

# C# Essentials: Veelgebruikte classes

Lector: Tom Quareme

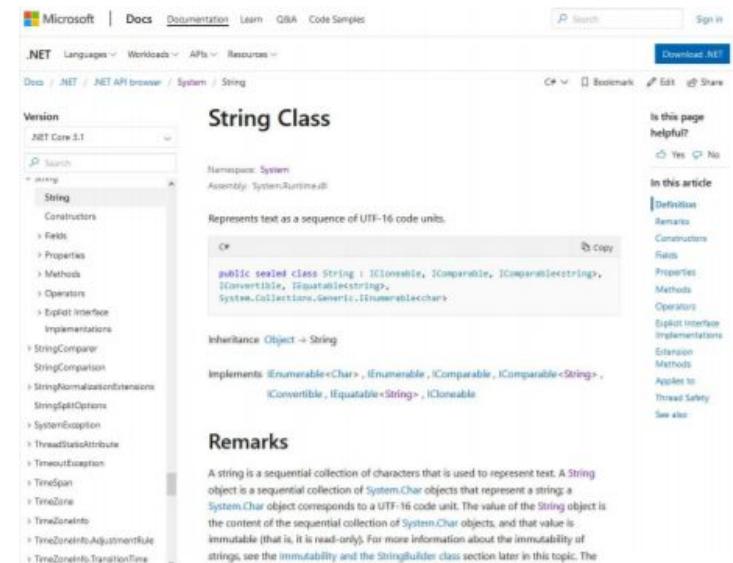
# Overzicht ingebouwde classes C#

- System.Math
- System.String
- System.Text.StringBuilder
- System.Random
- System.DateTime
- System.TimeSpan
- System.Windows.Threading.DispatcherTimer
- System.Windows.Forms.MessageBox

# Tip: zelf documentatie opzoeken tijdens coderen

- Meer uitleg bij ingebouwde methods krijgen:
  - CTRL+Klik op de method binnen Visual Studio.
  - Of class opzoeken op: <https://docs.microsoft.com>

```
//  
// Summary:  
//     Returns the logarithm of a specified number in a specified base.  
//  
// Parameters:  
//     a:  
//         The number whose logarithm is to be found.  
//  
//     newBase:  
//         The base of the logarithm.  
//  
// Returns:  
//     One of the values in the following table. (+Infinity denotes System.Double.PositiveInfinity,  
//     -Infinity denotes System.Double.NegativeInfinity, and NaN denotes System.Double.NaN.)  
//  
//     anewBase Return value a > 0 (0 <newBase< 1) lognewBase(a) a<  
//     0 (any value) NaN (any value) newBase< 0 NaN a != 1 newBase = 0 NaN a != 1 newBase  
//     = +Infinity NaN a = NaN (any value) NaN (any value) newBase = NaN NaN (any value)  
//     newBase = 1 NaN a = 0 0 <newBase< 1 +Infinity a = 0 newBase> 1 -Infinity a =  
//     +Infinity 0 <newBase< 1 -Infinity a = +Infinity newBase> 1 +Infinity a = 1 newBase  
//     = 0 0 a = 1 newBase = +Infinity 0  
public static double Log(double a, double newBase);
```



# System.Math (namespace: using System;)

- Constantes:
  - Math.PI  $\Rightarrow$  3.14159265358979 (double)
  - Math.E  $\Rightarrow$  2.71828182845905 (double)
- Functies::
  - Absolute waarde:
    - Math.Abs(-12.34)  $\Rightarrow$  12.34
  - Minimum en maximum (kleinste en grootste getal bepalen):
    - Math.Min(-12.34, 10)  $\Rightarrow$  -12.34
    - Math.Max(-12.34, 10)  $\Rightarrow$  10
  - Afronden naar dichtsbijzijnde **even!!!** getal:
    - Math.Round(10.5)  $\Rightarrow$  10, niet 11 want 10 is even!!!
    - Math.Round(10.51)  $\Rightarrow$  11, 11 is niet even maar ligt er het dichtst bij!
    - Math.Round(11.5)  $\Rightarrow$  12
    - Math.Round(-12.51)  $\Rightarrow$  -13
    - Math.Round(-12.3456, 2)  $\Rightarrow$  -12.35

