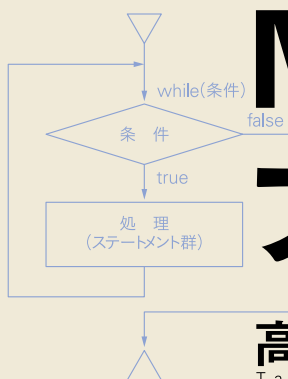


知的LEGO

Mindstorms NXT プログラミング入門

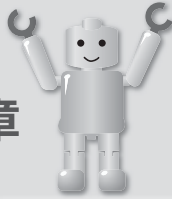
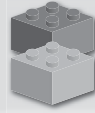


高本孝頼 [著]
Takayori Takamoto

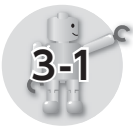




第3章

BricxCC
統合開発環境

本章では、BricxCCの開発環境についての構築方法と基本的な環境設定、それに主な機能群について紹介します。このBricxCCの機能に何があるかを知っているだけでも効率アップにつながります。まずは、ここに記述されている内容を一読しておいてください。



3-1 BricxCCのプログラミング開発環境構築

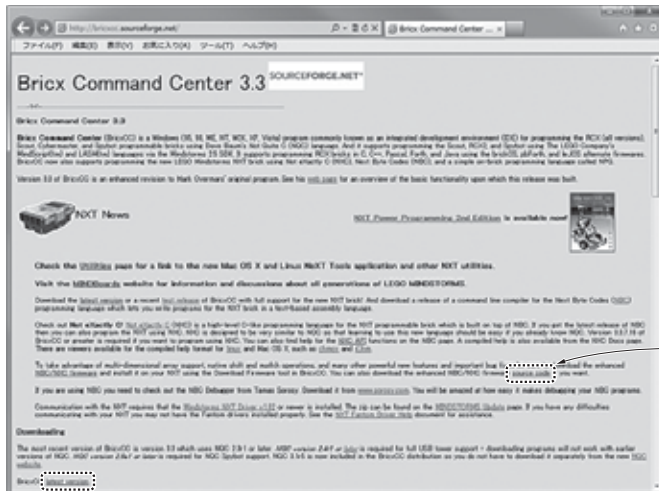
まず、BricxCCを利用するためのPC上の環境構築の説明を行います。ここでは、BricxCCのインストールから、立ち上げ時の操作、およびBricxCCの画面説明について紹介していきます。このインストールは、Windows PC上の環境構築となるので、正しい手順に沿って作業を進めてください。

(1) BricxCCのインストール

それでは、BricxCCのインストールを行います。PCをインターネットに接続し、以下の「Bricx Command Center 3.3」サイトにアクセスしてください。

<http://bricxcc.sourceforge.net/>

アクセスすることで図3-1-1に表示された画面が出てきます。BricxCCをインストール開始する

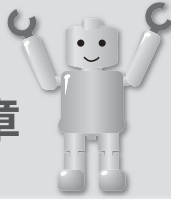
拡張NBC/NXC
ファームウェア

BricxCC最新バージョン・サイト

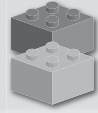
図3-1-1
Bricx Command Center 3.3のWeb サイト



第4章



ロボットを動かす



本章では、PC上で簡単なプログラムを作成し、ロボットを動かすまでの流れを説明していきます。ここでは、2-3節で説明したNXTロボットの3輪車モデルTribot（トリボット）を使い、両輪のモータ（アクチュエータ）を動かし、前進・後退するプログラムの作成事例を紹介します。このモータは、NXTロボットにとっては出力機能となります。

ここでは、プログラムの作成から、コンパイル、エラーが出たときのデバッグ（プログラムのミスを見つけて修正すること）、再コンパイル、そしてNXTロボットへのプログラムのダウンロード、さらにプログラムを実行するまでの一連のプロセス（過程・工程）を学んでいきます。

また、完成したプログラムを修正し、NXTが新たな動きをする、より複雑に動かすにはどうするかを学んでいきましょう。

プログラムが理解できるようになれば、どんどん先に進んでみてください。さらに課題で実力を確認してみてください。プログラム作成が楽しくなるには、自らが独創的に考え、思うようにロボットが動かせるようになることです。「よく考える」ことこそ、プログラム作成では重要で、将来のための自らの財産と思って、トレーニングを積んでください。



