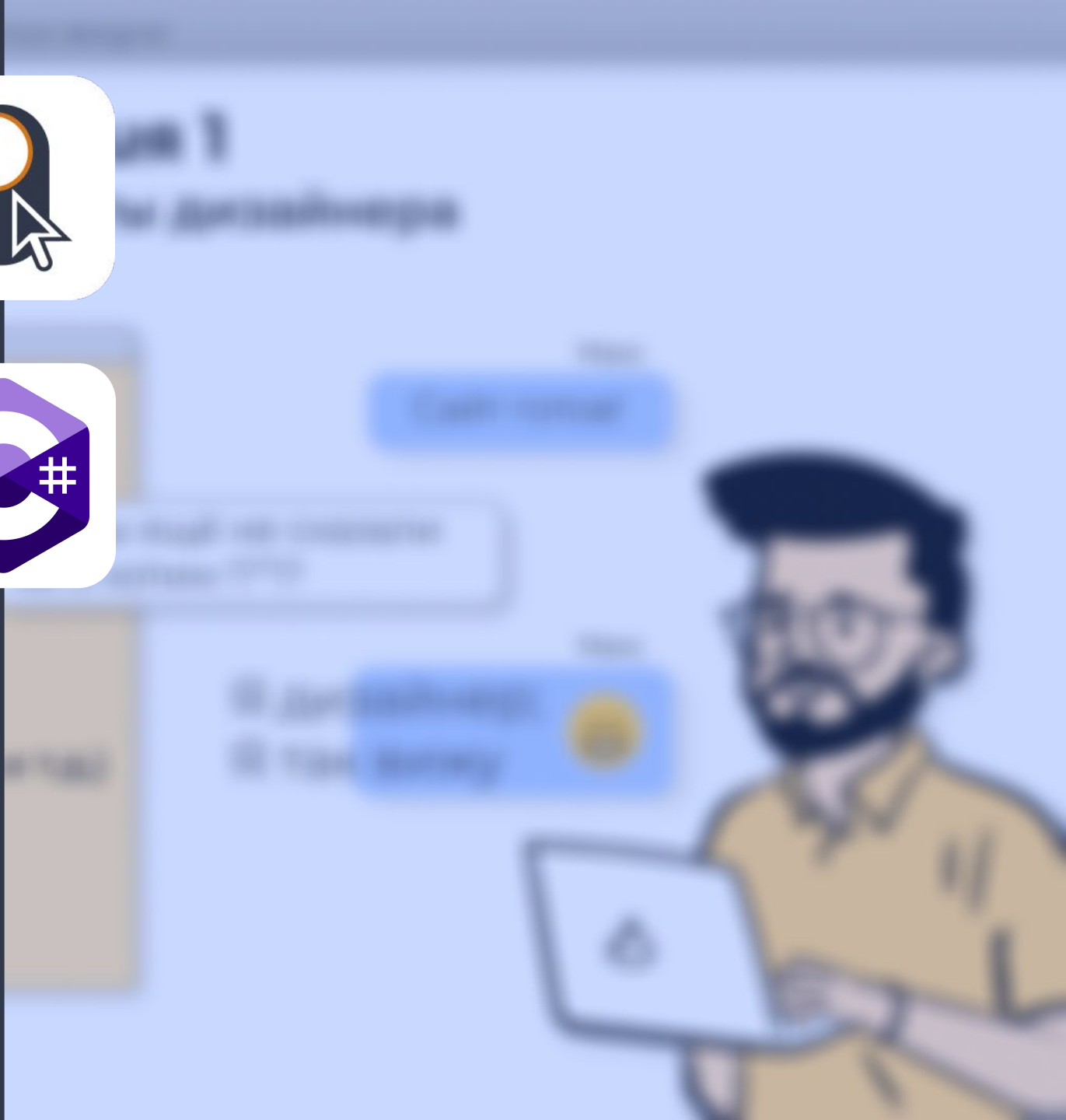


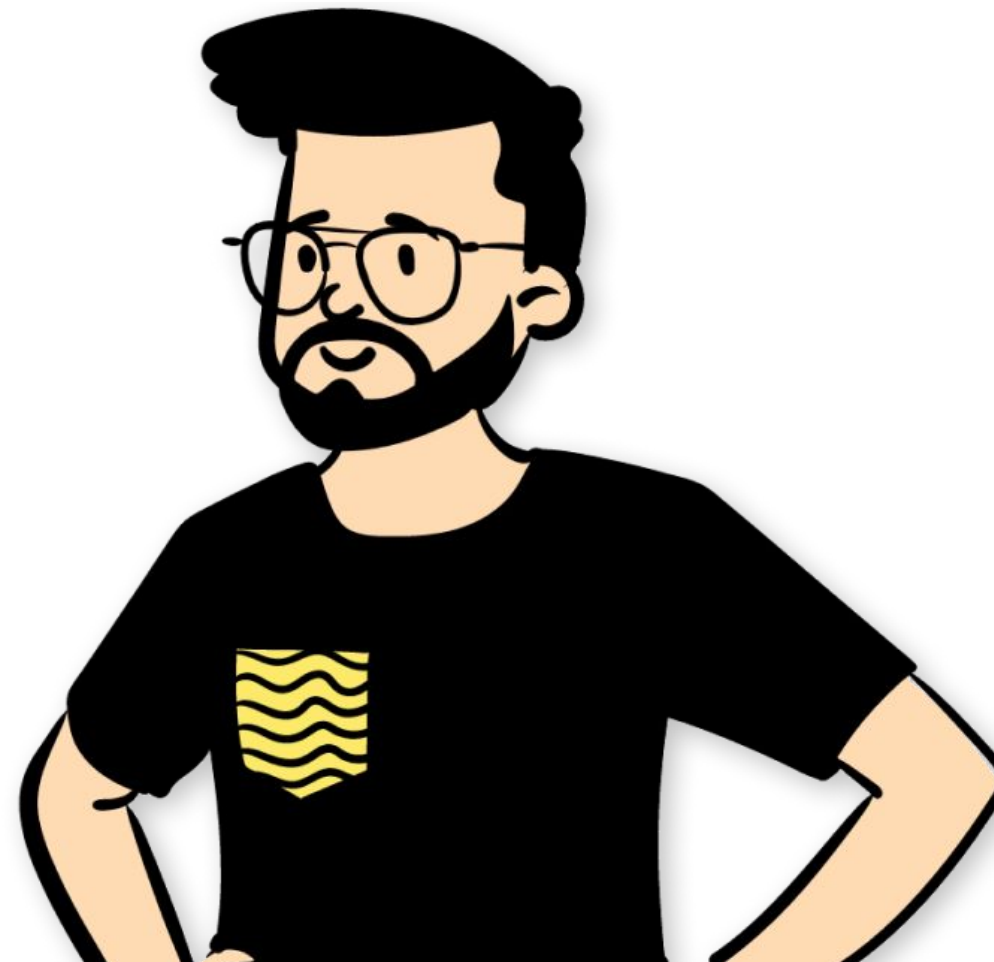
# 2

Программирование  
на C#  
Переменные и  
типы данных в  
языке C#



# Что такое переменные?

**Переменные** невероятно важны, так как позволяют хранить информацию и использовать её в дальнейшем. Вначале может быть не совсем понятно зачем вообще что-то записывать в переменную, если можно просто оперировать значениями без них. Понимание переменных придет немного позже, когда мы начнем создавать более сложные программы и нам потребуется хранить информацию в каком-либо месте.