# System.Math

- Functies::
  - Afronden naar boven:
    - `Math.Ceiling(-11.5) ⇒ -11`
    - `Math.Ceiling(11.5) ⇒ 12`
  - Afronden naar beneden:
    - `Math.Floor(-11.5) ⇒ -12`
    - `Math.Floor(11.5) ⇒ 11`
  - Machtsverheffing:
    - `Math.Pow(2, 3) ⇒ 8 (2*2*2)`
  - Logaritme:
    - `Math.Log(8, 2) ⇒ 3` (tot de hoeveelste macht 2 nemen om 8 te krijgen)
  - Vierkantwortel:
    - `Math.Sqrt(16) ⇒ 4`

# System.String (namespace: using System;)

- string str = "Visual CSharp";
- **Is een string leeg?**
  - if (str.Length == 0) {}
  - (NIET DOEN: str == "")
- **Strings vergelijken;**
  - str.Equals("Visual CSharp");
  - (NIET DOEN: str == "Visual CSharp")

In tegenstelling tot de '=='-operator geeft Equals een foutmelding als een string null is.  
In een later hoofdstuk zien we hoe we die foutmelding opvangen.

# System.String (namespace: using System;)

- Functies:

<code>string.Concat("pannen", "koek");</code> ⇒ "pannenkoek"	strings samenvoegen
<code>string.Compare("aa", "AA");</code> <code>string.Compare("aa", "AA", true);</code>	Strings vergelijken 0 ⇒ gelijk -1 ⇒ linkse kleiner dan rechtse 1 ⇒ linkse groter dan rechtse True negeert hoofdletters!
<code>string.CompareOrdinal("A", "a");</code>	Vergelijk ASCII-code ( $65 < 97 \Rightarrow -1$ )
<code>string.Equals("abcd", "Abcd");</code>	Zijn strings gelijk? (true/false)
<code>string.Equals("A", "a", StringComparison.OrdinalIgnoreCase);</code>	Zijn strings gelijk? (true/false) Negeer hoofdletters

# System.String

- Methods:

"Visual CSharp".Substring(2);	Substring uit string halen vanaf 2de teken (start vanaf 0!!!)
"Visual CSharp".Substring(2,5);	Substring uit string halen vanaf 2de teken tot en zo 5 tekens lang (start vanaf 0!!!)
"Visual CSharp".PadLeft(20); "Visual CSharp".PadRight(20);	Vul string aan langs links/rechts met spaties tot string 20 chars is
"Visual CSharp".PadLeft(20,'*'); "Visual CSharp".PadRight(20, '*');	Vul string aan langs links/rechts met * tot string 20 chars is

# System.String

- Methods/properties:

"Visual CSharp".Length; ⇒ 13	Lengte van string (property)
"Visual CSharp".Contains("CSharp"); ⇒ true "Visual CSharp".Contains("C#"); ⇒ false	Kijk of string substring bevat (true/false)
"Visual CSharp".IndexOf("blabla"); ⇒ -1 "Visual CSharp".IndexOf("CSh"); ⇒ 7	Geef positie vanaf 0 vanaf waar de substring in de string zit. -1 indien niet gevonden.
" Visual CSharp ".Trim(); ⇒ "Visual CSharp"	Verwijder langs links en rechts whitespace (spaties, newlines,...)
" Visual CSharp ".TrimEnd();	Enkel aan het einde trimmen
" Visual CSharp ".TrimStart();	Enkel aan het begin trimmen

# System.String

"Visual CSharp".Remove(4); $\Rightarrow$ "Visu"	Verwijder <u>vanaf</u> het 4de karakter (tellend vanaf 0)
"Visual CSharp".Remove(4, 3); $\Rightarrow$ "VisuCSharp"	Verwijder 3 karakters <u>vanaf</u> het 4de karakter (tellend vanaf 0)
"Visual C#".StartsWith("Visual") $\Rightarrow$ true	Start de string met een bepaalde substring? (true/false)
"Visual CSharp".Insert(2, "YO"); $\Rightarrow$ "ViYOusual CSharp";	Insert een string "YO" vanaf het 2de karakter (tellend vanaf 0)
"Visual CSharp".Replace("a", "o");	Vervang elke a door o in string
"Visual CSharp".ToUpper();	String in hoofdletters
"Visual CSharp".ToLower();	String in kleine letters

