

日本語プログラム言語 「なでしこ」公式バイブル

ver1.5対応版

クジラ飛行机 [著]

日本語で話すように書くから、
分かりやすい。
基本文法からリファレンスまで
みっちり収録。



「なでしこ」
作者による
ユーザー必携
の1冊!



日本語プログラム言語 「なでしこ」公式バイブル

ver 1.5対応版

クジラ飛行機 [著]

●サンプルファイルのダウンロード

<http://book.mynavi.jp/support/pc/5184/>

●そのほかの情報について（以下はどちらも同じページにつながります）

<http://nadesi.com/top/index.php?18&go>

<http://nadesi.com/top/index.php?> なでしこ本について

●本書は

『日本語プログラミング言語「なでしこ」公式ガイドブック』（2005年9月刊行）を元にした電子版/オンデマンド版です。

●本書中では、電子版/オンデマンド版制作にあたり、全体のプログラムが電子版制作時のバージョン（ver 1.543）で動作することを確認しています。

●本書の紙版にはディスクが付属していましたが、電子版/オンデマンド版には付属しておりません。上記本書サポートサイトから、データをご入手ください。

●書籍中に、ディスクに関する記述や解説がある場合は、適宜読み替えをお願いします。

●そのほかの本書中の情報は、基本的に上記書籍執筆段階（2005.8）のものです。ソフトウェアのバージョン、URL、製品のスペックなどの仕様や情報は、すべてその原稿執筆時点でのものです。執筆以降に変更されている可能性がありますので、ご了承ください。

・本書中の記述は、Windowsの基本操作をマスターしていることを前提としています。Windowsの機能および操作法に関してはWindowsの入門書などを参考にしてください。

・本書中の実行図は、一部を除くすべてがWindowsXPを活用したものです。他のバージョンを活用した場合、実行図が変わることがありますのでご注意ください。

・サンプルファイルの使用に際しては、必ず本書7頁の「本書の使い方」をお読みの上でご利用してください。

・本書のサンプルファイルに含まれるデータやプログラム、ファイルはすべて使用者の責任においてご利用ください。本書の掲載内容や、サンプルプログラムのデータ、プログラム、ファイルの使用によって発生した、いかなる損失、損害、その他いかなる事態についても、弊社及び著作権者は責任を負いかねます。

・本書に登場する製品の機能や、URL情報などは本書執筆時点でのものです。執筆以降、予告なく変更されることがあります。

・本書に記載された内容は、情報の提供のみを目的としております。したがって、本書を用いての運用はすべてお客様自身の責任と判断において行ってください。

・本書中の会社名や商品名は、該当する各社の商標または登録商標です。本書中では™および®マークは省略させていただいております。

・なでしこはオープンソースのソフトウェアで、著作権はクジラ飛行機にあります。また、なでしこは多くの有益なオープンソースのライブラリを利用しています。なでしこで作ったアプリの配布や、各ライブラリ利用に関しましては、なでしこ同梱のライセンスをご覧ください。



はじめに

本書は、日本語のプログラミング言語「なでしこ」の公式解説本です。なでしこの作者、クジラ飛行機が誠心誠意をこめて書きました。

日本語プログラミング言語「なでしこ」は、IPA(情報処理推進機構)の2004年度未踏ユースプロジェクトに採択されました。なでしこの、コンセプトは「誰でも簡単プログラミング」です。

なでしこの特徴：

- (1) 『日本語』を使ってプログラムを作ることができます。
- (2) プログラミング入門に最適です。
- (3) ネットワークやファイル処理など役に立つ豊富なライブラリがあります。

一般的なプログラミング言語は、英語を基本にしているので、プログラミングの前に難解な英単語の壁を越えなくてはなりません。しかし、なでしこなら、普段私たちが話している日本語を使ってプログラムを作ることができます。小学生からお年寄りまで、ワープロ感覚で手軽にプログラミングを始めることができます。

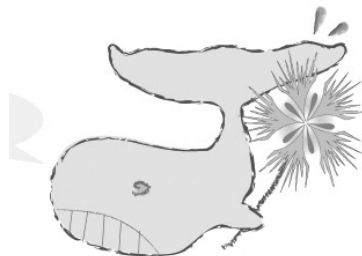
プログラミング初心者の行く手を阻むのは、英語だけではなくありません。プログラミング言語が持つ複雑な文法や構文表現が襲いかかります。なでしこの文法は単純です。そして、できるだけプログラムを日本語として読めるように工夫しています。それで、自分がどんな命令を書いているかが分かりやすく、プログラミング入門にぴったりなのです。

加えて、なでしこはオフィスでも役立つさまざまなライブラリを持っています。これは、もともと「なでしこ」の開発動機が、私自身の事務処理を手軽にプログラムで片付けたかったからです。パソコンで行う単純作業を自動処理するための汎用性の高いツールが欲しかったのです。そのため、なでしこには、メールを送信したり、ファイルを圧縮したりと、実用的な用途に使える命令が900個以上備わっているのです。

本書はプログラミングの入門から、オフィスでの仕事を自動化するためのアイデア、面白いゲームを作るための方法までを一通り解説します。実際に自分の作ったプログラムが、趣味や仕事に役に立ったときの喜びといった格別です。

自由自在にプログラムが作れるようになった暁には、パソコンがより便利なものに感じられるでしょう！！本書を通して、皆さんにプログラミングの楽しさをお届けできればいいなあと思います。

2005.8.1 クジラ飛行機





日本語プログラミング言語「なでしこ」

目次

はじめに	3
目次	4
本書の使い方	7

第1章

基礎編

9

STEP-1	なでしこについて	10
STEP-2	なでしこのセットアップ	14
STEP-3	はじめてのプログラム	19
STEP-4	なでしこを電卓として使いたい	23
STEP-5	変数を使いたい	30
STEP-6	ユーザーからの入力を得たい	34

第2章

フロー制御編

39

STEP-1	条件によってプログラムの動きを変えたい	40
STEP-2	インデントについて	47
STEP-3	プログラムを繰り返し実行したい	50
STEP-4	条件に応じて繰り返ししたい	55
STEP-5	制御構文のまとめ	60

第3章

データ編

63

STEP-1	データの型について	64
STEP-2	インデントについて	69
STEP-3	二次元配列変数	76
STEP-4	ハッシュ変数について	83

第4章

関数とグループ編

89

- STEP-1 関数について90
- STEP-2 グループについて95
- STEP-3 グループ機能とオブジェクト指向99

第5章

GUI 編

103

- STEP-1 GUI部品の基本104
- STEP-2 ボタン部品109
- STEP-3 エディタについて112
- STEP-4 リストについて121
- STEP-5 グリッドについて125
- STEP-6 メニューについて128
- STEP-7 ツリーについて134
- STEP-8 ブラウザ部品について137
- STEP-9 タイマーの使い方139
- STEP-10 イメージ部品について140
- STEP-11 フォーム部品について142

第6章

オフィス編

147

- STEP-1 Excel・Wordと連携する148
- STEP-2 重要データのバックアップ術154
- STEP-3 文書の整形160
- STEP-4 ワイルドカード170
- STEP-5 正規表現179
- STEP-6 テキストマイニング184
- STEP-7 CSV/TSVの処理188
- STEP-8 データベース195
- STEP-9 HTML/XML198
- STEP-10 データ形式の変換202

第7章

ゲーム編

209

STEP-1	じゃんけんゲームを作ろう	210
STEP-2	クイズゲームを作ろう	218
STEP-3	シューティングゲームを作ろう	225
STEP-4	アドベンチャーゲーム	231
STEP-5	RPGを作ろう	237

第8章

ツール編

243

STEP-1	FTP機能付TODOメモを作ろう	244
STEP-2	お絵かきチャットを作ろう	251
STEP-3	ニュースリーダーを作ろう	260
STEP-4	CGIを作ろう	265
STEP-5	メールソフトを作ろう	273
STEP-6	本書だけの特別プラグイン	281

付録

283

リファレンス	284
配布用実行ファイルの作り方	420
文法ガイド	421
記号の意味	426

コラム

質問掲示板を有効有効利用しよう！	17
プログラムが動かないときは なでしこ流デバッグ その1	102
プログラムが動かないときは なでしこ流デバッグ その2	146
上達のヒント	208
あとがき	282



本書の使い方

本書を適切に理解するには ● ● ●



本書の目標

本書の一番の目標はプログラミング大好き人間を作ることです。プログラミングができれば便利で楽しいことがいっぱいです。大好きなプログラミングで仕事を片付けることができれば、そんな素敵なことはありません。

執筆コンセプトは以下の通りです。

- お手軽プログラミングでプログラミングの楽しさを満喫すること
- ファイル処理・画像処理・Excel連携などの便利な命令を使って仕事を自動化すること
- オリジナルツールやオリジナルゲームを作ってフリーソフトとして発表すること

また、本書は「なでしこ」の解説本ですが、できるだけ、他のプログラミング言語を覚えたときにも役に立つような知識を提供するように配慮しています。なでしこを習得した後に、さらに他の言語に挑戦するのも良いと思います。

最近では「パソコンは詳しいけれど、プログラミングはしたことがない」という人がほとんどです。パソコンに触れる時間が長い人にこそ、プログラミングをオススメしたいと思います。プログラミングができることで役に立つことがたくさんあるからです。



本書の対象とする読者

本書は以下のような人を対象としています。

- これからプログラミングをはじめたい人
- ちょっと気の利いたツールやゲームを作りたい人
- Excelマクロ、VB、JAVA、C言語に挫折した人
- パソコンで仕事を楽にしたい人
- 日本語とプログラミングが好きな人



本書の構成

本書では日本語プログラミング言語「なでしこ」を使ってプログラムを作ります。プログラムのソースコードが出てきますが、ほとんどが日本語です。プログラム部分には網をかけ、解説部分と区別しています。例題となるプログラムはできる限り短くする一方で、最大限の効果が出せることを目標にしています。

基礎編、オフィス編、ツール作成編、ゲーム編、ネットワーク編と分かれています。

まず、基礎編でプログラムの基礎をマスターしましょう。基礎さえしっかり身につけばあとは応用です。オフィス編（定型処理編）では仕事を自動化するために「なでしこ流仕事術」を覚えましょう。ツール作成編では実際に役に立つツールを作ってみます。次にお楽しみのゲーム編です。ゲーム作成のためには、データ処理やグラフィックなどの技術が必要になりますので定型処理編で覚えた技術を応用します。そして、最後にネットワーク編でインターネットの世界へ旅に出ましょう。



本書だけの特典

なでしこは、インターネットのなでしこ公式サイトから自由にダウンロードできますが、公式サイトからは入手できない、便利なツールや、追加プラグインを本書だけに収録しています。



付録のサンプルファイルについて

このサンプルファイルには以下のファイルが収録されています。

- サンプルファイルの構成

- + sample (本書の全ソースコード)
- + sozai (本書のプログラムから使う素材)
- + plug-ins (特別プラグイン)
- + ReadMe.txt (解説ファイル)
- + index.htm (解説ファイル)

<sample>フォルダ : 本書で解説したプログラムを章ごとに分けて収めています。ファイルはZIP形式で圧縮されているので圧縮解凍ソフトを使って解凍してください。

<plug-ins>フォルダ : 本書だけの特別プラグインが収録されています。なでしこのplug-insフォルダにコピーすることで、なでしこに命令が追加されます。各プラグインの説明はそれぞれのフォルダにある ReadMe.txt を読んでください



