

notes plastiques

<[\[映画\]Babel](#) | [\[私事\]中学のとき](#)>

February 01 (Mon), 2010

C#基礎文法最速マスター

本当は増田に書く予定だったのですが、なぜか書き込み拒否されるので [ブログ](#)に書いてしまいました。

C#の文法一覧です。他の言語をある程度知っている人はこれを読めばC#の基礎をマスターしてC#を書くことができるようになっています。簡易リファレンスとしても利用できると思いますので、これは足りないと思うものがあれば教えてください。

1. 基礎

classの作成

プログラムはclassに記述します。たとえばSampleという名前のclassを作る場合、Sample.csファイル内に次のように書きます。(C#の場合、ファイル名とクラス名は同一でなくても良い。複数のクラスを書いても良い)

```
public class Sample {  
  
}
```

Mainメソッドの作成

プログラムはclass内のMainメソッドの先頭から実行されます。Mainメソッドは次のように書きます。

i am...



aont

C# PC,iPhone,C#,クラシック音楽, 数学,物理,NARUTOなどについて

articles

えっと...

[\[私事\]中学のとき](#)

[\[C#\]C#基礎文法最速マスター](#)

[\[映画\]Babel](#)

[\[雑記\]\[私事\]もう...](#)

[\[数学\]スペクトル法なんてのがあるらしい](#)

[\[数学\]回帰分析の標準偏差ってなに？](#)

[\[ネット\]\[Windows\]Google](#)

[Chromeをシステムレベルでインストールする方法 暫定版](#)

[\[雑記\]\[数学\]3次元関数の幾何学的性質](#)

[\[雑記\]\[私事\]\[C#\]実験つらい...](#)

tag cloud

C# Console Excel FileStream

GPU Google Google Chrome Hello

world IP InternetExplorer Microsoft

Ryuichi Sakamoto StreamReader

StreamWriter String **Windows**

ブログ プログラム 関

```
public class Sample {  
  
    public static void Main( String[] args ) {  
        // 処理を書く  
    }  
  
}
```

Console.WriteLineメソッド

文字列を表字するメソッドです。

```
Console.WriteLine( "Hello world" );
```

コメント

コメントです。

```
// 一行コメント  
  
/*  
 複数行コメント  
*/
```

変数の宣言

変数の宣言です。変数の宣言時にはデータ型を指定します。

```
// 変数  
int num;
```

データ型

データ型です。C#のデータ型にはプリミティブ型と参照型があります。以下はプリミティブ型のデータ型です。

```
// int(整数)型  
int num;  
// char(文字)型  
char c;  
// float(単精度浮動小数点)型  
float value;  
// double(倍精度浮動小数点)型  
double value;  
// bool(論理)型  
bool flag;
```

以下は参照型のデータ型です。

数 数学

aont at...

mixi
Twitter
DropBox

search

 詳細 一覧

count

83128

awasete yomitai

なんとなく日記
Bug Catharsis
my seamless monologue*
Unknown -要するに読書日記
と人力飛行機-
MCmieのcafeweddingと恵比
寿 * 代官山cafelife
D.I.'s Memorandum
川の果ての更に果てに
恵比寿法律新聞
リコログ
無視線～だんだん君がスキに
なる～

comments

2010-01-13 aont
2010-01-13 fwga1002
2010-01-07 aont
2010-01-07 りよー
2009-08-23 aont

trackback

2010-02-01 なんとなく日記 -
基礎文法最速マスターシリー
ズのまとめ
2010-02-01 Life like a clown -
はてな的プログラミング言語人
気ランキング
2010-02-01 なんとなく日記 -

```
// String型
String s;
// Date型
Date d;
// 配列型
String[] array;
```

プログラムのコンパイル

プログラムをコンパイルするには、コマンドラインで以下のようにします。

```
csc Sample.cs
```

プログラムの実行

プログラムを実行するには、コマンドラインで以下のようにします。

.net framework on Windowsの場合

```
Sample.exe
```

Mono.frameworkの場合

```
mono ./Sample.exe
```

2. 数値

数値の表現

int、float、double型の変数に数値を代入できます。int型には整数だけ代入できます。float、double型には整数でも小数でも代入できます。

```
int i = 2;
int i = 100000000;

float num = 1.234f;

double num = 1.234;
```

四則演算

四則演算です。

```
num = 1 + 1;
num = 1 - 1;
num = 1 * 2;
num = 1 / 2;
```

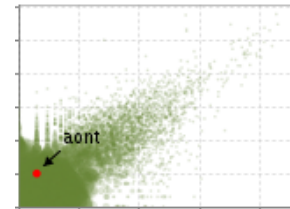
商の求め方です。割る数と割られる数が両方とも整数の場合、計算結果の小数点以下が切り捨てられます。

コンパイルが必要な手続き型言語

2010-02-01 燈明日記 - 基礎文法最速マスターぞくぞくキターー！

2010-02-01 きまぐれメモ - 各種言語による基礎文法最速マスターまとめ

top hatenar



shiori on air

各種言語による基礎文法最速マスターまとめ - きまぐれメモ「JPEGファイル回転プログラム azure」の紹介

EPS-conv

Twitter / Takeshi K. Jiro: なんか Java 屋さんが無理して C# のこと書い ...