# System.String

- **string.Join(", ", mijnArray):** voeg arrayelementen samen tot string gescheiden door ,
- **mijnStringVariabele.Split(','):** splits de string van de variabele in stukken strings volgens de ,

```
string[] element = {"PCVO Limburg", "Campus PXL Hasselt", "Elfde-Liniestraat 26", "3500 Hasselt"};  
  
Console.WriteLine("\n ===== Afdruk lus === ");  
for (int i = 0; i < element.Length; i++)  
{  
    Console.Write(element[i] + ", ");  
}  
  
Console.WriteLine("\n\n\n ===== Afdruk string.Join() === ");  
  
Console.WriteLine(string.Join(", ", element));  
  
Console.WriteLine("\n\n ===== Afdruk string.Split() ===");  
string adres = "PCVO Limburg,Campus PXL Hasselt,Elfde-Liniestraat 26,3500 Hasselt";  
string[] delen = adres.Split(',');  
  
for (int i = 0; i < delen.Length; i++)  
{  
    Console.WriteLine($"Element {i}: {delen[i]}");  
}
```

# System.Text.StringBuilder (using System.Text;)

- StringBuilder gebruiken is performanter dan concatten (+ of +=) op strings.
- Voeg bovenaan in .cs file de namespace toe:
  - **using System.Text;**

- Gebruik:

```
StringBuilder sb = new StringBuilder(); // start van lege string
sb.Append("Nieuwe tekst:\r\n");
sb.AppendLine("Punt:").Append(" 10").AppendLine();
string str = sb.ToString(); // op het einde omvormen naar een string voor gebruik
```

- Append():
  - nieuwe tekst toevoegen
- AppendLine():
  - nieuwe tekst toevoegen en op het einde nieuwe regel starten

# System.Text.StringBuilder

- Andere functies: (probeer eens uit)

```
sb.Insert(0, "StringBuilder:").Insert(14,Environment.NewLine); // vooraan 14 kar toevoegen  
sb.Replace(':', '-',15,sb.Length-15); // vanaf positie 15 ':' door '-' vervangen  
sb.AppendFormat("{0}, {1}", i, j); // i en j worden onderaan toegevoegd  
Console.WriteLine(sb);  
sb.Clear(); //lengte = 0, dus leeg  
sb.Remove(0,14); // eerste 14 karakters (Stringbuilder:) verwijderd
```

# System.Random

- Random (willekeurig) getal genereren:
  - Maak eerst een instantie aan van Random class die ons een stroom aan getallen levert.
  - Lege constructor Random() geeft telkens nieuwe getallen op basis van systeemtijd:
    - `private Random rnd = new Random(); // verschillende stroom telkens`
  - Seed meegeven aan constructor levert dezelfde stroom getallen:
    - `private Random rnd = new Random(100); // dezelfde stroom telkens`
- Genereer daarna nieuw getallen:
  - `int getal = rnd.Next(1, 7); // genereer nieuw getal van 1 t.e.m. 6 (7 is EXCLUSIEF)`
  - `double getal = rnd.NextDouble(); // getal >= 0.0 en getal < 1.0`

# System.Windows.MessageBox

- MessageBox: berichtenvenster om een mededeling te tonen aan gebruiker.
  - **using System.Windows; // ENKEL in WPF, niet in console applicatie!!!**

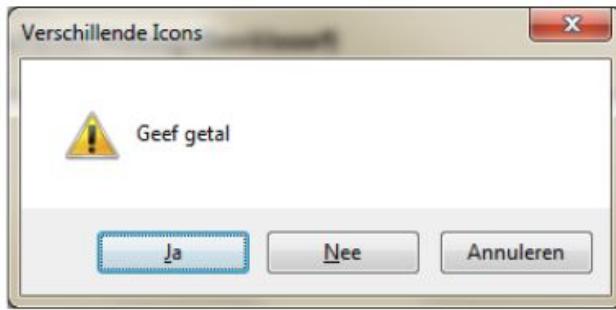


```
// Standaard MessageBox
MessageBox.Show("Geef een getal in", "Standaard");
```

