

# INTRO til C# - Opgavesæt 01

## S1 - Programming Introduction

### Opgavens fokusområder og læringsmål

- ☐ Du skal tilegne dig viden om at oprette en Console Application
- ☐ Du skal demonstrere oprettelse af en Console Application
- ☐ Du skal udvikle en simpel Console Application
- ☐ Du skal udvikle en Console Applikation der kan udskrive til konsollen
- ☐ Du skal oprette variabler og tildele dem værdier
- ☐ Du skal kunne manipulere/ændre værdien i en variabel



### Kort fortalt

Formålet med opgaven er, at give dig et indblik i, hvordan man opretter et nyt projekt i Visual Studio. I denne opgave skal du benytte projektypen: Console App (.NET Framework)

Til hver opgave skal du oprette et nyt projekt, som skal navngives som beskrevet i hver opgave.

### Praktiske oplysninger

Alle dine løsninger skal gemmes under følgende struktur: C:\CodeFolder\S1\Console  
Hvis denne struktur ikke er oprettet på din PC, skal du oprette den.

### Ressourcer

Du skal benytte Visual Studio 2022 Enterprise Edition – opdateret til nyeste version.

## Opgave 01

Åben Visual Studio 2022 (VS)

Klik på knappen *Create a new project*

I det næste vindue der åbnes, får du mulighed at vælge den type projekt du vil lave.

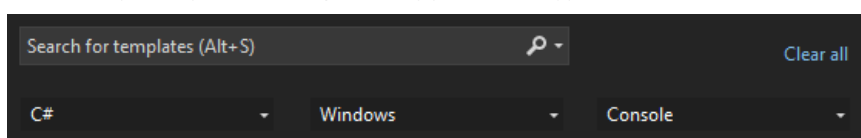
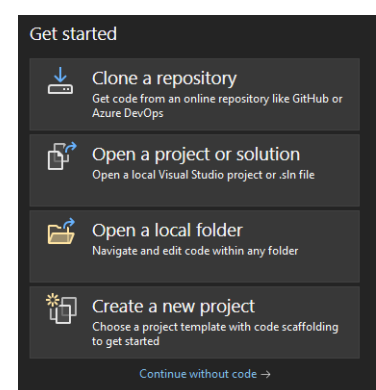
For at minimere mulige valg, kan man ved hjælp af tre dropdown bokse specificere hvilken type man skal bruge.

I dette tilfælde skal du lave en konsol applikation, derfor skal du vælge:

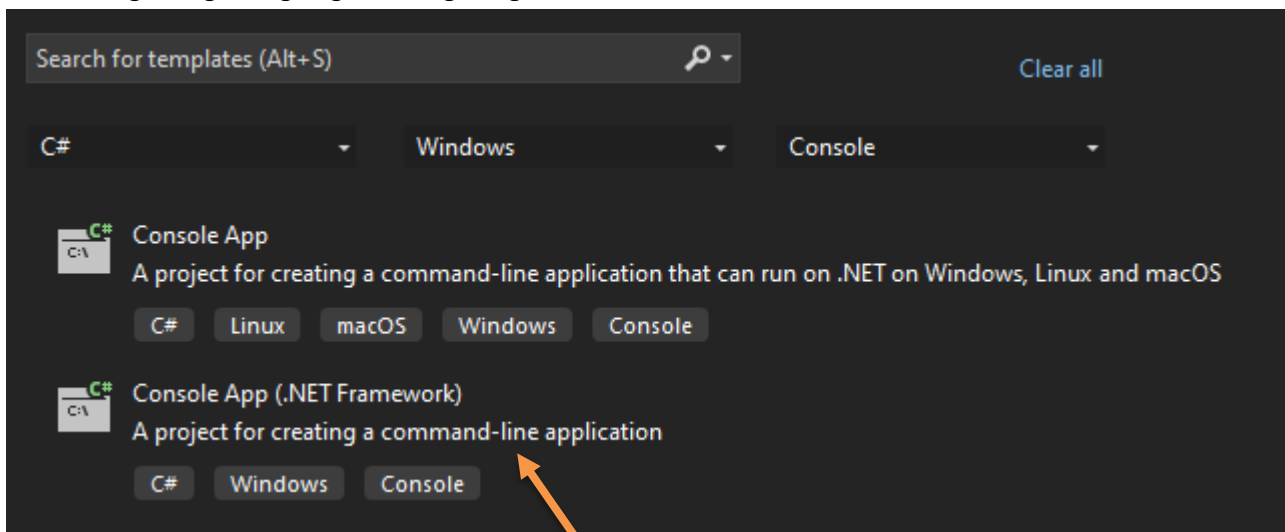
I den første dropdown vælger du sprog – C#

