**Data Types:**

***- Data types*** represent the different types of information that we can use in our programs and how they should be used.  
- Without data types, computers would try and perform processes that are impossible, like squaring a piece of text or capitalizing a number.  
- Tell us how data can be stored, what operations we can perform with it, different methods it can be used with

A screenshot of a computer program

Description automatically generatedA computer screen with text and numbers

Description automatically generated

Most common data types:

A screenshot of a computer

Description automatically generated  
  
- C# is **strongly-typed**, meaning it requires us to specify the data types for values that we’re using.   
- It is also **statically-typed**, which means it will check that we used the correct types before the program even runs.  
- Both language features are important because they help write scalable code with fewer bugs.

- When we use data in our programs, it’s good practice to save them in a **variable**. A variable is basically like a box in our computer memory where we can store values used in our code.  
- Every time we specify a variable, we have to declare what data type that variable is going to hold. Once defined, it can be used throughout the whole program  
- Can declare variables on one or two lines:

A black background with white text

Description automatically generated A black background with white text

Description automatically generated

- **Variables** can only contain underscores, letters, and digits  
- Variables cannot contain **Reserved Keywords**

**Errors in C#:**

A screen shot of a computer code

Description automatically generated

Errors in C# come in three parts:

1. The full local path of the file where the error occurred, the numbers that follow in parentheses denote the line and character of the file responsible for the error, respectively
2. Next is the actual error, which will inform us what in our code is incorrect. Every kind of error in C# has a unique identifier, like “CS0103” which can be found in the [Microsoft documentation](https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/compiler-messages/cs0103)
3. Finally, a C# program requires a special type of file using the .csproj extension, which dictates build and deploy metadata for the project.

**Converting Data Types:**

- C# checks to make sure that no data is lost during conversion, otherwise it won’t be allowed  
- **Implicit Conversion** - happens automatically if no data will be lost in the conversion (int 🡪 double)  
**- Explicit Conversion -** requires a cast operator to convert a data type into another one. If we do want to convert a double to an int, we could use the operator (int)

Using Built in Methods:

- For most data types, there is a built in method for conversion: Convert.ToX(), Convert.ToString(), Convert.ToDouble()  
- Full list can be [found here](https://learn.microsoft.com/en-us/dotnet/api/system.convert?view=netframework-4.7.2)

**Notes:**

- Console.ReadLine() will always return a string even if values entered are integer or decimal  
- [How to Convert a String to a Number](https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/types/how-to-convert-a-string-to-a-number)boo article