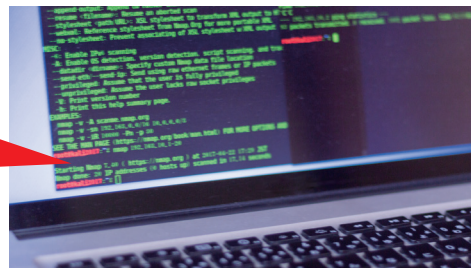


プログラミング言語発展の歴史を知らう ～今と昔のプログラミング言語～

プログラミング言語って何だろう？



現代社会において、私達の身の回りに存在する様々な種類のコンピュータは、すべて「プログラム」によって動いている。これらの「プログラム」は人間が思い通りにコンピュータを動かすための、言わば「指示書」のようなもので、「プログラム」を作成することを「プログラミング」と言い、プログラミングをする人を「プログラマー」と呼んでいる。
コンピュータを思い通りに動かすためには、本来、コンピュータが理解できる言語である「機械語」を使って、その指示を伝える必要があるのだが、機械語は「0」と「1」だけを使った数字の羅列なので、私達人間が理解するのは非常に困難である。そこで機械語を私達が理解しやすい「文字列」に変換した言語が必要となった。これが「プログラミング言語」である。



プログラミング言語には様々な種類がある【プログラミング言語発展の歴史】

一説によると300種類近くあるとも言われる「プログラミング言語」だが、現在プログラミングの現場で使われている主な言語はせいぜい20種類くらいである。どんな種類のプログラミング言語があり、どのように発展してきたのか下の表を見て確認してみよう。

第1世代 (低級言語)

第2世代 (高級言語)

第3世代 (高級言語)

機械語

(マシン語)

2進法 (0と1のみ) で書かれたプログラミング言語。数字の羅列なので、プログラムを書いた人以外は理解しずらく「低級言語」と言われる。そのため、英単語や記号・数字などを組み合わせて、コンピュータに指示を出すための言語が開発される事となった。これを「高級言語」という。

アセンブリ

「アセンブリ」はプログラミング言語の中で最も機械語に近い言語である。そのため、使いこなすのがとても難しい。主に、他のプログラミング言語ではプログラムを書くことができない場合などに「アセンブリ」が使われる。

1954年～ FORTRAN

(フォートラン)

開発者: IBM ジョンバックス
史上初の高級言語で、その後に現れた多くのプログラミング言語が FORTRAN の影響を受けている。名前の FORTRAN は「formula translation (数式翻訳)」に由来。科学技術計算に向けた言語で、非常に多くの数学関数やサブルーチンを持つ。

1959年～ COBOL

(コボル)

開発者: GODASYL (データ言語協議会)
名前の COBOL は「Common Business Oriented Language (共通事務処理用言語)」に由来しており、事務処理用のソフトに多く使われている。非常に多くのプログラムが COBOL で書かれ、それらの多くが現在も稼働中であるため、COBOL プログラムは今でも多く存在し、需要もかなりある。現在でも重要な言語である。

1960年～ LISP

(リスプ)

開発者: MIT (マサチューセッツ工科大学) ジョン・マッカーシー
名前の LISP は「list processing (リスト処理)」に由来し、関数型言語に分類される。

1964年～ BASIC

(ベーシック)

開発者: 米国ダートマス大学の数学者 ジョーン・ケニーとトーマス・カッツ
名前の BASIC は「Beginners All-purpose Symbolic Instruction Code (初心者向け汎用記号命令コード)」に由来。一時はアマチュアプログラマーに定番のプログラミング言語だった。日本では今でも、高校の情報授業などで使われることも多い。

1968年～ Pascal

(パスカル)

「構造化プログラミング」をサポートする初期の代表的なプログラミング言語。もともと教育用に開発されたため、使われる場所としては大学の授業などが多い。歴史的に重要な言語で、根強いファンがいる。

1972年～ C

(シー)

