

Отличие метода “Parse” от “TryParse” состоит в том, что “Parse” пытается преобразовать значимый тип без проверки возможно ли это и возвращает преобразованное значение указанного типа, а в случае неудачи выдаёт ошибку.


Метод “TryParse” проверяет возможно ли преобразование и если возможно, то записывает его в указанную переменную, возвращает true, если преобразование возможно и false если невозможно. При неудачном преобразовании для всех типов кроме bool записывает в переменную в аргументах null, для типа bool в переменную записывается значение false.

Как видно из скриншота ниже “TryParse” всегда возвращает значение булевского типа, метод “Parse” возвращает указанный программистом тип (в случае успешного преобразования).

```
class Program
{
    static void Main(string[] args)
    {
        int a, b;
        a = int.Parse("1");
        int.TryParse("1", out b);

        char c, d;
        c = char.Parse("1");
        char.TryParse(null, out d);

        bool e, f;
        e = bool.Parse("true");
        bool.TryParse("true", out f);
    }
}
```



```
// [10 13 - 10 32]
IL_0001: ldstr      "1"
IL_0006: call       int32 [System.Runtime]System.Int32::Parse(string)
IL_000b: stloc.0        // a

// [11 13 - 11 57]
IL_000c: ldnull
IL_000d: ldloc.s     b
IL_000f: call       bool [System.Runtime]System.Int32::TryParse(string, int32&)
IL_0014: call       void [System.Console]System.Console::WriteLine(bool)
IL_0019: nop

// [14 13 - 14 33]
IL_001a: ldstr      "1"
IL_001f: call       char [System.Runtime]System.Char::Parse(string)
IL_0024: stloc.2        // c

// [15 13 - 15 59]
IL_0025: ldnull
IL_0026: ldloc.s     d
IL_0028: call       bool [System.Runtime]System.Char::TryParse(string, char&)
IL_002d: call       void [System.Console]System.Console::WriteLine(bool)
IL_0032: nop

// [18 13 - 18 36]
IL_0033: ldstr      "true"
IL_0038: call       bool [System.Runtime]System.Boolean::Parse(string)
IL_003d: stloc.s     e

// [19 13 - 19 42]
IL_003f: ldstr      "true"
IL_0044: ldloc.s     f
IL_0046: call       bool [System.Runtime]System.Boolean::TryParse(string, bool&)
IL_004b: pop
```

Метод “Parse” следует использовать только тогда, когда мы наверняка знаем, что обрабатываемое значение обработается в значение такого же типа, как и переменной, в которую мы записываем это значение.

Метод “TryParse” следует использовать если есть вероятность некорректного обрабатываемого значения, чтобы избежать возможной ошибки в программе при обработке некорректных данных.

При использовании для типа char при удачном преобразовании Parse - возвращает код символа в Unicode, TryParse – записывает код в указанную в аргументах переменную и возвращает true. Для удачного преобразования необходим строго один символ.

При использовании для типа bool для удачной конвертации необходимо чтобы одна из строк имела значение “true” либо “false”. Только эти две строки способны сконвертироваться без ошибок.