4-1 ロボットの前進と後退

ここでは初心者向けに、簡単にロボットを前進させたり、後退させたりするプログラムの作成から、コンパイル、ダウンロードなどについて説明していきます（図4-1-1）。

（1）BricxCCを使ってNXCプログラムの作成

それでは、PC上のBricxCCを立ち上げ、NXCプログラムを作成していきます。リスト4-1-1に記述してあるプログラムを作成してみてください。このプログラムは、ロボットを2秒前進させ、その後2秒後退させる単純なプログラムです。

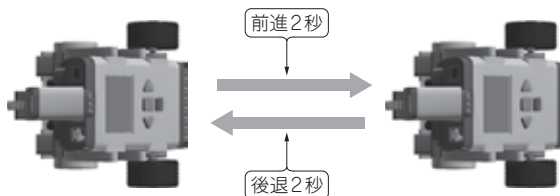


図4-1-1
ロボットの前進と後退

リスト4-1-1 前進と後退のプログラム

```
1. task main()
2. {
3.   OnFwd(OUT_AC,75);
4.   Wait(2000);
5.   OnRev(OUT_AC,75);
6.   Wait(2000);
7.   Off(OUT_AC);
8. }
```

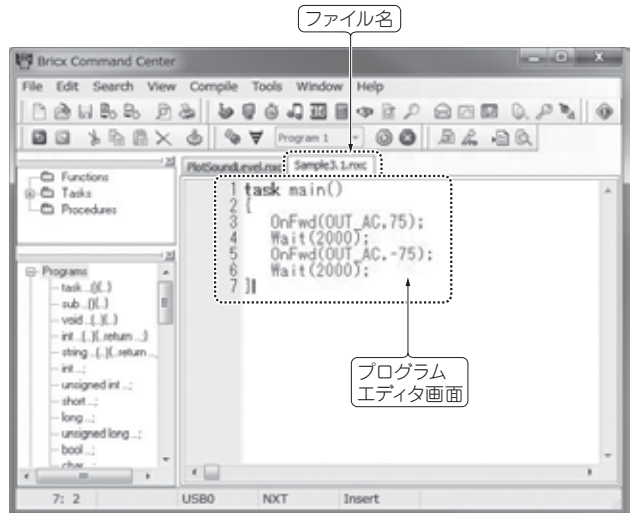


図4-1-2 エディタ画面上的NXCコーディング

作成手順としては、まずBricxCCを立ち上げ、図4-1-2にあるメニュー・バーの「File」から「New」を選択し、プログラム・エディタ画面でプログラムを作成していきます。

次に、プログラム内容について詳細な説明をしていきます。

まず、1行目の「task」は、**タスク**（仕事）と呼び、2行目と8行目の中括弧（波括弧）「{」と「}」内に一つひとつのステートメント（文）を定義していきます。「task main()」の「main」（メイン）は**タスク名**で、NXCプログラム中で最初に実行するタスクとなり、開発するプログラム内に必ずこの「task main()」（メイン・タスク）が一つのみ定義する必要があります【注意：この「task main()」は標準C言語の「void main()」にあたる】。

次の3行目「OnFwd(OUT_AC,75);」の「OnFwd」は、あらかじめNXCに組み込まれた**内部関数**で、モータを駆動させるものです。ここではカンマで区切られた二つの**引数**（ひきすうパラメータ）を設定しています。最初の引数「OUT_AC」は、後で説明する**出力ポートA**と**C**を両方定義したもので、両輪のモータを設定していることを意味します。次の「75」は、モータの**パワー**を設定する数値パラメータとなります。

次の4行目の「Wait(2000);」は、**2秒間待機**（継続）を意味します。この「Wait」もNXCの内部関数となります。

それでは、ここでの内部関数をまとめておきます。

【内部関数】

前進 : OnFwd(出力ポート, パワー);
後退 : OnRev(出力ポート, パワー);
停止 : Off(出力ポート);
待機 : Wait(継続時間);