動作環境について

なでしこは、以下の環境で動作が確認されています。

- Windows 2000
- Windows 2003
- Windows XP
- Windows Vista
- Windows 7
- Windows 8 / 8.1



サンプルソースコードについて

- ・ サンプルファイルに含まれるデータやプログラム、教材はすべて著作物であり、著作権はそれぞれの著作権者が保有します。本書購入者が学習用として個人で閲覧する以外の使用は一切認められておりません。ご注意ください。
- ・ 本書の内容に関して、営利的・個人的使用にかかわらず、印刷物、ホームページ等へのデータの転載を禁じます。
- ・ サンプルファイルに収録されているデータやプログラム、ファイルはすべて読者の責任においてご利用ください。本書サンプルファイル収録のデータやプログラム、ファイルの使用により発生したいかなる損失や損害、その他いかなる事態についても、著者およびマイナビは責任を負いかねます。

基礎編

いざ、なでしこの扉を開きましょう。
基礎編では、なでしこ自身の紹介、
それから、ゼロからはじめるための
方法を紹介します。

日本語のプログラミング言語「なでしこ」は
日本語で話すようにプログラムが書けるのが魅
力です。プログラミングができると便利なこ
と・良いことがたくさんあります。なでしこな
ら、気負わずはじめられるので、ぜひ、挑戦し
てみてください。

一章では、はじめてのプログラムから、なで
しこを計算機として使う方法、また、変数を使
って計算機以上のものとして使う方法を紹介し
ています。プログラミングの基礎となる部分で
すが、ここだけでも、ぐっとパソコンをより活
用できるようになるでしょう。



● なでしこについて ● ● ●

まずは、日本語プログラミング言語「なでしこ」について紹介します。なでしこの生い立ちから応用分野まで見ていきます。他のプログラミング言語と比較してみて長所・短所も確認してみます。



なでしこの生い立ち

” なでしこで誰でも簡単プログラマー”

こんなキャッチフレーズを掲げて、誰にでも使えるプログラミング言語を目指して開発を行いました。なでしこは、2004年度IPA（情報処理推進機構）の未踏ユースプロジェクトで採択されたプロジェクトの一つです。

主な開発者は、私（クジラ飛行機）です。ですが、未踏ユースやオープンソースのイベントや、なでしこのWEBページに寄せられた意見感想など、多くの人の助言がいっぱい詰まっています。



私は、なでしこの前に、「ひまわり」という言語を作っていました。これは、2001年から作っていたのですが、新たにゼロから作り直したので、名前を変えて再出発することになったのです。

「ひまわり」から「なでしこ」になって、プログラムがより日本語らしくなりました。そして、実行速度もずいぶん高速になりました。



なでしこはプログラミング入門に最適

なでしこは、プログラミングの入門用として最適です。他のプログラミング言語と比べても覚えることが少ないので、ちょっと始めてみようという人にぴったりです。プログラムも上から下へと実行される単純な作りなので、プログラムの動きを把握しやすいと言えます。

他のプログラミング言語は英語を基本としているので、英単語が難しく、ちょっとしたプログラムでも、何をしているものが確認するのも大変です。また、高度なプログラミング技法やたくさんルールを覚える必要もあるので、壁が高いのです。

いきなり、C言語やJAVAなどの本格派プログラミング言語を学ぶ前に、気軽に始めることのできる「なでしこ」で『プログラミングのいろは』を勉強してみてください。プログラミングをはじめるのに必要なエディタや実行環境も全て無料で手に入れることができますし、日本語なので、ワープロ感覚で、ちょっと遊んでみるのにぴったりです。

企業のプログラミング研修でも、日本語のプログラミング言語を利用すると、C言語など一般的なプログラミング言語から学習をはじめるのよりも、より早くプログラミングを習得できるそうです。



仕事に役に立つプログラムも作れる

元とは言えば、なでしこは、会社で雑用担当だった私が、自分の事務雑用を片付けるためのツールとして作ったところから始まっています。ですので、はっきり言って、大規模なソフトの開発には向いていません。その代わりに、事務作業の自動化のためのプログラムを作るのは得意です。ほんの数行のプログラムで、いろいろな仕事を片付けることができます。

例えば、WordやExcelの連携も得意なので、Excelで作った名簿を元に、Wordで招待状やDMを自動作成したりすることもできます。面倒なホームページの更新も楽々です。ページの自動作成から、WEBへのファイル転送まで、全てを自動化できます。また、毎日、仕事の後にデータを自動でバックアップさせたり、パソコンのメンテナンスを自動で行うようにしたり…仕事に役立つ800個以上のライブラリを備えていますので、オフィスでの大活躍は間違いありません。



おもちゃっばい？

多くのプログラマーが集うイベントでなでしこを展示したり、ネットの掲示板などを見ると、なでしこが「おもちゃっばい」という評判を聞きます。以下のプログラムを見てください。これは恋人の誕生日まで何日なのかを調べるプログラムです。



誕生日まで何日.nako

- ```
01: 恋人の誕生日は「{今年}/1 2 / 2 4」。
```
- ```
02: 今日から恋人の誕生日までの日数差を表示。
```

見ての通り全部日本語です。英語のプログラミングに慣れた人には、とても奇異に映るようです。それで、なでしこについて、賛否両論いろいろな意見を聞くことができます。しかし、「おもちゃっばい」ことこそ、なでしこにとって最大の魅力なのです。おもちゃならば、気軽に楽しみながらプログラミングを始めることができるというものです。

また、おもちゃで遊んでいるうちに、趣味や仕事に役立つプログラムを作ることができたなら、なんて素晴らしいことでしょうか。日本語だからと言って、敬遠せず、多くの人に使ってもらえたらいいなあと思います。



趣味のプログラミングにも最適

なでしこは、趣味のプログラミングにも最適です。ある程度、なでしこに慣れてくれば、RPGやアドベンチャーゲーム、パズルゲームなど、ゲームを作ることができるでしょう。

その代わりに、3Dやアクションゲームなど高速に画面を切り替えるのは若干苦手です。プログラミング言語には、それぞれ得意とする分野が異なります。このようなゲームを作るなら、フリーソフトのHSPを使ったり、C言語など本格的なプログラミング言語を使うと良いでしょう。しかし、これらの言語を覚える時も、なでしこで得た知識がそのまま活かせると思います。



数行のプログラムでもスゴイことができる

なでしこには、パッケージ化された命令が多く標準で搭載されています。普通のプログラミングで数百行書かなくてはならないようなプログラムでも、なでしこなら、ほんの数行で実現できるのです。

2005年春、『日本語で10行プログラミング』と題して、マイナビのWEBサイト『マイナビニュース』(<http://news.mynavi.jp/column/nihongoprog/>)にてコラムを執筆していました。

これは、なでしこを使って、毎回、10行以内でプログラムを作るという内容なのですが、10行以内という短いプログラムで、ニュースリーダーや画像ビューワーを作ったりしています。プログラマーの友人にも、どうして10行であんなすごいものが作れるのかとよく驚かれています。

その秘密が、なでしこのパッケージ化された命令にあるのです。よく使うプログラムを短い日本語の命令に縮めているのです。そのため、なでしこの魅力は、『日本語の部分ではなく、豊富に用意された実用的な命令にある』とまで言う人もいます。

それでは、実際のプログラムを見てみましょう。ファイルの圧縮なんて、たった一行です。



ファイル圧縮.nako

```
01: マイドキュメントを「backup.lzh」へ圧縮。
```

MP3などの音楽ファイルを演奏するのも簡単です。



音楽演奏.nako

```
01: 「音楽.mp3」を再生。
```

簡潔にプログラムが書けるので、何のプログラムが書いてあるのか一目瞭然です。他の人の作ったプログラムでも、簡単に改良ができます。



豊富な命令を一覧からD & Dで貼り付けるだけ

なでしこの専用エディタには、命令の一覧表や命令の検索機能など便利な機能があります。命令の一覧表や検索結果から、「表示」や「演奏」などの命令をドラッグ&ドロップで貼り付けることもできます。命令を貼り付けるとプログラムの雛形が挿入されます。先ほど紹介したプログラムも、一覧から命令を貼り付けて、部分的にちょっと書き換えただけのものです。

ドラッグ&ドロップでプログラムのひな形を簡単に挿入できます。

➡ エディタの命令一覧から簡単に雛形を挿入できます！！





応用分野について

なでしこを何に使うのかは、アイデア次第です。プログラミングに対する敷居の低さを利用して、プログラミング教育にも使えるだろうと思います。中学校のパソコンクラブで使っているという報告や、技術の時間にプログラミングの授業をやったという先生の報告も頂きました。

また、開発セットはUSBメモリなどのストレージに入れて持ち運べるサイズなので、漫画喫茶やネットカフェなど、出先でちょっとした仕事をこなすのに使ったり、関数電卓やポケコンの代わりに使ったりもできます。

ゲームの作成はもちろんのこと、名簿帳や家計簿を作ったり、ブラウザやメールソフト、エディタなど、自分だけのアイデアを実現するオリジナルソフトの作成にも使えます。



プラグインで機能の拡張が簡単

なでしこの便利な命令はプラグインの形で提供されています。既に、ファイル、ネットワーク、オフィス連携、文字列処理、RS232Cなどのプラグインがあります。

ですので、もし、C言語やDelphiなど他のプログラミング言語ができる人なら、それらの言語を使って、なでしこを拡張するプラグインを作ることができます。なでしこは、オープンソースですので、既存のプラグインのソースを読むこともできますし、プラグインの作り方のドキュメントも用意されています。



オンラインソフトを公開してみんなに使ってもらおう！

なでしこで作ったプログラムは、なでしこをインストールしてないと動かないのかと言えばそんなことはありません。なでしこで作ったプログラムは、実行ファイル(EXE)形式に変換することができます。実行ファイルにすれば、なでしこを持っていない人のWindowsでも動かすことができます。

なでしこで作ったソフトを、オンラインソフトとしてWEBで公開しましょう。ソフトに対する感想やお礼のメールをもらうことができます。ソフトを『使う側』から『作る側』になるのです。自分の作ったソフトがきっかけとなって、多くの人と友達になれるかもしれません。

プログラミングの楽しさを体験したら、次に、オンラインソフトの作者になって、新しい世界の扉を開いてみてはどうでしょうか。

● なでしこのセットアップ ● ● ●

「なでしこ」はインターネット上からダウンロードできます。さっそくダウンロードして、インストールしてみましょう。



インストールとアンインストール

インストール ● ● ●

なでしこのダウンロードは、以下のURLから行うことができます。ZIPアーカイブをダウンロードし解凍したら、セットアップファイル「SetupNako.exe」を実行します。すると、開発環境のエディタや実行環境が整えられます。

ダウンロードページ

<http://nadesi.com/top/index.php?ダウンロード>

アンインストール ● ● ●

コントロールパネルの『プログラムの追加と削除』からなでしこをアンインストールできます。



➔ アンインストール

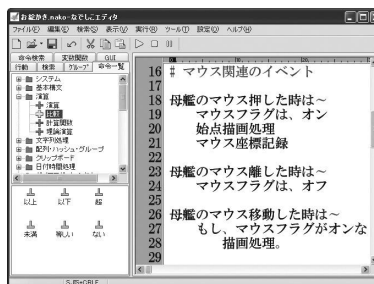


サンプルプログラムを実行してみよう！

なでしこエディタ ● ● ●

なでしこの開発では「なでしこエディタ」を使います。スタートメニューから『すべてのプログラム-日本語プログラミング言語「なでしこ」-なでしこエディタ』を起動してみましょう。プログラミングは「習うより慣れろ」が鉄則です。実際にいくつかのプログラムを実行して遊んでみましょう。

