DateTime   t1   =   new   DateTime(100);     
  DateTime   t2   =   new   DateTime(20);     
      
  if   (DateTime.Compare(t1,   t2)   >     0)   Console.WriteLine("t1   >   t2");       
  if   (DateTime.Compare(t1,   t2)   ==   0)   Console.WriteLine("t1   ==   t2");       
  if   (DateTime.Compare(t1,   t2)   <     0)   Console.WriteLine("t1   <   t2");   

1。比较时间大小的实验

string st1="12:13";

string st2="14:14";

DateTime dt1=Convert.ToDateTime(st1);

DateTime dt2=Convert.ToDateTime(st2);

DateTime dt3=DateTime.Now;

if(DateTime.Compare(dt1,dt2)>0)

msg.Text=st1+">"+st2;

else

msg.Text=st1+"<"+st2;

msg.Text+="\r\n"+dt1.ToString();

if(DateTime.Compare(dt1,dt3)>0)

msg.Text+="\r\n"+st1+">"+dt3.ToString();

else

msg.Text+="\r\n"+st1+"<"+dt3.ToString();

2。计算两个时间差值的函数，返回时间差的绝对值：

private string DateDiff(DateTime DateTime1,DateTime DateTime2)

{

string dateDiff=null;

try

{

TimeSpan ts1=new TimeSpan(DateTime1.Ticks);

TimeSpan ts2=new TimeSpan(DateTime2.Ticks);

TimeSpan ts=ts1.Subtract(ts2).Duration();

dateDiff=ts.Days.ToString()+"天"

+ts.Hours.ToString()+"小时"

+ts.Minutes.ToString()+"分钟"

+ts.Seconds.ToString()+"秒";

}

catch

{

}

return dateDiff;

}

3。实现计算DateTime1－36天＝DateTime2的功能

TimeSpan ts=new TimeSpan(40,0,0,0);

DateTime dt2=DateTime.Now.Subtract(ts);

msg.Text=DateTime.Now.ToString()+"-"+ts.Days.ToString()+"天\r\n";

msg.Text+=dt2.ToString();

use "DateTime.Compare" static method  
DateTime.Compare( dt1, dt2 ) > 0 : dt1 > dt2  
DateTime.Compare( dt1, dt2 ) == 0 : dt1 == dt2  
DateTime.Compare( dt1, dt2 ) < 0 : dt1 < dt2  
        /// <summary>  
         /// 计算两个日期的时间间隔  
         /// </summary>  
         /// <param name="DateTime1">第一个日期和时间</param>  
         /// <param name="DateTime2">第二个日期和时间</param>  
         /// <returns></returns>  
         private string DateDiff(DateTime DateTime1, DateTime DateTime2)  
         {  
             string dateDiff = null;  
             
             TimeSpan ts1 = new TimeSpan(DateTime1.Ticks);  
             TimeSpan ts2 = new TimeSpan(DateTime2.Ticks);  
             TimeSpan ts = ts1.Subtract(ts2).Duration();  
             dateDiff = ts.Days.ToString()+"天"  
                 + ts.Hours.ToString()+"小时"  
                 + ts.Minutes.ToString()+"分钟"  
                 + ts.Seconds.ToString()+"秒";  
             
             return dateDiff;  
         } 说明：  
1.DateTime值类型代表了一个从公元0001年1月1日0点0分0秒到公元9999年12月31日23点59分59秒之间的具体日期时刻。因此，你可以用DateTime值类型来描述任何在想象范围之内的时间。一个DateTime值代表了一个具体的时刻  
2.TimeSpan值包含了许多属性与方法，用于访问或处理一个TimeSpan值  
下面的列表涵盖了其中的一部分：  
Add：与另一个TimeSpan值相加。  
Days:返回用天数计算的TimeSpan值。  
Duration:获取TimeSpan的绝对值。  
Hours:返回用小时计算的TimeSpan值  
Milliseconds:返回用毫秒计算的TimeSpan值。  
Minutes:返回用分钟计算的TimeSpan值。  
Negate:返回当前实例的相反数。  
Seconds:返回用秒计算的TimeSpan值。  
Subtract:从中减去另一个TimeSpan值。  
Ticks:返回TimeSpan值的tick数。  
TotalDays:返回TimeSpan值表示的天数。  
TotalHours:返回TimeSpan值表示的小时数。  
TotalMilliseconds:返回TimeSpan值表示的毫秒数。  
TotalMinutes:返回TimeSpan值表示的分钟数。  
TotalSeconds:返回TimeSpan值表示的秒数。