ここでの引数となる出力ポートとパワーは、次のような具体的な設定となります。

出力ポート：NXT 本体の上部にある接続コード記号（A, B, C の三つを指す）

OUT_A, OUT_B, OUT_C,

OUT_AB, OUT_AC, OUT_BC, OUT_ABC の7通り


パワー：モータの動力の設定パワー（単位はなし）

0～100（-100～0 の設定も可能）


（2）NXC プログラムの保存とコンパイル

作成したプログラムは、**ファイル名**を付けて**保存**しておく必要があります（ファイルの拡張子は「nxc」となる）。メニュー・バーの「File」から「Save as...」を選択して、ファイル名を付けて、保存してください【**注意**：ファイル名が付けられていない場合、コンパイル・エラーが出ることもある】。

次にPCとNXT本体との接続が必要となります。この接続は、USBケーブルによるものか、無線によるBluetooth機能を使うかのいずれかとなります。初心者はUSBケーブルにて接続する方法をお勧めします。ケーブルを接続してから、メニュー・バー「Tools」→「Find Brick」でNXT接続設定を行ってください。接続されたかどうかは、同じ「Tools」の「Find Brick」がグレー表示されているか、ステータス・バーに接続された情報が表示されるかで確認できます。正しく接続されている場合には、ステータス・バーの**接続名**に「USB0」（USBケーブルの場合）などか「BTH」（Bluetoothの場合）の表示のいずれかとNXTの名前（**NXT名**）が表示されます（**図4-1-2**参照）。

次にコンパイルを行います。このコンパイルは、メニュー・バー「Compile」→「Compile」を選択して実行します。または、ファンクション・キー「F5」を押すか、アイコン・メニューの中の「」を選択してもコンパイルを行うことができます。

【コンパイル】（以下の方法のいずれか）

- ① メニュー・バー「Compile」→「Compile」
- ② ファンクション・キー「F5」
- ③ アイコン・メニュー「」

エディタ画面の下に**コンパイル・エラー**が発生した場合には、コーディングになんらかの文法的な間違いがあることが考えられます。またはファイル名を付けていない場合もエラーが表示される場合があります。

よくあるエラーとしては、処理文の終わりの**セミコロン「;」**の抜けや、中括弧「{」や「}」の抜け、内部関数やコーディングのスペル・ミスなどが考えられます。また、**2バイトの空白文字**（全角スペース）が入っている場合には、エラーを見つけ出すのが難しく、よく初心者になりがちなエラーとなります（**参考**：全角スペースを2文字の半角スペースに変換する場合には、BricxCCのエディタ上で「Ctrl+R」キーを押し、「Replace text」（文字列の変換）画面を表示させ置換する）。

本章の終わりの「**COLUMN 4-1** プログラムのデバッグ」には、デバッグしやすい環境設定についてまとめてあるので参考にしてください。

(3) NXTロボットへのプログラムのダウンロード

正しくプログラムがコンパイルされたら、次はNXT本体にプログラムをダウンロードします。前述したように、BricxCC画面のステータス・バーに、PCとNXTが正しく接続されているかどうかを確認してください。また未接続の場合には、接続名に「no port」と表示されます。この場合には、USBケーブルが未接続か、NXTの電源がOffの状態などが考えられるので、確認・修正し、メニュー・バーの「Tods」→「Find Brick」もしくは「Shift + ctrl + F3」で再接続を実行してください。

それではNXTにプログラムをダウンロードするために、メニュー・バー「Compile」→「Download」を選択するか、ファンクション・キー「F6」を押します。または、アイコン・メニューの「▼」を選択します。

【ダウンロード】（以下の方法のいずれか）

- ① メニュー・バー「Compile」→「Download」
- ② ファンクション・キー「F6」
- ③ アイコン・メニュー「▼」

【注意：NXT本体に、プログラムやファイルなどを64個以上保存しようとしたり、256Kバイト以上のファイル容量を保存しようとしたりすると、新たなプログラムのダウンロードはできなくなる。この場合にはメニュー・バー「Tools」→「NXT Explorer」（NXTエクスプローラ）で、NXT本体の中の不要なファイルを削除するか、PCに移動する】

(4) NXTロボットのプログラム実行

それではNXTにダウンロードされたプログラムを実行してみましょう。USBケーブルが接続されている場合やBluetoothが接続されている場合には、メニュー・バーの「Compile」→「Run」を選択するか、ファンクション・キー「F7」を押します。またはアイコン・メニューの「▶」を選択します。

ここでUSBケーブルを引き抜いた状態でも実行できます。この場合には、NXT本体のディスプレイ上の「My Files」を選択し、「Software Files」の中から、PC側で付けたファイル名を表示させて実行（オレンジ）ボタンを押します。

