

C# プログラミング入門 1

この入門書は C# 言語によるコンピュータ プログラムの作り方について説明します。プログラムを書いた経験のない方を対象に、サンプルを交えながら C# 言語の構文規則やプログラミングの際に知っておく必要がある事柄についてテーマごとに説明しています。

そもそもプログラムとは何でしょうか。プログラムという言葉は、普段の生活でもよく見聞きする言葉です。テレビやラジオの番組、催し、行事などの進行表もプログラムと呼びます。どちらのプログラムも、物事を行う順番が記述されている点で共通しています。また、料理本のレシピと調理手順にも似ています。コンピュータのプログラムに限って言えば、「コンピュータに対しての指示が順番に記述されている手順書」になります。コンピュータに実行して欲しい作業を、決められた言葉で書くことができれば、複雑な計算や文書の編集処理、他のコンピュータとの通信、さらには音楽などの各種メディアの再生や編集など、コンピュータにできることであれば何でも実現できます。

プログラムはプログラミング言語というコンピュータに理解できる言葉で書く必要があります。今回の入門では C# というプログラミング言語について学習します。C# 言語は 2000 年に Microsoft 社が発表した、比較的新しい開発言語です。既存の言語に含まれていた問題点を取り除き、また最新の機能や概念を取り入れて統合することで、プログラミングしやすく、また複雑な処理内容も少ないコードで実現することができます。各種標準化団体による標準化もされており、教育機関での学習用途に向いています。また多くの企業で採用が進められています。

本入門では開発ツールとして Visual Studio を使用します。Visual Studio には入門者向け、プロフェッショナル プログラマ向け、企業や団体向けなどの、多くのエディションがあります。入門者向けのエディションである「Visual C# 2005 Express Edition 日本語版」は以下の URL のサイトから無料でダウンロードできます。

<http://www.microsoft.com/japan/msdn/vstudio/express/vcsharp/>

* Microsoft、Windows、Windows Vista、Visual Studio、Visual C#、Visual Basic は、米国 Microsoft Corporation の米国及びその他の国における登録商標または商標です。

* Windows の正式名称は、Microsoft Windows Operating System です。

© 2007 Microsoft Corporation. All rights reserved.

文書情報

タイトル	ねらい	章題	キーワード
1. 初めてのプログラミング	<ul style="list-style-type: none"> ● 開発環境である Visual Studio に慣れます。 ● プログラミングのおおまかな流れについて把握します。 	<ol style="list-style-type: none"> 1. Visual Studio の起動 2. 新規プロジェクトの作成 3. Visual Studio の画面説明 4. コード入力 5. 実行 (デバッグ) 6. Visual Studio の終了 演習課題 	プログラム 統合開発環境 C# 言語 コンソール アプリケーション コマンド プロンプト ソース ファイル コード エディタ 実行
2. プログラムが動くしくみ	<ul style="list-style-type: none"> ● プログラムがコンピュータ上で動く仕組みの概要を理解します。 ● プログラムを実行するためにはビルドが必要であることを理解します。 	<ol style="list-style-type: none"> 1. コンピュータの中身 2. プログラムの実行 3. プログラムのビルド 	CPU メモリ ハードディスク キーボード マウス ディスプレイ OS ソース コード ソース ファイル ビルド 実行形式のファイル
3. Main メソッド	<ul style="list-style-type: none"> ● C# の一般的なコードの構造について理解します。 ● プログラムの開始地点である Main メソッドについて理解します。 	<ol style="list-style-type: none"> 1. C# プログラムの一般構造 2. Main メソッド 	名前空間 Main メソッド メソッド
4. コメント	<ul style="list-style-type: none"> ● コメント文の書き方について理解します。 	<ol style="list-style-type: none"> 1. コメント文について 2. コメントの書き方 演習課題 	コメント
5. 標準出力	<ul style="list-style-type: none"> ● 標準入出力について理解します。 ● Console.Write、Console.WriteLine メソッドの使い方を理解します。 	<ol style="list-style-type: none"> 1. Console クラス 2. Write/WriteLine メソッド 演習課題 	標準入出力 Console クラス 書式情報
6. 標準入力	<ul style="list-style-type: none"> ● Console.Read、Console.ReadLine メソッドの使い方を理解します。 	<ol style="list-style-type: none"> 1. Read/ReadLine メソッド 	Read メソッド ReadLine メソッド
7. 変数とリテラル	<ul style="list-style-type: none"> ● 変数とリテラルについて理解します。 	<ol style="list-style-type: none"> 1. 変数 2. リテラル 演習課題 	変数 代入 リテラル 参照 変数名