➔ なでしこエディタ



サンプルプログラムを実行 ● ● ●

なでしこエディタを起動したら、まず、なでしこで何ができるのか、サンプルプログラムを実行してみましょう。

サンプルプログラムは、なでしこをインストールしたフォルダにある<sample>フォルダ以下にあります。

メニューの[ファイル]-[開く]をクリックして、サンプルプログラムを読み込んでみましょう。



注意！

エディタのメニューの「ファイル-サンプルフォルダを開く」でサンプルプログラムのあるフォルダを開けます。



↑ サンプルプログラムを開いてみよう！

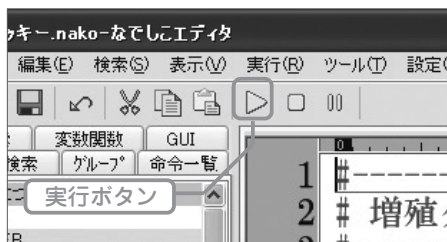
まずは、<sample>フォルダの中の<グラフィック>フォルダから『増殖クッキー.nako』を読み込んでみましょう。(ここでは、ローカルディスク (C:))

➡ インストールしたフォルダのフォルダの中にサンプルがたくさんあります。ここでは、Cドライブの「Program Files—nadesiko—sample—グラフィック—増殖クッキー.nako」を選んでいきます。

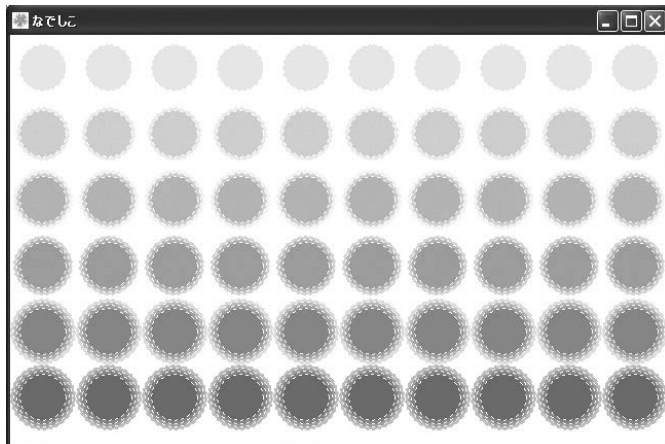


これは、クッキーを模した丸い円が浮かんで消えていくプログラムです。ファイルを読み込んだら、画面上部にある実行ボタンを押してみましょう。プログラムが実行されます。

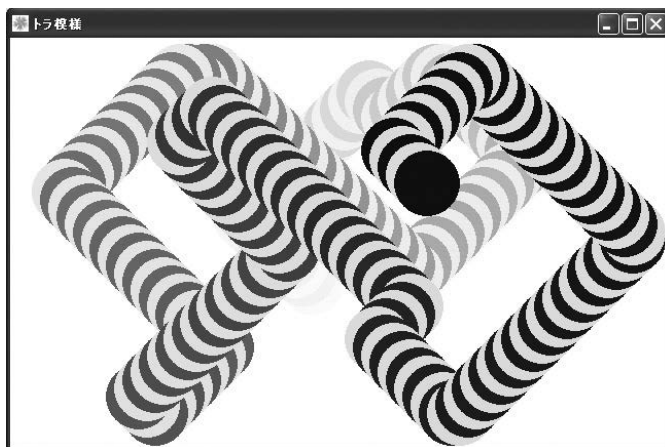
➡ 実行ボタンを押すとプログラムを実行できます。



プログラムが実行されました。閉じる場合は、ウィンドウの閉じるボタンを押してください。



次に同じフォルダにある「トラ模様.nako」を同様に読み込んで実行してみましょう。黄色と黒色のトラ模様を描いてクネクネと丸い円が次々と描かれます。



うまく動かすことができたでしょうか。他にもゲームなどの面白いサンプルがあるので、遊んでみてどんなことができるのか確認してみましょう。



インストールをしたくない場合

インストールの手順で紹介したように、Webからなでしこをダウンロードし、セットアップを行えば、ショートカットを作ったり、スタートメニューを作成したりなど、簡単に行えるのですが、Windowsへインストール情報を登録する、インストールの作業を行わずに『ちょっと使いたいとき』があります。

ちょっと使いたいときは、出張先のパソコンや、会社のパソコン、インターネットカフェなどでちょっと使いたい時です。ちょっとだけ使うのに、わざわざインストールするのは大げさです。しかも、ネットカフェなどではインストールができないように設定されているパソコンもあります。

なでしこはUSBメモリなどで持ち運んで使うことも考慮して作ってあります。むろんインストールしたほうが便利ですが、インストールしなくても、なでしこの配布ファイルを解凍して、デスクトップなどにコピーするだけで、なでしこエディタ (nakopad.exe) を実行することができます。

また、なでしこのエディタを使わなくても、メモ帳やお気に入りのエディタを使ってプログラムを作る事ができます。次の頁の「インストールなし+エディタなしでは」をご覧ください。



最新版をWEBでチェックする

なでしこの開発は、引き続き積極的に行われています。インターネットで常に最新版をチェックしましょう。なでしこの公式Webサイト「<http://nadesi.com>」には、なでしこに関する情報がたくさん載せられていますので、時々チェックしてみてください。



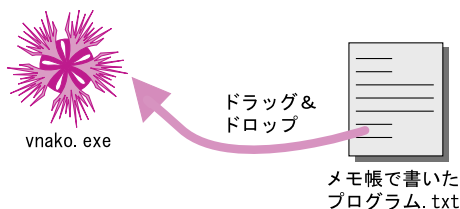
インストールなし+エディタなしでは

一般的に、なでしこの実行に必要なのは、vnako.exe + dnako.dll + <plug-ins>フォルダ + <lib>フォルダです。これだけあれば、メモ帳にプログラムを書いて実行させることもできます。

なでしこのプログラムを書いて保存すると、「(ファイル名).nako」という名前で保存されますが、これはただのテキストファイルです。

ですので、メモ帳にプログラムを書いてプログラムを実行することができます。

例えば、メモ帳に『こんにちは』と言う。』とだけ書いて"hello.nako"という名前で保存してください。そして、保存した"hello.nako"ファイルを、vnako.exeにドロップすると、「こんにちは」とメッセージダイアログが表示されます。こんな感じでプログラムすることもできます。



↑ テキストファイルをvnako.exeにドロップするとプログラムを実行することができます。

もし、お気に入りのテキストエディタがあるのでしたら、それを使って「なでしこ」用のプログラムを書くこともできるわけです。



クジラのコラム

質問掲示板を有効利用しよう！ ● ● ●

WEBならではの強みとしては、なでしこについて質問できる質問掲示板が用意されています。分からないことがあれば、思い切って質問してみると良いでしょう。質問掲示板は、なでしこトップページ (<http://nadesi.com>) から、コミュニティ>質問する板とリンクをたどると見ることができます。

しかし、質問掲示板に質問を書き込んでも、なかなか答えがもらえないこともあります。偶然、他のユーザーさんが忙しかった場合に答えがもらえないという理由もありますが、それよりも、質問の仕方を工夫することによって、より早く答えをもらうことができるかも知れません。そのポイントをお教えしましょう。

● 掲示板を利用するポイント

- 一度にいくつも質問しない・質問は1つずつ
- 問題点は何なのか考え質問を単純にする
- 何ができなかったのか自分が試したことを書き加えておく

まず、一点目、1つの書き込みで質問することは1つにしましょう。いくつも質問すると、何に答えてよいのかポイントが分かりにくくなり、欲しかった答えが得られなくなります。どうしても、いくつかの質問があるときは、何回かに分けて質問を書き込んだ方が良いでしょう。

そして二点目、『何が分からないのか分からない』という質問がよくあります。私も時間がある限り、質問に答えようと思っているのですが、質問に答えようと思っても、何が分からないのか分からない、また、一体何がやりたいのかが分からないということがあります。答えてあげたくても、答えられないの

です。できるだけ何をしたいのか類推して答えようと思っていますが限界があります。自分が回答する人の立場にたって、何が分かれば問題が解決するのかを考えましょう。

最後の三点目、とにかく、質問すれば、誰かが答えてくれる…というのは甘い考えです。『自分はこれだけ調べたけど、それでも分からなかった』という点を書き加えることで、調べ方・考え方があっていいのかだけでなく、何をしかったのが相手に伝わりやすくなります。

これは、なでしこの掲示板だけに限ったことではありません。なでしこ以外にも、いろいろな質問掲示板がありますが、どこへ何を質問するのでも、この3点をしっかり意識しておけば、回答が得られやすくなります。

全てのことが分かっている人なんていません。どんなに頭が良い人でも、誰かに聞かないと解決しないことがたくさんあります。そして、WEBには親切な人がたくさん居ます。誰かの質問に答えたくてウズウズしている人もいます。そんな人の興味を引き付けることができれば、思ったこと以上の情報を得ることができます。答える側の立場にたった質問を心がけましょう。

●プログラムの掲示板で腕を磨こう

なでしこのホームページからプログラムの掲示板にもリンクが張られています。プログラムの掲示板では、誰でも自分の作ったプログラムを投稿することができます。苦労して作ったプログラムがあれば、ここに投稿してみましょう。手塩にかけて作ったプログラムであれば、誰かに使ってもらったり感想やアドバイスをもらえたら良いですね。プログラムの掲示板はそんな場所を提供してくれます。プログラム掲示板を有効利用して、どんどん腕を磨きましょう。

また、なでしこでどんなことができるのか、プログラム掲示板に投稿されているプログラムを実行して試してみるのもいいかもしれません。

●何からはじめたら良いのか

質問掲示板でよく質問されるものに、「プログラミングをマスターしたいんだけど、何からはじめたら良いですか?」というものがあります。この質問に対する答えは様々です。「まずはマニュアルに沿ってやってみよう」とか「サンプルプログラムを自分なりに改造してみよう」などの回答がつきます。私が奨めるのは、本書の前半を読み終わったところで、実際に自分で簡単なものを何か作ってみるのが良いと思います。また、本書のサンプルを本を見ずに自分で考えて作ることができるか試してみましょう。(プログラミング上達のヒントも後ろのページにあります。)

● はじめてのプログラム ● ● ●

プログラム実行の手順の再確認と簡単なプログラムの説明をします。何事もはじめが肝心ですが、気負わず、コーヒー・紅茶を片手にリラックスしてはじめてみましょう。また画面に文章を表示するいろいろな方法も覚えましょう。



とにかくプログラムを書いて実行してみたい！

プログラミングというのは、『習うより慣れる』の部分が多いので、なかなか、本を読んだだけでは使いこなせるようになりません。そこで、多少手間はかかりますが、自分で入力してプログラムの動作を確かめながら進んでください。