====================================================<BR>int jg=72;//设置一个增加的时间  
    DateTime dt=Convert.ToDateTime("2006-4-23 12:22:05");// 设置一个初始化的时间  
    DateTime newdt=dt.AddHours(jg);//初始化时间加上增加的时间  
    DateTime nowt=DateTime.Now;//现在的时间  
    Response.Write("现在时间是："+nowt+"<br>");  
    Response.Write("数据库时间是："+dt+"<br>");  
    Response.Write("新的时间是："+newdt+"<br>");  
    if(newdt<nowt)//如果相加后的时间大于现在的时间  
    {  
     Response.Write("可以");  
    }  
    else//否则  
    {  
     Response.Write("不行");  
    }

======================================================  
// 得到某年某月的天数  
   public static int GetDaysInMonth(int rYear,int rMonth)  
   {  
    DateTime dt1 = DateTime.Parse(rYear+"-"+rMonth+"-01");  
    DateTime dt2 = dt1.AddMonths(1);  
    TimeSpan ts = dt2-dt1;  
    return (int)ts.TotalDays;  
   }

//得到星期Text  
   public static string GetDayOfWeekHtml(int rDayOfWeek)  
   {  
    switch(rDayOfWeek)  
    {  
     case (int)DayOfWeek.Sunday:  
      return "<font color=\"#ff0000\">星期日</font>";  
     case (int)DayOfWeek.Monday:  
      return "<font color=\"#000000\">星期一</font>";  
     case (int)DayOfWeek.Tuesday:  
      return "<font color=\"#000000\">星期二</font>";  
     case (int)DayOfWeek.Wednesday:  
      return "<font color=\"#000000\">星期三</font>";  
     case (int)DayOfWeek.Thursday:  
      return "<font color=\"#000000\">星期四</font>";  
     case (int)DayOfWeek.Friday:  
      return "<font color=\"#000000\">星期五</font>";  
     case (int)DayOfWeek.Saturday:  
      return "<font color=\"#008800\">星期六</font>";  
     default:  
      return "";  
    }  
   }

//   得到某年某月的起止日期，格式为0000-00-00  
   public static string[] GetBeginEndDate(int rYear,int rMonth)  
   {  
    string[] arr = new string[2];  
    DateTime dt1 = DateTime.Parse(rYear+"-"+rMonth+"-01");  
    arr[0] = dt1.ToLongDateString();  
    DateTime dt2 = dt1.AddMonths(1).AddDays(-1);  
    arr[1] = dt2.ToLongDateString();  
    return arr;  
   }

==========================  
在ASP中日期比较使用：DateDiff( "d", "2006-1-30", now )>0;在C#中使用:TimeSpan

代码如下：  
using System;  
using System.Collections;

public class DatediffClass  
{  
public static void Main()  
{  
   DateTime dt1 = DateTime.Parse("2006-04-01");  
   DateTime dt2 = DateTime.Parse("2006-05-01");

   TimeSpan ts = dt2.Subtract(dt1);  
   Console.WriteLine(ts.TotalDays);  
   Console.ReadLine();  
}

}

如果是比较大小：DateTime.Compare(t1, t2) >   0

===================================  
1、DateTime 数字型  
System.DateTime currentTime=new System.DateTime();  
1.1 取当前年月日时分秒  
currentTime=System.DateTime.Now;  
1.2 取当前年  
int 年=currentTime.Year;  
1.3 取当前月  
int 月=currentTime.Month;  
1.4 取当前日  
int 日=currentTime.Day;  
1.5 取当前时  
int 时=currentTime.Hour;  
1.6 取当前分  
int 分=currentTime.Minute;  
1.7 取当前秒  
int 秒=currentTime.Second;  
1.8 取当前毫秒  
int 毫秒=currentTime.Millisecond;

s