I den anden dropdown vælger du platform – Windows

I den tredje dropdown vælger du applikationstype - Console



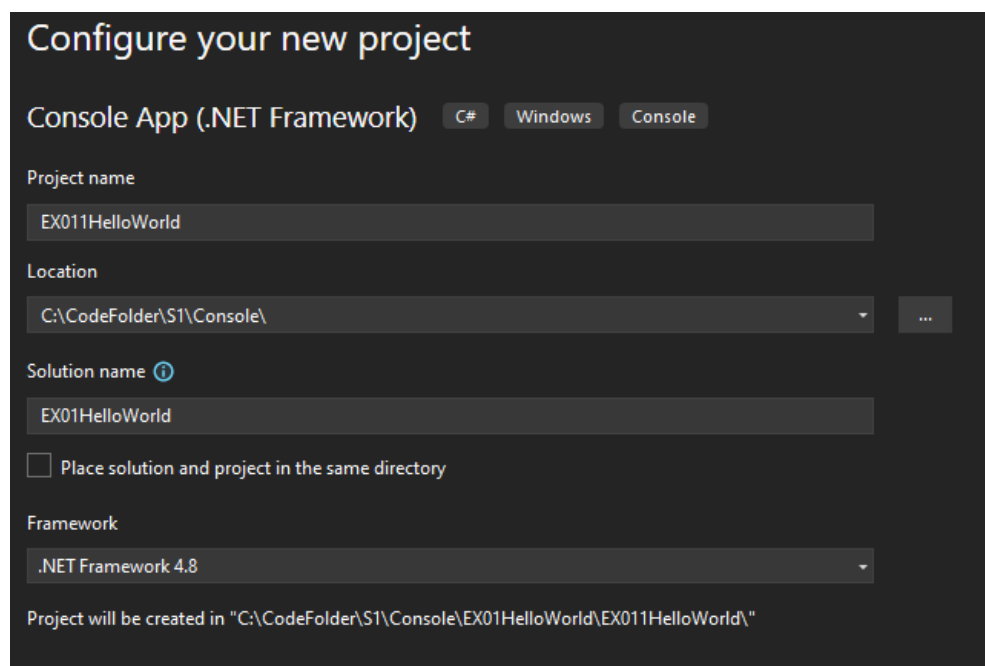
Det skulle gerne give dig følgende valgmuligheder:



Du skal klikke på: *Console App (.Net Framework)* og klik på knappen *Next* i nederste højre hjørne.

Nu fremkommer der et nyt vindue, hvor du skal konfigurere dit nye projekt.

Du får vist følgende:



Du skal nu navngive dit projekt feltet: *Project name*

Du skal indtaste følgende som projektnavn: EX01HelloWorld

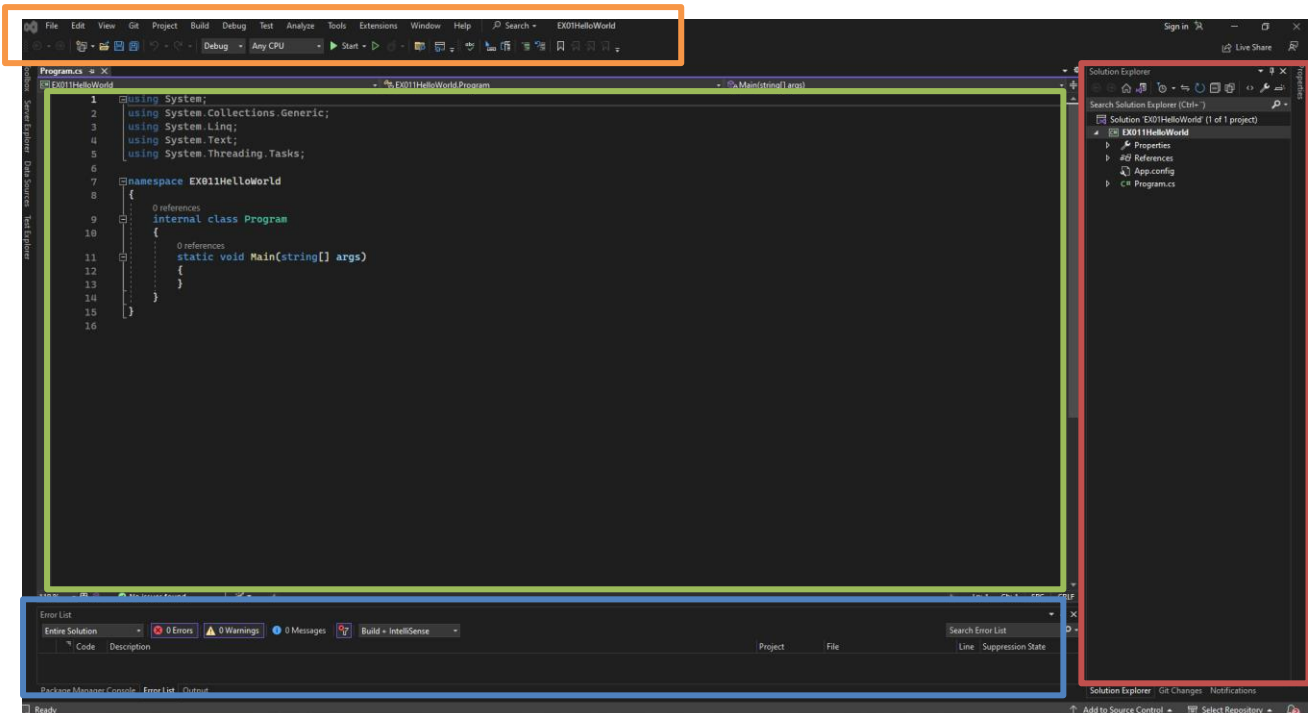
I feltet *Location*, skal du angive stien til den mappe hvor projektet skal oprettes.

