

# HoGent

BEDRIJF  
EN  
ORGANISATIE

## Hoofdstuk 1: Inleiding

# Hoofdstuk 1: Inleiding

---

1. Overzicht en praktische inlichtingen
2. .Net Core
3. De ontwikkelomgeving
4. De productieomgeving
5. Inleiding tot Visual Studio
6. En ... wat nu?
7. Appendix

# Overzicht en praktische inlichtingen

# Situering

Applicatieontwikkeling

Netwerken en systeembeheer

Gegevensbehandeling en beheer

Probleemanalyserend denken en Business

Communicatie

Probleemonderzoekend en oplossend denken

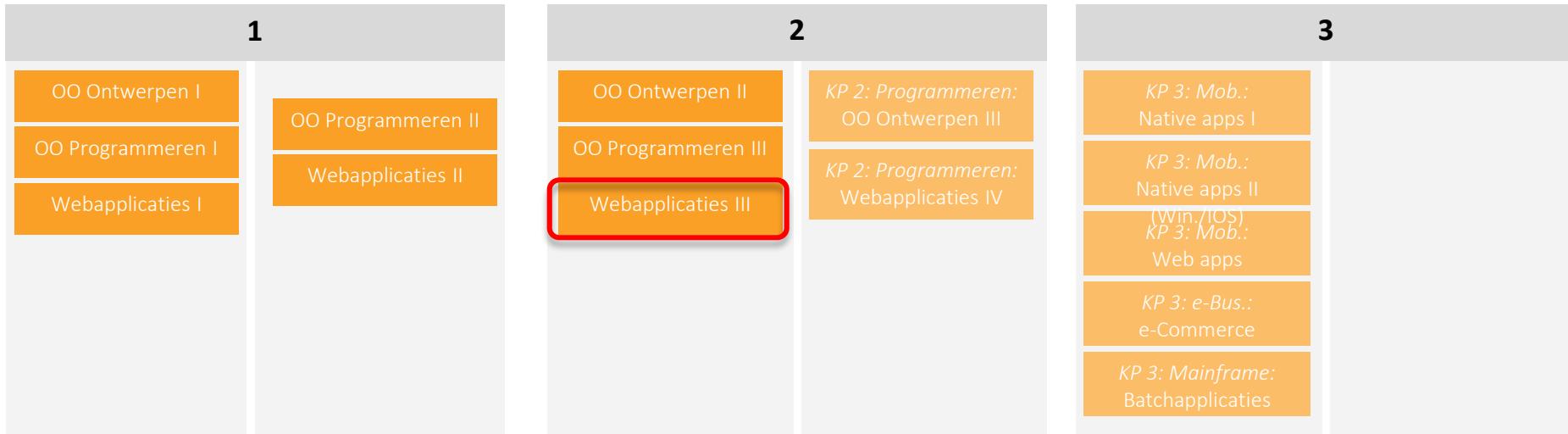
Professionele en individuele groei

# Situering

	Applicatieontwikkeling	Netwerken en systeembeheer	Gegevensbehandeling en beheer	Probleemanalyserend denken en Business	Communicatie	Probleemonderzoekend en oplossend denken	Professionele en individuele groei
1	OO Ontwerpen I						
	OO Programmeren I	Computerarchitectuur	Databanken I	Analyse I	Internationale communicatie I	Math4IT	
	Webapplicaties I						
2	OO Programmeren II	Computernetwerken I		Management		Probleemplossend denken I	iTalent Projecten-workshops I
	Webapplicaties II						
3	OO Ontwerpen II			Analyse II			
	OO Programmeren III	Besturingssystemen		Bedrijfsmanagement	Internationale communicatie II		
	Webapplicaties III						
4	KP 2: Programmeren: OO Ontwerpen III	KP 2: Netwerken: Computernetwerken II	Databanken II	IT2Business		Probleemplossend denken II	iTalent
	KP 2: Programmeren: Webapplicaties IV	KP 2: Netwerken: Computernetwerken III				Onderzoekstechnieken	Projecten-workshops II
	KP 3: Mob.: Native apps I	KP 3: Syst. & Net.: Windows Server	Databanken III	Analyse III			
5	KP 3: Mob.: Native apps II (Win./ios)	KP 3: Syst. & Net.: Enterprise Linux		KP 3: e-Bus.: Business software		KP 3: Mob./Syst. & Net./e-Bus.: Project III	
	KP 3: Mob.: Web apps	KP 3: Syst. & Net.: Computernetwerken IV	KP 3: Mainframe: Databanken en transactiesystemen	KP 3: e-Bus.: Ondernemen	Internationale communicatie III	KP Algemeen: Studium Generale	
	KP 3: e-Bus.: e-Commerce	KP 3: Mainframe: Mainframe	KP 3: e-Bus.: Bus. Intelligence & Big Data	KP 3: e-Bus.: e-Marketing		Artificiële intelligentie	
	KP 3: Mainframe: Batchapplicaties	KP 3: Mainframe: Inleiding mainframe					
						Bachelorproef	iTalent
							KP Algemeen: IP Project
							Stage

# Situering - Applicatieontwikkeling

- ▶ Eindcompetentie: de student kan **kwaliteitsvolle IT-oplossingen efficiënt en autonoom ontwerpen, ontwikkelen, documenteren en testen**, rekening houdend met nieuwe ontwikkelingen en toepassingsdomeinen.



# Situering - Doelstellingen

## Doelstellingen

- Kan site design maken met behulp van layout pages, HTML5 en CSS3
- Kan MV\* toepassen en webapp aanmaken
- Kan OO programmeren/LINQ toepassen in C#
- Kan unit testen voor Model en Controller aanmaken
- Kan database aanmaken en data in een database manipuleren a.d.h.v. ORM Entity Framework
- Kan authenticatie en autorisatie integreren in de applicatie
- Kan een kwaliteitsvolle applicatie implementeren met aandacht voor best practices en efficiëntie

# 1. Overzicht Webapplicaties III

---

- ▶ Webapplicaties I en II behandelen enkel de technologieën aan de client side van een webapplicatie: html5 – css3 – javascript
- ▶ Webapplicaties III behandelt een **server side technologie**:
  - ASP.NET Core MVC
- ▶ Enkele andere server side technologieën:
  - Java via JavaServer Pages (\*.jsp)/Spring
  - PHP (\*.php) - Open Source Scripting
  - Node.js (\*.js)
  - Python, e.g. via Django (\*.py)
  - Ruby, e.g. Ruby on Rails (\*.rb, \*.rbw)
  - ...

# 1. Overzicht Webapplicaties III

---

▶ Doel cursus:

## Maken van **database gestuurde webapplicaties**

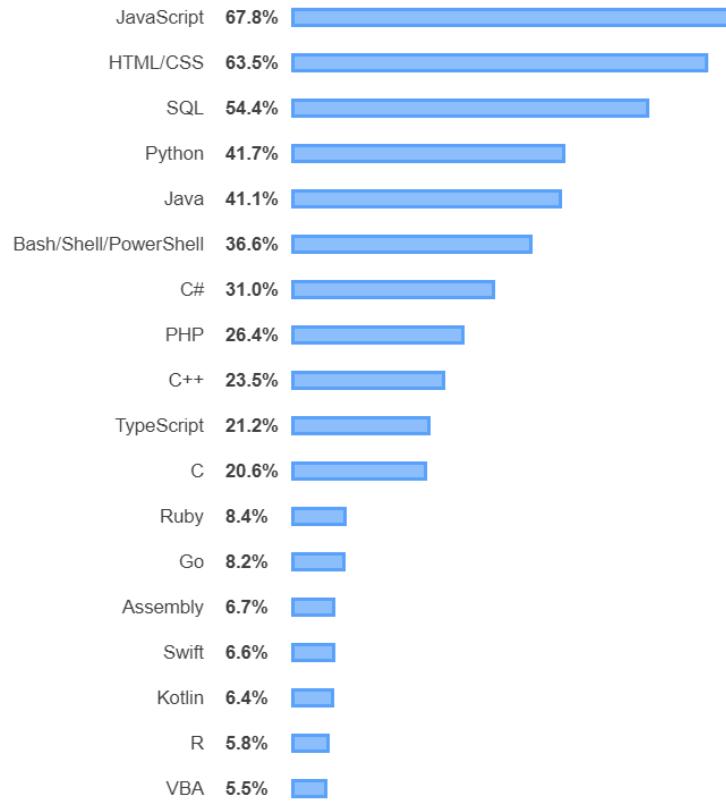
- Ontwikkelomgeving: Visual Studio 2019 Enterprise (met ingebouwde Web Server (IIS Express)) en SQL Server 2017 Developer edition
- Programmeertaal: C# 8.0
- ASP.NET Core MVC: Een klassenbibliotheek binnen .NET Core framework voor het maken van web applicaties
- Entity Framework Core: Object Relational Mapper Framework

# 1. Overzicht Webapplicaties III

▶ <https://insights.stackoverflow.com/survey/2019#technology>

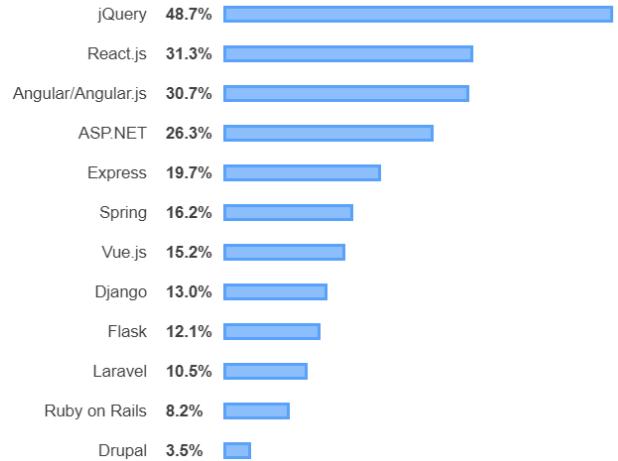
Programming, Scripting, and Markup Languages

All Respondents Professional Developers



Web Frameworks

All Respondents Professional Developers



# 1. Overzicht Webapplicaties III

- ▶ Overzicht cursus Webapplicaties III
  - .Net Core
  - .NET IDE (Ontwikkelomgeving)
  - C#
    - Data Types, Arrays, Collections, Klassen
    - Controlestructuren, Methodes
    - Debugging en Foutafhandeling
    - LINQ (Language Integrated Query)
  - ASP.NET Core MVC
    - Model
    - Controller: action methods, action results, httpContext
    - View: html5, css3, bootstrap, tag helpers, validatie, partial views
  - Unit testen
    - Unit testen van Model en Controller
  - Gegevenstoegang
    - ORM Entity Framework Core
  - Security: Authenticatie/Authorisatie – XSS
  - Ajax en partial views

# 1. Overzicht Webapplicaties III

---

- ▶ Slides op Chamilo
- ▶ Optioneel Handboek
  - Pro ASP.NET Core MVC
    - Auteurs: Adam Freeman
    - Uitgeverij: Apress
    - ISBN 978-1-4842-0397-2
    - Ook als e-book beschikbaar,  
<https://www.apress.com/gp/book/9781484203972>
    - updates en source code: <https://github.com/apress/pro-asp.net-core-mvc>

# 1. Overzicht Webapplicaties III

- ▶ ASP.NET Core
  - ASP.NET Core documentatie: <https://docs.microsoft.com/en-us/aspnet/core/>
  - .Net technische documentatie: <https://docs.microsoft.com/en-us/dotnet/>
  - De code op github: <https://github.com/aspnet/AspNetCore>
- ▶ Pluralsight: e-learning site
  - <http://www.pluralsight.com/>
  - Meer op <https://help.pluralsight.com/help/do-you-offer-student-discounts>
- ▶ Microsoft Learn: free online training for developers
  - <https://docs.microsoft.com/en-us/learn/>