【プログラムの実行】（以下の方法のいずれか）

■USBケーブルかBluetooth接続状態の場合

- ① メニュー・バー「Compile」→「Run」
- ② ファンクション・キー「F7」
- ③ アイコン・メニュー「▶」

■USBケーブルを切り離した場合

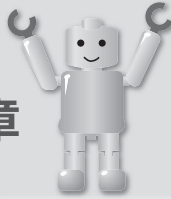
- ④ NXTの【実行】（オレンジ）ボタンを数回押す

思ったとおりに前進と後退ができたでしょうか。





第5章



ロボットを 複雑に動かす



前章では、簡単なプログラミングとして、NXTロボットの前進と後退、それに停止について学びました。本章では、NXTロボットを少し複雑に動かしてみましよう。例えば、旋回させたり、渦巻き状に動かしたり、正方形上に動かしたりしてみましよう。さらに、初歩的な段階での、センサの使い方や、NXT本体のディスプレイ機能やサウンド機能についても学んでいきます。応用編では、より高度な使い方を紹介しているので、部分的にのぞいてみてはいかがでしょうか。

ここでは、C言語の基本制御(コントロール)を行う文(ステートメント)を学んでいきます。この基本制御の中の**繰り返し文**は、プログラム処理を繰り返すときに使うもので、ロボットを複雑に動かすことができるようになります。

また、ここではプログラムの**見やすさ**を学びます。プログラムは、自分自身が関係したり、場合によってはほかの人が関係したりして、時間が経った後でも利用することがあります。このためにプログラムの見やすさは重要な意味をもちます。この方法として、コメントの記述方法や、中括弧「{……}」を使った**入れ子の表現**、それに各行の段下げを行う**インデント**について学びます。

それでは、NXTロボットの動きをまず想定し、どうプログラミングしていくかを学んでいきましょう。

プログラミング作業は、想定した動きを実現化する作業となります。モノづくりでは、この想定・計画するものを初期段階で図化したり文書化したりします。これが**設計段階**と呼ばれ、ときにみんなで議論し、知恵を絞って、具体化していく作業となります。場合によっては、試行錯誤といったことも行いながら作業を進めていくこともあります。ここでは、一部**フローチャート**を記載しています。このフローチャートは、設計の初期段階での図化、つまり「見える化」の作業となり、プログラム設計の一つの手法となります。



5-1 ロボットの旋回(簡単なプログラミング)

前述したロボットTribotの片方の車輪を一時停止したり、逆回転で動かすことで、ロボット自体を回転させたり、旋回させたりすることができます。

このことで、ロボットの動きに、直進、左旋回、直進といった動きを与えてみましよう(図5-1-1)。どういったプログラムになるでしょうか。

すでに、第4章で学んできた内容で、ここでのプログラムはできます。トライしてみてください。

回答は一つとは限りません。リスト5-1-1にその例のプログラムを記載しておきます。

それでは、プログラムをコンパイルし、NXT本体にプログラムをダウンロード・実行してみま

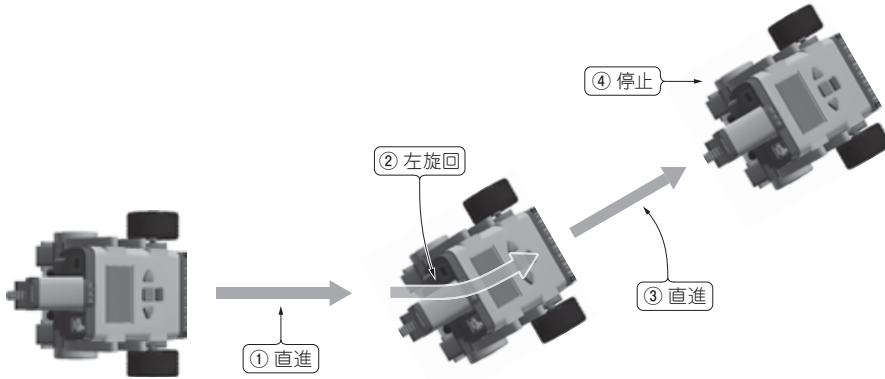


図5-1-1 左旋回直進

リスト5-1-1 直進+左旋回+直進のプログラム

```

1. task main()
2. {
3.   OnFwd(OUT_AC, 75); } ← ① 直進
4.   Wait(1000);
5.   Off(OUT_C); } ← ② 左旋回
6.   Wait(1000);
7.   OnFwd(OUT_AC, 75); } ← ③ 直進
8.   Wait(1000);
9.   Off(OUT_AC); ← ④ 停止
10. }
```