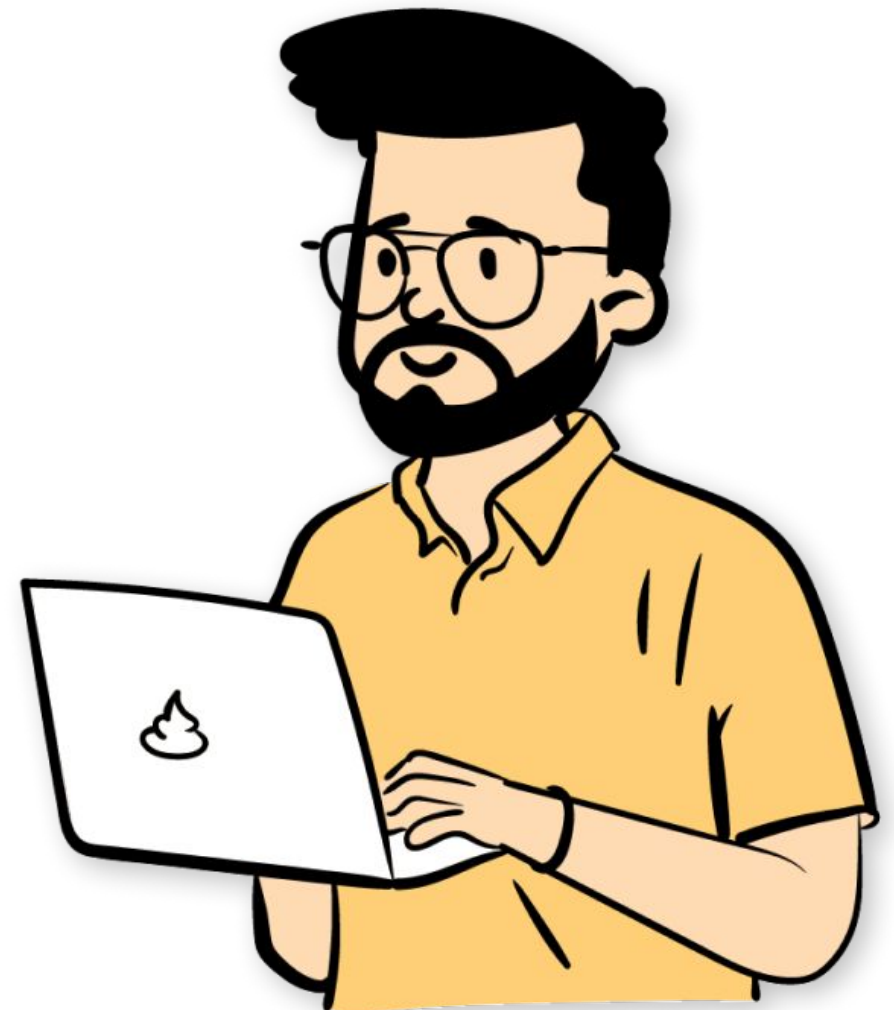


# Какие типы переменных существуют в C#?

Типы переменных в языке **C#** указываются перед названием переменной. От конкретного типа данных будет зависеть содержимое переменной. Если мы укажем что переменная с типом данных `int` (*целые числа*), то в неё мы не сможем записать строку или число с точкой.

В ходе программы мы можем записывать новые значения в переменную, но тип данных должен оставаться неизменным:

```
float some_value = 2.95f;  
some_value = 6.9f; // Записали новое значение
```



# Целые числа

Рассмотрим все типы данных для создания переменных.

## Целые числа:

- **byte** - предназначен для хранения целых чисел. Диапазон чисел от 0 до 255, если число будет большим 255 или меньшим 0 – выдастся ошибка;
- **short** - предназначен для хранения целых чисел. Диапазон чисел от -32 768 до 32 767;
- **ushort** - предназначен для хранения целых положительных чисел. Диапазон чисел от 0 до 65 535;
- **int** - предназначен для хранения целых чисел. Диапазон чисел от -2147483648 до 2147483647;
- **long** - предназначен для хранения целых чисел. Диапазон чисел от -9 223 372 036 854 775 808 до 9 223 372 036 854 775 807.



# Числа с точкой

Разница между типами заключается только в диапазоне чисел, который можно записать в переменную. Также не используйте большие типы данных по типу long, если число маленькое, так как чем больше диапазон, тем больше памяти требуется компьютеру на конкретную переменную.

## Числа с точкой:

- **float** - для создания чисел с плавающей точкой.  
Диапазон чисел от от  $-3.4 \cdot 10^{38}$  до  $3.4 \cdot 10^{38}$ ;
- **double** - для создания чисел с плавающей точкой.  
Диапазон чисел от от  $\pm 4.9 \cdot 10^{-324}$  до  $\pm 1.8 \cdot 10^{308}$ .