# 1. Overzicht Webapplicaties III

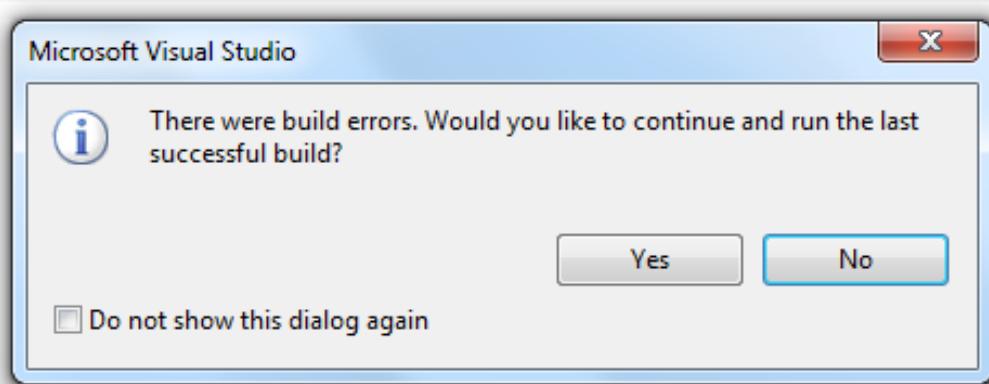
---

- ▶ Lesrooster
  - 2 x 2 lesuren per week: 2u theorie en 2u oefeningen.
- ▶ Verloop lessen:
  - Theorie adhv praktische voorbeelden.
  - Oefeningen: zelfstandig uitwerken van de oefeningen – sluiten onmiddellijk aan bij de theorie.
- ▶ Studietijd
  - Werklast in studiefiche: 48u college en 120u zelfstudie
- ▶ Vereiste voorkennis
  - HTML5 , CSS3 , JavaScript, Bootstrap
  - Java (collections)
  - (Design patterns)
  - (TDD en mocking)

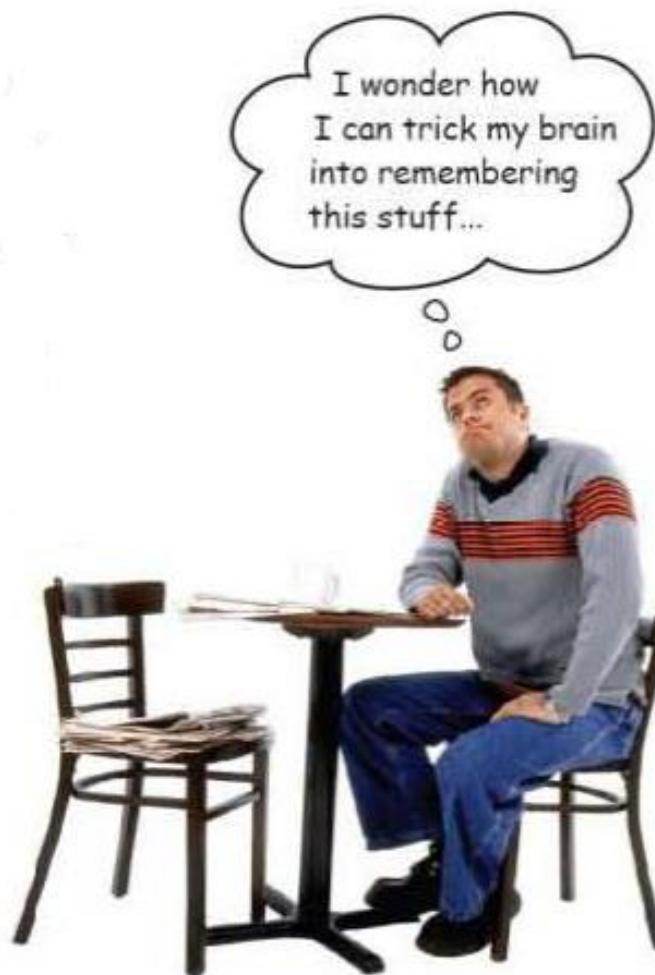
# 1. Overzicht Webapplicaties III

## ▶ Puntenverdeling

- 100% examen
  - op PC (3u)
  - 1 grote oefening, een MVC web applicatie waar hier en daar een stukje code moet worden aangevuld
  - open boek.
    - Slides zijn digitaal beschikbaar
    - Oefeningen,... afprinten en bundelen in een mapje
    - Handboeken mogen gebruikt worden
  - **Compilatie fouten = 0/100**



# 1. Overzicht Webapplicaties III



# 1. Overzicht Webapplicaties III

---

- ▶ Enkele tips die kunnen helpen om te slagen voor Web III
  - Kom naar de les en VOLG de les.
  - De theorie wordt aangebracht door het opbouwen van een applicatie. De applicatie wordt in de les stap per stap opgebouwd (zie slides).
    - Per commit/branch: Doe mee, schrijf code en voeg commentaar toe aan de code
    - OF Completed applicatie (master): Neem notities, voeg commentaar toe aan de code
  - Zorg er eerst voor dat je de theorie hebt doorgenomen (les) en de leerstof begrijpt! Stel vragen in de les.
  - Maak de oefeningen!!!
    - Gebruik de voorbeeldcode + je commentaar als basis voor het oplossen van de oefeningen!! (slides enkel om eens extra informatie op te zoeken, maar vul je voorbeeldcode hier verder mee aan!)
    - In de oefeningenles – vraag ondersteuning aan de lector.
    - Werk de oefening thuis verder af
    - Maak de oefeningen en demo applicatie desnoods herhaaldelijk.
    - Op het internet staat ook heel veel materiaal, tutorials (zie slide 8)

# Exploring .NET Core

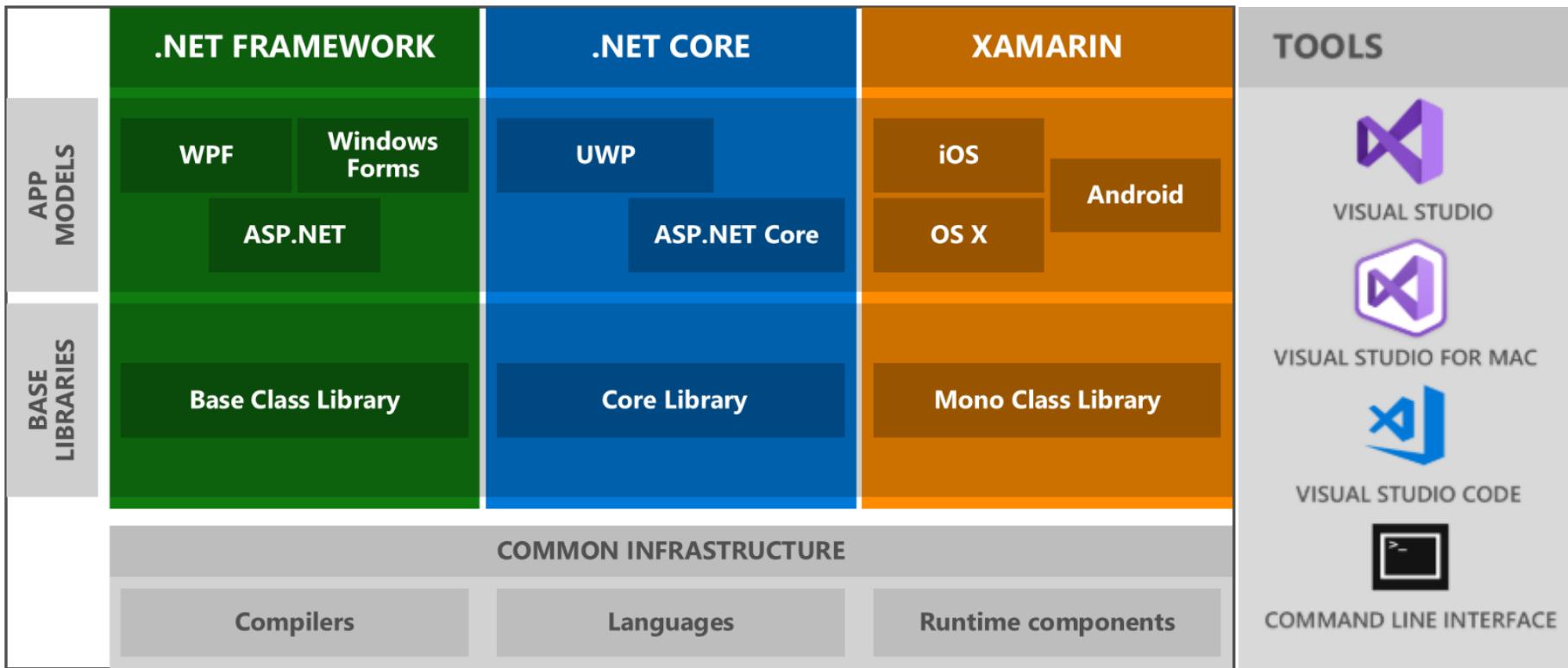


HoGent

# 2. .NET – Things to know

## ▶ .NET

- .NET framework, geïntroduceerd in juni 2000
- .NET Core is de cross-platform versie (Windows, Linux en Mac) van het .NET Framework, geïntroduceerd in 2016.

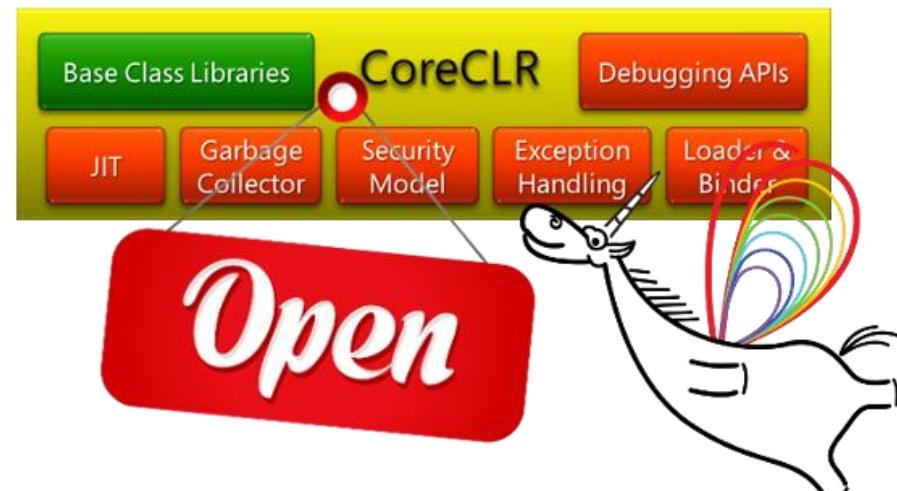


# 2. .NET – Things to know

- ▶ **1. .NET Core: software ontwikkelomgeving**
  - .NET Core is een open source **ontwikkelingsplatform** dat door Microsoft en de .NET community op GitHub wordt onderhouden. Het is **cross platform**, ondersteund Windows, MacOS en Linux, en kan gebruikt worden in apparaat, cloud en embedded / IoT scenario's.
  - Het bevat
    - .NET Core runtime
    - .Net core framework libraries
    - Asp.net core
    - .Net core CLI tools en language compilers (Roslyn en F#)
    - De dotnet tool
  - Repo: <https://github.com/dotnet/core>
  - Gedistribueerd via de **.Net Core SDK** (Software Development Kit) : download op <https://dotnet.microsoft.com/download/dotnet-core>

