

## 「DNCL」について

クジラ飛行機

### 1. DNCLとは何か？

大学入学共通テストに「情報Ⅰ」が導入されるのと同時に話題になっているプログラミング言語があります。それが「DNCL」です。これは、正式名称を「共通テスト手順記述標準言語」と言います。つまり、試験用のプログラミング言語です。

そもそも情報教育において、採用されるプログラミング言語は多種多様なものとなっています。どのプログラミング言語を学ぶのかは、学校ごとに選択することになっており、Python, JavaScript, VBAなど、さまざまな言語が選ばれています。

こうした背景より、大学入学共通テストにおいて、特定の言語を用いるのではなく、仮想的なプログラミング言語を用いてアルゴリズムを記述するという目的で用意されたのが、DNCLなのです。

つまり、DNCLは、他のプログラミング言語とは違って、実際に何かしらのアプリを作ったり、実務で使ったりするものではなく、試験のためにだけに用意された言語とすることができます。

### 2. DNCLの実行環境について

仮想的なプログラミング言語であるDNCLですが、実際に動かすことができれば、学習の助けになることでしょう。

実際にDNCLのプログラムを入力して動かすことのできる実行環境が開発されています。

筆者が開発している日本語プログラミング言語「なでしこ」(<https://nadesi.com>) や、オンラインプログラミング学習環境「どんぐり」(<https://>

[dolittle.eplang.jp/dncl](https://dolittle.eplang.jp/dncl)), 「つちのこ」(<https://tdaimon.jp/tsuchinoko/>), 「WaPen」, 「XTetra」など、いくつかあります。

いずれの実行環境も、ブラウザ上でプログラムを動かすことができるよう工夫されており、面倒なインストール作業も必要ありません。

ブラウザで動かすことができるというのは、つまり、パソコンだけでなくスマートフォンやタブレットでも動かすことができるということです。

最近では、学校にiPadやChromebookが納入されていることも多くなっていますので、ブラウザさえあればどんな環境でもインストールすることなく動かせるというのは、大きなメリットです。

次の画面は、DNCLをブラウザ上で実行しているところです。



図1 「なでしこ」でDNCLのプログラムを実行したところ



図2 「どんぐり」でDNCLのプログラムを実行したところ

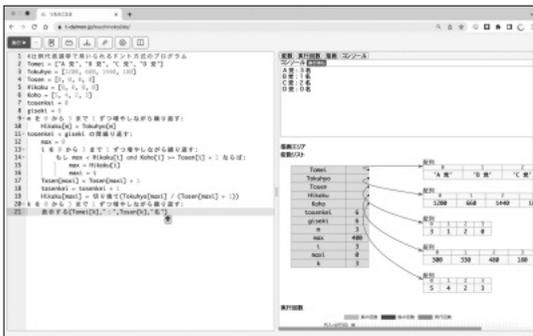


図3 「つちのこ」でDNCLのプログラムを実行したところ

### 3. DNCLの仕様書について

なお、DNCLには公式の仕様書が存在しています。上記で紹介したDNCL実行環境も、その仕様書に沿ったプログラムが動くようになっています。DNCLの仕様書はこちら (<https://www.dnc.ac.jp/albums/abm.php?d=67&f=abm00000819.pdf>) で公開されています。

ただし、大学入試センターによって2020年11月に公開された「試作問題」、2021年3月に公開された「サンプル問題」では、DNCLとは違った仮想言語が利用されていました。これは通称「DNCL2」と呼ばれています。原稿執筆時点で、DNCL2についての仕様書は公開されていません。

### 4. 次期DNCL (仮称DNCL2) について

それでも、上記の試作・サンプル問題より、このDNCL2がどんな言語であるかはわかってきています。

既存のDNCLは比較的VBAなどのBASIC言語に近いイメージでしたが、DNCL2ではPythonに近いプログラムを記述できるようになっています。

ちなみに、Pythonとは、昨今、最も注目を集めているプログラミング言語です。プログラミング言語の利用動向を調査するTIOBE Index (<https://www.tiobe.com/tiobe-index/>) でも、2021年後半から現在(2022年11月)まで、最も利用されているプログラミング言語として栄冠を手にしています。