もうなでしこはインストールしてありますね？デスクトップやスタートメニューにある「なでしこエディタ」をダブルクリックしてエディタを起動してみましょう。

エディタが起動したら、図1のようにエディタ部分へ以下のプログラムを入力して、画面上部にある実行ボタンを押してみてください。



図1 エディタへプログラムを入力

このプログラムを実行すると図2のように「こんにちは」と画面に表示されます。

図2 実行してみたところ



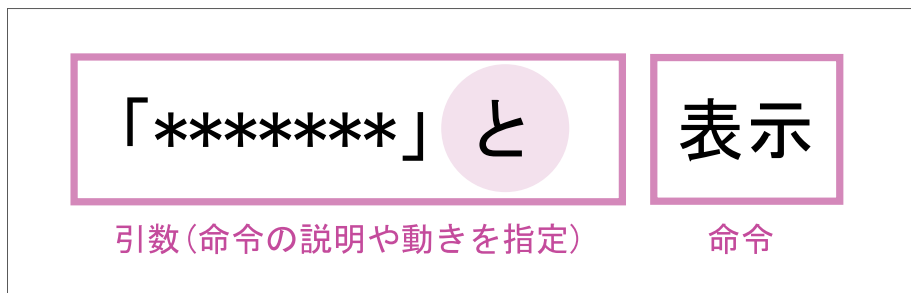
うまく動かなかった場合 ● ● ●

プログラムを実行しても画面に何も表示されなかったという人はいませんか？それは『「こんにちは」と表示』のカギカッコ「…」を書き忘れた場合です。なでしこでは、カギカッコや丸カッコ、タブや改行がプログラムとして意味を持ちます。これを書き忘れてないか確認してみてください。

プログラムの構造 ● ● ●

このプログラムは図3のような構造になっています。『「***」と表示』と書くと、***に指定した文字を表示します。

図3 プログラムの構造



『表示』というのが命令で『「***」と』の部分で命令の説明や動作を表す引数部分です。ここで命令語句と説明語句を区切るの、「と」や「を」などの助詞です。なでしこでは、助詞により語句の意味が区切られるということを覚えておいてください。

他の文章を画面に表示する方法

ダイアログに表示 ● ● ●

次に画面ではなく、メッセージ表示用のダイアログを表示して挨拶をさせてみます。前回と同じようにプログラムを入力し、実行させてみてください。図4のように画面にメッセージが表示できれば成功です。



図4『言う』命令でダイアログへ文字を表示したところ

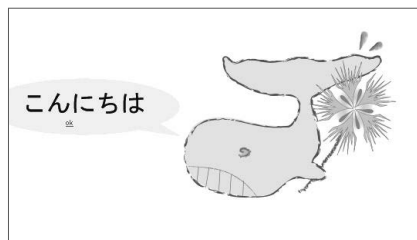


クジラが話す? ● ● ●

次にクジラを使って挨拶をさせてみます。上のプログラムと同じようにプログラムを入力して実行してみましょう。これを実行すると図5のようにクジラが表示されて「こんにちは」と挨拶をします。



図5 クジラがメッセージを話します





適当に書いても動きません。

このように「なでしこ」では、あたかも日本語で誰かに命令するようにプログラムを書いて動かすことができます。しかし、適当に書けば動くというわけではありません。例えば、さっき、クジラと書いていたところを、適当に『エリマキトカゲ』にしてどうなるのか実験してみましょう。

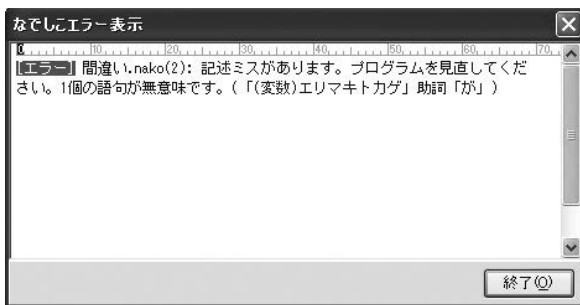


間違い.nako

01: エリマキトカゲが「こんにちは」と言う。

このプログラムを実行すると、図6のようなダイアログが出て『記述ミスがあります。プログラムを見直してください。』と怒られてしまいます。

なでしこは、プログラミング言語です。日本語をただ適当に書いていただけでは動きません。プログラミング言語では、ある一定の規則に従ってプログラムを記述する必要があります。それでこのように、知らない単語が出てくると間違いがあることを知らせるメッセージがでるようになっていきます。



↑ 図6 でたらめな語句を入れるとエラーがでます。



好きな言葉を表示させてみよう！

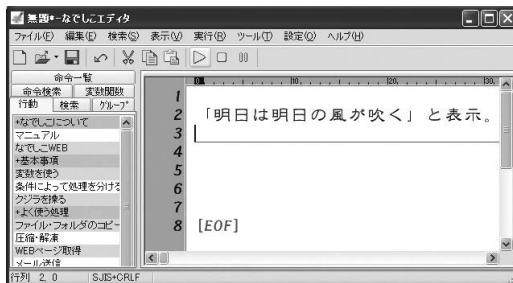
画面に文字を表示するいろいろな方法を見てきました。そこで今度は自分で好きな言葉を画面に表示させるプログラムを作ってみてください。『「★★★」を表示』の★★★のところを書き換えるだけで好きな言葉を表示できます。例えば、以下のプログラムは画面に「明日は明日の風が吹く」という文章を表示します。



好きな言葉.nako

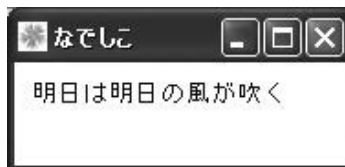
01: 「明日は明日の風が吹く」と表示。

➡ エディタ部分にプログラムを書きます。



実行すると「明日は明日の風が吹く」という文章を表示します。

➡好きな言葉に書き換えて実行してみよう。



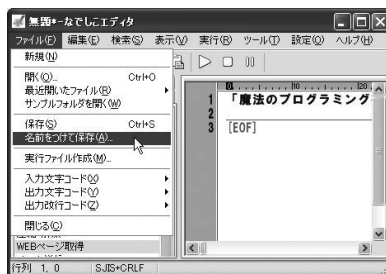
プログラムを保存しよう ●●●

プログラムを保存しておけば、次回使いたい時に、読み込んで再度実行することができます。

プログラムを保存するには、メインメニューから[ファイル - 名前を付けて保存]をクリックします。

するとプログラムの一行目にある行が名前の候補として表示されます。

保存したいフォルダへ移動し、もし適切な名前であれば修正して[保存]ボタンをクリックします。



↑ ファイルメニューから名前を付けて保存を選択します。



用語の確認

ここで今までに出てきた用語や一般的に使われるプログラミング用語をまとめてみます。

プログラム（英語:program）●●●

一般的に『プログラム』と言えば、運動会や演劇の進行手順を順を追って書いたものですが、コンピューターの世界でも、コンピューターに命令するための進行手順を書いたものがプログラムです。

映画やアニメのSFに出てくる『プログラム』は、でたらめな意味の無い記号の羅列であることが多いので、得体のしれないものという印象がありますが、実際に使われている『プログラム』はできるだけ分かりやすく書くものです。

プログラミング（英語:programming）●●●

プログラムを作成することを『プログラミング』と呼びます。また、プログラムを作成する人を『プログラマー』と呼びます。なでしこ開発時のキャッチフレーズは『なでしこで誰でも簡単プログラマー』です。なでしこで簡単にプログラマーになりましょう！

文法（英語:syntax）●●●

プログラムはコンピュータに動作手順を記したものであるので、コンピュータが理解できるように、ある一定の規則で記述しなければなりません。この規則のことを『文法』と呼び、各プログラミング言語によって多少の違いがあります。

エラー（英語:error）●●●

プログラムの間違いをエラーと言います。そしてプログラムにエラーがあることを、バグ(英語:bug)があるとも言います。バグとは虫とかばい菌という意味です。エラーもバグも、どこかで聞いたことのある単語ではないでしょうか。

ソース（英語:source）●●●

日本語で『ソース』と言えば焼きそばなどにかける調味料のことですが、プログラムのことを『ソースコード』や『ソース』と呼ぶこともあります。英語の"source"とは根源とか源の意味です。コンピューターの動く源ということでしょうか。また、プログラムの書かれているファイルのことを『ソースファイル』と呼びます。

● なでしこを電卓として使いたい ● ● ●

コンピューターのことを電子計算機とも言います。パソコンは計算が得意なのです。そこで、なでしこを電卓代わりに使う方法を紹介します。しかし、ただの電卓の代わりでは面白くないので、時間計算機や関数電卓として使う方法も紹介します。

STEP-4



足し算と引き算

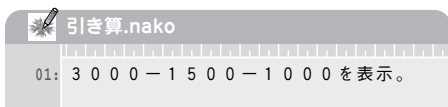
まずは、足し算からやってみます。以下のプログラムを入力して実行してみましょう。



➡ 足し算の結果「7」が表示されれば足し算ができていることを確認できます



次に引き算をやってみましょう。引き算も同じように書く事ができます。今度はちょっと大きな桁で試してみます。



➡ 引き算の結果「500」が表示されれば引き算ができていることを確認できます



この例のように『A-B-Cを表示』や『A+B-C+D+Eを表示』など、計算式はいくつでも並べて書く事ができます。



注意！

このSTEPの画像は、見やすくするために文字を大きく表示しています。文字を大きく表示する方法は、110頁を参照してください。



四則演算

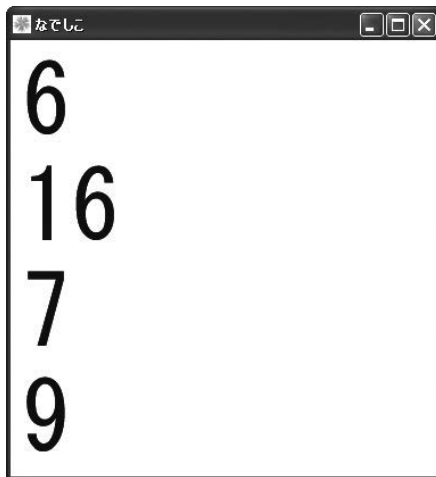
かけ算や割り算も、上の足し算と同じように書くことができます。以下のプログラムを実行して結果を確認してみましょう。



四則演算.nako

```
01: 3 × 2 を表示。
02: 4 × 8 ÷ 2 を表示。
03: 1 + 2 × 3 を表示。
04: ( 1 + 2 ) × 3 を表示。
```

➡計算結果が順に表示されています



実行結果は予想通りでしたか？かけ算や割り算は問題ないですね。ちょっと気になる点だけ補足します。

プログラムの3行目の計算結果を見てください。1 + 2 × 3 の答えが7になってます。足し算とかけ算があった場合、足し算よりもかけ算が先に計算されているのを確認してください。

4行目のように、丸カッコ (...) をつけるとカッコの中を優先して計算するようになります。普通の電卓では、電卓で計算出来るようにあらかじめ頭の中で組み立てないといけませんが、なでしこでは、一般的な計算式の順番にそって計算されるのです。

省略記号 ● ● ●

ところで、「×」という文字を入力するには、漢字変換で「かける」と入力して変換すると出てきます。また「÷」を入力するには「わる」と入力して変換します (Windows標準の漢字変換ソフトIME2003を使った場合)。

これでは、変換がちょっと面倒なので、もっと簡単にできる方法を紹介します。

計算式の中で「×」を「*」(アスタリスク)で、「÷」は「/」(スラッシュ)で代用できるようになっています。これは、他のプログラミング言語やExcelなどの表計算ソフトでも同じです。

ここで、Windowsに標準でついてくる電卓を起動してみてください (スタートメニューのアクセサリにあります。または、なでしこで『「calc」を起動。』と書いて実行するとすぐ起動します。)