Twitter / Takeshi K. Jiro:

Browsing: C#基礎文法最速マスター - n ...

my tweets

aont50: @takeshik Java マスターをばくったのでJavaつぼくなってしまいました。すみませんw こんど追記いたします ちなみに自分はJavaはほとんど使いません。

aont50: RT @takeshik: なんか Java 屋さんが無理して C# のこと書いた空気が出まくりなんだけど...? [TWNv814]

aont50: @jukucho01 リクエストありがとうございます！すみませんが、@aont_todoはメモ専用なのでこちらならフォローしていただいてもかまいませんが？

```
num = 1 / 2; // 0
```

割る数と割られる数のどちらかが小数の場合、計算結果の小数点以下が切り捨てられません。

```
num = 1.0 / 2; // 0.5  
num = 1 / 2.0; // 0.5  
num = 1.0 / 2.0; // 0.5
```

余りの求め方です。

```
// 余り  
mod = 4 % 2
```

インクリメントとデクリメント

インクリメントとデクリメントです。

```
// インクリメント  
++i;  
  
// デクリメント  
--i;
```

3. 文字列

文字列の表現

文字列はダブルクォートで囲みます。

```
String str = "abc";
```

文字列操作

各種文字列操作です。

```
// 結合  
String join = "aaa" + "bbb";  
  
// 分割  
String[] record = "aaa,bbb,ccc".Split( ", " );  
  
// 長さ  
int length = "abcdef".Length();  
  
// 切り出し  
"abcd".Substring( 0, 2 ) // abc  
  
// 検索  
int result = "abcd".IndexOf( "cd" ) // 見つかった場合は
```

aont50: SSD+HDDで長寿命化が可能な3.5インチアダプタが発売 <http://bit.ly/cBJL3S> これはHDDの先頭セクタからSSD容量分だけSSDにコピーしてSSDをキャッシュとして使うらしい。ってことはシステムパーティションをSSDと同じサイズにしておけばよいのかな？
aont50: C#基礎文法最速マスターは無いのね

4. 配列

配列変数の宣言

配列です。

```
// 配列の宣言  
int[] array;
```

配列の生成

配列の生成です。配列の生成時には要素数を指定するか、初期データを指定します。

```
int[] array;  
  
// 要素数を指定して配列を生成  
array = new int[5];  
  
// 初期データを指定して配列を生成  
array = new int[] { 1, 2, 3 };  
  
// 宣言と同時に配列を生成  
int[] array2 = new int[5];
```

配列の要素の参照と代入

配列の要素の参照と代入です。

```
// 要素の参照  
array[0]  
array[1]  
  
// 要素の代入  
array[0] = 1;  
array[1] = 2;
```

配列の要素数

配列の要素数を取得するには以下のようにします。

```
array_num = array.Length;
```

配列のコピー

配列の要素を別の配列にコピーするには以下のようにします。

```
int[] from = new int[] { 1, 2, 3 };  
int[] to = new int[5];  
  
from.CopyTo(to, 0);
```

5. 制御文

if文

if文です。

```
if ( 条件 )  
{  
  
}
```

if ~ else文

if ~ else文です。

```
if ( 条件 )  
{  
  
}  
else  
{  
  
}
```

if ~ else if 文

if ~ else if文です。

```
if ( 条件 )  
{  
  
}  
else if ( 条件 )  
{  
  
}
```

while文

while文です。

```
int i = 0;  
while ( i < 5 )  
{  
  
    // 処理  
  
    ++i;  
  
}
```

for文

for文です。

```
for ( int i = 0; i < 5; ++i )
{
    // 処理
}
```

| for-each文

for-each文です。配列の各要素を処理できます。

```
int[] fields = new int[] { 1, 2, 3 };

foreach (int field in fields)
{
    // 処理
}
```

| 6. メソッド

C#では関数をメソッドと言います。メソッドを作るには次のようにします。戻り値を返却するにはreturn文を使います。

```
static int sum( int num1, int num2 )
{
    int total;

    total = num1 + num2;

    return total;
}
```

| 9. ファイル入出力

ファイル入出力です。ファイル入出力を行うには、プログラムの先頭に以下を記述します。

```
using System.IO;
```

以下がファイル入力の雛形になります。ファイルのオープンや読み込みに失敗した場合、catch節に処理が移ります。

```
String filename = "text.txt";
StreamReader reader = null;
try
{
    reader = new StreamReader(new FileStream(filename,

    String line;
    while ((line = reader.ReadLine()) != null)
    {

    }

}
catch (IOException e)
{
    // エラー処理:

}
finally
{
    if (reader != null)
    {
        try
        {
            reader.Close();
        }
        catch (IOException e) { }
    }
}
```