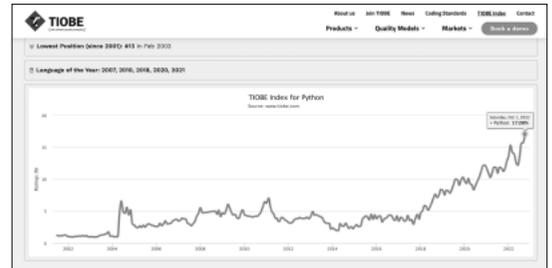


図4 2018年以後、Pythonは絶対的な人気を誇っている —TIOBE Indexより

Pythonは、2018年頃、人工知能の分野で注目を集めたのをきっかけに、一気に普及したプログラミング言語です。Webアプリケーションの開発から、定型業務の自動化まで、幅広く利用されています。文法が容易であり、プログラミング初心者でも扱いやすいというメリットがあります。また、豊富なライブラリが用意されており、高度な科学計算から、Excelの自動化やゲーム開発に至るまで幅広い用途で利用できます。

Pythonの勢いを表すエピソードの一つを紹介しましょう。筆者は本職ではプログラミング技術書籍の執筆を行っていますが、ここ数年書いた書籍の大半がPythonに関連するものでした。本屋のプログラミング言語のコーナーを覗いてみると、Python関連のものが非常に多いので納得されることでしょう。

そのため、DNCL2でPythonに似た構文が採用されたことは歓迎されるべきことであり、世界的なPython人気に追随したものとと言えます。

## 5. DNCL2へ対応した実行環境は？

なお、本稿ですでに紹介したDNCLの実行環境である「なでしこ」「どんぐり」「つちのこ」では、仕様書の公開を待つことなく、DNCL2へ対応しています。詳細な仕様が開示されていないことから、それぞれの環境で多少の差異はあるものの、基本的にはどの環境でも、DNCL2のプログラムを動かすことができます。

DNCLとDNCL2の最大の違いは、Pythonが言語のベースになったことによる文法の差異です。DNCL2では、Pythonと同じようにブロック表現にインデント（字下げ）を利用します。また、配列変数の添字が0起点になりました。

「なでしこ」では、DNCL2を使いたい場合にはプログラムの冒頭に「!DNCL2」と記述することで、DNCL2のプログラムを実行できるようになります。次の画面は、DNCLだけでなく、DNCL2にも対応していることを示すものです。

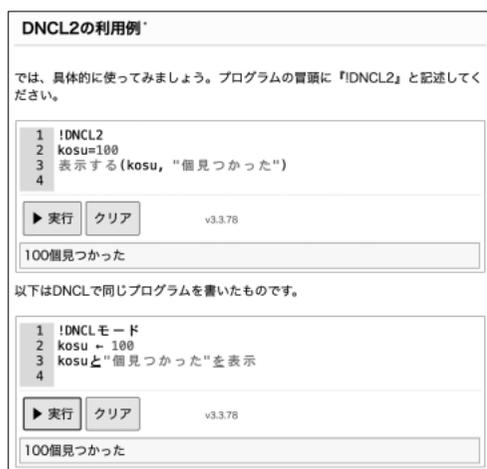


図5 DNCLに加えてDNCL2にも対応している

## 6. プログラミング言語は一つ覚えれば 二つ目の習得も容易

多くのプログラマーは、複数のプログラミング言語を使い分けています。それは、業務ごとに扱うプログラミング言語が異なるからです。例えば、スマートフォンで動くアプリを作りたいとします。この時、iPhoneのアプリを作るには、

Object-CまたはSwiftというプログラミング言語を使う必要があります。Androidのアプリを作るには、JavaまたはKotlinという言語を使う必要があります。同じスマートフォンのアプリを作るのであっても、対象とするOSによって違う言語を使う必要があるのです。

いくつものプログラミング言語を使い分けるためには、それらの言語を覚えないとはいけません。そう聞くと大変だと感じますが、実際のところ、プログラミング言語は自然言語（英語や中国語など）とは根本的に異なります。まず、プログラミング言語の概念は多くの言語で共通です。当然、言語ごとの特徴はあるものの、全く概念が異なるということはまれです。それに加えて、プログラミング言語で扱う語彙や文法は限られています。そのため、一つのプログラミング言語を学んだ後で、二つ目の言語を学ぶのは容易なのです。

この点を踏まえて、プログラミング言語の歴史を眺めるなら、いろいろなプログラミング言語の栄枯盛衰を見ることができます。栄枯盛衰が世の常であり、現在これほど流行しているPythonもいつかは勢いを失うことがあるでしょう。