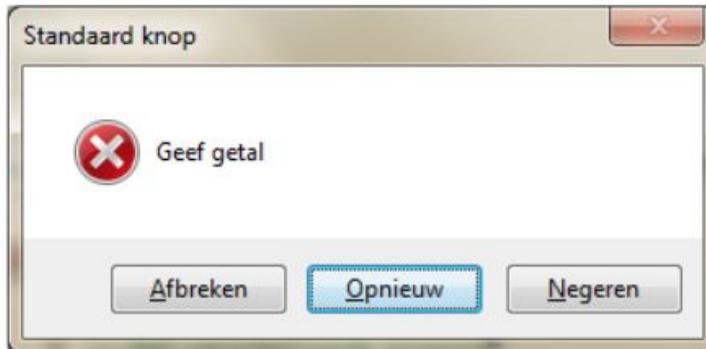


```
// Met verschillende knoppen
MessageBox.Show("Geef getal", " Verschillende knoppen ",
MessageBoxButtons.YesNo);
```

# System.Windows.Forms.MessageBox



```
// Met verschillende icons  
MessageBox.Show("Geef getal", "Verschillende Icons ",  
    MessageBoxButtons.YesNoCancel, MessageBoxIcon.Exclamation);
```



```
//Standaard knop  
MessageBox.Show("Geef getal", "Standaard knop ",  
    MessageBoxButtons.AbortRetryIgnore, MessageBoxIcon.Stop,  
    MessageBoxResult.Retry);
```

# System.Windows.Forms.MessageBox

- Aangeklikte resultaten van MessageBox gebruiken:

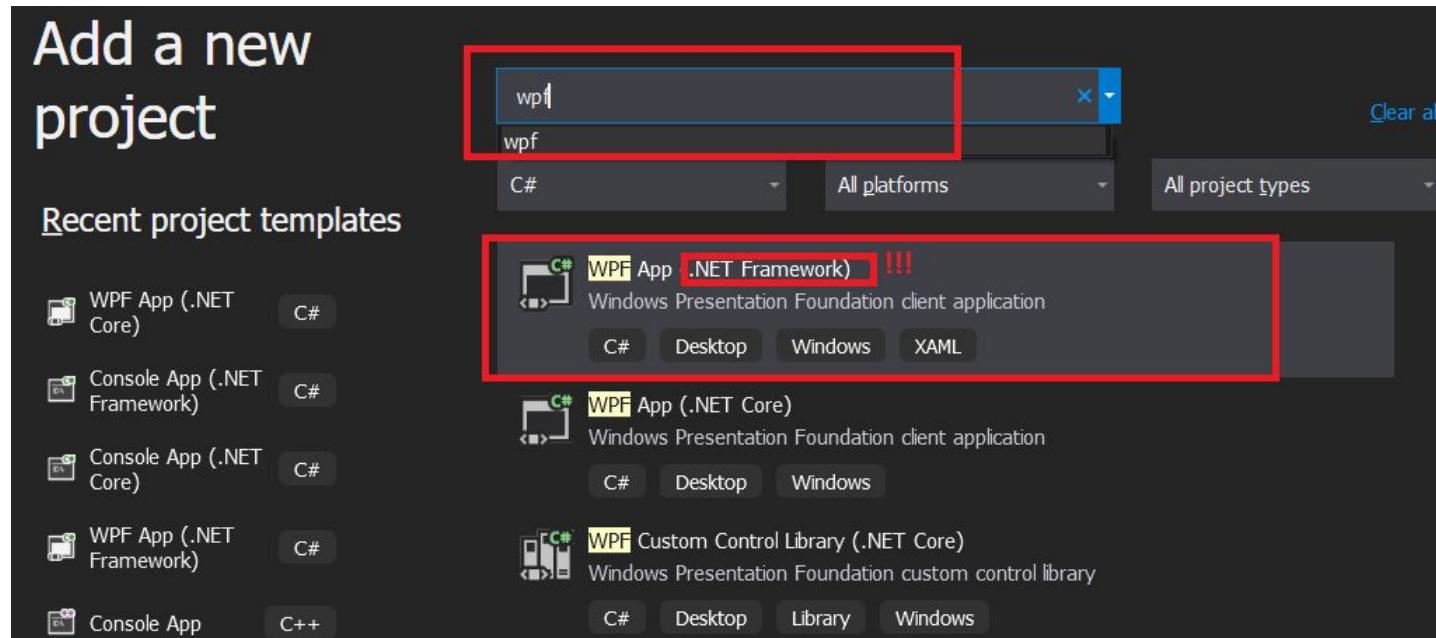
```
// Resultaatwaarden
MessageBoxResult antw = MessageBox.Show("Wil je echt afsluiten?", "Project afsluiten.",
    MessageBoxButtons.YesNo, MessageBoxIcon.Question, MessageBoxResult.No);

if (antw == MessageBoxResult.Yes)
{
    e.Cancel = false;
}
else
{   // Cancel het Closing-event en window wordt niet gesloten.
    e.Cancel = true;
}

if (antw == MessageBoxResult.Retry) TxtResultaat.Text = "Opnieuw";
if (antw == MessageBoxResult.Abort) TxtResultaat.Text = "Afbreken";
if (antw == MessageBoxResult.Ignore)TxtResultaat.Text = "Negeren";
```

