeek.Saturday pour un samedi
DayOfYear	Int	Jour de l'année (1 à 366)
Month	Int	Mois
Year	Int	Année
Hour	Int	Heure
Minute	Int	Minute
Second	int	Seconde
Millisecond	Int	Milliseconde
TimeOfDay	TimeSpan	Durée depuis minuit

Propriété statique (DateTime.)	Type	Rôle
Now	DateTime	Date et heure du jour
Today	DateTime	Date du jour (sans l'heure)

3. Les formats de date utilisable dans la méthode ToString

Code	Valeur
d	14/09/2017
D	Lundi 18 septembre 2017
f	Lundi 18 septembre 2017 13:10
F	Lundi 18 septembre 2017 13:10:45
g	18/9/2017 13:10
M	14 septembre
t	13:10
Y	Septembre 2017
:	Séparateur des heures
/	Séparateur de date
d	Jour, entre 1 et 31
dd	Jour, entre 01 et 31
ddd	abréviation du jour (lun pour lundi)
dddd	nom complet du jour
M	Mois, de 1 à 12 (déclinaison identique à d mais en majuscule)
y	Année de 1 à 99
yyyy	Année de 1 à 9999
h ou hh ou H ou HH	Heure, de 0 à 11 ou 00 à 11 ou 0 à 23 ou 00 à 23
m ou mm	Minute, de 0 à 59 ou de 00 à 59
s ou ss	Seconde, de 0 à 59 ou de 00 à 59

On peut aussi utiliser tout autre caractère dans le format. Pour éviter qu'un caractère soit interprété, il faut le placer entre apostrophe (il est donc impossible de placer une apostrophe dans le format)

Exemple :

```
DateTime dt = DateTime.Now;
string s = dt.ToString("'Bonjour, nous sommes le ' dddd dd MMMM yyyy, 'il est actuellement'
hh 'heures et ' mm 'minutes'");
```

4. Exemple d'utilisation

Initialisation d'une date à partir de ses trois paramètres
DateTime uneDate = new DateTime(2020,03,07)
Ajouter une journée à une date
uneDate = uneDate.AddDays(1)
Tester si la date est un dimanche
If (uneDate.DayOfWeek == DayOfWeek.Sunday)
Récupérer le nombre de jour entre deux dates (2 solutions)
int d = (int) (fin.Date - debut.Date).TotalDays; int res = (fin - debut).Days;
Initialisation d'une date au premier janvier de l'année
DateTime uneDate = new DateTime(DateTime.Now.Year,1,1)
Mise en forme d'un objet TimeSpan : leTemps
string elapsedTime = String.Format("{0:00}:{1:00}:{2:00}.{3:00}", leTemps.Hours, leTemps.Minutes, leTemps.Seconds, leTemps.Milliseconds);

5. Calculer le temps d'exécution d'un programme

Il n'est pas recommandé d'utiliser la classe DateTime pour cela car la précision de sa propriété Now est trop faible.

Il faut utiliser la classe Stopwatch présente dans la bibliothèque System.Diagnostics

```
Stopwatch stopWatch = new Stopwatch();
stopWatch.Start();
// placer ici le traitement
stopWatch.Stop();

// Récupération du temps écoulé
TimeSpan leTemps = stopWatch.Elapsed;

// Affichage sous la forme Mise en forme du temps affiché hh:mm:ss.000
Console.WriteLine("Temps écoulé : {0:00}:{1:00}:{2:00}.{3:00}", leTemps.Hours,
leTemps.Minutes, leTemps.Seconds, leTemps.Milliseconds );
```

6. Webographie

[http://msdn.microsoft.com/fr-fr/library/System.DateTime\(v=vs.110\).aspx](http://msdn.microsoft.com/fr-fr/library/System.DateTime(v=vs.110).aspx)
[https://msdn.microsoft.com/fr-fr/library/system.diagnostics.stopwatch\(v=vs.110\).aspx](https://msdn.microsoft.com/fr-fr/library/system.diagnostics.stopwatch(v=vs.110).aspx)
<http://www.csharp-examples.net/string-format-datetime/>