よう。②の左旋回はどうなりましたか。どの程度左旋回したでしょうか。また、タイルやカーペット、畳などの違う床材の上で、どのように旋回するかも見てみましょう。床面が異なると、必ずしも同じ回転角度で旋回していないのがおわかりでしょうか。つまり、回転角度を思ったとおりにするには、いろいろと試行錯誤の調整が必要で、特に②の中の「Wait」の待機時間の数値を変えることとなります。

その他、モータ・パワーの大きさやバッテリー状態によっても、回転角度は変わることがあるので注意してください（第8章で紹介するNXCの高度なモータ制御関数群を使えば、精度を高めた回転角度が制御できるようになる）。

5-2 繰り返しの動き(わかりやすいプログラミング)

ここでも、先に図5-2-1にあるNXTロボットの動きを見てもらい、どうプログラミングしていくかを考えていきましょう。この図を見て、どのようなことが必要か、どうすれば解決できるか。常に考える習慣を身に付けてプログラミングするようにしてください。いきなりプログラムを作成す

リスト7-5-3 暗くなると音発生

```
1. #define LIMLIGHT 70
2. task main()
3. {
4.     SetSensorLight(S3,false);
5.     while(true)
6.     {
7.         if(SENSOR_3<LIMLIGHT)
8.             PlayTone(262,100); // 暗くなると音を出す
9.     }
10. }
```

リスト7-5-4 光センサ値に合わせた速度処理

```
1. task main()
2. {
3.     SetSensorLight(S3,false);
4.     while(true)
5.     {
6.         OnFwd(OUT_AC,SENSOR_3);
7.     }
8. }
```

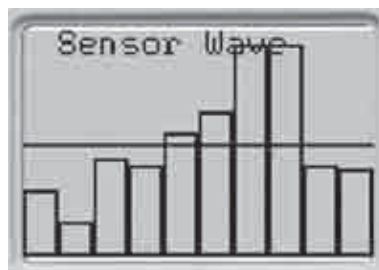


図7-5-1 光センサで取れた動的グラフの表示

て棒グラフを描いています。

また、このグラフを動的にするために、14行目と15行目とで、値を一つずつずらし、新たな光センサ値を最終配列値の「dx[9]」に設定しています。

それでは、実際に動かして動的になるかどうかを確認してみてください。

このプログラムは、いろいろと応用が効くので、ほかのセンサでも使いまわしてどうなるか試してみてください。また、22行目の待機時間も少し変更したらどうなるかも試してみてください。

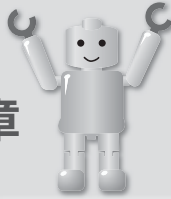
7-6 カラー・センサを使いこなす

カラー・センサは、すでに表7-2-5で紹介したように、色識別(カラー判定)機能と、光センサと同じ光測定機能の二つの測定値が取り出せます。また、カラー・センサには、3色(赤、緑、青)のLEDが付いていて、全LEDの点灯と消灯、それに単独のLEDの点灯の切り替えができ、全LED点灯のみ色識別を行い、その他は光測定となります。色識別の機能は、6色(黒:1、青:2、緑:3、黄:4、赤:5、白:6)を区別します。ただ、中間色や混合色だと、識別が正しくできない場合があります。





第8章



高度なモータ制御



モータは駆動部品の一つで、ロボットを動かす上で欠かせない出力装置です。ロボットをより高度で複雑に動かすには、いろいろと高性能なモータを使う必要があります。例えば、瞬時に回転させたり、速度を一定にしたり、また回転角度や回転数で制御したり、さらに複数モータの同期（シンクロ）をとって動かしたりするモータが必要となる場合があります。これらの機能を使って、センサを使った倒立ロボットや正確に移動するロボットなどが実現できるようになります。

2-1節で紹介したように、標準のNXT製品には、高性能な光エンコーダを内蔵するアクチュエータと呼ばれる動力部品が3個付属しています。このアクチュエータは、回転角度を制御したり、互いの同期をとったりすることができます。このことで正確な回転・距離移動や複雑な制御などが可能となります。