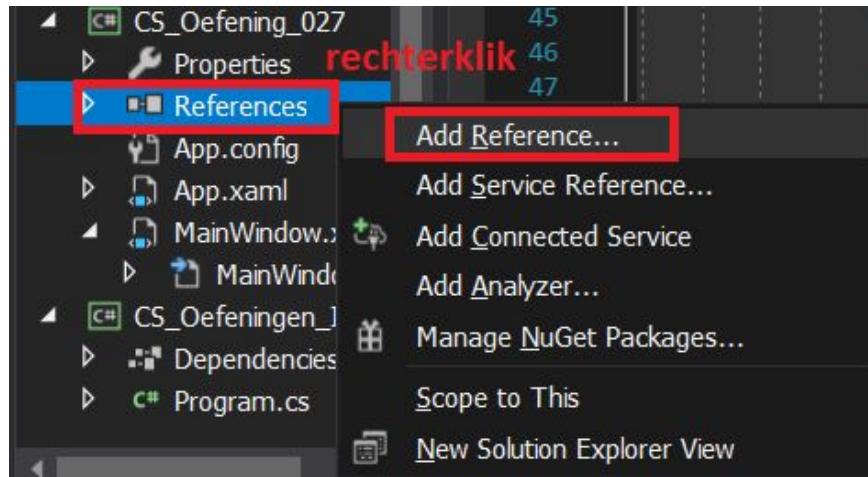
# Microsoft.VisualBasic.Interaction

- In C# bestaat een InputBox dialoogvenster niet ⇒ Visual Basic importeren.
- **OPMERKING:** Maak een nieuw **WPF (.NET Framework) project** aan!!!
  - Dit werkt voorlopig nog niet in .NET Core!