## Прочие типы данных

- **bool** - логический тип данных. Предназначен для записи истинного (true) или ложного (false) значения;
- **char** - тип данных для работы с символами. Позволяет поместить в одинарных кавычках какой-либо символ;
- **string** - тип данных для текста. В двойных кавычках можно сохранить одно или множество слов.



# Типы данных в языке C# в формате таблицы:

Тип	Тип .NET	Размер	Байт	Диапазон значений	
				MIN	MAX
<b>Числовые (целые типы)</b>					
<code>byte</code>	<code>Byte</code>	8-разрядное значение без знака;	1	0	255
<code>sbyte</code> <sup>1</sup>	<code>SByte</code>	8-разрядное значение со знаком;	1	-128	127
<code>short</code>	<code>Int16</code>	16-разрядное значение со знаком;	2	-32 768	32 767
<code>ushort</code> <sup>2</sup>	<code>UInt16</code>	16-разрядное значение без знака;	2	0	65 535
<code>int</code>	<code>Int32</code>	32-разрядное значение со знаком;	4	-2 147 483 648	2 147 483 647
<code>uint</code> <sup>2</sup>	<code>UInt32</code>	32-разрядное значение без знака;	4	0	4 294 967 295
<code>long</code>	<code>Int64</code>	64-разрядное значение со знаком;	16	9 223 372 036 854 775 808	9 223 372 036 854 775 807
<code>ulong</code> <sup>2</sup>	<code>UInt64</code>	64-разрядное значение без знака;	16	0	18 446 744 073 709 551 615
<b>Числовые (с плавающей точкой)</b>					
<code>float</code>	<code>Single</code>	32-разрядное значение с плавающей точкой в стандарте IEEE. Точность 4 – 9 цифр;	4	$\pm 1,5 \times 10^{-45}$	$\pm 3,4 \times 10^{38}$
<code>double</code>	<code>Double</code>	64-разрядное значение с плавающей точкой в стандарте IEEE. Точность 15 – 17 цифр;	8	$\pm 5,0 \times 10^{-324}$	$\pm 1,7 \times 10^{308}$
<code>decimal</code>	<code>Decimal</code>	128-разрядное значение с плавающей точкой, повышенной точности. Точность 28 – 29 цифр;	16	$\pm 1,0 \times 10^{28}$	$\pm 7,9228 \times 10^{28}$
<b>Символьные типы</b>					
<code>char</code>	<code>Char</code>	16-разрядный символ <code>Unicode</code> ( <code>char</code> иногда не представляет 8-разрядное значение, как в неуправляемом коде на C++);	2		?
<code>string</code>	<code>String</code>	Массив символов;	N/A	N/A	N/A
<b>Символьные типы</b>					
<code>bool</code>	<code>Boolean</code>	Логический оператор, принимает значения: <code>true</code> – истина и <code>false</code> – ложь;	1	<code>false</code>	<code>true</code>
<b>Особые типы</b>					
<code>object</code>	<code>Object</code>	Базовый тип для всех типов;	N/A	N/A	N/A
<code>dynamic</code>	N/A	Для CLR тип <code>dynamic</code> идентичен типу <code>object</code> . Однако компилятор C# позволяет переменным типа <code>dynamic</code> участвовать в динамическом разрешении типа с упрощенным синтаксисом;	N/A	N/A	N/A
<b>Отдельные типы</b>					
<code>enum</code>	N/A	Тип перечисления. Определяется пользователем. Содержит в себе заданное количество различных значений, называемого списком перечислений;	N/A	N/A	N/A
<code>struct</code>	N/A	Обычно используется для инкапсуляции небольших групп связанных переменных;	N/A	N/A	N/A

<sup>1</sup> Буква "s" обозначает со знаком. То есть число может быть отрицательным.

<sup>2</sup> Буква "u" обозначает без знака.

Для создания переменной необходимо указать тип данных и её название. Также можно сразу же добавить к ней значение:

```
int a = 0; // Добавление значения сразу  
byte c; // Создание переменной без значения  
short b, y = 342; // Создание нескольких переменных
```





# Практика

- Создайте 3 переменных для работы с целыми числами
- Создайте 2 переменных для работы с текстом
- Выведите все переменные в консоль
- Приведите примеры, где можно применять переменные типа Bool