## 2. .NET – Things to know

- Core Common Language Runtime, CoreCLR
  - De source code: <https://github.com/dotnet/coreclr>
  - Een runtime execution environment: uitvoeren van de verschillende .Net applicaties (Web applicaties, UWP,...)
  - Het bevat de garbage collector, JIT compiler, primitieve data types and low-level classes.
  - Is een Virtuele Machine met JIT compilatie (compiles code on the fly tijdens uitvoering)



## 2. .NET – Things to know

- .Net Core Base Class Library, CoreFX:
  - Een klassenbibliotheek System.\* met 1000den types die je kan gebruiken bij het bouwen van een applicatie (zoals klassen voor collections, file systems, console, XML, async,...)
  - De klassen worden gegroepeerd in **namespaces**. Een **namespace** is set van logisch gegroepeerde klassen, zo kunnen objecten met een andere herkomst maar met dezelfde naam door elkaar gebruikt worden
    - System.Data: alle klassen voor gegevensmanipulatie
    - De source code: <https://github.com/dotnet/corefx>
    - De API reference: <https://docs.microsoft.com/en-us/dotnet/api/index?view=netcore-2.2>
  - Een nieuwe feature release elke 6 maanden, continue verbetering elke 2 maanden

## 2. .NET – Things to know

- **ASP.NET Core** : cross platform framework voor het bouwen van cloud based web applicaties voor Windows, Mac or Linux (zie verder)
- Een verzameling van **.Net Core CLI Tools** en compilers.
  - .Net Core Command Line tools (CLI): <https://github.com/dotnet/cli> : command line tools voor de ontwikkeling van apps (compiling, nuget package management, running, testing, ...)
  - The Roslyn compiler: <https://github.com/dotnet/roslyn>
- De **dotnet tool** : installer packages voor de .net core runtime en libraries.
  - <https://docs.microsoft.com/en-us/dotnet/articles/core/tools/dotnet>
  - Het selecteert en host de runtime, laadt de assemblies en start de app. Dezelfde host wordt ook gebruikt om SDK-tools op dezelfde manier te starten.
- De talen
  - C#, F#, VB

# 2. .NET – Things to know

## 2. .NET is taal-onafhankelijk

- Ontwikkeling .Net Core applicaties kan in verschillende talen
  - C# 8.0, F#, Visual Basic
  - Programma's gecreëerd in verschillende talen, kunnen met elkaar communiceren daar de **Roslyn** compiler ze allen naar **Microsoft Intermediate Language (MSIL)** compileert.
  - IL is een ECMA en ISO Standaard
- De **.NET Standard Library**
  - .NET Standard is a set of APIs that all .NET platforms have to implement. This unifies the .NET platforms; So code sharing for .NET developers across all platforms is possible: desktop applications, mobile apps & games, and cloud services

## 2. .NET – Things to know

- ▶ 3. .NET is gecompileerd, niet geïnterpreteerd
  - 2 compilaties
    - Code wordt gecompileerd door de **Roslyn compiler** naar de **Microsoft Intermediate Language (MSIL)**, bij de eerste request van pagina of op voorhand
      - Dit wordt ook CIL (Common Intermediate Language) genoemd, en is gedefinieerd in the standard ECMA-335.
      - De gecompileerde file = **assembly**
      - Definieert instructies voor (Core)CLR.
      - Zo is taalonafhankelijkheid mogelijk
    - MSIL wordt vertaald naar machine code door CoreCLR, meerbepaald de **RyuJIT compiler (Just in time IL compiler)**, juist voor het uitvoeren van code (JIT). Een JIT kent de feitelijke hardware (zo dient de developer geen verschillende implementaties aan te maken voor de verschillende OS,...)

## 2. .NET – Things to know

---

- Assembly
  - .dll of .exe bestand op je harde schijf, met gecompileerde klassen, interfaces,...
  - = unit of deployment, versie beheer, security
  - Om een klasse te kunnen gebruiken in een applicatie
    - Leg een referentie naar de assembly (voor de common assemblies worden reeds automatisch een referentie toegevoegd, zie later de Reference folder)
    - Importeer de namespace

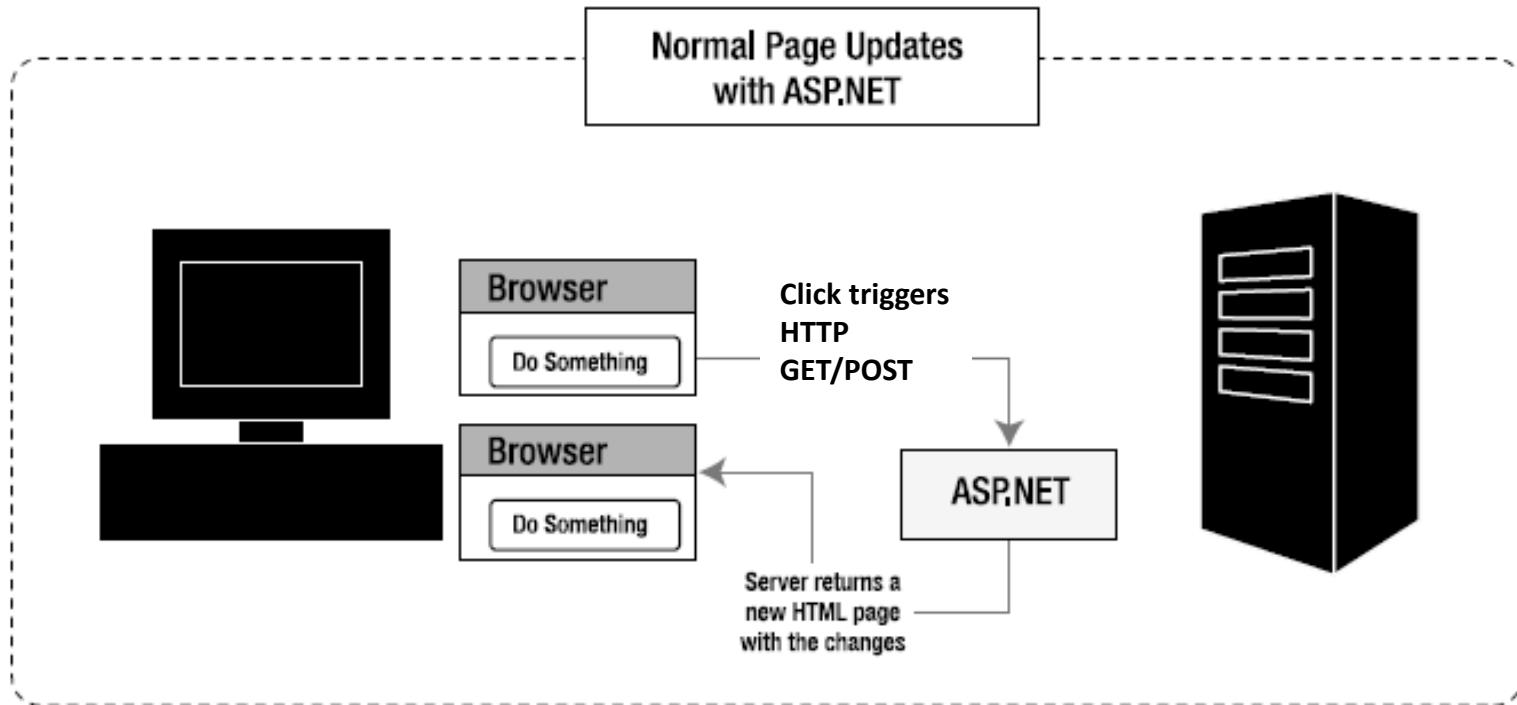
## 2. .NET – Things to know

### 4. ASP.NET Core is een technologie om webapps te maken

- Server-side rendering (traditional web apps)
  - Wordt op de server uitgevoerd, waar de html pagina (html, javascript,...) wordt gegenereerd
  - Vergelijkbare technologien: php, spring,...
  - 2 manieren
- - ASP.NET Core MVC : Model – View - Controller
  - Razor Pages (new) : page-focused coding
- Client-side rendering (SPA) => Web IV
  - technologieën waar de code op de client wordt uitgevoerd in combinatie met een backend
  - Client side : React, Angular, Vue, ...
  - Backend: Web API's, gRPC services en worker services in asp.net core

## 2. .NET – Things to know

- Werking ASP.NET Core MVC – traditionele web applicatie (server side)



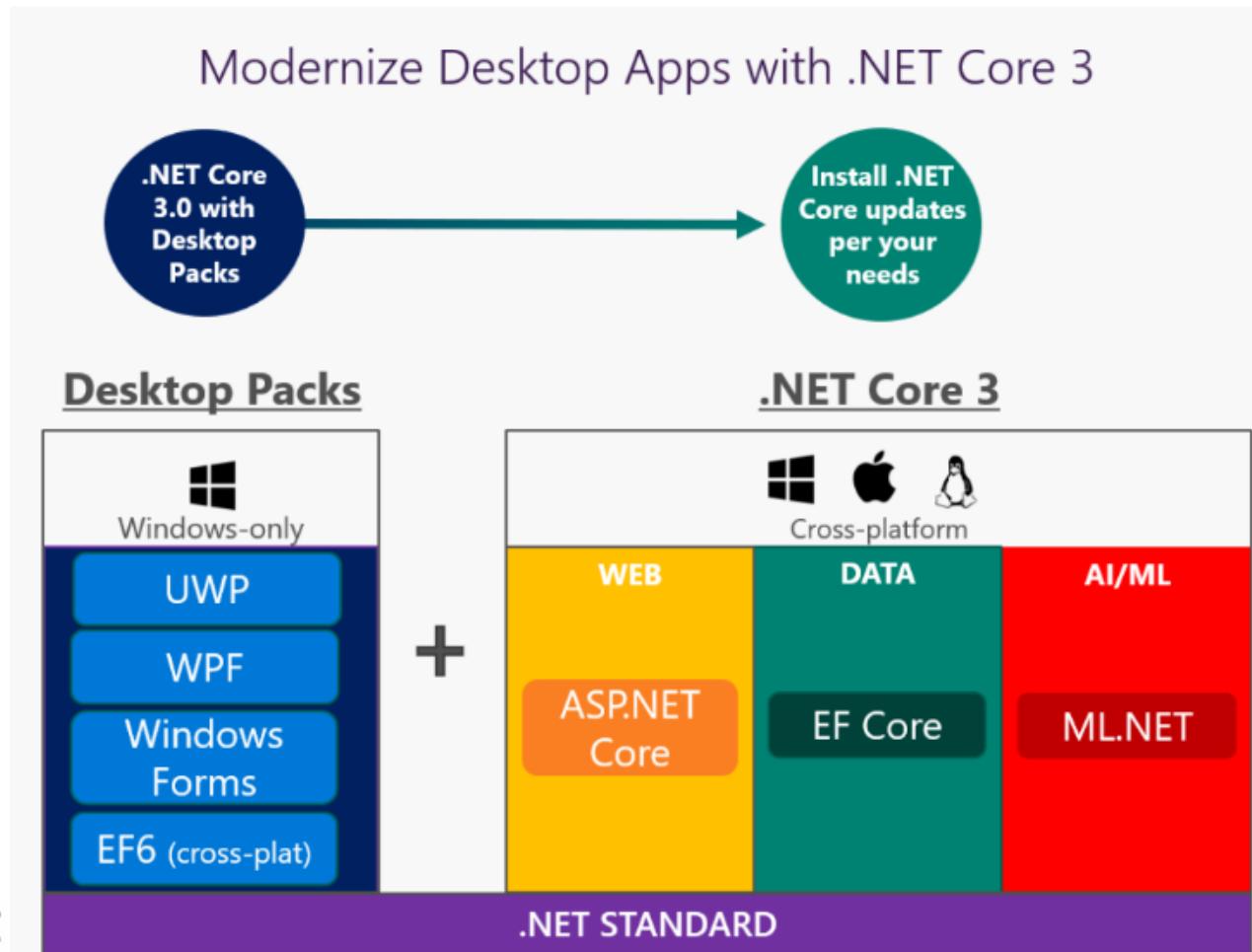
- Werking ASP.NET Core Web API – SPA (Single Page Application) => Web IV