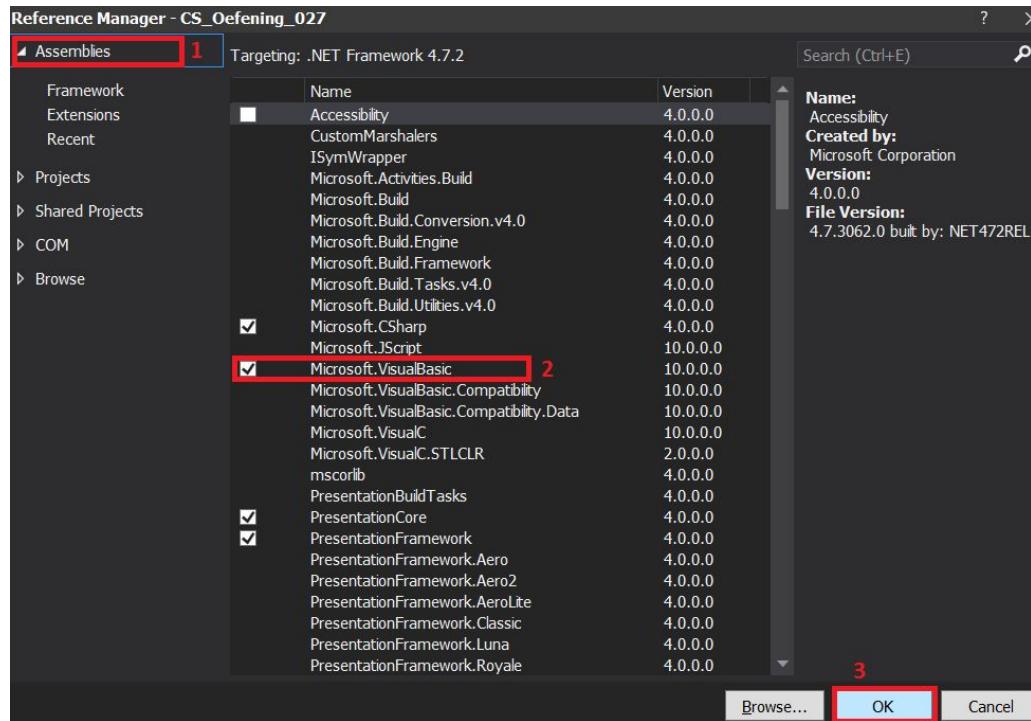
# Microsoft.VisualBasic.Interaction

- Doe rechterklik op References, Add Reference...



# Microsoft.VisualBasic.Interaction

- Duid bij Assemblies aan: Microsoft.VisualBasic. En druk op OK.



# Microsoft.VisualBasic.Interaction



```
using Microsoft.VisualBasic;           prompt      titel      horizontale
string antwoord = Interaction.InputBox("Geef een getal", "Invoer", "50", 500);    positie
// Testen tot er een invoer is.                                         standaardantwoord
while (string.IsNullOrEmpty(antwoord)) // Cancel, Sluiten of lege string
{
    MessageBox.Show("Geef een getal in", "Foutieve invoer");

    antwoord = Interaction.InputBox("Geef een getal", "Invoer", "50", 500);
};
```

# Microsoft.VisualBasic.Interaction

- De positiecoördinaten voor het invoervenster worden weergegeven in pixels.
- Wanneer je geen coördinaten opgeeft dan wordt het invoervenster automatisch gecentreerd.
- Een pixel (samentrekking pictures en element) is een punt van een digitaal scherm.
- Het aantal pixels is bepalend voor de resolutie (vb.  $1280 \times 1024$  pixels ) van het beeldscherm.

# System.DateTime

- DateTime: is een datatype voor een datum tussen:
  - 1 januari 0001 en 31 december 9999 en tijd tussen 0:00:00 en 23:59:59.

`DateTime datum //variabele van datatype DateTime`

<code>DateTime.Today</code>	23/10/2019 0:00:00
<code>DateTime.Now</code>	23/10/2019 15:54:10
<code>datum.Date</code>	23/10/2019 15:54:10;5545904
<code>datum.Day</code>	23
<code>datum.DayOfWeek</code>	Wednesday
<code>datum.DayOfYear</code>	277
<code>datum.Hour</code>	15
<code>datum.Minute</code>	54
<code>datum.Month</code>	10
<code>datum.TimeOfDay</code>	15:54:10.5545904
<code>datum.Year</code>	2019

# System.DateTime

- **Methods:**

datum.AddDays(36)	voegt dagen toe of trek dagen af
datum.AddMonths(12)	voegt maanden toe of trek maanden af
datum.AddYears(2)	voegt jaren toe of trek jaren af
datum.Subtract(datum)	geeft verschil in dagen, uren en minuten (geen maanden, jaren)
datum.Subtract(datum).Days	geeft verschil in dagen tussen de opgegeven datums
DateTime.Parse (string)	zet string om naar datum
datum.ToString("d")	geeft lange datum: dddd d mmmm yyyy
datum.ToString("G")	geeft korte datumnotatie: d/mm/yyyy
datum.ToString("T")	geeft lange tijdsnotatie: hh:mm:ss
datum.ToString("t")	geeft korte tijdsnotatie: hh:mm