開発者: ベル研究所デニス・リッチーなど
OSなどの基礎部分のソフトウェア作りや、ゲーム開発、家電向けなどで活躍する言語。「C」は応用範囲が広く、昔はこの言語から勉強するのが一般的だった。しかしながら初心者には高難易度の言語でもある。「C」という名前の由来は、先に「B」という言語があり「B」の次の言語という意味で名前が付けられたと言われる。

1982年～ C++

(シー・プラス・プラス)

開発者: ベル研究所のコンピュータ科学者 ビャーネ・ストラウスツップ
「C」の後継として、「C++オブジェクト指向」の言語として誕生。「C」にプラスしたものという意味で「C++」と名付けられた。「C」と同じく、活用範囲が広いのが特徴で、特に実行速度の速さが重要とされるゲーム業界では根強い人気があり、様々な家庭用ゲーム開発に使われている。

1983年～ Objective-C

(オブジェクティブ・シー)

開発者: ブラッド・コックス
「C」をオブジェクト指向的に拡張した言語。「small talk」というオブジェクト指向言語を引き継いでおり、同じく「C」をオブジェクト指向的に拡張した「C++」とは異なる言語である。スティーブ・ジョブズに認められ、アップルの macOS X や iOS アプリの公式言語に採用されたことで爆発的に広まった。

1987年～ Perl

(パール)

開発者: ラリーウォール
「軽量化言語」や「スクリプト言語」の父的存在で「PHP」や「Ruby」も影響を受けている。文字列処理を得意としており、Web サーバー側のプログラムで活躍。いろいろな書き方ができるほか、参考書も多いため、初心者にもお勧めな言語。

1991年～ Visual Basic

(ビジュアルベーシック)

開発者: マイクロソフト社
入門向けプログラミング言語「Basic」の進化形。主に Windows のアプリケーションを開発するために作られた言語で、一体化された統合開発環境は使いやすく、直感的にプログラミングが可能なので初心者にも最適。

1991年～ Python

(パイソン)

開発者: グイド・ヴァン・ロッサム
Python は「オブジェクト指向スクリプト言語」と言われる。言語仕様が少なく文法がシンプルなので、初心者にも優しい。ビッグデータを扱いたい人や人口知能の分野で注目されるディープラーニングを研究したい人など、データサイエンティストには必須の言語とも言える。

1992年～ Java

(ジャバ)

開発者: サン・マイクロシステムズ社のジェームズ・ゴスリンとビル・ジェイなど
オブジェクト指向の代表的な言語で最もプログラマー人口を誇る。最初に学ばれた言語 NO.1。公共システムや大企業の基幹システム、組み込み機器、Web アプリ・Android アプリなどあらゆる場面で使われている。

1993年～ Ruby

(ルビー)

開発者: まつもととゆきひろ
日本人が開発したオブジェクト指向スクリプト言語のため日本語で書かれた文書が多いという利点がある。しかし、日本だけではなく世界的にも人気を博している。

1995年～ PHP

(ピー・エイチ・ピー)

開発者: ラスマス・ロードフ
PHP は WEB ページ作成のために作られた言語で、初心者でも比較的簡単に WEB ページを作れるのが特徴。その名前は「Personal Home Page (個人的なホームページ)」に由来するとされる。

1996年～ JavaScript

(ジャバスクリプト)

開発者: ネットスケープ コミュニケーションズ社
Web ブラウザ向けにスタートした言語だが、現在はパソコンやスマホ・サーバー・組み込みなどあらゆる分野に浸透中で、いまや Web サーバー側アプリでも活躍。「JavaScript」は、「Java」に似たスクリプト言語だが、「Java」とは異なる言語。

2002年～ C#

(シー・シャープ)

開発者: マイクロソフト社
マイクロソフト社の .NET Frameworks (ドットネット・フレームワーク) 環境における主力言語。「C」よりもオブジェクト指向でありつつ「Java」の影響も強く受けている。また「C++」ほど複雑ではなく、万人が使いやすいように配慮されている。C# の名前は「C++」が由来で、+ が4つ合わさり、# になった。

