



日本語プログラミング言語「なでしこ」に関する解説

♡ 14



情報処理学会・学会誌「情報処理」

2021/03/15 10:44

...



クジラ飛行机（くじらはんど）

なでしことは？

日本語プログラミング「なでしこ」は、その名の通り日本語をベースとしたプログラミング言語です。正式版公開から今年で16年目になります。当初から開発スローガンは「なでしこで誰でも簡単プログラマ」です。

母国語の日本語をベースとしたプログラミングを行うことにより、老若男女を問わず、誰でもプログラミングができる環境を提供したいという気持ちで開発を続けてきました。開発の早い段階からオープンソースとして公開しており、ライセンスも自由度の高いMITライセンスを採用しています。次のWebサイトで公開しています（図-1）。



ちなみに、母国語でコンピュータを操作できると言っても「プログラミング言語」であって「自然言語」ではありません。もちろん、自然言語的なアプローチも考えたことはあるのですが、配布の容易さや実行内容の正確性の観点からプログラミング言語の形態が妥当だと考えました。

「なでしこ」は開発スローガンの「誰でも簡単」を実践するため、シンプルで覚えやすい文法を心がけて設計しています。これは英語圏におけるBASIC言語のように、初心者が覚えやすく、すぐに実務で使えることを念頭に置いています。また母国語のメリットを存分に活かすため「プログラムを読んでその動作が分かること」を重視しています。

たとえば、次のプログラムは設定された締め切り日まで、あと何日あるのかを調べるプログラムです。なでしこの文法をまったく知らないとしても、このプログラムを見て意味が分からないという方はいないのではないのでしょうか。

締め切りは「{今年}/12/20」

今日から締め切りまでの日数差を表示

このプログラムは、Webブラウザ上で動かすことができます（図-2）。エディタにプログラムを入力して「実行」ボタンを押すと以下のようにプログラムを実行して計算結果を表示します。



図-2 Webエディタでプログラムが実行できる

なぜ日本語プログラミング言語を作ったのか？

そもそも、筆者が「なでしこ」を開発しようと思ったのは、事務の自動化のための言語が欲しいと思ったところから始まっています。当時、筆者は中小企業でコンピュータに関する業務を一手に引き受けており、膨大な仕事をいろいろなツールを組み合わせでこなしていました。しかし、雑多なツールをその都度用意するのは大変でした。そこで業務を自動化できる簡単なプログラミング言語を作ろう思ったのがきっかけです。

一般的なプログラミング言語を開発した場合、結局自分だけがプログラムを作る

ことになりそうです。しかし、できれば、社内の同僚たちにも使ってもらえるものにしたいと思いました。そこで、日本語をベースとしたプログラミング言語を思いつきました。ゼロからプログラムを組むのは難しいとしても、プログラムを読んでその動作が分かるなら、修正して使ってもらえると考えたのです。

日本語プログラミング言語のメリット

日本語プログラミング言語のメリットは、学習コストの低さと、視認性の高さです。英語がベースのプログラミング言語だと、プログラミングを始めるにあたって英単語を覚えなくてはなりません。しかし日本語プログラミング言語であれば純粋にプログラミングに集中できます。すでにプログラミングができる者からすれば、プログラミングで使う英語など大して難しくはないと思うものです。しかし、実際に話を聞いてみると意外にも専門用語の英単語が覚えられない、英数記号が怖いという声があります。日本語ベースのプログラミング言語であれば、プログラムを日本語として読むことができ、読むとその動作が推測できます。

想定している対象ユーザは？

なでしこが対象としているのは「プログラミング初心者」と「処理の自動化を実現したい人」です。

なでしこ公開から16年が経過し、かつて学生だったユーザが、今では立派なプログラマに成長したという話を聞きます。「なでしこだからプログラミングを始めることができた」と嬉しいコメントを多数いただいています。なでしこを入り口としているいろんなプログラミング言語にステップアップしていくことができます。一度プログラミング技術を身につけてしまえば別のプログラミング言語の習得は容易です。

また、現在プログラミング入門として、Scratchなどビジュアルプログラミング

言語が用いられています。しかしビジュアルプログラミング言語から、本格的なプログラミング言語に移行するには大きな壁があります。そこで、クッションとして日本語プログラミング言語を使っていただければと思います。母国語で読んで意味の分かるプログラミング言語であれば壁は低くなります。