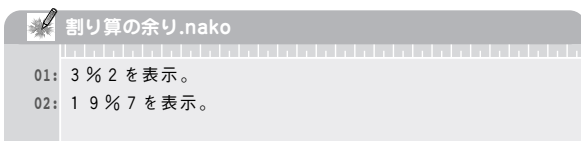
↑ Windows標準の電卓です

これを見ると分かりますが、Windows標準の「電卓」では、はじめから、「×」を「*」、「÷」を「/」と表しているのです。私は仕事柄パソコン初心者の方から質問を受けることがあるのですが「Windowsの電卓って掛け算と割り算できないんですか?」と質問を受けることもしばしばです。

「*」と「/」の記号でかけ算、割り算ができることを覚えておきましょう。

割り算の余り ● ● ●

それから、電卓をよく見てみると、右の方に「%」という記号があります。これは、割り算の余りを求めるときに使います。例えば「3%2」と書けば、3を2で割った余りを求めるという意味になります。ではプログラムを実行して確認してみましょう。

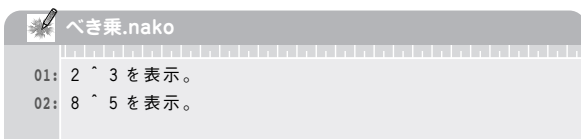


これを実行すると1と5が表示されます。

3を2で割ると1あまり1。19を7で割ると2あまり5なので、このように表示されるのです。

累乗（べき乗） ● ● ●

他によく使う計算として累乗（べき乗）があります。これは「^」という記号で計算できます。累乗とは、ある数に同じ数を次々と何回か掛けることです。



これを実行すると8と32768が表示されます。

『2^3』とは『2×2×2』と同じ意味で答えは8です。同じく『8^5』は『8×8×8×8×8』の意味で答えは32768です。

計算に使える演算子の一覧 ● ● ●

いろいろ計算記号が出てきましたので、使える記号（演算子）をまとめてみました。

記号	意味	使用例	結果
+	足す	3 + 2 を表示	5
-	引く	5 - 3 を表示	2
× または *	掛ける	3 × 2 を表示	6
÷ または /	割る	4 ÷ 2 を表示	2
%	割り算の余り	4 % 2 を表示	0
^	累乗	2 ^ 3 を表示	8

計算の練習 ● ● ●

これで一通り四則演算の計算ができるようになったので練習をしてみましょう。なでしこを使って以下の問題を解くプログラムを作ってください。

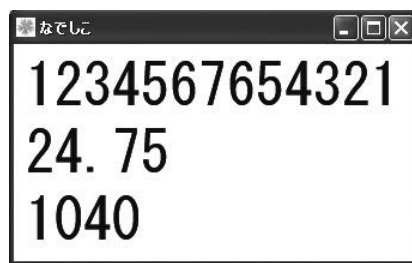
- (1) 1111111×1111111 の答えは？
- (2) $(11 + 22) \times 33 \div 44$ の答えは？
- (3) 100円の消しゴムを5つと、180円のサインペンを3つ買ったらいくら？

答えを見なくてもプログラムを作ることができたでしょうか？

```

計算練習.nako
01: #1 (1) の答え
02: 1 1 1 1 1 1 1 × 1 1 1 1 1 1 1 を表示。
03:
04: # (2) の答え
05: (1 1 + 2 2) × 3 3 ÷ 4 4 を表示。
06:
07: # (3) の答え
08: 1 0 0 × 5 + 1 8 0 × 3 を表示。

```



↑ 計算練習の実行結果です

コメントについて

ちなみに、上のプログラムで「#」(シャープ)から始まる行がありますが、これはコメントと言って、プログラムとして何も意味も持たない文です。注釈文とも言います。ちょっとしたメモ代わりとして使えます。以下のように文の途中にコメントを記述することもできます。

```

コメント.nako
01: # コメントのテスト
02: 「ここはコメントではない」と表示。# ここはコメント

```

『表示』命令についての補足

計算をするのに『表示』は必要？ ●●●

ここで『表示』を書かずに、 $3 + 4$ とだけ書いて実行したらどうなるのでしょうか？やってみると分かりますが、何も表示されません。これはどういうことでしょうか？

```

表示なし.nako
01: 3 + 4

```

➡ $3 + 4$ とだけ書いて実行してみたが何も表示されなかった



『 $3 + 4$ 』と書けば、 $3 + 4$ の計算はされるのです。しかし答えは画面に表示されません。なぜなら、計算

結果を必ず画面に表示してしまうと、たくさんの計算を行った場合によっては画面が数字だらけになってしまふことがあります。

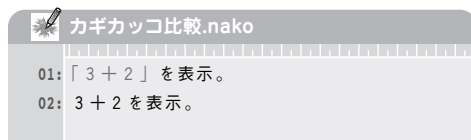
計算結果を画面ではなくファイルに保存したいこともありますし、ネットワークでつながったコンピュータに転送したいこともあるのです。

それで、計算結果を画面に表示したいときは『x x xを表示』と書く必要があるのです。

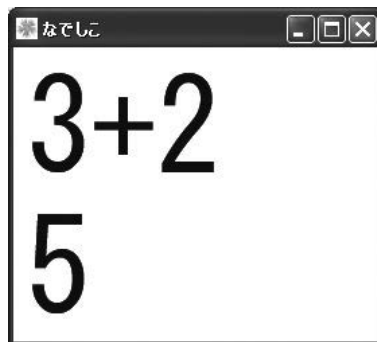
カギカッコは不要なの？ ●●●

『表示』を覚えたとき、文章を表示するときは『「***」と表示』のように、表示する内容をカギカッコで囲う必要がありました。しかし、今回のように計算を行うときには、カギカッコで囲う必要はありません。

確認のために、カギカッコを書いたときと、書かなかったときと、どのようにプログラムの動作が変わるのか比較してみましょう。



➡一行目がカギカッコをつけた場合、
二行目がカッコを書かなかった場合



実行してみて結果に納得ができたでしょうか。カギカッコを書くと、カギカッコの中に書いた内容がそのまま表示されます。そして、カギカッコを書かない場合で、計算式があれば、それを計算した結果が表示されます。

ついでに、ちょっと気の利いた表示にしてみましょう。



➡説明メッセージを計算結果に加えて表示したものです



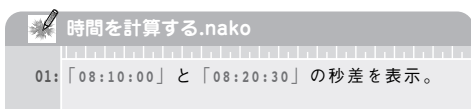
時間計算機・日付計算機として使う

さて、ここまでは「なでしこ」を普通の電卓の代わりとして使った訳ですが、普通の電卓では付いてない「時間計算機」や「日付計算機」として使う方法を紹介します。

時間の計算 ●●●

なでしこを使えば普通の電卓ではできないような計算も、らくらく計算することができます。以下は、8時10分00秒から8時20分30秒の秒差を計算するためのプログラムです。例えば、勉強の開始時間と終了時間

を入れて勉強時間を調べたり、会社の出社時間と退社時間を入れて勤務時間を調べるのに使えますね。

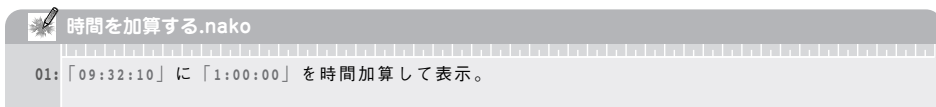


➡『秒差』命令で、8:10:00と08:20:30の秒差を計算して表示してみたところです。



プログラムを実行すると、図のように、計算結果の630が表示されます。時間や日付を指定するときは「08:10:00」のように時間や日付をカギカッコで囲う必要があります。もしうまく実行できなかった場合は、カッコや語句が合っているかを確認してみてください。

今のは時間の引き算でしたが、次に時間を足してみます。9時32分10秒に1時間を足すには以下のように書きます。

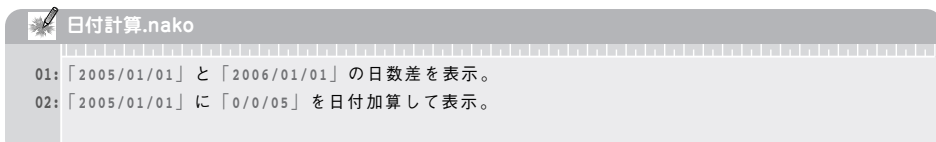


➡『時間加算』命令で09:32:10に1時間を足して表示してみたところです。

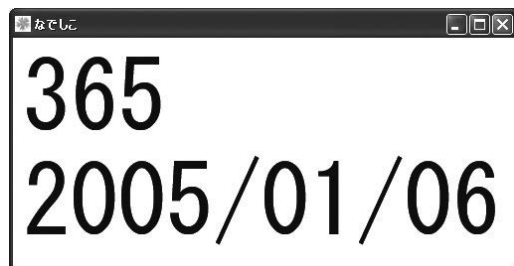


日付の計算 ● ● ●

次は日付についてやってみます。日付も時間と同じように計算ができます。以下のプログラムは、2005年1月1日から2006年1月1日までの日数差を表示します。そして、次いで、2005年1月1日に5日分加算した日付を表示します。



➡1行目が『日数差』命令を使って2005年元旦から2006年元旦までの日数差、2行目が『日付加算』命令を使って2005年元旦に5日足した日付です。



日数差は冒頭の紹介で恋人の誕生日まで何日が調べるのにも利用しました。私は、以前、資格試験を受験するときに、テストまであと何日かを調べ、勉強の進み具合を調べるのにこの命令を使いました。時間や日付の計算は日常生活でも便利なものですね。



関数電卓として使う

今度はなでしこを関数電卓として使う方法を見ていきます。関数電卓というのは、三角関数や自然対数、絶対値などの計算を行うことができる高度な電卓です。日常生活ではあまり使うことがありませんが、少し高度な計算をしたいときは便利です。

関数電卓によくある関数だけを抜き出してみました。以下が利用できる関数の一部です。

関数名	意味
SQRT(x)	xの平方根（ルート）
SIN(x)	x（ラジアン単位）の角の正弦（サイン）
COS(x)	xの角の余弦（コサイン）
ARCTAN(x)	xの角の逆正接（アークタンジェント）
DEG2RAD(x)	ラジアンxを角度に変換
RAD2DEG(x)	角度xをラジアンに変換
EXP(x)	e（自然対数の底）のx乗
LN(x)	xの自然対数（ $\text{LN}(x) = \ln(x)$ ）
LOG2(x)	xの対数（基数2）
LOG10(x)	xの対数（基数10）
ABS(x)	xの絶対値
INT(x)	xの整数部分を返す
FRAC(x)	xの小数部分を返す



関数電卓.nako

01: (SIN(PI/4)^3+COS(PI/3))/5を表示。

なでしこ

0.170710678118655



三角関数を使った計算もこの通り



計算機の代わりに「なでしこ」を使うメリット

Windowsの電卓も使いやすいのですが、電卓では数値を入力して「+」など演算子を入力すると前の数値が消えてしまいます。ちょっと複雑な計算をしようと思った時、どこまで計算したかを忘れてしまって、はじめから計算し直した経験がある人いませんか？

なでしこなら計算式がそのまま残るので、どこまで計算したか一目瞭然です。計算式を入力して間違いがないか確認してから計算結果を得ることができます。

●変数を使いたい●●●

変数とはプログラム中で使う数値や文字などのデータをしまっておくことができる箱のようなものです。変数を使うと数値だらけだった計算式を意味のある単語に置き換えることができるので、ぐっとプログラムが見やすくなります。