# System.DateTime

- Voorbeelden:

```
// declareren en initialiseren van variabele datum
DateTime vandaag = DateTime.Today;                                // huidige systeemdatum
DateTime datum = new DateTime(2018, 10, 3);                      // 3 oktober 2018
DateTime conversieDatum = DateTime.Parse("2018-10-3");        // converteert string naar datum
Console.WriteLine(DateTime.Now);                                  //3/10/2018 16:32:47
Console.WriteLine(vandaag.AddYears(2));                            //3/10/2020 16:32:47
Console.WriteLine(vandaag.TimeOfDay);                            //16:32:47.2911001
Console.WriteLine(vandaag.DayOfWeek);                           // Wednesday
Console.WriteLine((int)vandaag.DayOfWeek);                      // 3
Console.WriteLine(vandaag.Subtract(gbDatum).Days); // geeft dagen tussen vandaag en gbDatum
Console.WriteLine(vandaag.AddMonths(12));                         //3/10/2019 16:32:47
Console.WriteLine(vandaag.AddDays(-6));                          //27/9/2018 16:32:47
Console.WriteLine(vandaag.ToString());                           //woensdag 3 oktober 2018
Console.WriteLine(vandaag.ToShortDateString());                  //3/10/2018
Console.WriteLine(vandaag.ToString("T"));                        //16:32:47
Console.WriteLine(vandaag.ToString("t"));                        //16:32
```

# System.TimeSpan

- **TimeSpan:**
  - Tijdspanne of tijdsinterval.
  - `TimeSpan interval = new TimeSpan(12, 30, 45); // uren, minuten, seconden`
- **Voorbeeld:**
  - ```
// Defineer twee datums.  
DateTime date1 = new DateTime(2010, 1, 1, 8, 0, 15);  
DateTime date2 = new DateTime(2010, 8, 18, 13, 30, 30);  
  
// Bereken het interval tussen deze twee datums.  
TimeSpan interval = date2 - date1;  
Console.WriteLine("{0} - {1} = {2}", date2, date1, interval.ToString());
```

# System.Windows.Threading.DispatcherTimer

- Gebruik DispatcherTimer om taken op vaste tijdsintervallen uit te voeren.
  - DispatcherTimer aanmaken.
  - Ken de taak toe als een event.
  - Stel het tijdsinterval in.
  - Start de DispatcherTimer.
- Voeg eerst bovenaan MainWindows.xaml.cs de namespace toe:
  - **using System.Windows.Threading;**

# System.Windows.Threading.DispatcherTimer

- DispatcherTimer aanmaken:

- `private DispatcherTimer klokje = new DispatcherTimer();`
- Onderaan in MainWindow() onder de InitializeComponents():  
Of binnen private void Window\_Loaded(object sender, RoutedEventArgs e)  
  
`klokje.Tick += new EventHandler(DispatcherTimer_Tick);  
klokje.Interval = new TimeSpan(0, 0, 1); // uren, minuten, seconden  
klokje.Start(); // gebruik .Stop() om te stoppen`  
  
`// Bijvoorbeeld de tijd instellen:  
tijd = DateTime.Now;  
LblTijd.Content = $"{tijd.ToString("yyyy-MM-dd HH:mm:ss")};`

# System.Windows.Threading.DispatcherTimer

- Tijd updaten met een Tick event (typ dit zelf in je MainWindow bestand!):

```
private void DispatcherTimer_Tick(object sender, EventArgs e)
{
    LblTijd.Content = $"{DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss")}";
}
```

# Werken met characters

```
char karakter1 = 'A';
char karakter2 = '%'; // ALT+171
char karakter3 = '9';

Console.WriteLine(karakter1.CompareTo('B'));           // Output: "-1" ('A' < 'B')
Console.WriteLine(karakter1.Equals('A'));               // Output: "True"
Console.WriteLine(char.GetNumericValue(karakter2));     // Output: "0,5"
Console.WriteLine(char.GetNumericValue(karakter3));     // Output: "9"
Console.WriteLine(char.IsDigit(karakter2));             // Output: "False"
Console.WriteLine(char.IsDigit(karakter3));             // Output: "True"
Console.WriteLine(char.IsLetter(','));                  // Output: "False"
Console.WriteLine(char.IsLower('u'));                   // Output: "True"
Console.WriteLine(char.IsNumber(karakter2));            // Output: "True"
Console.WriteLine(char.IsNumber(karakter3));            // Output: "True"
Console.WriteLine(char.IsPunctuation('.'));             // Output: "True"
Console.WriteLine(char.Parse("S"));                     // Output: "S"
Console.WriteLine(char.ToLower('M'));
```