

Урок 1

Введение в C#

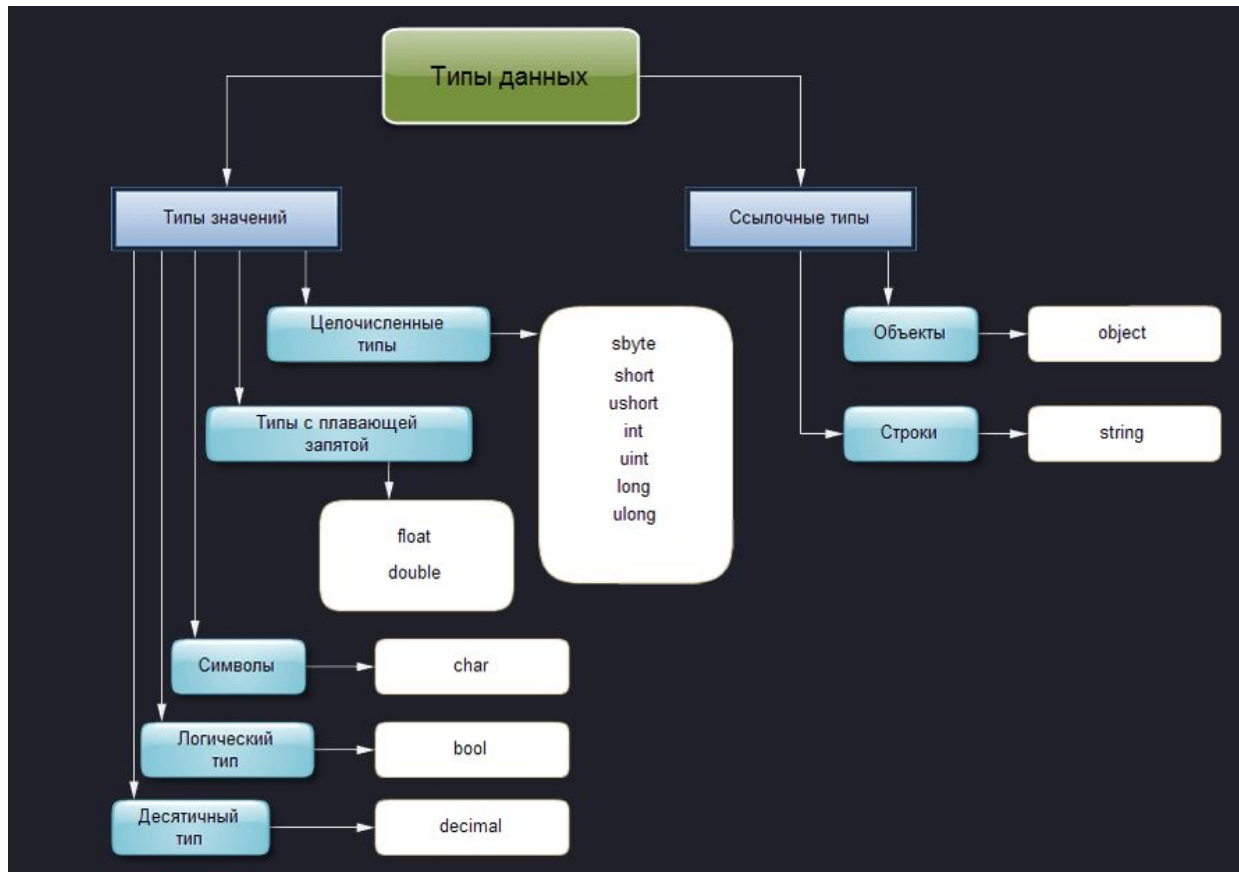
Инструменты разработчика,
написание первой программы

Платформа C#

- Простота
- Безопасность
- Переносимость
- Объектно-ориентированный характер программирования
- Надежность

Платформа C#

- Многопоточность
- Архитектурная независимость
- Интерпретируемость
- Высокая производительность



Компиляция и выполнение C# программы



Арифметические операции

Операция	Описание
+	Сложение
-	Вычитание
*	Умножение
/	Деление
%	Остаток от деления
++	Инкремент (приращение на 1)

Арифметические операции

Операция	Описание
+=	Сложение с присваиванием
-=	Вычитание с присваиванием
*=	Умножение с присваиванием
/=	Деление с присваиванием
%=	Остаток от деления с присваиванием
--	Декремент (отрицательное приращение на 1)

Формы записи условных операторов:

- if (условие) оператор;
- if (условие) {
 последовательность
 операторов } else {
 последовательность
 операторов }

Доступные операторы сравнения:

