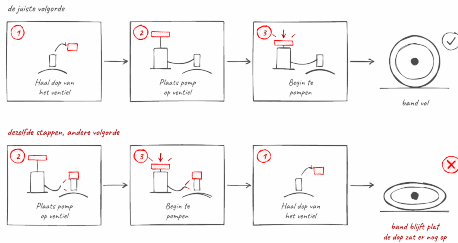




// 01

Programmeren en algoritmes

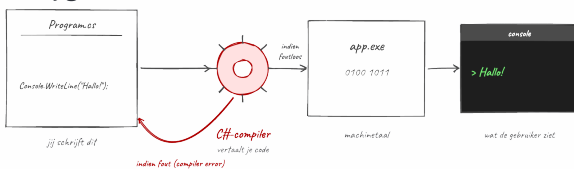
Een computer doet **exact** wat jij hem vertelt. Een **algoritme** is het recept: een reeks instructies die de computer uitvoert. De **volgorde** telt.



// 02

C# en de compiler

C# (spreek uit *siesjarp*) is een **hogere programmeertaal** van Microsoft, deel van **.NET**, bedacht door Anders Hejlsberg. De **compiler** vertaalt je code. Staat er ergens een fout in, dan maakt hij geen uitvoerbaar bestand.



Achter de schermen wordt C# eerst **IL-code**. De **CLR** zet die bij het uitvoeren *just-in-time* om naar machinecode.

// 03

Visual Studio, zonder A.I.

- **Visual Studio Community** is een **IDE**: editor, compiler en debugger in één
- Nieuw project: **Console App** met het groene C#-icoontje
- Vink **Do not use top-level statements** aan
- Starten met de **groene driehoek**: VS bewaart zelf. Nooit Save as...
- Layout kwijt? **Window** → **Reset window layout**
- Zet **IntelliCode** en **Copilot** voorlopig uit: je leert ook niet hoofdrekken met een zakrekenmachine

// 04

Je eerste programma

Dit skelet maakt Visual Studio voor je aan:

```
namespace Demo1
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

- **namespace**, **class** en **Main** negeer je voorlopig: je code komt **tussen de accolades van Main**
- Code loopt van **boven naar onder**, elke lijn eindigt op een **puntkomma ;**
- Alles na **//** is commentaar

// 05

WriteLine en ReadLine

Console.WriteLine toont tekst en springt naar een nieuwe lijn. **Console.ReadLine()** wacht tot de gebruiker op enter drukt en geeft de tekst terug. Toekennen gaat van rechts naar links.

```
Console.WriteLine("Wie ben jij?!");
string naam = Console.ReadLine();
Console.WriteLine(naam); // de inhoud
Console.WriteLine("naam"); // letterlijk
```

UITVOER

```
Wie ben jij?!
tim [enter]
tim
naam
```

Sneller typen: **cw** en twee keer tab geeft **Console.WriteLine()**;

// 06

Write, spaties en \$

Console.Write springt niet naar een nieuwe lijn. Spaties zijn ook tekens: zet ze **binnen** de aanhalingstekens. Met een **\$** voor de aanhalingstekens zet je een variabele tussen **{ }** in je zin (*string interpolation*).

```
Console.Write("Dag ");
Console.Write(naam);
Console.WriteLine(", hoe gaat het?");
Console.WriteLine($"Dag {naam}, hoe gaat het?");
```

UITVOER

```
Dag tim, hoe gaat het?
Dag tim, hoe gaat het?
```

// LET OP

Spaties buiten de " " negeert C#. Vergeet je de \$, dan staat er letterlijk **{naam}** op het scherm.

// 07

Fouten oplossen

Een fout krijgt een **rode kronkellijn**, en zolang die er staat, compileert je programma niet. Laat de **Error List** open: een klik brengt je naar de juiste lijn. Veel fouten tegelijk? Kijk net boven de eerste: meestal ontbreekt er een **;**. Het gele **lampje** stelt een oplossing voor, maar wees kritisch.

✗ **Console.Writeline**(Hallo) ✓ **Console.WriteLine**("Hallo");

// LET OP

C# is hoofdlettergevoelig: **Readline** en **Writeline** bestaan niet. En meldt VS bij het starten dat er nog fouten zijn: klik nooit op Yes en vink de checkbox niet aan. Dan draait je laatste werkende versie, niet je huidige code.

// 08

Kleuren

```
Console.BackgroundColor = ConsoleColor.Yellow;
Console.ForegroundColor = ConsoleColor.Black;
Console.WriteLine("Zwart op geel");
Console.ResetColor();
```

De kleur geldt voor de tekst die je **daarna** toont; wat al op het scherm staat, verandert niet. Alle kleuren staan in **ConsoleColor**, bv. **ConsoleColor.Red**. **ResetColor()** zet de standaardkleuren terug.