タイトル	ねらい	章題	キーワード
8. 変数の定義	<ul style="list-style-type: none"> ● 変数の定義の仕方について理解します。 	1. 変数の定義	
9. 使える文字	<ul style="list-style-type: none"> ● 変数名や、クラス名、メソッド名など、識別名として使える文字について理解します。 ● わかりやすい変数名にするための命名規則について理解します。 	<ol style="list-style-type: none"> 1. 識別名に使用できる文字 2. 命名規則 	
10. 予約語	<ul style="list-style-type: none"> ● 識別名として使用できない予約語について理解します。 	1. 予約語	
11. データ型	<ul style="list-style-type: none"> ● データ型の種類について理解します。 	<ol style="list-style-type: none"> 1. データ型の一覧 2. データ型の種類 	組み込み型 ユーザー定義型 値型 参照型 単純型
12. 単純型 【サンプル プログラム有り】	<ul style="list-style-type: none"> ● 整数、浮動小数などのデータ型の使い方について理解します。 	<ol style="list-style-type: none"> 1. データ型の一覧 2. 整数 <ol style="list-style-type: none"> 2.1 整数型 2.2 リテラル 3. 浮動小数点 <ol style="list-style-type: none"> 3.1 浮動小数点型 3.2 リテラル 4. 10 進数値 <ol style="list-style-type: none"> 4.1 decimal 型 4.2 リテラル 5. 文字型 <ol style="list-style-type: none"> 5.1 char 型 5.2 リテラル 6. ブール型 <ol style="list-style-type: none"> 6.1 bool 型 6.2 リテラル 7. 型変換 演習課題	整数リテラル 実数値リテラル Unicode 文字 エスケープ シーケンス true false 暗黙の型変換 明示的な型変換
13. 文字列型 【サンプル プログラム有り】	<ul style="list-style-type: none"> ● 文字列型の使い方について理解します。 	<ol style="list-style-type: none"> 1. 文字列型 2. 文字列リテラル 3. 文字列型と数値型の変換 演習課題	文字列リテラル エスケープ シーケンス Convert クラス ToString メソッド Parse メソッド
14. 算術演算子 【サンプル プログラム有り】	<ul style="list-style-type: none"> ● 足し算や引き算などの演算子の使い方について理解します。 	<ol style="list-style-type: none"> 1. 演算子 2. 算術演算子 3. インクリメント/デクリメント演算子 4. 演算子の優先順位 演習課題	演算子 オーバーフロー

タイトル	ねらい	章題	キーワード
15. 演算精度 【サンプル プログラム有り】	<ul style="list-style-type: none"> ● 浮動小数点などの実数の演算で気を付ける必要がある、演算精度の問題について理解します。 	<ol style="list-style-type: none"> 1. 浮動小数点の有効桁数 2. 情報欠落 3. 桁落ち 	丸め誤差 情報欠落 桁落ち
16. 文字列連結演算子 【サンプル プログラム有り】	<ul style="list-style-type: none"> ● 文字列の連結をおこなう + 演算子の使い方を理解します。 	<ol style="list-style-type: none"> 1. 文字列連結演算子 演習課題	
17. 関係演算子 【サンプル プログラム有り】	<ul style="list-style-type: none"> ● 等号、不等号などの関係演算子の使い方について理解します。 	<ol style="list-style-type: none"> 1. 等値演算子、非等値演算子 2. 大小比較 3. 優先順位 演習課題	
18. 代入演算子	<ul style="list-style-type: none"> ● 代入演算子の使い方について理解します。 	<ol style="list-style-type: none"> 1. 代入演算子 2. 複合代入演算 3. 優先順位 	
19. その他の演算子	<ul style="list-style-type: none"> ● メンバ アクセス演算子の使い方について理解します。 ● 今回取り上げる演算子と、その優先順位についておさらいします。 	<ol style="list-style-type: none"> 1. メンバ アクセス演算子 2. その他の演算子 3. 優先順位 	
20. 配列 【サンプル プログラム有り】	<ul style="list-style-type: none"> ● 配列の概念について理解します。 ● 配列の使い方について理解します。 	<ol style="list-style-type: none"> 1. 配列 2. 配列の宣言 3. 配列の初期化 4. 配列の要素 5. 配列のコピー 6. 配列のメンバ要素 	配列 配列の要素 インデックス
21. 条件分岐 【サンプル プログラム有り】	<ul style="list-style-type: none"> ● if ステートメントと switch ステートメントの使い方について理解します。 	<ol style="list-style-type: none"> 1. if ステートメント 2. switch ステートメント 演習課題	ステートメント 順次制御 if ステートメント switch ステートメント case ステートメント
22. ループ 【サンプル プログラム有り】	<ul style="list-style-type: none"> ● while、do、for、foreach の各ループの使い方について理解します。 	<ol style="list-style-type: none"> 1. while ループ 2. do ループ 3. for ループ 4. foreach ループ 5. break/continue ステートメント 演習課題	while ループ do ループ for ループ foreach ループ