そして、「なでしこ」の開発動機が事務の自動化だったことから「作業の自動化を実現したい人」にオススメです。豊富な自動化のための命令を利用して、作業を自動化できます。実際、昨年行ったなでしこユーザアンケートによると、ユーザの大半は30代から40代であり、多くの方が業務でも利用していることが分かりました。

なでしこでできること

現在、なでしこには、Windows上で事務作業に特化した「なでしこv1」と、Webブラウザ上で動作する「なでしこv3」の2種類があります。Windows版のなでしこv1には事務を自動化する1,335個の豊富な命令が搭載されています。Excel/Wordの自動化、ファイル処理、GUIプログラミングなどが可能です。

これに対して、Web版のなでしこv3では図形の描画からグラフ描画、GPS、テキストの読み上げなど、Webブラウザが備えるさまざまな機能を利用できます。また、Webサーバアプリ開発に使われるNode.js上でも動作します。これを使うと、ファイル処理やデータベースなどの事務自動化が可能で、機械学習やドローン操作なども可能です。

5分で理解するなでしこの文法

ここでは簡単に「なでしこv3」の基本文法を紹介します。日本語ベースのプログラミングが英語ベースの言語と根本的に異なる部分が、トークン（文法の最小要素）をどのように区切るかという部分です。英語ベースの言語ではもともとスペー

スや記号を利用して語句を区切ります。しかし、日本語では単語と単語を語句を明確に区切る習慣がありません。

筆者が最初に作った言語「ひまわり」では、使用者に単語ごとの分かち書きを強制していました。そのため、冒頭のプログラム例で言えば『今日から、締め切りまでの、日数差を、表示』のように不自然に読点を加えてプログラムを書く必要がありました。

「なでしこ」では、この不自然さを改善するため「助詞区切り」というルールを採用しました。これは、日本語の文章が基本的に「漢字・カタカナ+ひらがな」という形になっていることに注目したものです。助詞の「の」「と」「から」「まで」などのトークンを区切り記号と見なし自動で分割します（図-3）。この仕組みによって飛躍的にプログラムが自然な日本語に近づき視認性が高くなりました。

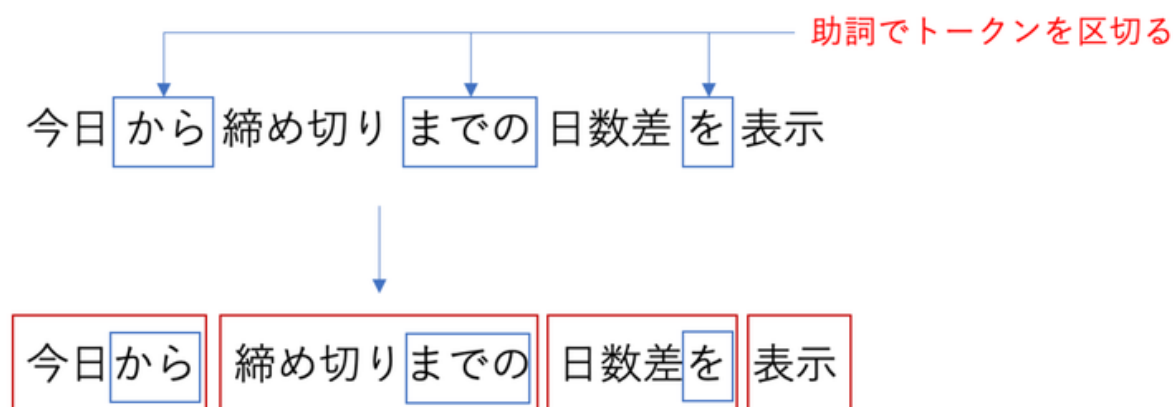


図-3 「助詞区切り」 - なでしこにおけるトークン分割の処理

それから、語尾の送り仮名は自由に記述できます。たとえば画面に値を出力する「表示」命令を使うとき、「表示する」「表示しろ」「表示しよう」「表示してください」と書くことができます。

また命令の引数である助詞に意味を持たせています。たとえば「AからBへファ

イルコピー」という命令があります。これはファイルをパスAからパスBへコピーします。このとき、「BへAからファイルコピー」のように引数の語順を変えても正しく動くようになっています。これにより引数の順番間違いを防ぐことができます。