## 2. .NET – Things to know

- ASP.NET Core – de geschiedenis
  - open-source framework voor het bouwen van Internet-connected applications (cloud based).
  - Is een complete rewrite van ASP.NET Framework
  - Cross platform: Windows, Linux, Mac OS X
  - ASP.NET Core 1.0 is released op 27/6/2016.
  - ASP.NET Core 2.0 released op 14/8/2017
  - ASP.NET Core 2.1 released op 30/5/2018
  - • ASP.NET Core 2.2 released op 4/12/2018

## 2. .NET – Things to know

- ASP.NET Core – de geschiedenis
  - ASP.NET Core 3.0 release tijdens .net conf, 23-25/9/2019



# 2. .NET – Things to know

- The future: .Net 5

(<https://devblogs.microsoft.com/dotnet/introducing-net-5/>)

.NET – A unified platform



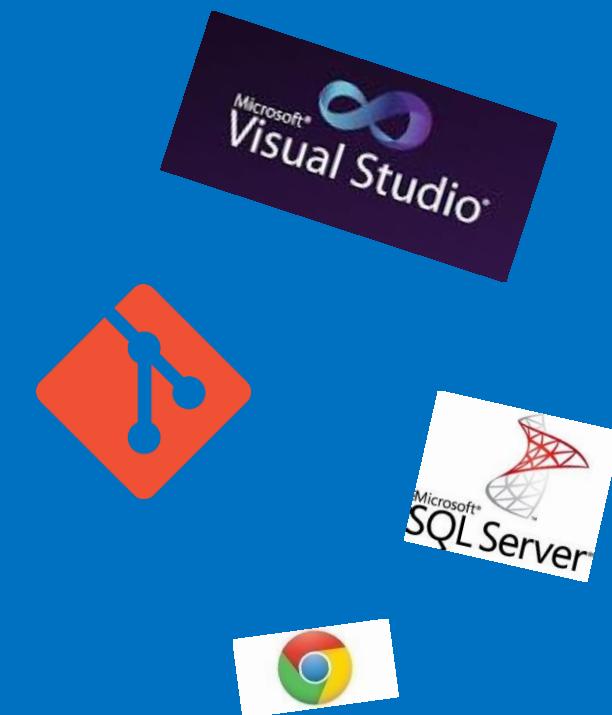
## .NET Schedule



## 2. .NET – Things to know

- HTTP: Hypertext Transfer Protocol
  - Request-response protocol binnen een client-server model
    - Request-verbs (client -> Server): GET, POST, PUT, DELETE, ...
    - Response (server -> client):
      - meestal HTML
      - maar ook Javascript, XML,CSS, beeld, geluid, JSON, e.a.
  - Is **STATELESS**
    - Elke request staat op zichzelf, server heeft geen geheugen: houdt geen status bij
    - Oplossingen:
      - Cookies
      - Hidden variables (alleen mogelijk bij forms)
      - Sessions
      - URL-rewriting (vb: /index/id2000)

# De ontwikkelomgeving



HoGent

# 3. De Ontwikkelomgeving

- ▶ Overzicht Visual Studio 2019: verschillende producten
  - <https://www.visualstudio.com/vs/compare/>
  - Wij gebruiken Visual Studio Enterprise 2019 (Windows) of Visual Studio for Mac (Mac)

Alle informatie i.v.m. installatie van software vind je terug in Hoofdstuk 00

# 3. De Ontwikkelomgeving

---

- ▶ Database
  - SQL Server 2017 Developer Edition :
  - SQL Server Management Studio V17.9.1
- ▶ GIT
  - source control and teamwork
- ▶ Microsoft docs (online help)
  - <https://docs.microsoft.com/en-us/>
- ▶ Een web server:
  - IIS Express wordt gebruikt binnen VS 2019(en wordt ook geïnstalleerd bij installatie VS2019)

# 3. De Ontwikkelomgeving

- ▶ Browser en developer tools in browser: naar keuze
  - In de les wordt Google Chrome gebruikt
- ▶ Extra:
  - Pluralsight (e-learning)
    - <https://help.pluralsight.com/help/do-you-offer-student-discounts>
  - MS Azure: \$100 credit te gebruiken binnen de 12 maanden
    - <https://azureforeducation.microsoft.com/devtools> (you can claim your Azure credit)
- ▶ Extra tools & extensies voor Visual Studio:  
<https://marketplace.visualstudio.com/>

# 3. De Ontwikkelomgeving

- ▶ Je kan ook gebruik maken van de **Command line tool** voor .NET
  - In command window, in de directory van je applicatie:
  - typ >dotnet - -help
    - dotnet new: Initializes a sample console C# project.
    - dotnet restore: Restores the dependencies for a given application.
    - dotnet build: Builds a .NET Core application.
    - dotnet publish: Publishes a .NET portable or self-contained application.
    - dotnet run: Runs the application from source.
    - dotnet test: Runs tests using a test runner specified in the project.json.
    - dotnet pack: Creates a NuGet package of your code
  - Meer op <https://docs.microsoft.com/en-us/dotnet/core/tools/?tabs=netcore2x>
  - Je kan ook gebruik maken van Visual Studio Code in combinatie met de CLI (een voorbeeld op <https://code.visualstudio.com/docs/languages/dotnet>)

# De productieomgeving



HoGent

# 4. De Productie omgeving

---

- ▶ Je kan de applicatie op verschillende platformen runnen
  - IIS, Microsoft Azure, Ubuntu, ...
  - Zie <https://docs.asp.net/en/latest/hosting/index.html> voor de hosting
  - Zie <https://docs.asp.net/en/latest/publishing/index.html> voor het publiceren en deployen van de applicatie

# Inleiding tot Visual Studio

Hello Visual Studio

een eerste console applicatie!

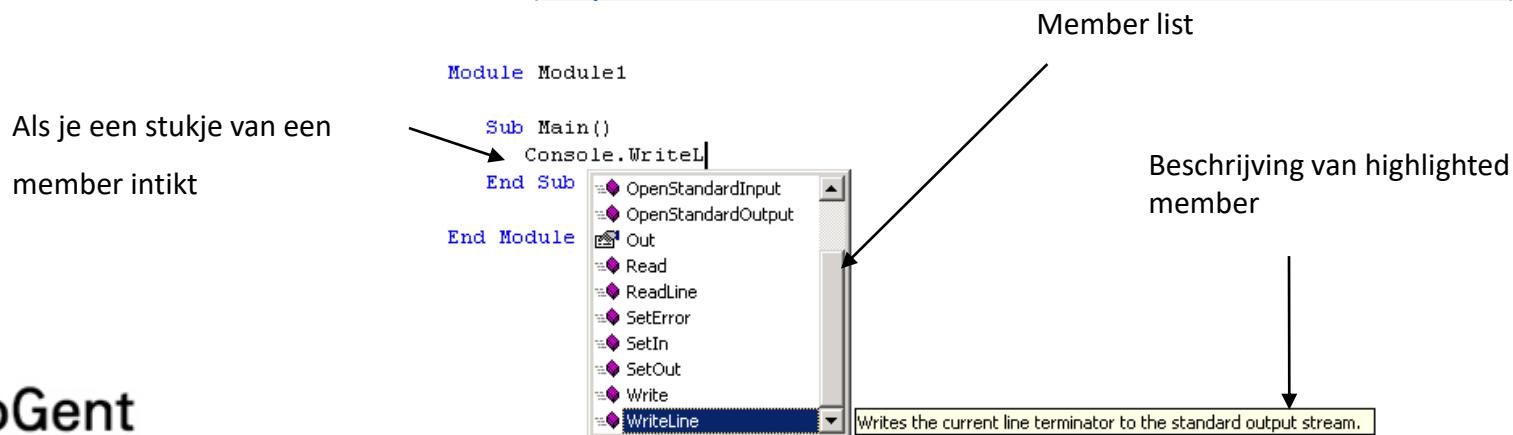


HoGent

# 5. Hello Visual Studio

## ▶ Visual Studio 2019

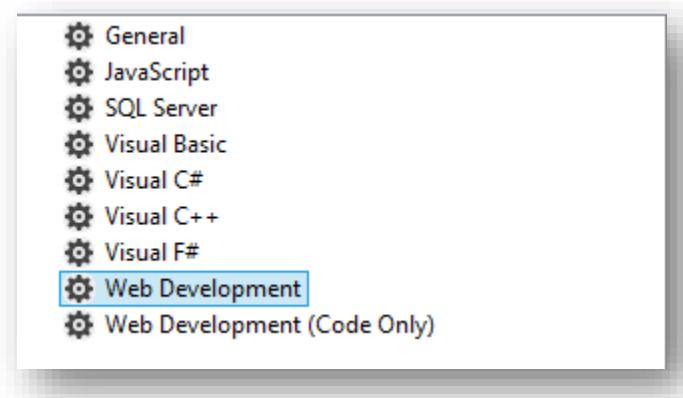
- Microsoft's Integrated Development Environment (IDE)
- Gebruikt om applicaties aan te maken, te debuggen en uit te voeren
- Visuele programmeeromgeving
  - IntelliSense/IntelliCode (tab of enter om geselecteerde member aan code toe te voegen)
    - IntelliCode maakt gebruik van AI. “IntelliCode trains across 2,000 open-source projects on GitHub—each with over 100 stars—to generate its recommendations”.(<https://visualstudio.microsoft.com/services/intellicode/>)