タイトル	ねらい	章題	キーワード
23. エラーと警告	<ul style="list-style-type: none"> コーディングの際に陥りがちなミスと、その際に表示されるエラー内容について説明します。エラーの表示内容から、問題の個所を見つけて対処できるようにします。 	<ol style="list-style-type: none"> エラーの種類 文法誤り 変数の初期化忘れ 誤った型の利用 範囲を超えた配列の参照 無限ループ ゼロによる割り算 	バグ コンパイル エラー 実行時例外
24. 関数 (メソッド) 【サンプル プログラム有り】	<ul style="list-style-type: none"> メソッドの使い方について理解します。 	<ol style="list-style-type: none"> メソッドとは メソッドの定義 メソッドの呼び出し 値渡しと参照渡し 演習課題	メソッド 戻り値 引数 修飾子 戻り値の型 メソッド名 引数の型 パラメータ
25. 変数のスコープ	<ul style="list-style-type: none"> 変数の有効範囲 (スコープ) について理解します。 	<ol style="list-style-type: none"> 変数のスコープ 	
26. 列挙型 【サンプル プログラム有り】	<ul style="list-style-type: none"> 列挙型の使い方について理解します。 	<ol style="list-style-type: none"> 列挙型とは 列挙型の定義 列挙型の使用 演習課題	列挙型 列挙子
27. 構造体 【サンプル プログラム有り】	<ul style="list-style-type: none"> 構造体の使い方について理解します。 	<ol style="list-style-type: none"> 構造体とは 構造体の定義 構造体の作成と初期化 構造体変数の使用 演習課題	構造体 フィールド
28. クラスの概要 【サンプル プログラム有り】	<ul style="list-style-type: none"> クラス概念と、基本的な使い方について理解します。 	<ol style="list-style-type: none"> クラスとは クラスの定義 インスタンスの生成 クラスの利用 	クラス メンバ インスタンス
29. 名前空間とクラス ライブラリ	<ul style="list-style-type: none"> 名前空間とクラス ライブラリについて理解します。 	<ol style="list-style-type: none"> .NET Framework について クラス ライブラリについて 名前空間 名前空間の定義 	.NET Framework 共通語ランタイム クラス ライブラリ 名前空間 完全修飾名 using ディレクティブ

タイトル	ねらい	章題	キーワード
30. 付録 便利なクラス 1 【サンプル プログラム有り】	● 標準のクラス ライブラリの中でよく使われるいくつかのクラスについて紹介しません。	1. String クラス (System.String) 2. Math クラス (System.Math) 3. DateTime 構造体 (System.DateTime) 4. File クラス (System.IO.File) 5. StreamReader/StreamWriter クラス (System.IO.StreamReader (StreamWriter))	
31. 付録 C# 言語の特徴	● C# 言語の特徴について理解します。	1. C# 言語について 2. オブジェクト指向 3. コンポーネント指向 4. タイプ セーフ 5. ガベージ コレクション 6. 例外処理 7. その他	オブジェクト指向言語 コンポーネント指向 タイプ セーフ ガベージ コレクション 例外処理

