

WEEK

task

C# Variables

- .int - stores integers (whole numbers), without decimals such as 123 or -123
- .double - stores floating point numbers, with decimals such as 19.99 or -19.99
- .char - stores single characters, such as 'a' or 'B'. Char values are surrounded by single quotes
- .string - stores text, such as "Hello World". String values are surrounded by double quotes
- .bool - stores values with two states: true or false



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

Line 1: using System means that we can use classes from the System namespace.
Line 2: A blank line. C# ignores white space. However, multiple lines makes the code more readable.
Line 3: namespace is used to organize your code, and it is a container for classes and other namespaces.
Line 4: The curly braces {} marks the beginning and the end of a block of code.
Line 5: class is a container for data and methods, which brings functionality to your program. Every line of code that runs in C# must be inside a class. In our example, we named the class Program.
Line 7: Another thing that always appear in a C# program is the Main method. Any code inside its curly brackets {} will be executed. You don't have to understand the keywords before and after Main. You will get to know them bit by bit while reading this tutorial.
Line 9: Console is a class of the System namespace, which has a WriteLine() method that is used to output/print text. In our example, it will output "Hello World".
If you omit the using System line, you would have to write System.Console.WriteLine() to print/output .text
Note: Every C# statement ends with a semicolon
Note: C# is case-sensitive: "MyClass" and "myclass" have different meaning.

..../footage/video.mp4

- نوع البيانات `char` يستخدم لتخزين حرف واحد فقط (مثل 'A' أو 'b' أو 'ا' أو '\$'). يتم تخزينه باستخدام 2 بايت (16 بت).
- لا يمكن استخدام علامة تصيير مزدوجة ("") لأنها تمثل `string`.
- وليس `char`.
- يمكن تحويل `char` إلى عدد (ASCII/Unicode) والعكس.
- يمكن استخدام دوال `char` لفحص نوع الحرف (IsLetter, IsDigit, etc).
- يمكن استخراج حروف من `string` وتحويل `char` إلى `string`.

Boolean Data Type

- `bool` نوع بيانات يستخدم لتمثيل القيم `true` و `false`.
- يُستخدم في العمليات المنطقية (&&, ||, !).
- يدخل في الشرط (if, while).
- يمكن مقارنته باستخدام ==, !=, <, >, <=, >=.
- القيم الرقمية يمكن تحويلها إلى `bool` أي رقم آخر (true, false).
- يمكن استخدام `bool?` للسماح بقيمة `null`.

WEEK

task

C# Variables

- .int - stores integers (whole numbers), without decimals such as 123 or -123
- .double - stores floating point numbers, with decimals such as 19.99 or -19.99
- .char - stores single characters, such as 'a' or 'B'. Char values are surrounded by single quotes
- .string - stores text, such as "Hello World". String values are surrounded by double quotes
- .bool - stores values with two states: true or false



C# Numeric Data Types

- الأعداد الصحيحة (...int, long, short) تُستخدم للأرقام بدون فاصلة عشرية.
- الأعداد العشرية (float, double, decimal) تُستخدم للأرقام مع فاصلة عشرية.
- العمليات الحسابية تشمل +, -, *, /, %. التحويل الضمني يحدث تلقائيًا، بينما التحويل الصريح يحتاج إلى .casting

Data Types	Memory Size	Range
char	1 byte	-128 to 127
signed char	1 byte	-128 to 127
unsigned char	1 byte	0 to 255
short	2 byte	-32,768 to 32,767
signed Short	2 byte	-32,768 to 32,767
unsigned Short	2 byte	0 to 65,535
int	4 byte	-2,147,483,648 to 2,147,483,647
signed int	4 byte	-2,147,483,648 to 2,147,483,647
unsigned int	4 byte	0 to 4,294,967,295
long	8 byte	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
signed long	8 byte	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
unsigned long	8 byte	0 to 18,446,744,073,709,551,615
float	4 byte	1.5×10^{-45} to 3.4×10^{38} (7 Digit)
double	8 byte	5×10^{-324} to 1.7×10^{308}
decimal	16 byte	-7.9×10^{-28} to 7.9×10^{28}

C# String Data Type

- نوع البيانات string يُستخدم لتخزين النصوص (مجموعة من الأحرف char). يعني بمحظوي على دوال وخصائص يسهل التعامل مع النصوص. كائن (object) يمكن تعريف string باستخدام علامة تنصيص مزدوجتين " ". " (نص فارغ) يعني أن النص موجود ولكن لا يحتوي على أي حروف. null يعني عدم وجود قيمة على الإطلاق.
- دمج النصوص (Concatenation) ""\$ (Interpolation) ""\$ (Length) حساب طول النص (char index) الوصول إلى حرف معين (@) يمكن استخدام @ لإنشاء نص يحتوي على أسطر متعددة

Arithmetic Operators

Arithmetic Operator	Name	Example
+	Addition	a + b
-	Subtraction	a - b
*	Multiplication	a * b
/	Division	a / b
%	Modulus	a % b
**	Exponentiation	a ** b
++	Increment Operator	a++
--	Decrement Operator	a--

WEEK

task

assignment operators



Assignment Operator	Name	Example
=	Assignment Operator	$c = a + b$
+=	Add AND assignment	$a += b$ is same as $a = a + b$
-=	Subtract AND assignment	$a -= b$ is same as $a = a - b$
*=	Multiply AND assignment	$a *= b$ is same as $a = a * b$
/=	Divide AND assignment	$a /= b$ is same as $a = a / b$

parsing

- تحليل النصوص (Parsing) يعني تحويل القيم النصية (string) إلى أنواع بيانات أخرى مثل `int`, `double`, `bool` مثل

C# comparison operators

تُستخدم معاملات المقارنة (Comparison Operators) في C# لمقارنة القيم وإرجاع `false` أو `true` كنتيجة.

Relational Operator	Name	Example
= =	Equal	$a == b$ is not true
!=	Not Equal	$a != b$ returns true
>	Greater Than	$a > b$ is not true
<	Less Than	$a < b$ is true
>=	Greater than or equal to	$a >= b$ is not true
<=	Less than or equal to	$a <= b$ is true

if

تُستخدم جملة `if` لتنفيذ كود معين بناءً على شرط معين. لو اتحقق الشرط، يتم تنفيذ الكود، ولو متحقق، يتم تخطيه أو تنفيذ `else` (إذا كانت موجودة).

WEEK

task

C# For Loop

تُستخدم حلقات `for` لتنفيذ كود معين عدة مرات طالما أن الشرط صحيح.

- ◆ **initialization** → تعيين متغير عداد البداية.
- ◆ **condition** → يكون صحيحاً ليستمر التكرار.
- ◆ **iteration** → تغيير قيمة العداد في كل دورة.



Arrays

المصفوفة (Array) هي مجموعة من العناصر ذات نفس النوع يتم تخزينها تحت اسم واحد في الذاكرة. يمكن الوصول إلى عناصر المصفوفة باستخدام الفهرس (index) الذي يبدأ من ٠.

```
string[] cars = {"Volvo", "BMW", "Ford",  
                 "Mazda"};  
Console.WriteLine(cars[0]); // Outputs Volvo
```