# 5. Hello Visual Studio – de IDE

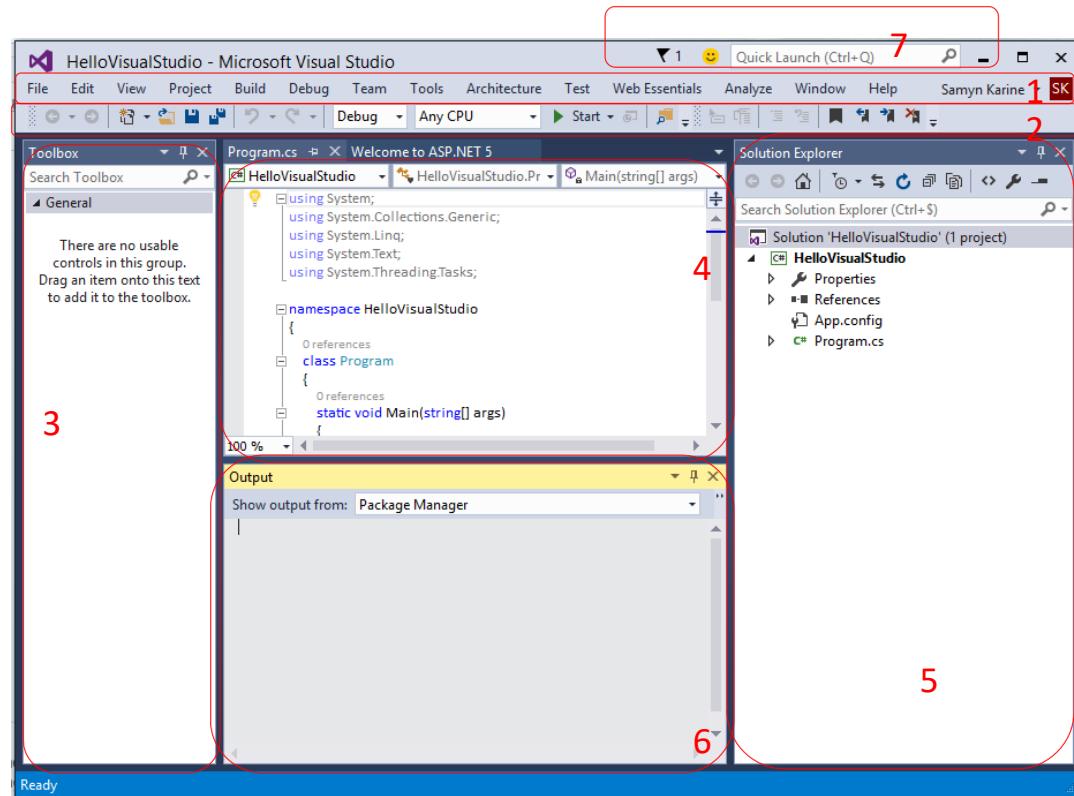
## ► De ontwikkelomgeving (IDE)

- Bij 1<sup>ste</sup> opstart Visual Studio,
  - dien je de omgeving in te stellen. Kies “Web Development”. (Kan je achteraf aanpassen via Tools > Export and Import settings > Reset all settings)



# 5. Hello Visual Studio – de IDE

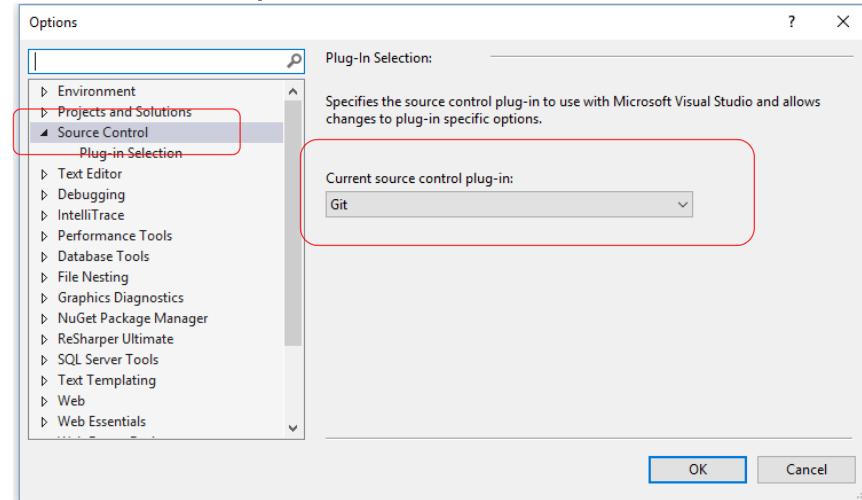
1. Menu: Standard windows menus
2. Toolbar: verzameling van toolbars , snelle toegang tot meest gebruikte features
3. Left pane: bevat meestal de Toolbox, gebruikt voor aanmaak van UI en Server explorer: connecteren met databases
4. Center pane: Start page, de documenten zoals C# files (tabbladen)
5. Right pane: bevat Solution Explorer, en mogelijk andere views
6. Bottom pane: bevat compiler output window en error list.
7. Quick Launch search box, notifications, feedback



# 5. Hello Visual Studio – de IDE

## ▶ Customisatie

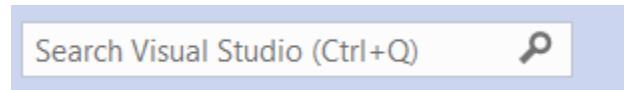
- Instellingen aanpassen: Tools menu > Options>Environment
  - General: Color Theme:
    - Light/Dark/Blue(Default)
  - Fonts and Colors:
    - lettertype, grootte en kleur
  - Source Control :
    - Kies GIT



- Aanpassen van toolbar: Tools > Customize
- Vensters kan je vergroten, verkleinen en vastpinnen
- Rechtsklik op scrollbar in center pane, Scrollbar options laat toe om scrollbar te customiseren
  - Pas eens aan in Map mode – medium
  - <http://msdn.microsoft.com/en-us/library/dn237345.aspx>

# 5. Hello Visual Studio – de IDE

- ▶ Quick Launch search box (Ctrl + Q):



- snel zoeken en uitvoeren van acties uit de IDE. Zoekt in menus, commands, options, installable components
  - Vb solution explorer, new project,...
- Het resultaat komt in 4 categoriën : All, Menu, Templates, Components
- Zie <https://docs.microsoft.com/en-us/visualstudio/ide/reference/quick-launch-environment-options-dialog-box?view=vs-2017&viewFallbackFrom=vs-2019>

# 5. Hello Visual Studio – de IDE

## ► Start Page

Creatie van  
nieuwe projecten,  
openen  
projecten,...

The screenshot shows the Visual Studio 2019 Start Page. On the left, there is a sidebar titled "Recente projecten" with an arrow pointing to the "FoodTruckNationApi.sln" entry in the "Open recent" list. The "Open recent" section lists several projects with their creation dates and locations:

- eShopOnWeb.sln (20-Aug-19 09:48) C:\Users\ksa607\Downloads\eShopOnWeb-master\eShopOnWeb-master
- projectapp\_backend.sln (25-Jun-19 18:24) D:\...\web4-backend-20182019-gent-2a2-tessawalraet\projectapp\_backend
- FoodTruckNationApi.sln (17-Jun-19 10:09) C:\temp\FoodTruckNationApi-master\FoodTruckNationApi-master\src
- Bierhalle.sln (17-Jun-19 10:06) C:\temp\Bierhalle\Bierhalle
- Bakery.sln (14-Jun-19 18:40) C:\temp\Bakery
- Beerhall.sln (14-Jun-19 17:56) C:\temp\beerhallOriginal\10thBeerhallMvcAdv
- YeeterAPI.sln (04-Jun-19 21:17) D:\Web4\A2Backend\SergeWannes

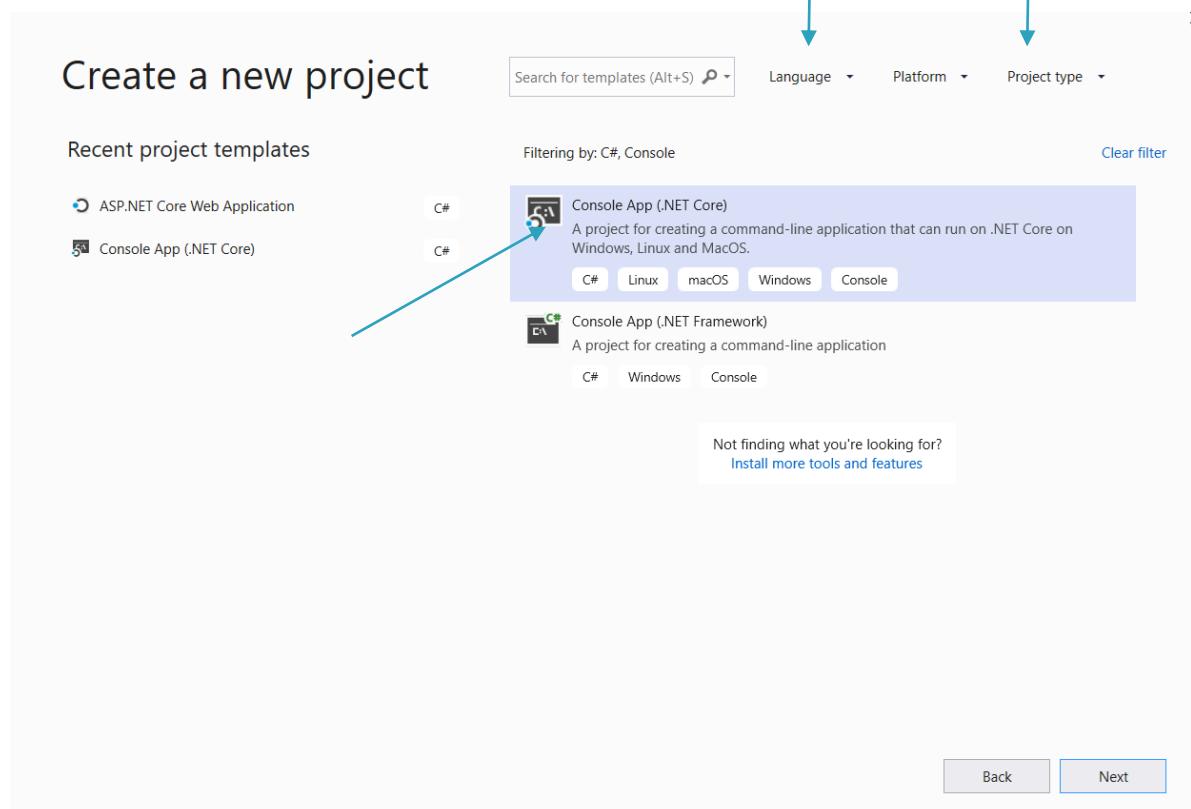
To the right, there is a "Get started" section with four options:

- Clone or check out code**: Get code from an online repository like GitHub or Azure DevOps.
- Open a project or solution**: Open a local Visual Studio project or .sln file.
- Open a local folder**: Navigate and edit code within any folder.
- Create a new project**: Choose a project template with code scaffolding to get started.

A blue arrow points downwards from the "Create a new project" option towards the bottom right corner of the page.