以下がファイル出力の雛形になります。ファイルのオープンや書き込みに失敗した場合、catch節に処理が移ります。


```

String filename = "text.txt";
StreamWriter writer = null;

try
{
    writer = new StreamWriter(new FileStream(filename)

    writer.WriteLine("abc");
    writer.WriteLine("def");
    writer.WriteLine("fgh");

}
catch (IOException e)
{
    // エラー処理:

}
finally
{
    if (writer != null)
    {
        writer.Close();
    }
}

```

知っておいたほうがよい文法

C#でよく出てくる知っておいたほうがよい文法の一覧です。

繰り返し文の途中で抜ける

繰り返し文の途中で抜けるにはbreak文を使用します。

```

for ( i = 0; i < 5; ++i ) {

    if ( 条件 ) {
        break;    // 条件を満たす場合、for文を抜ける。
    }

}

```

繰り返しの残り部分の処理をスキップする

残りの部分処理をスキップし、次の繰り返しに進むにはcontinue文を使用します。

```
for ( i = 0; i < 5; ++i ) {  
  
    if ( 条件 ) {  
        continue;    // 条件を満たす場合、残りの部分処理  
    }  
  
}
```

例外処理

例外を投げるにはthrow文を使用します。

```
throw new Exception( "Error message" );
```

例外処理をするにはtry ~ catch文を使用します。

```
try {  
  
    // 例外が発生する可能性のある処理  
  
} catch ( Exception e ) {  
  
    // 例外発生時の処理  
  
}
```

卒論で忙しい身なので後ほど追記いたします。

参考

- Java基礎文法最速マスター - 何かしらの言語による記述を解析する日記
- JavaScript基礎文法最速マスター - なんとなく日記

Permalink | コメント(0) | トラックバック(7) | 01:35  **16 users** 

[C#について徹底解説](#)


ITエンジニア必須の充実IT講座100選 無料会員登録でITpro新サービスを提供

[有限要素法 \(FEM\)開発ツール](#)

メッシュ作成、結果表示ソフト付属。ソルバーの開発のみに専念できます。

Ads by Google



 スパシク実力がつくと評判の微分方程式チャンネル...

マセマ出版社 馬場敬之 久池井茂
購入: 1人 クリック: 4回

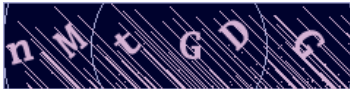


 「Hello, world.」オリジナルサウンドトラック

ニトロプラス
クリック: 2回

コメントを書く



画像認証

画像内の文字列を入力して下さい

投稿

トラックバック - <http://d.hatena.ne.jp/aont/20100201/1265042142>

[Diksam]Diksam基礎文法最速マスター

なんとなく日記 - JavaScript基礎文法最速マスター

きまぐれメモ - 各種言語による基礎文法最速マスターまとめ

Life like a clown - はてな的プログラミング言語人気ランキング

なんとなく日記 - 基礎文法最速マスターシリーズのまとめ

なんとなく日記 - コンパイルが必要な手続き型言語

燈明日記 - 基礎文法最速マスターぞくぞくキターーー！

idトラックバック

猫とC#について書くmatarilloの日記 - はてなC#er

リンク元

66 <http://matarillo.com/blood/>

42 http://pipes.yahoo.com/pipes/pipe.info?_id=faa858a20082ef6d25a

41 http://pipes.yahoo.com/pipes/pipe.info?_id=49361476efd1fe9761d

41 <http://reader.livedoor.com/reader/>

38 http://d.hatena.ne.jp/tt_clown/20100202/1265096776

36 <http://www.google.co.jp/reader/view/>

32 <http://d.hatena.ne.jp/gifnksm/20100131/1264934942>

24 <http://www.google.co.jp/reader/view/?hl=ja&tab=wy>

20 <http://codezine.jp/bookmark/>

18 <http://twitter.com/>

おとなり日記

2010-02-01 safsa's blog site 6/32 18%

2010-02-02 プログラミング言語を作る日記 6/90 6%

2010-01-31 ころの徒然～在る阿呆の日々～ 3/47 6%

2010-01-31 sakurako_sの日記 4/73 5%

<[映画]Babel | [私事]中学のとき>