2003年～ Scala

(スカラ)

名前の Scala は「scalable (スケーラブル)」= 小さいプログラムも大きいプログラムも記述できるという意味に由来し、名前の由来通り「いいとこ取り」の言語で「オブジェクト指向言語」と「関数型言語」を統合する言語とも言われる。

2014年～ Swift

(スウィフト)

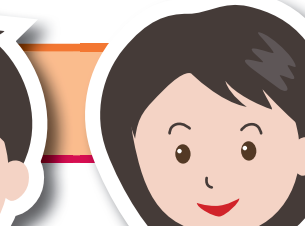
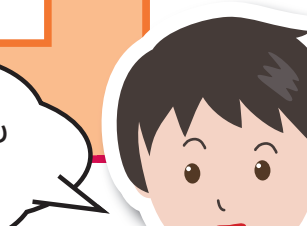
開発者: アップル社
「Objective-C」の後継で、2014 年よりアップルの macOS X および iOS アプリの公式開発言語に採用された新しい言語。いろいろな言語の良い部分を集めてできており、短くスッキリときれいなプログラムが書けるのが特徴。「関数型プログラミング言語」と言われる。

やっぱりプログラミングって難しいのかな？



そんな悩みを「ビジュアルプログラミング言語」で解決!!

これなら、僕たちもできそうだね!!



ビジュアルプログラミング言語とは

ビジュアルプログラミング言語とは、学習用に開発された「子ども向けプログラミング言語」の事。プログラマーなどが使っているプログラミング言語は、テキスト言語といえ、文字で「ソースコード」を入力するので、タイピングのスキルが必要。またソースコードは英単語からできているので、それが分からないとプログラムを理解することができない。そこで、タイピングスキルが無く、英単語を知らない子供でもマウスやタッチパネルの操作でプログラミングを楽しく学べるように、様々なビジュアルプログラミング言語が開発されたのだ。

ビジュアルプログラミング言語の例!!

Scratch (スクラッチ)

アメリカの MIT (マサチューセッツ工科大学) メディアラボが、子供たちの学習用として開発したビジュアルプログラミング言語で、学習用のプログラミング言語としてはユーザーが最も多い定番ツールである。HP で無償公開されており、パソコンの Web ブラウザ上で動作するため、だれでも気軽に使うことができるのが特徴。「動かす」「回転する」「音を鳴らす」などの命令がブロックになっており、これを自由に組み合わせるだけで、簡単にプログラムを作成することができ。また、条件分岐や数値などの高度な命令も用意されているため、簡単なゲームなどを作ることにも可能。

Scratch Jr (スクラッチ・ジュニア)

タブレットで使える。幼児向けの「Scratch (スクラッチ)」。文字の読み書きができなくても指先を使ったドラッグ操作で「動く絵本」を作ることができる。直感的な操作を可能にするデザインのプロットを採用しており、「Scratch (スクラッチ)」の入門編として使える。

プログラミン

文部科学省が2010年からHP上で無償公開しているビジュアルプログラミング言語。「Scratch (スクラッチ)」を参考に開発しているため、基本機能は非常に似ている。

Viscuit (ビスケット)

小学校へ就学する前の小さな子どもでも「遊び感覚」で理解できるビジュアルプログラミング言語。命令が「めがね」や「指」などのパーツで表現されており、文字や数値を使わないのが特徴。それぞれの命令はとてもシンプルだが、それらを組み合わせることで、ある程度複雑な動きも表現できる。

PROTCH (プロッチ)

山崎教育システム (株) が「Scratch (スクラッチ)」をベースに作成。スクラッチ同様、ブロック状の命令アイコンを組み合わせてプログラミングをする仕様。専用のロボットを追加し、計測・制御プログラミングを行えるのが特徴。

PROTCH [プロッチ] で楽しくプログラミング!!

