

型のサイズ

sizeof 演算子

アンセーフコードは、直接メモリを扱うことができるため

C 言語の様に、高度なポインタの演算ということもできるようになります

メモリを直接扱うということからも、場合によっては型のサイズが重要になります

特に、構造体の型破りな変換など、プロ的な(?)裏技をやる時などは必要になるでしょう

例えば、構造体の末尾に暗黙的なリソース情報を付加する時などです

型のサイズを取得する時は **sizeof()** 演算子を用います

sizeof(type)

type には、サイズを調べたい型の名前を指定します

この演算子は、指定した型のサイズを表す int 型を返します

ただし、type に指定する式は**アンマネージド型**でなければなりません

すなわち、ポインタ変数に代入できる型でなければならないのです

C# がデフォルトで備えている int や long というような型は

調べる前にコンパイラが理解できるため、この時は sizeof() 演算子は定数を返します

それ以外の型であれば、実装時に依存した値を返すようになります

```
struct Kitty {
    int nSize;
    long l;
}

unsafe class Test {
    public static void Main() {
        System.Console.WriteLine(sizeof(int));
        System.Console.WriteLine(sizeof(Kitty));
    }
}
```

このプログラムを実行すれば、int 型と Kitty 型のサイズを出力します

sizeof() 演算子で調べられるのは型のサイズだけであり

C 言語の様に変数を直接渡して調べることはできないので、注意してください

sizeof(type)

特定の型が使用するバイト数を返します

type - サイズを調べたいアンマネージド式を指定します

[前のページへ](#)

[戻る](#)

[次のページへ](#)