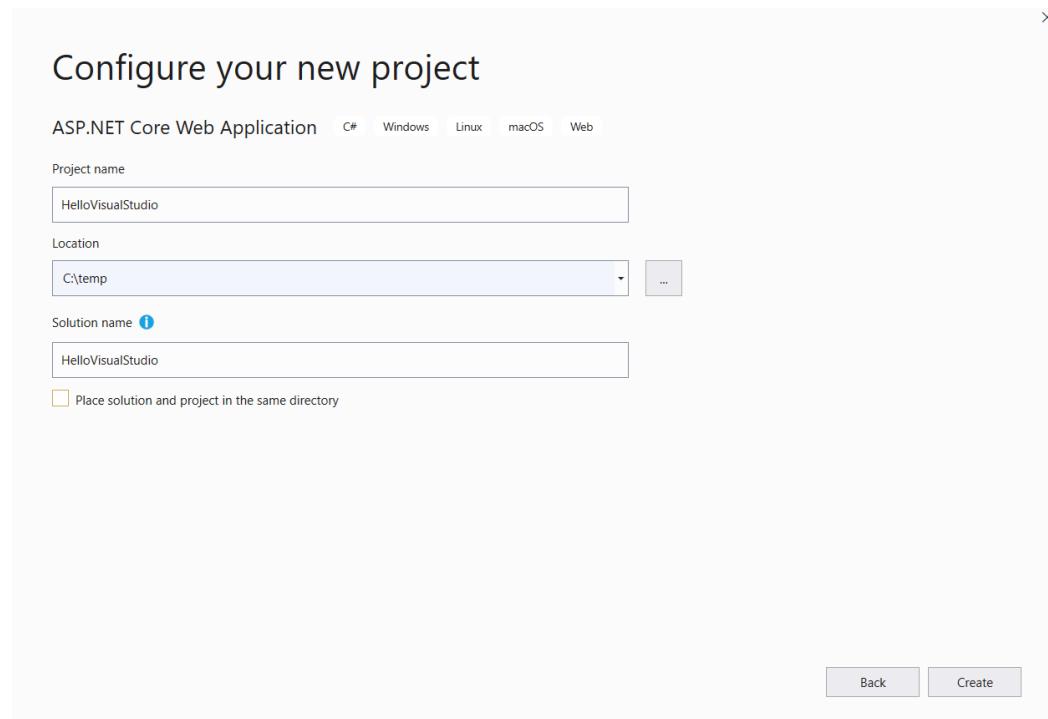
# 5. Hello Visual Studio - Console app

- ▶ Aanmaken van nieuw project “Hello Visual Studio”
  - **Op startpage > New Project of in VS : File > New > Project**
    - kies als taal C#, als Project Type Console, selecteer dan "Console App (.Net Core)" en klik next. Je kan bovenaan ook zoeken op Console



# 5. Hello Visual Studio - Console app

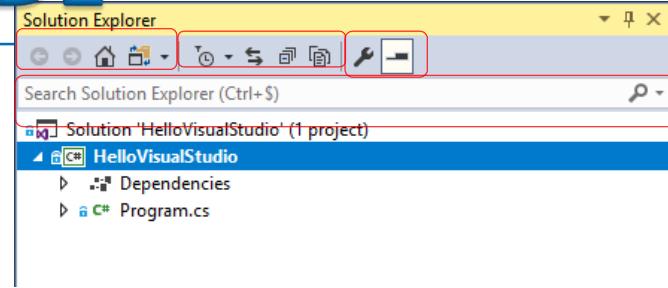
- ▶ Aanmaken van nieuw project “Hello Visual Studio”
  - **Op startpage > New Project of in VS : File > New > Project**
    - Vul Project Name (hier HelloVisualStudio ) in en kies locatie. Vink place solution and project in the same directory uit. Klik Create. Het project wordt aangemaakt.



# 5. Hello Visual Studio – IDE <sup>1</sup>

## ▶ Solution Explorer

- Openen via View > Solution Explorer
- Bevat een lijst met de verschillende folders en bestanden waaruit project is opgebouwd. Verder openklappen toont ook de logische elementen
  - Dubbelklik op een programmeerelement opent bestand in de overeenkomstige editor in center pane
  - Rechtermuisklik op 1 van de programmeer elementen geeft een context sensitive menu
- De toolbar bevat 3 groepen
  - Groep 1: Back, Forward, Home: werken samen met Search Solution Explorer box (erronder) om te navigeren tussen zoekresultaten. Home cleart search.
    - Dropdown naast zoek box: typ pro en kies in dropdown “search within file contents”, zoekt nu ook in de bestandsinhoud
    - Keuze tussen folder of solution view



# 5. Hello Visual Studio - IDE

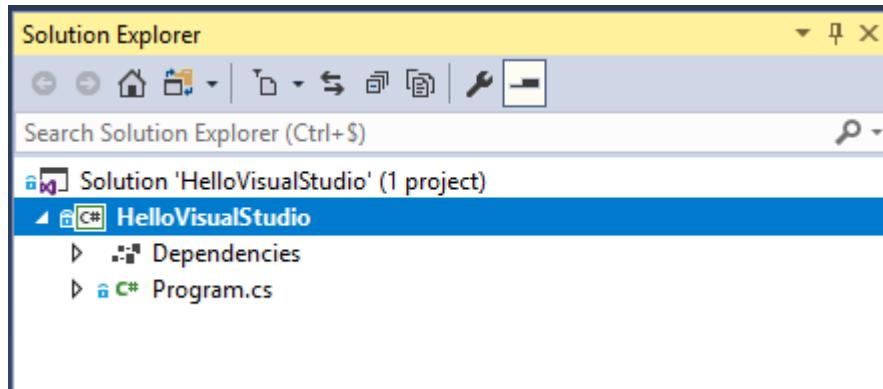
## ▶ Solution Explorer



- De toolbar bevat 3 groepen
  - **Groep 2:** bepaalt wat getoond wordt in de Solution Explorer.
    - Open Files filter:
      - klik op icoon: toont terug alle files
      - Dropdown - Pending changes: nog niet gecommitte files
      - Dropdown - Open Files: files die momenteel geopend zijn.
    - Sync with active document: toont huidig document in folderstructuur
    - Refresh: refresh van files
    - Collapse all: klap de folders in solution tree open of dicht
    - Show all files: toont alle bestanden in Solution directory, ook deze die niet behoren tot project (bvb bin folder)
  - **Groep 3:**
    - Properties: opent properties venster
    - Preview selected items: om snel de code te bekijken van een aantal files. Als enabled, 1 click opent item in eerste tab van middle pane. Klik op ander item, toont dan dit item in eerste tab,...
    - Enkel in Enterprise: genereert dependency graph geselecteerde file
- Opmerking
  - Ctrl + Tab heeft overzicht van geopende bestanden en vensters

# 5. Hello Visual Studio – Console app

- ▶ Structuur console app in Solution explorer
  - Dependencies: gebruikte assemblies. API's beschikbaar voor je app.
  - Program.cs: bevat Main

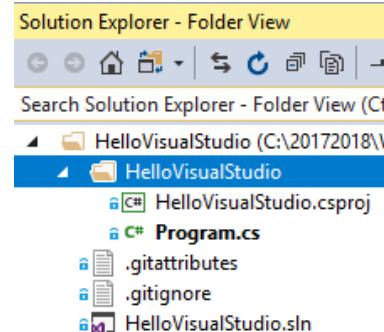
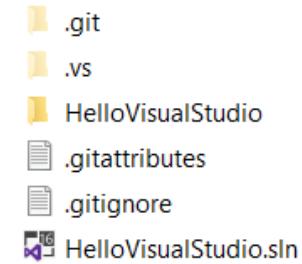


- Rechtsklik Solution > Add solution to source control als je de code wenst toe te voegen aan git

# 5. Hello Visual Studio – Console app

## ▶ Structuur console app in Windows Verkenner/folder view

- Folder HelloVisualStudio
  - .git, .gitattributes, .gitignore: info voor git
  - Een hidden folder .vs. Bevat o.a. een .suo file met de instellingen van je project (breakpoints,...)
  - <naam project>: folder met code project
    - .csproj : bevat het target framework. Hier de Netcoreapp2.2 meta package (is een Nuget package, zonder inhoud maar met een lijst van dependencies (andere packages). Bij compilatie worden de child packages geimporteerd. (Zie dependencies in solution folder)
    - Program.cs: Bevat Main methode
    - .sln: solution file met verwijzing naar projecten,... die solution bevat



# 5. Hello Visual Studio – Console app

## ▶ Program.cs

- C#

```
using System;

namespace HelloVisualStudio
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello C#");
        }
    }
}
```

## Java

```
package hellojava;

//import statements

public class HelloJava {

    public static void main(String[] args) {
        System.out.println("Hello java");
    }
}
```

- Zoek de verschillen.

# 5. Hello Visual Studio – Console app

## ▶ Program.cs

- De verschillen?

```
using System;

namespace HelloVisualStudio
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello C#");
        }
    }
}
```

using *namespace* (Java: import *package*)

Je hoeft ook de \* niet te vermelden

Visual Studio voegt automatisch de meest gebruikte namespaces toe. De niet gebruikte (in lichtgrijs) mag je verwijderen

namespace .... { } met daarbinnen de klasse. (Java: package ..., staat bovenaan)

Default access modifier is internal (enkel beschikbaar binnen de assembly). Je kan public toevoegen

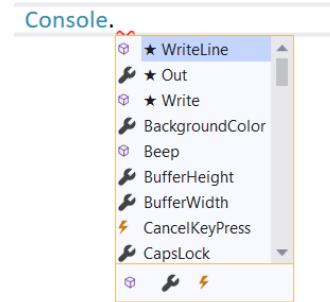
Alle methodes beginnen met een hoofdletter (java: kleine letter) – zie methodes Main en WriteLine

string met kleine letter. Is een alias voor de klasse String. Je mag ook String gebruiken

# 5. Hello Visual Studio – Console app

## ▶ IntelliSense/IntelliCode

- VS toont Member List bij ingave code of . (of Ctrl+Enter of Ctrl+J ) - tab of dubbelklik to select



- Ingave "(" toont overloadingen.A screenshot of the Visual Studio IDE showing a tooltip for the 'Console.WriteLine()' method. The tooltip header is 'Console.WriteLine()' with a small orange arrow icon. Below it, it says '▲ 1 of 18 ▼ void Console.WriteLine(string value)'. There is a star icon followed by the text '★ IntelliCode suggestion based on this context'. At the bottom, there is a description: 'Writes the specified string value, followed by the current line terminator, to the standard output stream.' and a note: 'value: The value to write.'
- Ingave parameter toont informatie over parameter
- Meer op <https://docs.microsoft.com/nl-nl/visualstudio/ide/using-intellisense>

# 5. Hello Visual Studio – Console app

## ▶ Hello Visual Studio

- Run de applicatie: F5 of  of Debug > Start Debugging.
- Visual Studio
  - compileert de applicatie
  - Start de console
  - Voert de code uit
  - Druk enter om de console te sluiten
- Klik  om te stoppen.

# 5. Hello Visual Studio – Console app

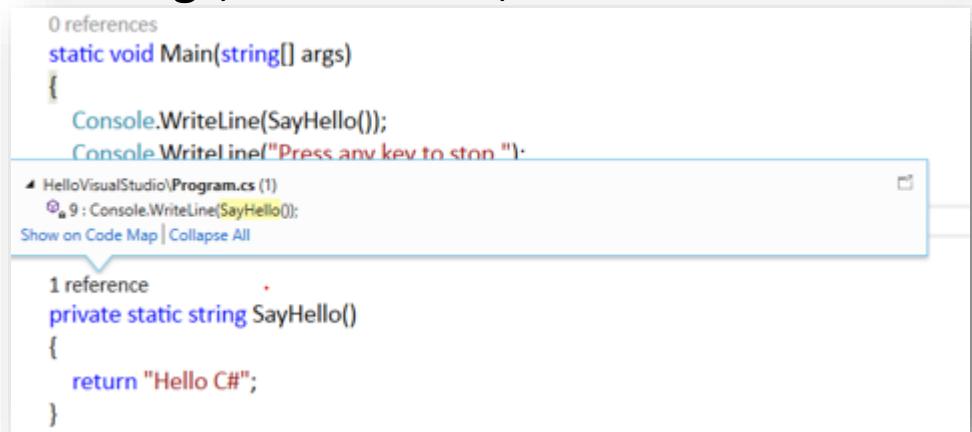
## ▶ Hello Visual Studio

- Maak een methode SayHello die de string “Hello C#” retourneert.
  - Maak gebruik van deze methode in de Main methode
  - Run
- 
- Extra : Vraag aan de gebruiker hoeveel maal Hello C# moet worden weergeven.