そして、日本語プログラミング言語では語順が重要です。英語圏で生まれた言語は「関数(引数1, 引数2, 引数3)」のように「動詞＋補語」の形をしています。しかし、日本語では「補語＋動詞」の形が一般的です。そこで、なでしこでは「引数1, 引数2, 引数3, 関数」のような形式でプログラムを記述します。ただし、計算式の中で数学関数を利用したい場合、この語順では不自然になってしまいます。そこで、関数の直後にカッコを記述した場合には、「関数(引数1, 引数2, 引数3)」と記述できるようにしています。たとえば、割り勘の計算を行うプログラムで切上を行う関数「CEIL」を使うプログラムは以下のように記述できます。

人数は3

一人分=CEIL(1000÷人数)

一人分を表示

次に、代入や計算式を次のように記述できます。日本語らしく「代入」を使う方法と、一般的なプログラミング言語のようにイコール(=)を利用する方法の2種類が用意されています。これを状況によって使い分けできます。なお「#」から始まる行はコメントです。

日本語らしく記述

1に2を足して3を掛けて答えに代入

一般的な言語と同じように記述

答えは $(1 + 2) \times 3$

答え = $(1 + 2) * 3$

そして、プログラミングの重要な要素にフロー制御があります。条件分岐は次のように記述できます。これは一般的な言語におけるif文を翻訳しただけと言えます。

天気 = 「晴れ」

もし、天気が「晴れ」ならば

「ピクニックへ行こう」と言う

違えば

「家で読書しよう」と言う

ここまで

繰り返し構文には「回」文を使います。何回目の繰り返しなのかは変数『回数』に自動的に代入されます。次のプログラムを実行すると、画面に「1回目のワン」「2回目のワン」「3回目のワン」と出力されます。見たまま、読んだままに実行されるので繰り返しの概念が理解しやすいと思います。

3回

「{回数}回目のワン」と表示。

ここまで

ほかにも、forに相当する『繰り返す』，foreachに相当する『反復』，whileに相当する『(条件)の間』など基本的な制御構文を備えています。

それから、「なでしこ」ならではの機能である特別変数「それ」を紹介します。これは関数を実行すると命令の実行結果が変数「それ」に代入されるという機能です。また、関数の第一引数を省略すると、変数「それ」の内容が補完されて使われます。これが何の役に立つのかというと、コマンドラインにおけるパイプのような処理を自然な日本語として記述できるようになります。

たとえば、以下のプログラムは文字列処理を行うものです。プログラムを実行すると「タンゴ」と画面に表示されます。

```
それは「りんご」
空白除去
カタカナ変換
「リ」を「タ」に置換
表示
```

前後の余分なスペースを削除する「空白除去」関数、ひらがなをカタカナに変換する「カタカナ変換」関数、文字列の置換を行う「置換」関数を順次実行します。変数「それ」は冒頭に1回しか現れませんが、関数の実行結果が「それ」に代入され、さらに次の命令の第一引数に「それ」の内容が補完されます。命令をただ羅列しているだけですが、次々と値が加工されていきます。

同様に、先ほども紹介した日本語らしい計算式もこの仕組みを利用しています。

以下のプログラムを実行すると20が表示されます。読んだままに計算が行われるのですが、その裏側で変数「それ」が補完されています。

```
# 変数「それ」の省略を利用  
2に3を足して4を掛けて表示
```

そして、関数も定義できます。関数を定義するには、『●(引数)関数名とは...ここまで』のような書式で記述します。たとえば次の例は2割引の処理を行う「割引処理」という関数を定義してこれを使う例です。100の2割引で80と表示されます。

```
# --- 関数の定義 ---  
  
●(値段を)割引処理とは  
    値段に0.8を掛けて切捨てて戻す  
  
ここまで  
  
# --- 関数を使う ---  
  
100を割引処理して表示
```

以上、簡単になでしこの文法を紹介しました。基本的に語順やトークンの区切り位置が違っただけで一般的なプログラミング言語と大きく変わることはありません。しかし日本語になっているだけでプログラムの意味がずいぶん分かりやすく感じるのではないのでしょうか。