Du skal angive følgende sti: C:\CodeFolder\S1\Console\

De øvrige felter skal du IKKE ændre noget i.

Klik på knappen *Create*.

Nu bliver der oprettet et nyt projekt i VS, hvor du kan skrive C# kode i en Console Application.



Brugergrænsefladen i VS har under udvikling af konsol applikationer 4 vigtige funktionsområder:

1. **Hovedmenu:** Indeholder en tekst og en ikon baseret menulinje.  
Den tekstbaserede menu er til alle funktioner som kan benyttes i denne type projekt.  
Den ikon baserede menu er til de funktioner fra menuen som oftest bliver benyttet.  
**Den ORANGE firkant**
2. **Solution Explorer:** Giver dig et overblik over de projekter din solution indeholder, og hvilke filer hvert projekt indeholder.  
**Den RØDE firkant**
3. **Compiling in- & output:** Her får man vist de eventuelle fejl man har skrevet i sin kode. Hvis du ikke kan se den i din VS, skal du benytte hovedmenuen og klikke følgende: **View – Error List**  
Så skulle den komme frem. Brugen af den kommer vi til senere.  
**Den BLÅ firkant**
4. **Code Editor:** Det er her du skal skrive din kode. Helt præcist hvor og hvordan kommer vi til om lidt, hvor du vil få en mere udførlig introduktion til *Code Editor*.  
**Den GRØNNE firkant**

Der er mange flere menuer og funktioner i VS, men de fire du har fået vist her er dem du får brug for i denne og de næste opgaver.

Det næste du skal have fokus på er det område indkredset af den grønne firkant, nemlig din Code Editor.

```

Program.cs  X
C# EX011HelloWorld  EX011HelloWorld.Program

1  using System;
2      using System.Collections.Generic;
3      using System.Linq;
4      using System.Text;
5      using System.Threading.Tasks;
6
7  namespace EX011HelloWorld
8  {
9      0 references
      internal class Program
10     {
11         0 references
         static void Main(string[] args)
12     {
13     }
14 }
15 }
16

```

Code Editoren opdeler sit indhold i to overordnede områder:

- **Linje nr. 1 – 5:** Her vises hvilke biblioteker din applikation som standard implementerer fra det store .NET Framework. Disse biblioteker indeholder de funktioner der skal benyttes for at kunne udføre de helt basale funktioner i en applikation.
- **Linje nr. 7 – 15:** Her vil der altid blive indsat den standard kodestruktur der skal benyttes i denne type applikation.
  - **Linje 7:** C# er et objektorienteret programmeringssprog (OOP) og i linje 7 angives det ydre objekt for vores applikation som kaldes *namespace* og er navngivet som projektet. Objektet *namespace* er indkapslet med "{" og "}" i linje 8 og 15.
  - **Linje 9:** Inde i objektet *namespace* vil der altid være en til flere objekter af typen *class*. Et objekt af typen *class* kan indeholde de elementer der kan kompileres til maskinkode og vil blive udført på computeren. Objektet *class* er indkapslet med "{" og "}" i linje 10 og 14.
  - **Linje 11:** Inde i objektet *class* vil der altid være en til flere *metoder*. I projekter af typen Console App (.NET Framework), vil der altid være en metode med angivelsen: `static void Main(string[] args)`  
Det er den metode som VS's compiler vil starte ud fra, og det er her du skal skrive din kode. Du skal dog først lige lave lidt plads til det, det gøres ved at klikke til højre for { i linje 12 og taste *enter* 3 gange.

```
static void Main(string[] args)
{
}
```

```
static void Main(string[] args)
{
    //
}
```

Du er nu klar til at skrive din første kode der vil frembringe noget på din skærm.