# 5. Hello Visual Studio - IDE

## ► Editor tools

- CodeLens (enkel in VS Enterprise)
  - Toont informatie over code references, unit test en als je in team werkt (Team Foundation Server of GIT): aanpassingen aan de code, linked bugs, work items, code reviews



- Instellen via Tools > Options >Text Editor > All languages > CodeLens
- <https://docs.microsoft.com/nl-nl/visualstudio/ide/find-code-changes-and-other-history-with-codelens?view=vs-2017>

# 5. Hello Visual Studio - IDE

## ► Editor tools

- Hoveren over lijn, geeft detail

The screenshot shows the Visual Studio code editor with the following C# code:

```
namespace HelloVisualStudio
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine(SayHello());
        }

        static string SayHello()
        {
            return "Hello C#";
        }
    }
}
```

A tooltip is displayed over the `Console.WriteLine(SayHello());` line, providing information about the `System.Console` class:

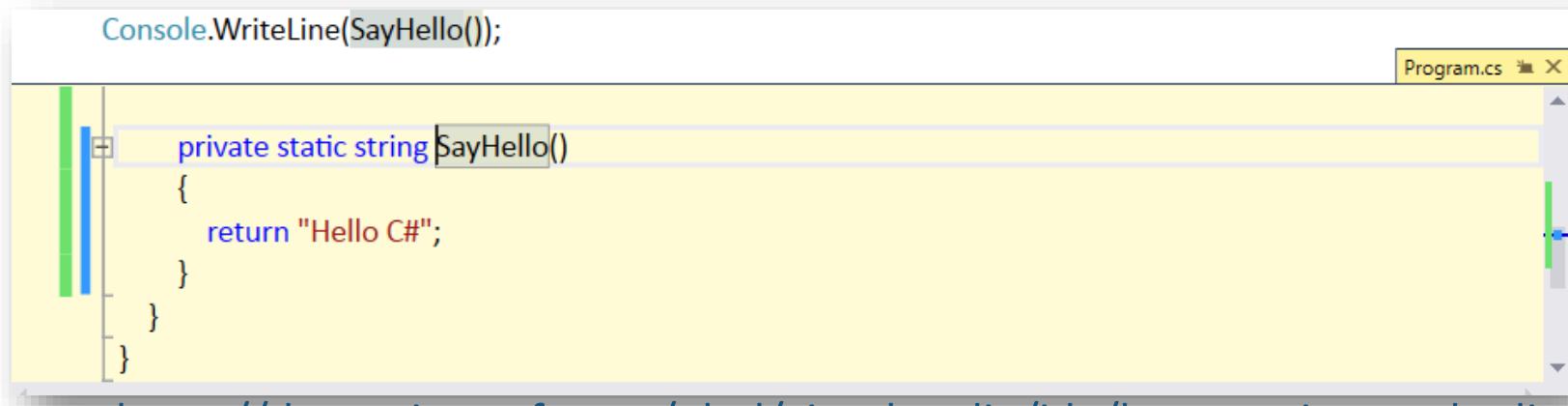
class System.Console  
Represents the standard input, output, and error streams for console applications. This class cannot be inherited.

- Kleuren in de marges
  - **No colour** = Line unchanged since you opened the file
  - **Green** = Line changed since you opened the file, but saved
  - **Yellow** = Line changed since you last saved

# 5. Hello Visual Studio - IDE

## ► Editor tools

- Peek definition (Alt + F12)
  - Bekijk en editeer code in een methode
  - Selecteer een methode, rechtsklik > Peek definition of Alt F12. Toont popup met de code.
  - Vb. Selecteer Hello() en dan Alt F12. Esc of x om popup te sluiten



The screenshot shows the Visual Studio IDE with the code editor open. The code in the editor is:

```
Console.WriteLine(SayHello());
```

The `SayHello()` method is highlighted in blue. A tooltip window titled "SayHello()" is displayed above the method definition, containing the code:private static string SayHello()  
{  
 return "Hello C#";  
}

- <https://docs.microsoft.com/nl-nl/visualstudio/ide/how-to-view-and-edit-code-by-using-peek-definition-alt-plus-f12?view=vs-2017>
- Go to definition (F12): opent de definitie in een tab

# 5. Hello Visual Studio - IDE

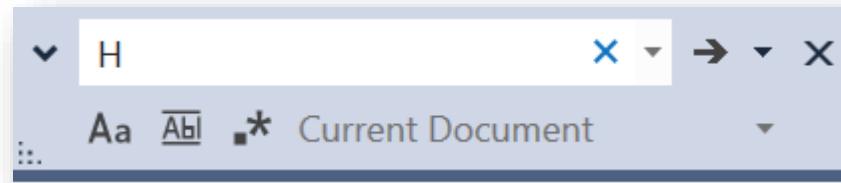
## ► Editor tools

- Go to All window, Edit > Go To : toont een search box
  - Go To Line (Ctrl +G) – quickly jump to a different line in your current document
  - Go To All (Ctrl + ,) or (Ctrl + T) – search results include everything below
  - Go To File (Ctrl 1, F) – search for files in your solution
  - Go To Type (Ctrl 1, T) – search results include:
    - Classes, Structs, Enums, Interfaces & Delegates
  - Go To Member (Ctrl 1, M) – search results include:
    - Global variables and global functions
    - Class member variables and member functions
    - Constants, Enum Items, Properties and Events
  - Go To Symbol (Ctrl 1, S) – search results include:
    - Results from *Go To Types* and *Go To Members*
  - *Meer op* <https://docs.microsoft.com/en-us/visualstudio/ide/navigating-code?view=vs-2019>

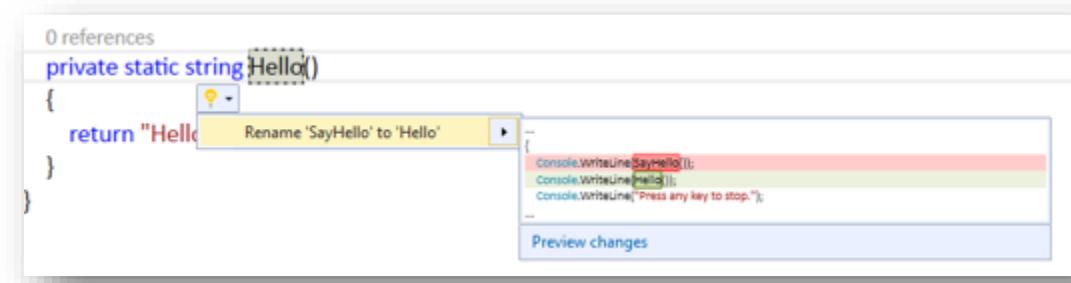
# 5. Hello Visual Studio - IDE

## ► Editor tools

- Find en Replace
  - In Menu Edit > Find and Replace
  - Quick Find: Ctrl F
  - Quick Replace: Ctrl H



- Als je de naam van een methode aanpast. Bvb SayHello in Hello, dan verschijnt een lampje. Mouse over lampje opent menu, zo kan je dit overal laten aanpassen



# 5. Hello Visual Studio - IDE

---

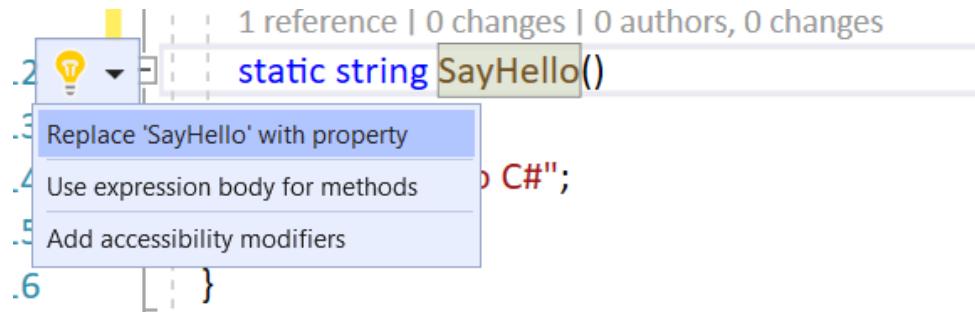
## ▶ Editor tools

- Refactoring: verbeteren van de code. Selecteer de code > Rechtsklik > Context menu (afhankelijk van de geselecteerde code)
  - [Change Method Signature](#)
  - [Encapsulate Field](#)
  - [Extract Interface](#)
  - [Extract Method](#)
  - [Inline Temporary Variable](#)
  - [Move Type to Matching File](#)
  - [Rename](#)
  - [Sync Type and Filename](#)
  - Meer op <https://docs.microsoft.com/nl-nl/visualstudio/csharp-ide/refactoring-csharp>

# 5. Hello Visual Studio - IDE

## ► Editor tools

- Quick Actions laten toe om snel je code te refactoren
  - Als je cursor in een code lijn plaatst en er zijn quick actions beschikbaar, dan verschijnt
    - screwdriver icon : er zijn acties beschikbaar, maar je hoeft ze niet toe te passen
    - light bulb : acties beschikbaar, die je best toepast om je code te verbeteren
    - error light bulb : acties beschikbaar die de fout in je code oplossen.
  - Om de acties te bekijken : Ctrl+; of klik dropdown bij icon
  - Bvb selecteer de methodeNaam SayHello

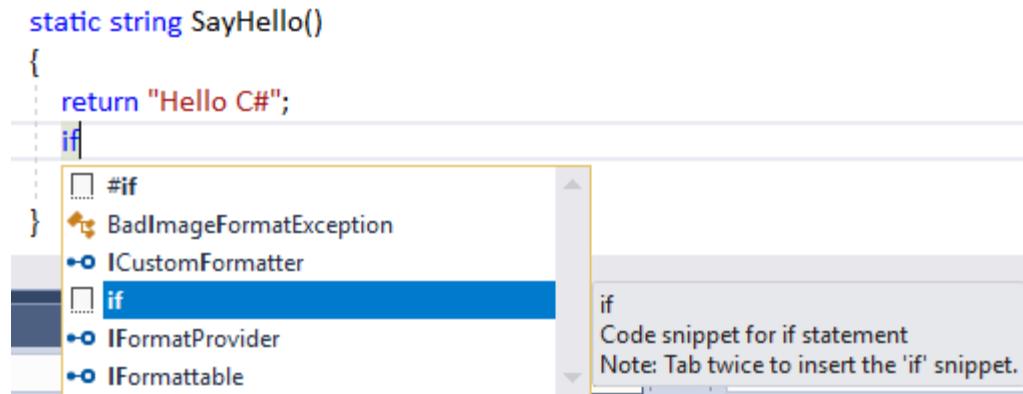


- Voor een overzicht van de common quick actions:  
<https://docs.microsoft.com/en-us/visualstudio/ide/quick-actions>
- De coding styles kan je instellen via Tools > Options > Text Editor > C# > Code Style > General

# 5. Hello Visual Studio - IDE

## ▶ Code snippets

- <https://docs.microsoft.com/nl-nl/visualstudio/ide/visual-csharp-code-snippets>
- Toevoegen van “ready made code” door snippet inserter
- Bvb typ if, en dan 2 \* tab

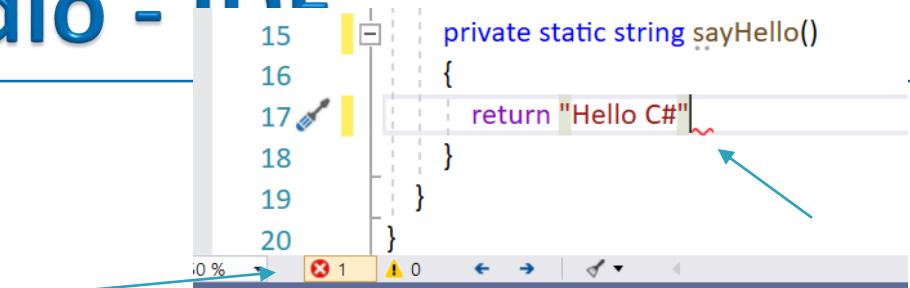


# 5. Hello Visual Studio - IDE

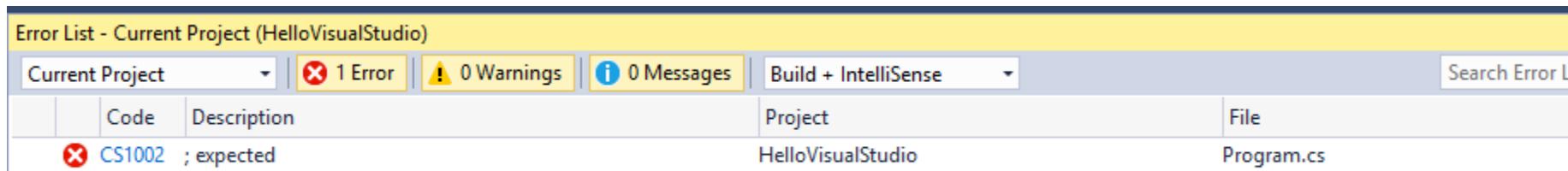
## ► Editor tools

- ErrorList

- Toont fouten, warnings tijdens editeren en compileren
- Dubbelklik een fout en je komt op de plaats in de code terecht
- Doe eens de ; weg bij SayHello methode
- In de editor krijg je een indicatie van de fouten
- Bij F5, Je krijgt popup venster, klik NO



- Hover over de foutindicatie of Ga naar ErrorList (Klik op fout icon onderaan of View > Error List): dubbelklik op fout, dan ga je naar de foutieve code.



