Enumeraties

Zo maak je een enum aan:

```
namespace Programmeren {
enum Weekdagen {Maandag,
Dinsdag, Woensdag, Donderdag,
Vrijdag, Zaterdag, Zondag}
}
```

Je kan er vervolgens variabelen met aanmaken:

```
Weekdagen variabeleNaam;
```

Vervolgens kan je er waarde aan toekennen:

```
variabeleNaam = Weekdagen.Ma-
andag
```

Je kan het vervolgens omzetten in getal of gewoon een string:

```
Console.WriteLine($"{(int)
Weekdagen.Maandag}. {Weekdage-
n.Maandag}");
```

Symbolen classes

Groen leeg bolletje = public attribuut
Rood leeg vierkantje = private attribuut
Onderlijnt = static
Groen vol bolletje = methode

DateTime

Geeft de huidige tijd weer:

```
DateTime variabeleNaam = DateTi-
me.now;
```

Geeft een custom tijd weer:

```
DateTime variabeleNaam = new
DateTime(year, month, day);
DateTime variabeleNaam = new
DateTime(year, month, day,
hours, minutes, seconds, millis-
econd);
```

Voeg data toe:

- AddDays
- AddHours
- ddMilliseconds
- AddMinutes
- AddMonths
- AddSeconds
- AddTicks
- AddYears

voorbeeld:

```
DateTime variabeleNaam = DateTi-
me.Now;
DateTime variabeleNaam = timeNo-
w.AddYears(5);
```

DateTime Extra

Enkele Properties:

- Date
- Day
- DayOfWeek
- DayOfYear
- Hour
- Millisecond
- Minute
- Month
- Second
- Ticks
- TimeOfDay
- Today
- UtcNow
- Year

voorbeeld:

```
DateTime moment = new DateTime(-
1999, 1, 13, 3, 57, 32, 11);
```

Datum & tijd formatteren:

```
WriteLine(now.ToString("d")); //
short date
WriteLine(now.ToString("D")); //
long date
WriteLine(now.ToString("F")); //
full date and time
WriteLine(now.ToString("M")); //
month and day
WriteLine(now.ToString("o")); //
date en time separated by T and
time zone at the end
WriteLine(now.ToString("R")); //
RFC1123 date and time
WriteLine(now.ToString("t")); //
short time
WriteLine(now.ToString("T")); //
long time
WriteLine(now.ToString("Y")); //
year and month
```

EXTRA:

d -> Represents the day of the month as a number from 1 through 31.
dd -> Represents the day of the month as a number from 01 through 31.
ddd-> Represents the abbreviated name of the day (Mon, Tues, Wed, etc).
dddd-> Represents the full name of the day (Monday, Tuesday, etc).
h-> 12-hour clock hour (e.g. 4).
hh-> 12-hour clock, with a leading 0 (e.g. 06)
H-> 24-hour clock hour (e.g. 15)

DateTime Extra (cont)

HH-> 24-hour clock hour, with a leading 0 (e.g. 22)
m-> Minutes
mm-> Minutes with a leading zero
M-> Month number(eg.3)
MM-> Month number with leading zero(e-g.04)
MMM-> Abbreviated Month Name (e.g. Dec)
MMMM-> Full month name (e.g. December)
s-> Seconds
ss-> Seconds with leading zero
t-> Abbreviated AM / PM (e.g. A or P)
tt-> AM / PM (e.g. AM or PM
y-> Year, no leading zero (e.g. 2015 would be 15)
yy-> Year, leading zero (e.g. 2015 would be 015)
yyy-> Year, (e.g. 2015)
yyyy-> Year, (e.g. 2015)

Localized time:

```
CultureInfo russianCI = new
CultureInfo("ru-RU");
Console.WriteLine($"Current time
in Russian style is: {now.ToSt-
ring("F", russianCI)}");
```

en-US English
nl-BE Dutch (Belgium)

String parsen naar DateTime:

```
string variabeleNaam = "8/11/2-
016"; //dit zou dus ook door
gebruiker kunnen ingetypt zijn
DateTime variabeleNaam = DateTi-
me.Parse(date_string);
```

Schrikkeljaar:

```
bool isLeap = DateTime.IsLeap-
Year(today.Year);
if(isLeap == true) {
Console.WriteLine("This year is
a leap year");
}
```

Rekenen met DateTime:

```
DateTime variabeleNaam1 =
DateTime.Today;
DateTime variabeleNaam2 = new
DateTime(year, month, day);
TimeSpan variabeleNaam = variab-
eleNaam1 - variabeleNaam2;
```