それでも、プログラミング言語を学び、身につけた問題解決能力やアルゴリズムの構築能力が無駄になることは決してありません。新しいプログラミング言語が流行したとしても、その基本となる部分に大きな変化はないからです。

そのため、どんな言語を学ぶとしても、プログラミングの考え方や、問題解決の方法など組み立てたアルゴリズムがどのように動くのかに注目して学んでいくと良いでしょう。

また、プログラミングは非常に楽しいものです。プログラミングを、文法をなぞるだけの単なる勉強で終わるのではなく、クリエイティブな作業であり、アイデアを実現するための手段として、楽しみながら学んでもらえたらと思います。

## 7. 仮想言語DNCLを踏まえて プログラミングを楽しむ方法

すでに述べたとおりDNCLは大学入学共通テス

トのための言語であり、その仕様に限定された範囲では、実際に役立つプログラムを作ったり、ゲームを作ったりすることは困難です。

しかし、プログラミングを身につける最速の方法は、自分で既存のプログラムを改良して動かしてみることです。そのため、プログラミング言語の学習初期段階から、DNCLを利用するのは、あまり良い方法ではないでしょう。

さまざまな機能を持つ、PythonやJavaScriptなど、実際の言語を使って一通りプログラミングの「いろは」について学び、自由に使いこなせるようになってから、DNCLの仕様を確認する方が現実的です。と言うのも、DNCLのような仮想言語では、学んだことを活かして役立つプログラムを作ったり、ゲームを作ったりできないからです。そのため、Pythonなどの言語を学ぶ方が、試験対策にしか使えない言語を学ぶよりも、学生たちの興味を惹きつけられると考えられます。

しかし、時間的な面から、いろいろなプログラミング言語を学ぶ余力はないと考える方もいることでしょう。それでも、すでに述べたとおり、一度プログラミング言語を学んでしまえば、二つ目以降の言語を学ぶのは難しくないのです。しかも仮想言語であるDNCLの文法や機能は非常に限定的です。プログラミングと試験対策は別物と考えるのが一つの方法です。

あるいは、もう一つの方法として、最初から、いろいろな機能をもったDNCLの実行環境を選択するという手もあります。

この点で「なでしこ」は、もともと実用的な機能を多く備えたプログラミング言語です。仕事の自動化からゲーム作成まで幅広い機能を備えています。

例えば、なでしこには、ブラウザのキャンバスという機能を利用して、いろいろな図形や画像を描画できます。また、データファイルを読み込み、それを描画する機能も備えています。他にも、サウンドの再生や、テキストの読み上げなど、ブラウザ版では500を超える命令を備えています。

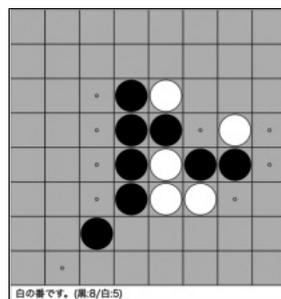


図6 なでしこで作ったリバーシ (https://n3s.nadesi.com/id.php?1776 にて公開中)

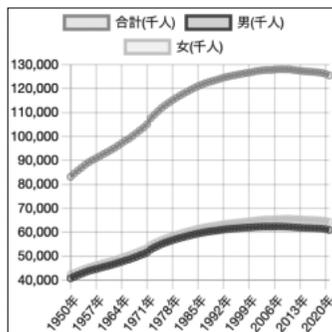


図7 データを読み込んで描画した東京の人口推移のグラフ (https://n3s.nadesi.com/id.php?1057 にて公開中)

## 8. 作ってこそ楽しいプログラミング

ところで、2010年頃の話ですが「プログラマーの仕事に応募してくる大多数の人は、繰り返しと条件分岐を使った簡単なプログラムですら作ることができない」「情報系の大学を出ていても書けない」という記事が話題になりました。これは「Fizz Buzz問題」として有名な話です。10年前の話で、現在の状況ではないのですが、「プログラミングを学ぶこと」と実際に「プログラムが書けること」の間には大きな溝があることがわかる話です。

とは言え、せっかく、プログラミングを学んだのなら、実際にプログラムを作れる方が楽しいことでしょう。しかも、最近では身近にスマートフォンがあり、作ったプログラムを気軽に動かすことができます。何でも良いので作品を作り、それを身近なスマートフォンで動かすなら、きっとプログラミングの愉しさに気づいてもらえると思います。そして、楽しみながら学ぶならきっとプログラミング学習も苦にならないのではと思います。