Оператор	Значение
<	Меньше
<=	Меньше или равно
>	Больше
>=	Больше или равно
==	Равно
!=	Неравно

Доступные операторы сравнения:

- if (условие) оператор;
- if (условие) {
 последовательность
 операторов } else {
 последовательность
 операторов }

1

```
static void Main(string[] args)  
{  
    int a = 5;  
    int b = 2;  
    c = a + b;  
    Console.WriteLine(c);  
}
```

2

```
public static void main(String[] args) {  
    char c1 = "Hello";  
    Console.WriteLine(c);  
}
```

3

```
static void Main(string[] args)
    string hw = "Hello, World!";
    Console.WriteLine(hw);
}
```

4

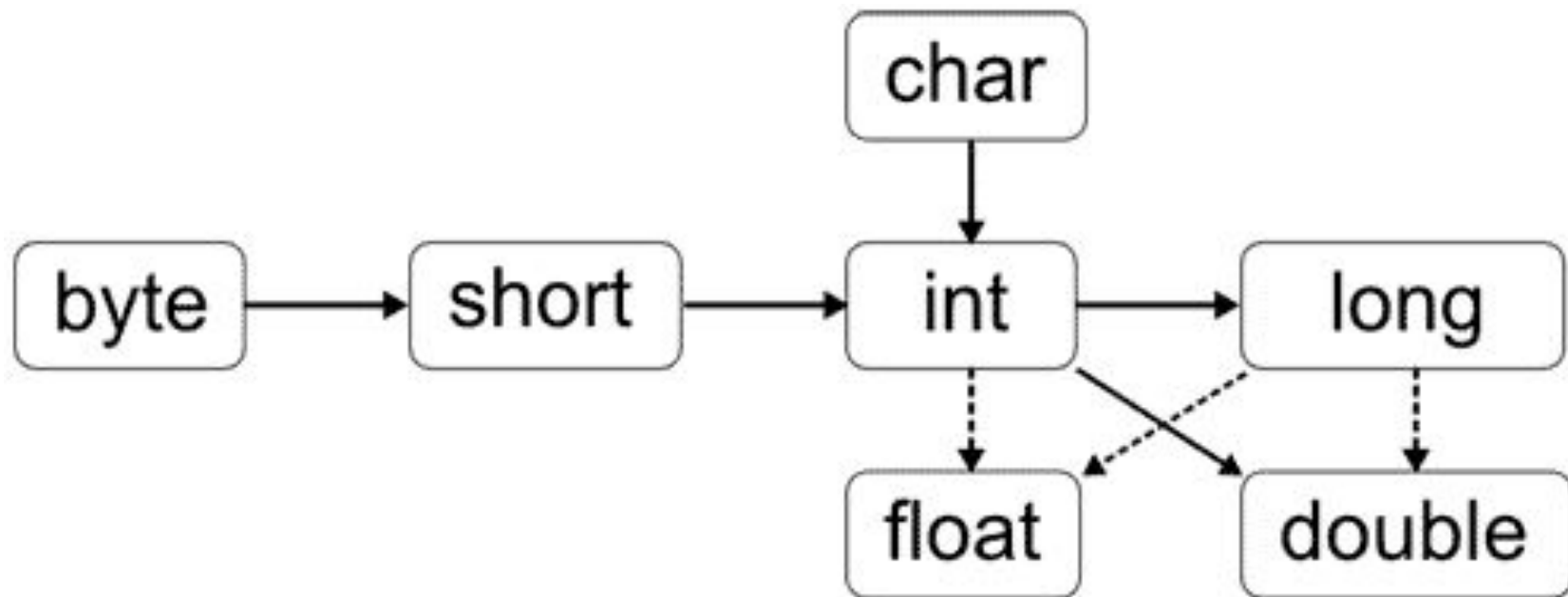
```
static void Main(string[] args) {
    string q = 'Hello';
    Console.WriteLine(q);
}
```

5

```
static void Main(string[] args) {  
    int a = 10;  
    int b = 20;  
    int c = a * 2 - b + 5;  
    Console.WriteLine("c = " + c);  
}
```

6

```
static void Main(string[] args) {  
    int a = 20;  
    string b = "myNumber = ";  
    a *= a;  
    Console.WriteLine(b + a);  
}
```



Приведение типов данных

1. **Неявное приведение типов** происходит автоматически при попытке преобразовать значение из более узкого типа данных в более широкий

```
byte b = 50; short s = b;
```

2. **При явном приведении типов** из более широкого в более узкий тип данных C# потребует от вас подтверждения ваших действий.

```
short s = 100; byte a = (byte)s;
```