「PROTCH (プロッチ)」とは、プログラミング学習教材「プロロボ」のプログラム作成ソフトを、フローチャート式から「スクラッチ」ベースのブロック式に変更したものです。「プロロボ」の「スクラッチ」バージョンという開発コンセプトから「PROTCH (プロッチ)」と命名されました。ブロック状の命令アイコンをつないでいくだけで、誰でも簡単に楽しくプログラミングを行います。

プログラミング学習教材 「プロロボ」

2018年秋発売予定 開発のため仕様変更の可能性があります。

プロッチ・エディター画面

※プロッチ・エディターは、米国マサチューセッツ工科大学メディアラボが開発したオープンソースソフトウェア「SCRATCH・スクラッチ」を、カスタム化したブロックプログラミング学習ソフトです。

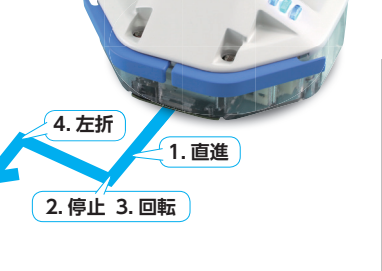
PROTCH [プロッチ] を使うと、こんなことができる!!

プログラム作成ソフトが「スクラッチ」ベースになったことで、PROTCH [プロッチ] では、計測・制御プログラミング以外にも、様々な事ができるようになりました。

1 計測・制御のプログラミング

オリジナル機能「プロッチモード」

ロボットの制御に必要な命令ブロックのみを表示するモードで、簡単に計測・制御プログラミングが可能です。

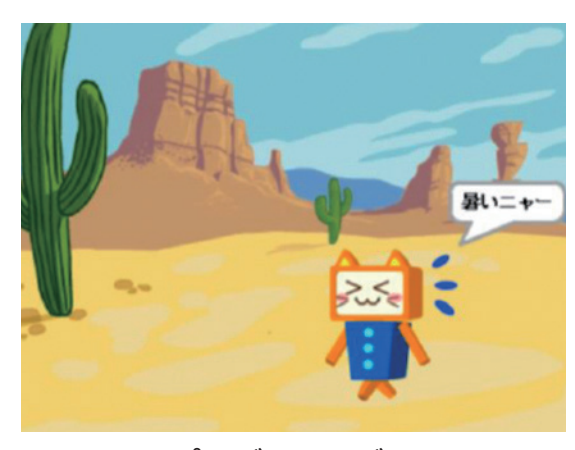


「計測・制御のプログラミング」を行い、専用のロボットを操作できます。



プロッチを2台接続することで、チャットプログラムの体験ができます。

2 ネットワークを利用した双方向性のあるコンテンツのプログラミング

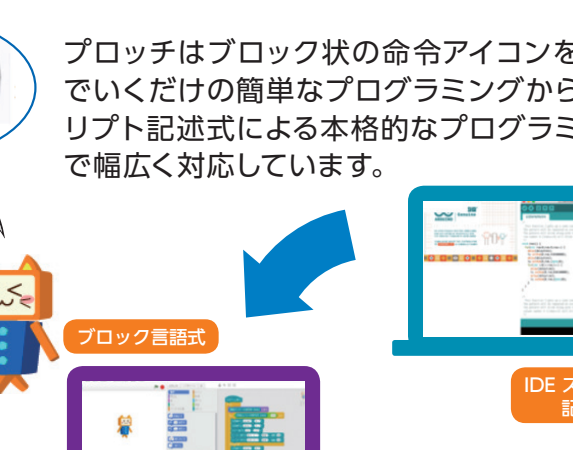


コンテンツプログラミングで、アニメーションの作成などが行えます。



双方向性のあるゲームプログラミングが行えます。 ※ロボットの「接触センサ」や「光センサ」をスイッチとして使用し、画面内のキャラクターを動かすことが可能です。

3 コンテンツのプログラミング



プロッチはブロック状の命令アイコンをつないでいくだけの簡単なプログラミングから、スクリプト記述式による本格的なプログラミングまで幅広く対応しています。

4 スクリプト記述式のプログラミング



プロッチで楽しくプログラミングにチャレンジしよう!!