NXCでもこのアクチュエータを制御する関数群が多く用意されていて、簡単な前進・後退・停止から、両輪の同期、正確な距離移動、それに複雑なPID制御ができるようになっています。

本章では、これらNXCの関数群でも簡単に操作・制御できるいくつかの関数群を紹介しましょう。

特に正確な移動距離の制御ができれば、直角に曲がる方法や、円を描くことができるようになります。ただし、注意事項として、バッテリーが少なくなったり（パワー不足）した場合や、床の材質が滑りやすかったり摩擦が多かったりした場合には、直角に曲げたり、円を描いたりすることは難しくなります。その点を気にしながら、プログラミングを進めていってください。

この章でもTribot（トリボット）を使って説明していきます。



8-1 調整機能をもつモータ制御 (OnFwdReg)

第4章では、二つのパラメータ（出力ポートとパワー）でモータを制御する「OnFwd」と「OnRev」を紹介しました。この関数群では、床の材質などによっては、左右のモータが同じような動きをせずに、右にずれたり、左にずれたりし、直線で動かすことが難しかったと思います。

ここでは、NXCが用意しているモータ制御関数のうち、モータを調整する制御機能の「OnFwdReg」と「OnRevReg」について紹介しましょう。

この二つの関数は、モータの調整（Regulate）を取りながら制御していく関数で、以下の引数（出力ポート、パワー、調整）をもったものとなります。

【調整機能をもつモータ制御関数】

```
OnFwdReg(出力ポート, パワー, 調整); // モータ前進の調整制御  
OnRevReg(出力ポート, パワー, 調整); // モータ後退の調整制御  
ここで調整は, 「出力ポートの調整モード定数」で以下のとおり.
```

```
OUT_REGMODE_IDLE      : 調整なし  
OUT_REGMODE_SPEED     : スピード調整  
OUT_REGMODE_SYNC      : 同期調整
```

この関数の引数で, 出力ポートとパワーは, すでに4-1節で説明してきました. ここでは, 調整に関する引数について三つの調整パラメータ「OUT_REGMODE_IDLE」, 「OUT_REGMODE_SPEED」, 「OUT_REGMODE_SYNC」を紹介します.

まず「OUT_REGMODE_IDLE」は, アイドリング(idle)状態のことで, モータの調整をしない機能となります.

次の「OUT_REGMODE_SPEED」は, スピードを調整する制御で, 実際にはパワーを一定に保つように制御する機能となります. つまり, 前の「OUT_REGMODE_IDLE」では, 動いている状態のパワーは指定した値ではなく, 近い値で動きますが, この「OUT_REGMODE_SPEED」は, 指定されたパワーで制御を保ちます.

また, 「OUT_REGMODE_SYNC」は, 指定された出力ポートにおいて, 複数のモータの同期(Synchronize)を取って制御します. 例えば, 出力ポートに「OUT_AC」としていた場合には, 出力ポートAとCに接続されたモータの回転数は同じ状態になるよう同期を取りながら動きます.

それでは, これらの三つのパラメータの違いを, 以下のモータ状態表示関数を使って, プログラムを作成し確認してみましょう.

【モータの状態表示関数】

```
char MotorActualSpeed(出力ポート); // モータのスピード値取得  
long MotorRotationCount(出力ポート); // モータの回転数取得  
long MotorTachoCounter(出力ポート); // モータの回転速度値取得
```

リスト8-1-1のプログラムで調整パラメータを変更させながら値がどう変わるかを観察してみてください.

8-2 同期をとるモータ制御(OnFwdSync)

次に, 複数のモータの同期を取る制御関数「OnFwdSync」を紹介します. この制御関数は, 複数のモータが常に同じ回転数や回転比率になるように調整しながら前進したり, 後退したりします. このことで, よりまっすぐ前進・後退ができるようになります.



見本

ISBN978-4-7898-1895-7

C3055 ¥2600E

CQ出版社

定価：本体2,600円（税別）



9784789818957



1923055026001

このPDFは、CQ出版社発売の「知的LEGO Mindstorms NXTプログラミング入門」の一部見本です。

内容・購入方法などにつきましては以下のホームページをご覧ください。

内容 <http://shop.cqpub.co.jp/hanbai/books/18/18951.htm>

購入方法 <http://www.cqpub.co.jp/hanbai/order/order.htm>