

C#の覚書【入門1】

とりあえずインストールはC# 2008 Express Edition

どうやって書くのちよっとわかんない(ほぼCしか知らない)からネットで検索して、
<http://ufcpp.net/study/csharp/index.html#intro>
このサイトに決めた！！

まずは最初から読んでいって……なかなかおもしろそうだ！

【表示と変数】

コンソールアプリケーションでプロジェクトの雛形作成

自動記述された部分のメインの中で

cwと打ってtabを2回押すと自動的に Console.WriteLine();

ここまで記述されるなんて便利なんだ！コードスニペットというらしい。

調べてみると…偉大な先輩方が

「同じプログラムを何度も書いている！！よし、サブルーチンだ！！」

「別プログラムだが使いたいコードがある！！よし、ライブラリだ！！」

「基本的なコードは同じだけど内容が違う！！よし、デザインパターンだ！！」

「ライブラリやデザインパターンにするまでもないがもっと細かいレベルでの反復コードがある…
が数が膨大だ…どうしよう…そうだ！！！！よし、コードスニペットだ！！」
って感じらしい。

ってかコードスニペットが生まれたのは2005年くらいなのか…まだまだ勉強がたりないなw

プログラミングに戻るか…

```
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("hello world!!");

            int i=10;
            Console.WriteLine(i);
            double x=1.0;
            Console.WriteLine(x);

            var y=20;
            Console.WriteLine(y);

        }
    }
}
```

ここまで書いて、var 型が気になる……型推論というものらしい！

以下ウィキペディア参照情報

ほとんどの言語においては、すべての値は、その値がもつデータの種類を表す型をもつ。実行時にならないと型がわからない言語を動的型付けの言語という。一方、コンパイル時に型がわかる言語を静的型付けの言語という。静的型付けの言語において、関数の入力および出力の型やローカル変数の型は、通常は明示的に表記する必要がある。例えば、次は C 言語の例である。

```
int addone(int x) {
    int result;

    result = x+1;
    return result;
}
```

関数定義の最初の行 `int addone(int x)` では、`addone` は整数一つを入力としてとり、整数を結果として出力する、と宣言している。`int result;` の行では、ローカル変数 `result` が整数型であることを宣言している。

型推論の機能がある言語では、例えば次のように書くことができるだろう。

```
addone(x) {
    var result;

    result = x+1;
    return result;
}
```

この書き方は動的型付けの言語に非常によく似ているが、しかしながらすべての型はコンパイル時に定められる。この仮想的な例では、`+` 演算子は常に二つの整数を一つの整数に変換する。これから、`x+1` の値が整数であることが型推論によって推論される。故に `result` の型は整数であり、`addone` の戻り値が整数であることがわかる。同様に `+` 演算子が入力として整数をとることから `x` が整数であることがわかり、`addone` の引数の型が整数であることがわかる。

以上ウィキペディア情報。

ほほう、変数の型についても奥が深いな！