# 5. Hello Visual Studio - IDE

## ► Editor tools

- Als de code rood onderlijnd is, dan verschijnt er meestal een error light bulb en ga dan naar potential fixes voor quick actions

A screenshot of the Visual Studio code editor showing a syntax error. The code is:

```
string PrintResult(int i, string s)
{
    int x = i * Math.; // Error here
    var resu s
}
```

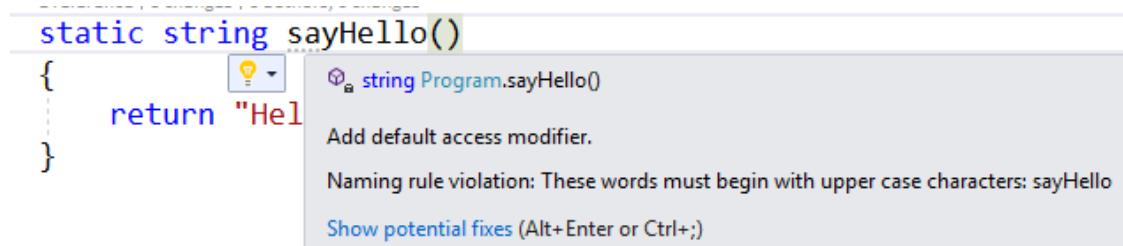
The word "Math" is underlined with a red squiggly line, indicating an error. A lightbulb icon appears over the period after "Math". A tooltip box is open, displaying the message: "The name 'Math' does not exist in the current context". Below the tooltip, there is a link labeled "Show potential fixes". At the bottom left of the editor, it says "0 references".

# 5. Hello Visual Studio - IDE

## ► Editor tools

### ◦ Analyzers

- Runnen in de background en analyseren de source code. Als ze code vinden die niet voldoet aan de regels wordt dit onderlijnd in de editor en staat er een warning/message in de Error List. Er verschijnt ook meestal een light bulb met de mogelijke code fixes bij hoveren
- Bvb start de naam van de methode SayHello met een kleine letter, wordt in groen onderlijnd of met stippelijnen

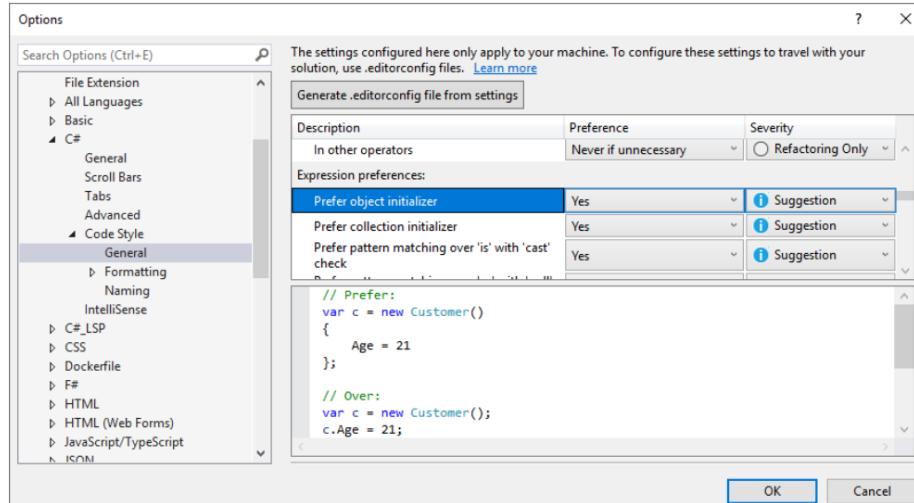


- Kan je configureren via de code styles

# 5. Hello Visual Studio - IDE

## ► Editor Tools

- Code styles
  - Tools > Options > Text Editor > C# > Code Style > General/Formatting/Naming
  - Voor elke code style kan je een preference en severity instellen. De Severity activeert de quick actions.
  - Van toepassing op je Visual Studio personalization account. Kan je exporteren naar een editorconfig file zodat ze van toepassing zijn voor iedereen die aan het project werkt (heeft voorrang)

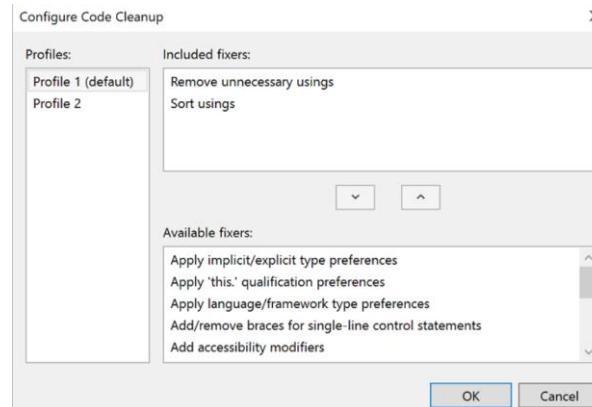


# 5. Hello Visual Studio - IDE

## ► Editor Tools

- Code styles

- Code cleanup button (onderaan)  of **Ctrl+K, Ctrl+E** past de code styles toe op de openstaande file.
  - Maar eerst dien je op te geven welke style je wenst toe te passen. Klik pijl naast broom icon, Configure code clean up.



- Code cleanup voor het volledige project. Rechtsklik solution in solution explorer > Analyze and Code Cleanup > Run Code Cleanup.
  - Meer op <https://docs.microsoft.com/en-us/visualstudio/ide/code-styles-and-code-cleanup?view=vs-2019>

# 5. Hello Visual Studio - IDE

## ▶ Editor tools

- Help
  - Staat online: <https://docs.microsoft.com/en-US/>
  - Help: Help > View Help (Visual studio help)
  - Context-sensitive help (F1)
    - Je selecteert een stukje code (bvb WriteLine) en drukt F1

## Console.WriteLine Method

Namespace: [System](#)

Assemblies: System.Console.dll, mscorlib.dll, netstandard.dll

Writes the specified data, followed by the current line terminator, to the standard output stream.

### Overloads

---

[WriteLine\(String, Object, Object\)](#)

Writes the text representation of the specified objects, followed by the current line terminator, to the standard output stream using the specified format information.

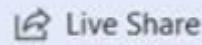
---

[WriteLine\(String\)](#)

Writes the specified string value, followed by the current line terminator, to the standard output stream.

# 5. Hello Visual Studio - IDE

## ▶ Live Share



- Eens aangemeld met je school account kan je een live share sessie starten
  - Klik op de Live Share optie om een collaboration session te starten : toont een “Sharing” menu en invitation link die je kan delen met andere team members.
  - Een team member kan een Live Share session joinen via “File > Join Collaboration Session” en ingave van de Live Share Invitation Link.
  - Tools > Options > Live Share > User account : instellen van je account
  - Meer op <https://youtu.be/dKLJsiK1QU8>

# En ... wat nu?

# 6. En ... wat nu?

---

- ▶ Hoofdstuk 2: C# (syntax, controle structuren, collections) Dit is zelfstudie.
- ▶ Het **Model** (Hoofdstuk 3):
  - Klassen – associaties – Overerving – Polymorfisme - Abstracte klasse – Interface - Statische members - Generic klasse - Naming Conventions
  - TDD: we maken ook de unit testen aan
- ▶ Hoe debug ik mijn code? Hoofdstuk 4
- ▶ Daarna verder met C# (Hoofdstuk 5: Linq)
- ▶ Dan **Controller-View** (Hoofdstuk 6)

# 6. En ... wat nu?

---

- ▶ Hoofdstuk 2 is zelfstudie
  - Neem de slides door
- OF
- Pluralsight : C# fundamentals in C#6.0
- Of
- <https://docs.microsoft.com/en-us/dotnet/csharp/tutorials/intro-to-csharp/index> (interactieve tutorials)
- ▶ Neem dit door tegen de volgende les
- ▶ Maak de oefening op volgende slide

# 6. Oefening bij hoofdstuk 2

## ▶ Oefening

- De rij van Fibonacci is genoemd naar Leonardo van Pisa, bijgenaamd Fibonacci, die de rij noemt in zijn boek Liber abaci uit 1202. De rij begint met 0 en 1 en vervolgens is elk volgende element van de rij steeds de som van de twee voorgaande elementen. De eerste elementen van de rij zijn dan als volgt:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, ...

- Vraag de gebruiker welk getal in de rij hij wenst te kennen. Bvb 5<sup>de</sup> getal. Resultaat is 3.
- Het programma berekent dan dit getal en geeft dit weer
  - Oplossing 1: bereken het getal
  - Oplossing 2: maak een array aan met de eerste 1000 getallen. Gebruik dan de index voor ophalen van het element

# Appendix

# Appendix: Community Resources

- ▶ Latest .net core documentation:  
<https://docs.asp.net/en/latest/>
- ▶ .net core repo: <https://github.com/dotnet/core/>
- ▶ Core clr repo: <https://github.com/dotnet/coreclr/>
- ▶ Core fx repo: <https://github.com/dotnet/corefx/>
- ▶ .Net Core documentation:  
<https://docs.microsoft.com/en-us/dotnet/core/index>
- ▶ .Net Core tutorials:  
<https://docs.asp.net/en/latest/tutorials/index.html>
- ▶ VS documentatie : <https://docs.microsoft.com/nl-nl/visualstudio/>

# Appendix: Community Resources

- ▶ ASP.NET Website - <http://www.asp.net/mvc>
- ▶ ASP.NET Community Standup: laatste nieuwtjes  
<https://live.asp.net/>
- ▶ MSDN: <http://msdn.microsoft.com>
- ▶ MSDN Blogs: <http://blogs.msdn.com>
- ▶ Vragen: <http://stackoverflow.com>
- ▶ Support: <http://support.microsoft.com>
- ▶ E-learning: <https://channel9.msdn.com/>
- ▶ .NET Glossary:<https://docs.microsoft.com/en-us/dotnet/standard/glossary>
- ▶ .Net Architecture :  
<https://www.microsoft.com/net/learn/architecture>