計算式を変数で分かりやすく

変数を使うことで、計算式を分かりやすくプログラムにすることができます。例えば、以下のような計算を、なでしこを使ってプログラムにしてみましょう。

気のあった友人が3人集まって食事会へ行きました。そこで、500円のビールを2杯と350円のウーロン茶を1杯、それから、1000円の Pasta を3皿、900円のピザを2枚頼みました。お会計は仲良くワリカンに（3等分）することにします。一人分の支払いはいくらになるでしょう？

ビールを飲まなかった人がちょっと損をしますが、気の合う仲間同士なので気にしなかったことにします。単純にプログラムにすると以下ようになります。



計算式.nako

01: $(500 \times 2 + 350 \times 1 + 1000 \times 3 + 900 \times 2) \div 3$ を表示。

➡ 食事会の計算結果



次に、この計算を変数を使ってプログラムにしてみます。



計算式2.nako

```
01: ビールは、500円
02: ウーロン茶は、350円
03: パスタは、1000円
04: ピザは、900円
05:
06: 人数は、3人
07:
08: 合計金額は、ビール×2+ウーロン茶×1+パスタ×3+ピザ×2
09: 合計金額÷人数を表示。
```

実行結果は同じ2050ですが、プログラムはずいぶん違うものになりました。変数を使っていない数字の羅列と、変数を使って数値を置き換えたものと、どちらが分かりやすいでしょうか。

簡潔に書いてあるのは前者ですが、後者の変数を使った方が、計算式の意味が分かりやすく、間違いがあったときにも、修正が簡単にできそうです。

例えば、もし、以前この店に来たときは、パスタもピザも値段が同じ1000円だったとします。それで、ピザもパスタも1000円で計算したのですが、実際に会計を済ませると、パスタの値段が1300円に値上がりしていました。この場合、前者の変数を使わない方法だと、どの1000が何を表しているのか分からないので式を組み立てなおす必要がありますが、後者の変数を使った方法では、パスタの値段を書き換えるだけで良いのです!!

このように、変数を使うと複雑な計算に意味を持たせ、プログラムを分かりやすくすることができるのです。



算数の公式に当てはめてみる

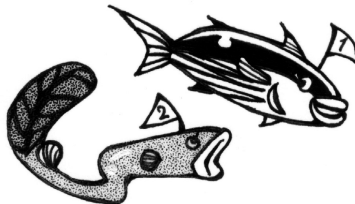
次に、簡単な公式を変数を使ってプログラムにしてみようと思います。以下は速さに関する公式です。この公式はこれだけで既になでしこのプログラムとして動かすことができます。これを使って、小学校の5年生の問題を解いてみましょう。

- 速度＝距離÷時間
- 時間＝距離÷速度
- 距離＝速度×時間

魚のかけっこ問題 ●●●

海の中を泳ぐ魚は、いったいどのぐらいのスピードで泳いでいるのでしょうか。魚の中でいちばん速いのはメカジキです。なんと時速100kmで泳ぎます。つぎはマグロで80kmです。そしてカツオは60kmです。ウナギなどはゆっくりなので時速4kmです。

そこで問題です。カツオとウナギがスタート地点からゴールまで同じ距離を同時に走りました。その時、ウナギはゴールするのに30時間もかかりました。それでは、カツオは何時間かかったのでしょうか？



魚のかけっこ.nako

```
01: # (1) ゴールまでの距離を求める
02: 速度は 4 # ウナギの速さ
03: 時間は 30 # ゴールまでかかった時間
04: 距離＝速度×時間
05:
06: # (2) カツオがゴールまでかかる時間を求める
07: 速度は 60 # カツオの速さ
08: 時間＝距離÷速度
09:
10: 時間を表示。
```

小学生の問題と言ってもなかなか高度です。まずは、ウナギの泳ぐ速さとゴールまでにかかった時間を使って、ゴールまでの距離を求めます。速さと時間が分かっているので『距離＝速度×時間』の公式で求められま

す。次に、ゴールまでの距離が出れば、『時間＝距離÷速度』でカツオがゴールすまでにかかった時間が求められます。

これを実行すると2が表示され、カツオは2時間でゴールできるということが分かります。

地球から月までの距離 ● ● ●

次の問題です。地球から月までの距離を光が往復するのに、2.56秒かかるそうです。光の速さは秒速30万kmとされています。それでは地球から月までの距離は何kmでしょうか？



地球から月までの距離.nako

```
01: 速度は300000 # 秒速30万km
02: 時間は2.56 # 地球から月までかかった時間
03:
04: 距離＝速度×時間
05: 距離を表示。
```

さっきの問題が解ければ、今度のは簡単ですね。ちょっと単位が大きいのので入力間違いに注意すれば問題ないでしょう。『距離＝速度×時間』の公式に当てはめるだけです。プログラムの実行結果は、768000と表示されます。



変数の代入方法

既にここまで見てきた中で、変数へいろいろな数値を代入してきました。なでしこでは、右の3つの方法で代入を行うことができます。

	形式	例
(1)	(変数)は(値)	値段は300
(2)	(変数)=(値)	値段=300
(3)	(変数)に(値)を代入	値段に300を代入

この3つの方法はどれを使っても同じ意味になります。一般的なプログラム言語では、(2)の方法で代入を書きますが、なでしこでは、好きな方法を選べます。



文字列の代入と変数の埋め込み

文字列の代入 ● ● ●

また、変数には文字を代入することもできます。プログラムの世界では、語句や文章などのデータのことを、『文字列』と呼びます。文字が列として連なったデータなので文字列です。

プログラムで数値を指定する時は、そのまま、30とか77とか書きますが、文字や文章などの文字列を指定するときには、カギカッコで囲くという規則があります。例えば、「こんにちは」とか「有言実行」のように書きます。いままでも、文章を表示するときは、カギカッコで囲っていましたがね。



➡ 文字列は必ずカギカッコで囲います

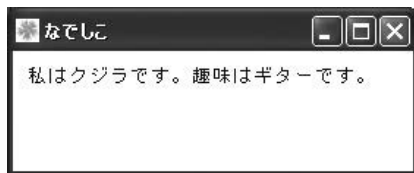


変数を文字列に埋め込む方法 ● ● ●

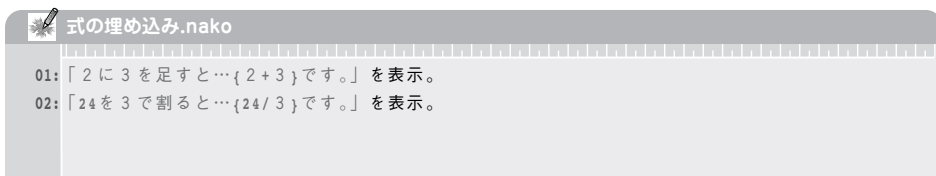
文字列は変数に埋め込むことができます。例えば、上の『文字列の代入』のプログラムにある自己紹介を自然な文章で表示してみます。



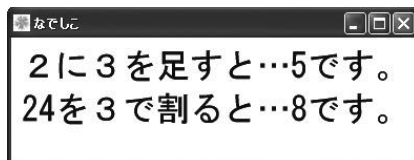
➡ 文字列に変数を埋め込んで表示したところ



このように、文字列の中で「...{変数}...」と書くことにより、変数を文字列の中に埋め込むことができます。また、変数だけでなく、以下のように計算式を埋め込むことも可能です。



➡ 文字列に式を埋め込んで表示したところ



● ユーザーからの入力を得たい ● ● ●

さて今度はプログラムの途中でユーザーから何かデータを入力してもらうプログラムを作ってみます。これによりプログラムを書き換えなくても変数の内容を変えて表示結果を変えることができます。



『尋ねる』命令を使う

ユーザーから何か数値や文字列を入力してもらうには『尋ねる』命令を使うのが一番簡単です。尋ねる命令は以下のように使います。



尋ねる.nako

```
01: 「年齢は？」と尋ねる。
02: それを表示
```

一行目、『「年齢は？」と尋ねる』と書くと、図1のような入力ダイアログが出ます。ここには、数値や文字列などを入力できます。そして『尋ねる』命令で入力された文字列は、『それ』という変数に自動的に代入されます。そこで、『それを表示』と書くと、ユーザーの入力した値がそのまま画面に表示されます。



図1 『尋ねる』命令を使うと入力ダイアログが出て値を入力できます

➡ 18を入力して[決定]ボタンを押すと入力したものが表示されます



インチ→センチ変換プログラム

『尋ねる』の練習として、インチ(Inch)→センチ(cm)の変換プログラムを作ってみようと思います。1インチは2.54cmです。インチと言えば、電気屋さんで、TVの大きさや、パソコンのモニターの大きさを表すのに使われていますよね。以下が変換プログラムです。



インチ→センチ.nako

```
01: 「何インチですか？」と尋ねる。          # 1
02: インチはそれ。                          # 2
03: センチはインチ×2.54                     # 3
04: 「{インチ}インチは{センチ}cmです。」を表示。 # 4
```

プログラムを解説します。1行目の『尋ねる』文では、ユーザーへ「何インチですか？」と質問します。2行目ではユーザーが入力した値を変数『インチ』へ代入しています。3行目では2.54をかけてインチをセンチへと変換しています。

そして最後の4行目では、変換結果を表示しています。ここでは、前項でやった文字列へ変数を埋め込む技を使って分かりやすくプログラムの結果を表示しています。



↑ インチ→センチの変換プログラムで16を入力してみたところ

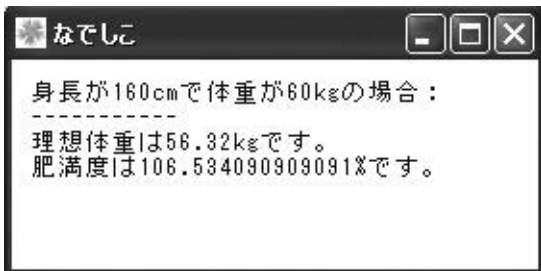
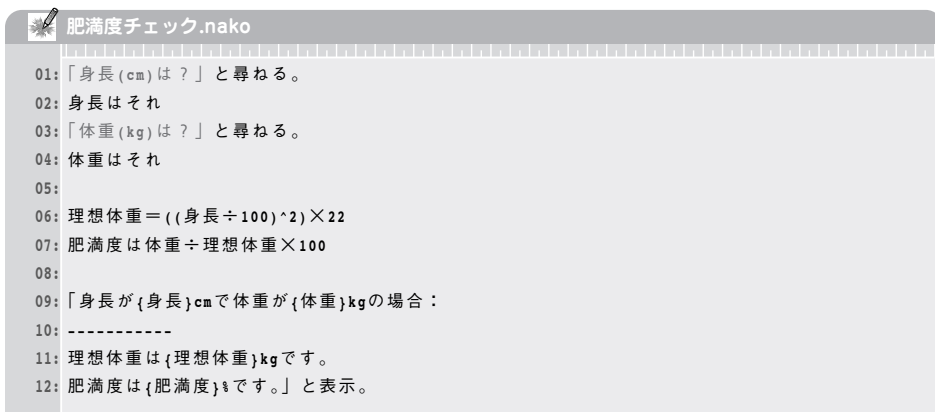
健康チェック～BMIで肥満度を確認

さて、いくつか『尋ねる』命令を使ってプログラム作成の練習をしてみようと思います。手始めに、なでしこで、肥満度チェックをやってみます。簡単な健康チェックの方法にBMIというのがあります。BMIを使うと理想体重を調べることができます。そこでこれを使って肥満度を算出してみます。

