Переменная - именованная область памяти, к которой можно обращаться за данными, используя идентификатор (имя переменной). Данные, находящиеся в переменной (то есть по её адресу в памяти), называются значением этой переменной.

В C# все типы данных делятся на две большие категории: **значимые типы (Value Types)** и **ссылочные типы (Reference Types)**. Основное различие между ними заключается в том, где хранятся их данные в памяти и что происходит при их присваивании или передаче в метод.

Переменная значимого типа напрямую содержит свое значение.

- значимые: int, double, bool, char

Переменная ссылочного типа хранит не сам объект, а **ссылку (или адрес)** на место в памяти, где этот объект находится.

- Ссылочные: `string`, все массивы (`int[]`, `string[]`), все классы (`class`), `object`, делегаты.

Как и во многих языках программирования, в C# есть своя система типов данных, которая используется для создания переменных. Тип данных определяет внутреннее представление данных, множество значений, которые может принимать объект, а также допустимые действия, которые можно применять над объектом.

C# строго типизированный язык. Это означает, что **компилятор проверяет код на ошибки, связанные с типами, ещё до его выполнения**.
Пример: в питоне или JS вы получите ошибку связанную с типом только во время итерации программы, в шарпе до запуска IDE нас предупредит.

IDE - Integrated development environment. То, где мы прогаем. Visual Studio в нашем случае.

- bool: хранит значение `true` или `false` (логические литералы). Представлен системным типом `System.Boolean`

```
    bool alive = true;
    bool isDead = false;
    ```

- byte: хранит целое число от `0` до `255` и занимает 1 байт. Представлен системным типом `System.Byte`
- sbyte: хранит целое число от `-128` до `127` и занимает 1 байт. Представлен системным типом `System.SByte`
```
```

- short: хранит целое число от `-32 768` до `32 767` и занимает 2 байта. Представлен системным типом `System.Int16`
- ushort: хранит целое число от `0` до `65 535` и занимает 2 байта. Представлен системным типом `System.UInt16`
- int: хранит целое число от `-2 147 483 648` до `2 147 483 647` и занимает 4 байта. Представлен системным типом `System.Int32`. Все целочисленные литералы по умолчанию представляют значения типа int:
- uint: хранит целое число от `0` до `4 294 967 295` и занимает 4 байта. Представлен системным типом `System.UInt32`
- long: хранит целое число от `-9 223 372 036 854 775 808` до `9 223 372 036 854 775 807` и занимает 8 байт. Представлен системным типом `System.Int64`
- ulong: хранит целое число от `0` до `18 446 744 073 709 551 615` и занимает 8 байт. Представлен системным типом `System.UInt64`
- float: хранит число с плавающей точкой от `-3.4*10^38` до `3.4*10^38` и занимает 4 байта. Представлен системным типом `System.Single`. 7 знаков после запятой.
- double: хранит число с плавающей точкой от `±5.0*10-324` до `±1.7*10308` и занимает 8 байта. Представлен системным типом `System.Double`. 16 знаков после запятой.
- decimal: хранит десятичное дробное число. Если употребляется без десятичной запятой, имеет значение от ±1.0*10-28 до ±7.9228*1028, может хранить 28 знаков после запятой и занимает 16 байт. Представлен системным типом `System.Decimal`. 29 значащих цифр.
- char: хранит одиночный символ в кодировке Unicode и занимает 2 байта. Представлен системным типом `System.Char`. Этому типу соответствуют символьные литералы:
- string: хранит набор символов Unicode. Представлен системным типом `System.String`. Этому типу соответствуют строковые литералы.
- object: может хранить значение любого типа данных и занимает 4 байта на 32-разрядной платформе и 8 байт на 64-разрядной платформе. Представлен системным типом `System.Object`, который является базовым для всех других типов и классов .NET.

```csharp
object a = 22;
object b = 3.14;
object c = "hello code";
```

```csharp
string name = "Tom";
int age = 33;
bool isEmployed = false;
double weight = 78.65;
```

Неявная типизация.

Мы можем использовать и модель неявной типизации

```csharp
var hello = "Hell to World";
```

```
var c = 20;
```