キーワード索引

キーワード	章題
英	
.NET Framework	29. 名前空間とライブラリ
C# 言語	1. 初めてのプログラミング
case ステートメント	21. 条件分岐
Console クラス	5. 標準出力
Convert クラス	13. 文字列型
CPU	2. プログラムが動くしくみ
do ループ	22. ループ
false	12. 基本型
for ループ	22. ループ
foreach ループ	22. ループ
if ステートメント	21. 条件分岐
Main メソッド	3. Main メソッド
OS	2. プログラムが動くしくみ
Parse メソッド	13. 文字列型
Read メソッド	6. 標準出力
ReadLine メソッド	6. 標準出力
switch ステートメント	21. 条件分岐

キーワード	章題
Tostring メソッド	13. 文字列型
true	12. 基本型
Unicode 文字	12. 基本型
using ディレクティブ	29. 名前空間とライブラリ
while ループ	22. ループ
あ	
値型	11. データ型
暗黙の型変換	12. 基本型
インスタンス	28. クラスの概要
インデックス	20. 配列
エスケープ シーケンス	12. 基本型 13. 文字列型
演算子	14. 算術演算子
オーバーフロー	14. 算術演算子
オブジェクト指向言語	31. 付録 C# 言語の特徴
か	
ガベージ コレクション	31. 付録 C# 言語の特徴
完全修飾名	29. 名前空間とライブラリ

キーワード	章題
キーボード	2. プログラムが動くしくみ
組み込み型	11. データ型
クラス	28. クラスの概要
共通言語ランタイム	29. 名前空間とライブラリ
クラス ライブラリ	29. 名前空間とライブラリ
桁落ち	15. 演算精度
構造体	27. 構造体
コード エディタ	1. 初めてのプログラミング
コマンド プロンプト	1. 初めてのプログラミング
コメント	4. コメント
コンソール アプリケーション	1. 初めてのプログラミング
コンパイル エラー	23. エラーと警告
コンポーネント指向	31. 付録 C# 言語の特徴
さ	
参照	7. 変数
参照型	11. データ型
実行	1. 初めてのプログラミング
実行形式のファイル	2. プログラムが動くしくみ
実行時例外	23. エラーと警告
実数値リテラル	12. 基本型
修飾子	24. 関数 (メソッド)
順次制御	21. 条件分岐
情報欠落	15. 演算精度
書式情報	5. 標準出力
ステートメント	21. 条件分岐
整数リテラル	12. 基本型
ソース コード	2. プログラムが動くしくみ
ソース ファイル	1. 初めてのプログラミング 2. プログラムが動くしくみ
た	
代入	7. 変数
タイプ セーフ	31. 付録 C# 言語の特徴
単純型	11. データ型
ディスプレイ	2. プログラムが動くしくみ
統合開発環境	1. 初めてのプログラミング

キーワード	章題
な	
名前空間	3. Main メソッド 29. 名前空間とライブラリ
は	
ハードディスク	2. プログラムが動くしくみ
配列	20. 配列
配列の要素	20. 配列
バグ	23. エラーと警告
パラメータ	24. 関数 (メソッド)
引数	24. 関数 (メソッド)
引数の型	24. 関数 (メソッド)
標準入出力	5. 標準出力
ビルド	2. プログラムが動くしくみ
フィールド	27. 構造体
プログラム	1. 初めてのプログラミング
変数	7. 変数
変数名	7. 変数
ま	
マウス	2. プログラムが動くしくみ
丸め誤差	15. 演算精度
明示的な型変換	12. 基本型
メソッド	3. Main メソッド 24. 関数 (メソッド)
メソッド名	24. 関数 (メソッド)
メモリ	2. プログラムが動くしくみ
メンバ	28. クラスの概要
文字列リテラル	13. 文字列型
戻り値	24. 関数 (メソッド)
戻り値の型	24. 関数 (メソッド)
や	
ユーザー定義型	11. データ型
ら	
リテラル	7. 変数とリテラル
例外処理	31. 付録 C# 言語の特徴
列挙型	26. 列挙型
列挙子	26. 列挙型