タートルグラフィックスを楽しもう

ここで具体的になでしこv3を使ってできることをいくつか紹介したいと思います。いくつかプログラムを紹介しますが、いずれもブラウザで気軽に実行できます。以下のURLにあるWebエディタを使うと便利です。

なでしこv3のWebエディタ

[URL] <http://nadesi.com/v3/new>

最初にタートルグラフィックスを紹介します。これは画面上に表示したカメをプログラミングで操作するプログラムの総称です。カメの動いた軌跡が図形として描画するという機能です。教育用プログラミング言語のLOGOについていた機能ですが、なでしこv3でも教育用途を念頭に置いているので実装されています。なお、この機能はPythonにも実装されており、プログラミング学習の初歩として良い教材となっていることが分かります。図-4に命令をまとめました。

命令	説明
カメ作成	タートルグラフィックスを開始
[x, y]へカメ起点移動	座標(x,y)へ起点を移動
Nだけカメ進む	Nピクセルだけカメを前に進める
Nだけカメ戻る	Nピクセルだけカメを後ろに戻す
Vだけカメ右回転	カメの進行方向をV度右に回転する
Vだけカメ左回転	カメの進行方向をV度左に回転する

タートルグラフィックスを利用して六角形の図形を描画するプログラムは次のようになります。

```

カメラ作成.
6回
    90だけカメラ進む
    (360÷6)だけカメラ右回転
ここまで
    
```

プログラムを実行すると図-5の左上のような六角形が描画されます。このように繰り返しを使うプログラムが自然に記述できます。

また、ちょっと変数を使って工夫すると面白い図形が描画できるので、プログラミングで絵を描いて楽しめます（図-5の右上）。

```

カメラ作成.
N=10
50回
    Nだけカメラ進む
    (360÷6)だけカメラ右回転.
    N=N+3
ここまで
    
```

そして、これはタートルグラフィックスでよく見かける例ですが、関数の再帰処理を使えば複雑な木を模した図形も描画できます（図-5の左下）。10行のプログラムですが、それなりに凝った図形を描画できるとプログラミング学習のモチベーションを上げることができるでしょう。

カメラ作成して[320, 500]にカメラ起点移動

50だけカメラ進み 28の100で枝描画

●(角度のLで)枝描画

もし $L < 10$ ならば戻る

Lだけカメラ進み 角度だけカメラ左回転

角度の $(3 \times L \div 4)$ で枝描画

(角度 $\times 2$)だけカメラ右回転

角度の $(3 \times L \div 4)$ で枝描画

角度だけカメラ左回転して Lだけカメラ戻る

ここまで

Ajaxとグラフの描画

次にAjaxとグラフ描画の機能を紹介します。なでしこv3ではWebアプリでよく使われる処理が手軽に実現できます。

たとえば、次のプログラムは、郵便番号から住所を取得するWeb APIを使うプログラムです。郵便番号221-0824から住所を調べて画面に住所を表示します。

ZIP = "221-0824"

API = "https://api.aoikujira.com/zip/zip.php?fmt=json&zn={ZIP}"

対象をJSONデコードしてJに代入.

J["result"]を表示.

ここまで

なお、助詞の『には』はコールバック関数を処理する特別な構文です。上記のように『には』から『ここまで』の間に書いたプログラムが、非同期通信が完了した時点で処理が実行されます。

次に厚生労働省がオープンデータとして公開しているコロナ陽性者数の推移を記したCSVファイルを利用してみましょう。これは「日付, 人数」が羅列されたCSVデータです。これをAjaxで取得してグラフ描画してみます。わずか4行でグラフが描画できます（ただし、ブラウザにはCORSというクロスドメインの制限があるため、一度CSVファイルをなでしこエディタにアップロードした上で実行する必要があります）。

URL=「/v3/storage/images/6.csv」 # アップしたファイル

URLにAJAX送信した時には

対象をCSV取得して線グラフ描画

ここまで

プログラムを実行すると図-5の右下のようなグラフが描画されます。「CSV取得」命令を使うとCSVデータを二次元配列に変換します。そして「線グラフ描画」命令でグラフを描画します。この命令は一般的なCSVファイルの形式を認識してグラフを描画するものとなっています。