Inden du er så langt, skal du dog have oprettet noget der kaldes en *variabel*.

En *variabel* skal opfattes som en elektronisk beholder der kan indeholde data.

En variabel kan kun indeholde en type data, indtil videre vil fokusere på to typer: tekst og tal.

- Datatypen til tekst hedder: **string**
- Datatypen til heltal (tal uden decimaler) hedder: **int** (forkortelse af integer).

Når man skal oprette en variabel, skal man altid gøre det efter en fast skabelon:

Først angiver man **datatypen**, så en **identifier** og til sidst giver man variabelen et **indhold** (value).

Datatype – Navn – Værdi  
**string schoolName = "AspIT"**

```
static void Main(string[] args)
{
    string schoolName = "AspIT";
}
```

**Bemærk at kommandolinjen afsluttes med et semikolon ; Det skal alle kommandolinjer altid !**

Nu skal vi have koden til at udskrive indholdet af variabelen *schoolName* i den *console* der åbnes når vi starter programmet. Det kan vi gøre ved at benytte en statisk klasse der er tilgængelig via .NET Framework biblioteket *System*.

Klassen du skal benytte er: *Console*

Klassen *Console* indeholder flere metoder, du skal først benytte metoden *WriteLine()*.

Det gøres på følgende måde – skriv: *Console.WriteLine()*;

Metoden *WriteLine()* vil udskrive indeholder der er angivet inden for metodens paranteser, det kalder man en parameter. Men du har jo ikke skrevet noget, det gør du nu = *Console.WriteLine(schoolName)*;

Hvis man vil være helt sikker på, at man får vist resultatet af sine anstrengelser, skal man lige tilføje en ekstra linje kode:

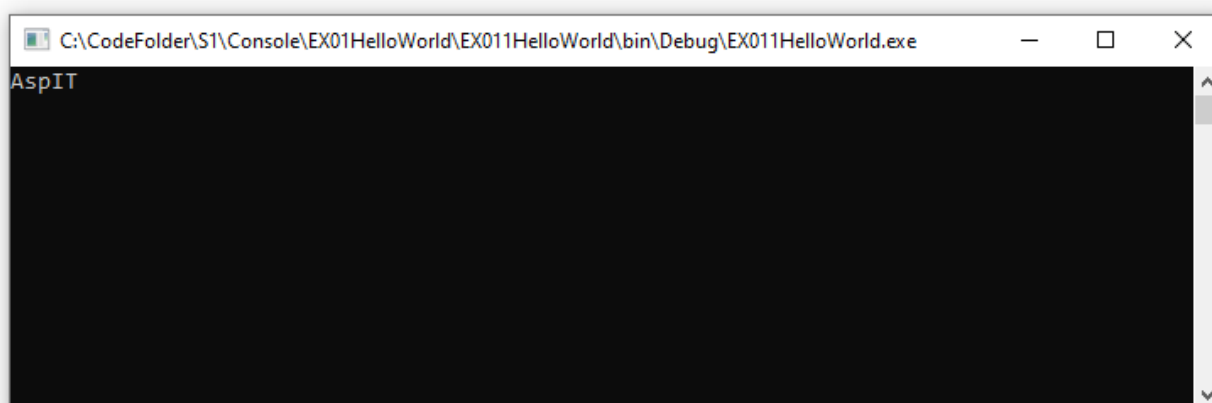
*Console.ReadLine()*;

Dette vil sikre, at konsolvinduet ikke lukker før brugeren taster enter.

```
static void Main(string[] args)
{
    string schoolName = "AspIT";
    Console.WriteLine(schoolName);
}
```

```
static void Main(string[] args)
{
    string schoolName = "AspIT";
    Console.WriteLine(schoolName);
    Console.ReadLine();
}
```

Tast F5 – og efter et kort øjeblik, skulle du gerne få vist et konsolvindue med et indhold som vist på næste side.



Hvis du får vist dette – TILLYKKE !!!!!