もともとBMIとはこの体重なら病気になる確率をもっとも低いという疫学調査から導き出されたものです。理想体重の求め方は以下のようにします。

$$\text{理想体重} = ((\text{身長} \div 100)^2) \times 22$$

この理想体重の算出方法を元にプログラムを作ってみます。ユーザーから身長と体重を尋ね、その入力を元に理想体重と肥満度を算出します。



← 身長160cm、体重60kg
を入力した場合の結果

プログラムの解説をします。これは前の『インチ→センチ変換』プログラムを応用したものです。『尋ねる』命令を使って身長や体重をユーザーから入力してもらい、それを変数『身長』『体重』に代入します。この変数を使って理想体重、肥満度を計算して、最後に計算結果を表示するというものになっています。

最後の『表示』命令ですがカギカッコの合間に改行を挟んでいます。なでしこでは、一度に複数の行を表示したい場合に、このようにカギカッコの中に改行を挟んでも良いことになっています。



ちょっと横道

ちなみに、本来なら男女差や職業による運動量の差も考慮しないといけないので、プログラムの実行結果については参考程度に考えてください。



歩いた歩数と消費カロリーをチェック

肥満度をチェックしたところで、次に歩いた歩数と消費カロリーのチェックをしてみようと思います。

昔から、『一日一万歩歩くと健康になる』と言われていますが、万歩計でも買わないと歩いている歩数なんか分かりません。そこで、体重60kgの人の場合、1分間100歩のペースで歩くと約3.3kcalを消費するというデータから、だいたいの歩いた歩数と消費カロリーを算出してみましょう。



歩数と消費カロリー.nako

```
01: 「今日は何分間くらい歩いたの?」と尋ねる。
02: 歩いた分数はそれ。
03: 歩数は 100 × 歩いた分数。
04: カロリーは、3.3 × 歩いた分数
05: 「{歩数} 歩、歩きました。
06: 消費カロリーは {カロリー} kcalです」と言う。
```



← 40を入力したところ。40分歩いただけでは1万歩には届かない

プログラムの際には「肥満度チェック」を応用したものになっています。そこで、ここでプログラムの基本的な流れを確認してみます。どんなプログラムでも基本的にはこの流れに沿ったものになります。

- (1) データの入力
- (2) 入力されたデータを元に計算など処理
- (3) プログラムの結果を出力

まず(1)のデータの入力というのは、今回で言えば、何分歩いたのかユーザーからの入力を得る部分です。そして、(2)が歩数やカロリーなどの計算部分、最後の(3)で、画面に計算結果を表示するという部分に当たります。

自分でプログラム作ろうと思ったときは、この流れを意識して作ると考えやすいと思います。



卒業年度を調べるプログラム

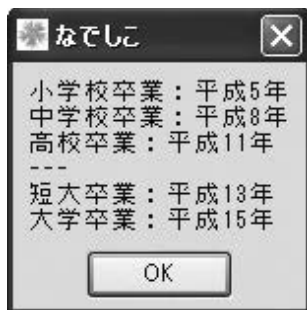
今度は卒業年度をチェックするプログラムを作ってみます。就職やアルバイトの履歴書を書くとき、いつも卒業年を調べるのは大変なことです。そこで、プログラムを使って手取り早く計算してしまおうというわけです。

プログラム自体はただの足し算ですが、ここでは、早生まれの判定はやってませんので、1～3月生まれの人には生まれた年を1つ減らして入力してみてください。この問題については、『もし～ならば～』構文を使うことで改良できますので、また後で改良することにします。



卒業年度チェック.nako

```
01: 「生まれた年度（西暦）は？」と尋ねる。 # (1)
02: 生年はそれ。
03:
04: 小学校は(生年+13)を和暦変換 # (2)
05: 中学校は(生年+16)を和暦変換
06: 高校は(生年+19)を和暦変換
07: 短大は(生年+21)を和暦変換
08: 大学は(生年+23)を和暦変換
09:
10: # 結果の表示 # (3)
11: 「小学校卒業：{小学校}
12: 中学校卒業：{中学校}
13: 高校卒業：{高校}
14: ---
15: 短大卒業：{短大}
16: 大学卒業：{大学}」と言う
```



← 1980を入力してみたところ。

プログラムの(1)の部分は生まれた年を西暦で質問しています。ユーザーから入力された値を変数『生年』に代入します。(2)では、卒業年度を計算し、それを『和暦変換』命令で和暦表記に変換しています。そして(3)では結果を表示します。



プログラムは保存して使おう！

関連付けで実行 ●●●

なでしこプログラムでファイルの拡張子は「.nako」です。インストールすると、拡張子「.nako」となでしこの実行ファイルが関連付けされます。関連付けされると、エクスプローラー上で「xxx.nako」というファイルをダブルクリックした時に、なでしこのプログラムが実行されるようになります。

例えば、以下のプログラムを入力したら、『文字数チェック.nako』というファイル名で保存してください。
(なでしこエディタの[ファイル]メニューから[名前をつけて保存]をクリックすると保存できます。)



文字数チェック.nako

- 01: ファイル選択
- 02: それを開いて文字数を表示。

エクスプローラー上でこの『文字数チェック.nako』をダブルクリックすると、直接プログラムが実行されます。このプログラムは選択したテキストファイルの文字数を数えて表示するという単純なものです。

いままでは、なでしこエディタを起動した後、プログラムを入力して実行という手順を踏んでいましたが、関連付けされていれば、ダブルクリックしただけでプログラムを実行することができるのです。

たった2行のプログラムでも、いちいち入力しなおしたり、エディタから開いて実行するのは面倒です。そこで、これをデスクトップなど実行しやすい場所にコピーしておけば、いつでもダブルクリックでプログラムを実行できるという訳です。



←コードを保存



↑文字数を表示



注意！

実行ファイルの作り方については、420ページ参照。

フロー制御編

プログラムというのは、ただ上から下に1つずつ実行されるだけではなく、途中で分岐させたり、繰り返したりすることができます。

.....

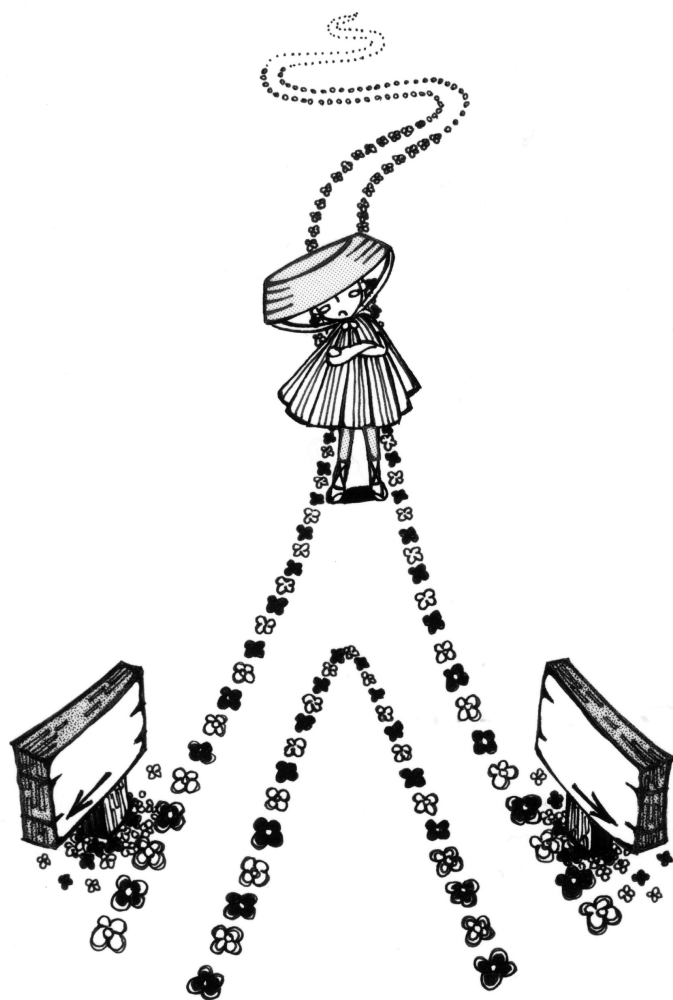
どのような道順で、プログラムを実行させるのか、指示することができるのです。このような、プログラムの走る道順を指示することを、フロー制御と言います。

フロー制御には、以下のバリエーションがあります。

- (1) 条件によって実行するプログラムを変える
➡ (もし…ならば、条件分岐)
- (2) 指定回数繰り返す
➡ (?回…、繰り返す)
- (3) 条件を指定して繰り返す
➡ (…の間…)
- (4) データを1つずつ繰り返す
➡ (反復)

ここではプログラムの走る道順を指示する方法をマスターします。

.....



STEP-1

●条件によってプログラムの動きを変えたい●●●

今までのプログラムは、ただ上から下へと順に実行するだけのものでしたが、今回は、条件によってプログラムの動きを変えてみようと思います。このために『もし…ならば…違えば…』構文を使います。

✿『もし…ならば…』構文

洗濯物 ● ● ●

条件によってプログラムを分けるには、『もし…ならば…』構文を使います。例えば、洗濯物を干すかどうか判別するプログラムを作りたいと思います。洗濯物というのは、晴れた日に干すものです。以下はれっきとした「なでしこ」のプログラムですが、日本語としても通じるのではないかと思います。

✿ 洗濯.nako

```
01: 天気は「晴れ」  
02: もし、天気が「晴れ」ならば「洗濯物を干す」と言う。
```

➡ 天気が「晴れ」だったので「洗濯物を干す」とメッセージが表示されます。



運動会 ● ● ●

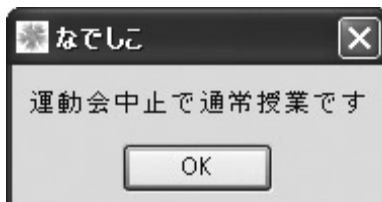
天気の続きで、今度は運動会をプログラムしてみようと思います。さっきの洗濯物は「晴れ」以外のときのことを書かなかったのですが、今度は、「晴れ」以外の場合も考慮してみます。

運動会というのは「雨天中止」と決まっています。つまり、晴れならば、運動会が開催されるのですが、雨が降ったときは、中止となり、通常の授業が行われるのです。

✿ 運動会.nako

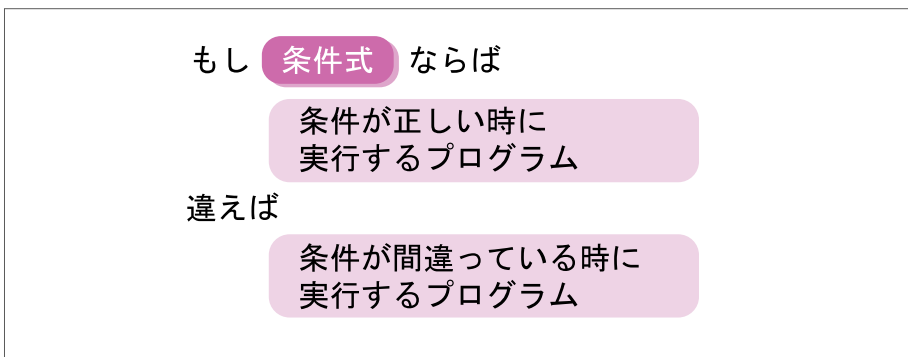
```
01: 天気は「雨」  
02: もし、天気が「晴れ」ならば  
03:     「運動会を開催します！」と言う。  
04: 違えば  
05:     「運動会中止で通常授業です」と言う。
```