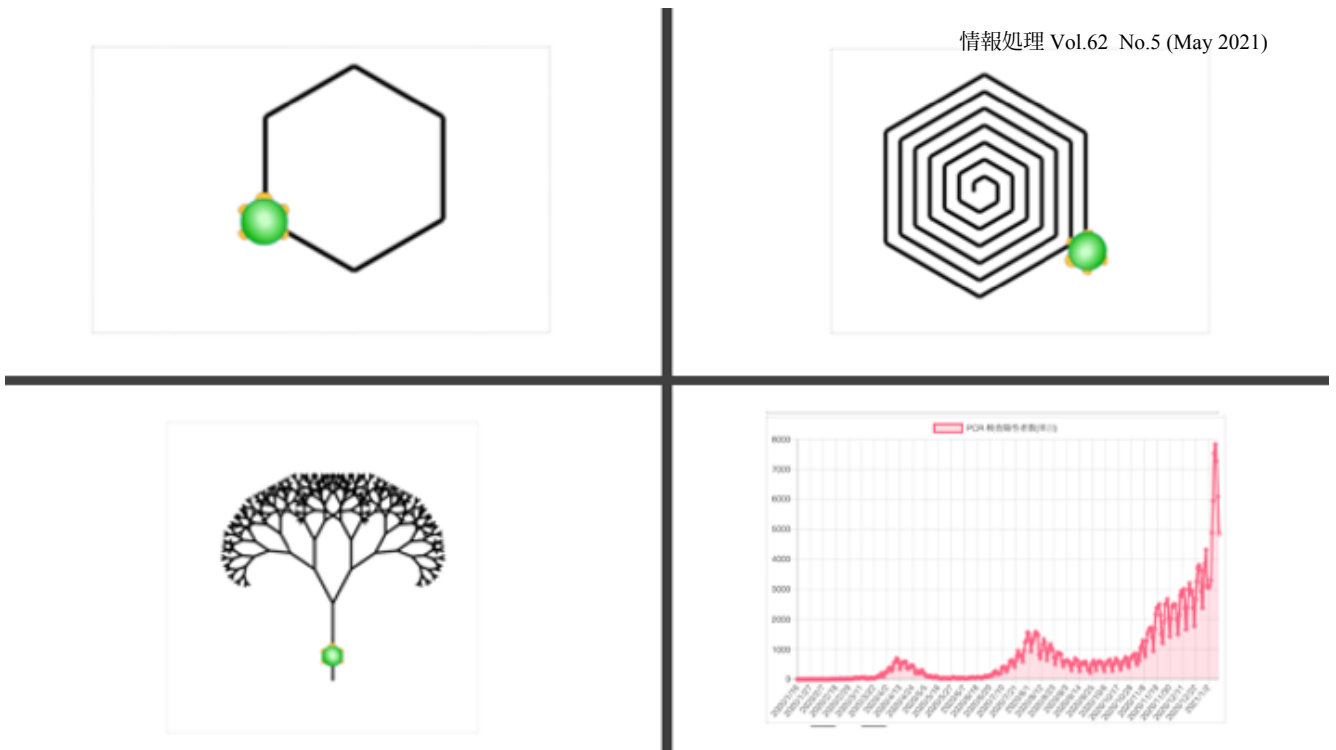


図-5 プログラムの実行結果

今後の開発計画について

これまで「誰でも簡単プログラマ」という目標に向かってコツコツと開発を行ってきました。今後も読みやすく分かりやすい言語を目指していきたいと思っています。今後はマニュアルの充実に加えて、タブレットやスマートフォンでも快適にプログラミングできるようブラウザ上の開発環境を改良したいと思っています。

最後に、本稿にて日本語プログラミング言語の魅力を少しでも知っていただけて嬉しく思います。ブラウザ上で気軽に作って楽しめますので一度試してみてください。

(2021年1月13日受付)

(2021年3月15日note公開)

※「情報処理」Vol.62 No.5 に著者のくじら飛行機氏の巻頭コラムを掲載予定です。

■クジラ飛行机

自由型プログラマ。代表作にテキスト音楽「サクラ」や日本語プログラミング言語「なでしこ」など。これまでに30冊以上の技術書籍を執筆。2001年オンラインソフト大賞入賞。2010年 IPA OSS貢献者賞受賞。