➡ 天気が「雨」なので「運動会中止…」のメッセージが表示されます。



上のプログラムで、先に具体的な例を見てきましたが『もし…ならば…違えば…』構文は以下の書式で記述します。条件を指定してその条件が正しいかどうかを判定し、その判定に応じた処理を行いたい時に使います。

▼『もし…ならば…違えば…』構文の書式



一行目に『もし (条件式) ならば』と書いたら、二行目の条件が正しい時に実行するプログラムは、必ず字下げします。また三行目の『違えば』の後の、条件が間違っているときのプログラムも必ず字下げします。

偶数奇数の判定

それでは、実践的なプログラムを作ってみようと思います。手始めに偶数が奇数かを判定するプログラムを作ってみようと思います。言うまでもありませんが、偶数とは2で割り切れる数のことで、奇数と言うのは、2で割り切れない数のことです。

ユーザーから適当な数値を入力してもらい、その数値が偶数か奇数かを判定して、その答えを画面に表示してもらうというプログラムを作ってみます。

```

偶数奇数.nako
01: 「判定する数値は？」と尋ねる。
02: 値はそれ。
03: 割り算余りは、値%2
04: もし、割り算余りが0ならば
05:     「{値} は偶数」と言う。
06: 違えば
07:     「{値} は奇数」と言う。
    
```



↑ 1192を入力すると
偶数と判定されます。

1行目ではユーザーから入力を得ます。そして、2行目で入力結果を、『値』という名前の変数に代入します。そして、3行目で『値%2』を計算して変数『割り算余り』に代入します。『%』の意味覚えてますか？『%』は割り算の余りを求める演算子でした。『値%2』は、値を2で割った余りを調べます。

4行目で『もし、割り算余りが0ならば』と条件が正しいのかどうか判定しています。条件『割り算余りが0』が正しければ、5行目のプログラムを実行し、間違っていれば、7行目のプログラムを実行します。



試し読みはお楽しみ
いただけましたか？

ここからはManatee
おすすめの商品を
ご紹介します。

Manatee Tech Book Zone 

3.6
2017

1

おすすめ
電子書籍

2

Manatee

体験型の本書でプログラミングの第1歩を踏みだそう!

『やさしくはじめるiPhoneアプリ作りの教科書 [Swift 3& Xcode 8.2対応]』は、iPhoneアプリを作ってみたく初心者のための入門書です。プログラミングが初めての人、苦手意識がある人でも楽しく学んでいけるよう、なるべくやさしく、イラストや図をたくさん使って解説しています。本書では実際にサンプルアプリを作りながら学んでいきますが、イラストによる解説で、一歩ずつ丁寧に、iPhoneアプリ作りの基本と楽しさを学べます。

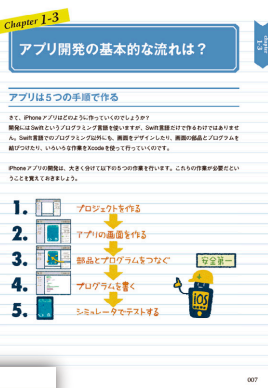
豊富なイラストで「なぜ?」を解消! Javaの第一歩を踏み出そう

『スッキリわかるJava入門 第2版』は、Javaの基礎から初学者には難しいとされるオブジェクト指向まで、膨らむ疑問にしっかり対応しました。Javaプログラミングの「なぜ?」がわかる解説と約300点の豊富なイラストで、楽しく・詳しく・スッキリとマスターできる構成となっています。「なんとなくJavaを使っているけれど、オブジェクト指向の理解には自信がない」「学習の途中で挫折してしまった」という方にもおすすめです。

「プログラミング」

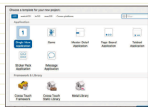
イラストによる解説で、プログラミングはじめての人でも学べる

300ものイラストで楽しく・詳しく・スッキリとマスター!



テンプレートを選ぶ

アプリを作るには、あらかじめ用意されているテンプレートを選びます。テンプレートはあらかじめ用意されているアプリの雛形です。テンプレートはあらかじめ用意されているアプリの雛形です。テンプレートはあらかじめ用意されているアプリの雛形です。



テンプレートを選ぶ

Single View Application: シングルビューアプリケーション。最も基本的なアプリの雛形です。Single View Application: シングルビューアプリケーション。最も基本的なアプリの雛形です。Single View Application: シングルビューアプリケーション。最も基本的なアプリの雛形です。

作成するサンプルアプリはシンプルで、意味を理解しながら作っている

「書くのは、わかっていても書けないから...」
「それは問題ないです!」「大丈夫!」の様に「心配がないから...」

書くのが苦手な人でも、この本で書けるようになります。書くのが苦手な人でも、この本で書けるようになります。書くのが苦手な人でも、この本で書けるようになります。

たくさん文章を表示しよう
画面は2行以上の文章を表示させることもできます。次のように2行を書き足してコンパイル・実行しましょう。

```
リスト 0-4
public class Main {
    public static void main(String[] args) {
        System.out.println("おはよう");
        System.out.println("こんにちは");
        System.out.println("さようなら");
    }
}
```

実行結果
おはよう
こんにちは
さようなら

会話のやりとりの中でも、開発現場でのヒントが詰め込まれている

やさしくはじめる iPhone アプリ作りの教科書 [Swift 3& Xcode 8.2 対応]

マイナビ出版
森崎尚 (著者)、
まつむらまきお (イラスト)
312 ページ 価格: 3,002 円 (PDF)



Swift
(iOS 開発)

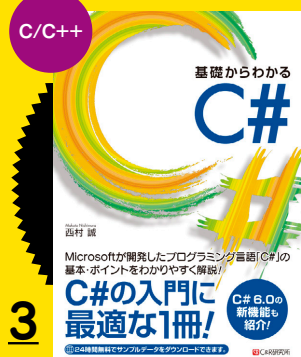
スッキリわかる Java 入門 第2版

インプレス
中山清義・国本大悟 (著者)
658 ページ
価格: 2,376 円 (PDF)



Java

**幅広いジャンルで活躍！
C# がキホンから学べる本**



基礎からわかる C#

本書はプログラミングの経験がある人を対象とした、プログラミング言語「C#」の入門書です。C# の概要から基本的な文法、特徴的な機能まで、わかりやすく解説しています。C# 6.0 の新機能についても解説しています。

シーアンドオール研究所
西村誠(著者)
168 ページ 価格: 1,944 円(PDF)

**プログラミング未経験でも
Android アプリを開発！**



**イラストでよくわかる
Android アプリの作り方
Android Studio 対応版**

親しみやすいイラストや、ステップバイステップでの丁寧な解説が基本コンセプト。開発環境「Android Studio」に対応し、Android のプログラムを作りながら、自然に Java というプログラム言語の知識が身につきます。

インプレス
羽山博・めじろまち(著者)
価格: 2,138 円(PDF)

**JavaScript を網羅的に
取り上げた骨太の 1 冊**



JavaScript 逆引きハンドブック

JavaScript の逆引きリファレンスの決定版。JavaScript の機能を網羅的に取り上げていて、骨太の 1 冊になっています。JavaScript の基本的な処理や便利な Tips はもちろん、HTML5 の API についても数多く掲載しています。

シーアンドオール研究所
古旗一浩(著者)
993 ページ 価格: 3,694 円(PDF)

**初めてのウェブ開発も安心
Ruby の文法を基礎から解説**



**改訂 3 版
基礎 Ruby on Rails**

Ruby の文法やオブジェクト指向の考え方を初歩から解説。アプリケーションのモックアップ作り、データベースを導入し、ログイン・ログアウト機能を加え、最終的にはメンバーや記事の管理ページまでできあがります。

インプレス
黒田努・佐藤和人(著者)
536 ページ 価格: 3,240 円(PDF)

**初歩から順に理解できる
PHP とデータベース**



いちばんやさしい PHP の教本

PHP とデータベースの基本を順番に学んで、実践的なプログラムを完成させていく PHP の入門書です。大事なポイントや勘違いしやすいポイントは講師がフォロー。セミナーを受けている感覚で読み進められます。

インプレス
柏岡秀男・池田友子(著者)
240 ページ 価格: 1,836 円(PDF)

**PHP5.4 の基本から
MySQL との連携まで！**



PHP+MySQL マスターブック

この一冊で PHP と MySQL の基本と Web アプリケーションの構築法について学習できる実践的なプログラミング入門です。現場必須のプログラム構築法、API の活用法から、セキュリティ技術まで詳しく解説します。

マイナビ出版
永田順伸(著者)
384 ページ 価格: 2,916 円(PDF)

はじめてプログラミングに
触れる前に読んでおこう



9

プログラミングの基本から
手取り足取りじっくり解説



10

&

必ずアルゴリズムの意味が
わかるようになる入門書!



11

プログラミングの世界へようこそ

全くの初心者がプログラミングを勉強したいとき、さまざまな疑問が湧いてきます。「どの言語を覚えればいいのか?」「文系でも大丈夫?」本書はプログラミングに触れる前に知っておきたい基本をイラスト付きで解説します。

マイナビ出版
尾川一行・中川聡(著者)
192 ページ 価格: 1,933 円(PDF)

目指せプログラマー!
プログラミング超入門

本書は Windows 開発の標準ツールとも言える「Visual Studio」を使い、C# というプログラミング言語でプログラミングの基本を学びます。最終的には、ちょっとしたアクションゲームが作れるくらいになるのが目標です。

マイナビ出版
掌田津耶乃(著者)
312 ページ 価格: 2,074 円(PDF)

楽しく学ぶ
アルゴリズムとプログラミングの図鑑

図解とイラストを豊富に使ったアルゴリズムの入門書。アルゴリズムとは「問題を解決するための考え方」です。それが分かってくれば、8 種類のプログラミング言語を使ったサンプルプログラムを実際に試しましょう。

マイナビ出版 森巧尚(著者)、まつむらまきお(イラスト)
300 ページ 価格: 2,689 円(PDF)

プログラミングの初心者が
Python 3 を学ぶのに最適



12

Go 言語の基礎から応用まで
ポイントがよくわかる



13

&

R 言語の機能を
目的から見つけ出せる!



14

基礎 Python 基礎シリーズ

プログラミングの初心者を対象にした Python 3 の入門書です。変数の取り扱いから、リスト、タプルといった Python 固有のデータの操作、制御構造や関数などについて、初心者でも基礎から学習できるように説明しました。

インプレス
大津真(著者)
312 ページ 価格: 2,894 円(EPUB)

改訂 2 版
基礎からわかる Go 言語

Google が開発したプログラミング言語「Go」の基礎から応用までをわかりやすく解説した 1 冊です。最新の Go 1.4 のバージョンに対応して改訂しました。Linux、Mac OS X、Windows の各環境に対応しています。

シーアンドアール研究所 古川昇(著者)
240 ページ 価格: 2,138 円(EPUB)

改訂 3 版
R 言語逆引きハンドブック

本書では、最新バージョンの R 3.3.0 に対応し、R 言語の機能を目的から探すことができます。統計が注目を集めるなか、R を利用するユーザーも増えています。初心者でも使えるように、導入から丁寧に解説しています。

シーアンドアール研究所
石田基広(著者)
800 ページ 価格: 4,860 円(PDF)