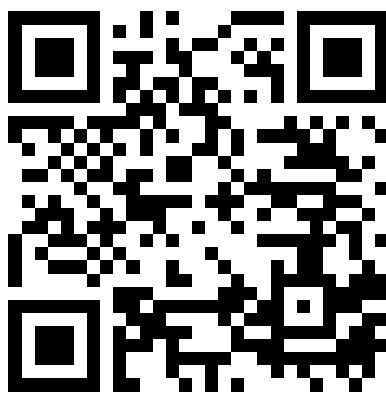


「群馬デジタルイノベーションチャレンジ」 地域ICTクラブ

第1回テキスト



2022年5月21日・22日開催

※本テキストは「ぐんまプログラミング教育推進協議会・ぐんまジュニアICTクラブ」の「プログラミングスクール入門編テキスト」に準拠するものです。

はじめに「プログラミングはたのしい」

みなさんは、ゲームをしたことが ありますか？

ゲームは つくるのも たのしいです。

きょうは、いっしょにゲームをつくります。

ところで「コンピュータ」ってことば、きいたこと ありますか？

みなさんの まえ にあるのは「パソコン」という コンピュータです。

ゲームき や、スマートフォンも コンピュータです。

コンピュータに、なくてはならないものが あります。

それが「プログラム」です。「ゲーム」も もちろんプログラムです。

そのほかにも、たくさんの プログラムが あります。

ゲームいがいの プログラムも つくってゆきましょう。

そして、プログラムが どのようにうごくか かんがえてみましょう。

プログラムとは？

プログラムは、「事前に」という意味の「pro」と、「書く」という意味の「gram」を組み合わせた言葉です。「前もって書く」という意味になります。

コンピュータ以外にもプログラムと呼ばれるものがあります。演奏会のプログラム、運動会のプログラム。これらも実際に演奏会や運動会が行われる「前」に、演目や種目を「書き」てある点で、「プログラム」といえるわけです。

プログラム・ソフトウェア・アプリケーション

「プログラム」と似た言葉で「ソフトウェア」や「アプリ(アプリケーション、アプリケーションソフト)」があります。同じように使われることもありますが、大まかには、複数の「プログラム」を組み合わせた比較的大規模なものを「ソフトウェア」と呼び、「ソフトウェア」の中でも利用者が直接触れるものを「アプリ」と呼ぶ、とお考え下さい。たとえば「ワープロソフトは、仮名漢字変換プログラムと、画面表示プログラム、印刷プログラムなどから構成されている」とか「iPhone のシステムである iOS は apple が開発しているが、アプリは誰でも自由に開発することができる」といった使い方をします。

コンピュータ、ソフトがなければただの箱

コンピュータには「ハードウェア」と「ソフトウェア」の双方が必要です。ゲーム機本体だけではゲームはできず、他にゲームソフトを追加することが必要です。「ハードウェア」とは目に見えて手で触れることのできる機器のことを指します。(もともと「ハードウェア」とは「金物」をさしていました)。それに対して「ソフトウェア」は、ゲームカセットや CD-ROM などの機器に格納されているプログラム本体、また音声や画像などのデータを指します。「ソフトウェア」はデジタルな情報なので、コピーしたり、インターネットを通じて配信したりすることができるわけです。

プログラムの3つのきほん

プログラムは、コンピュータへの めいれいです。

きほんが 3 つあります。

「うえからした」

「くりかえす」

「わかるる」

きょうは、この 3 つをまなびます。

どんな プログラムも、この 3 つの くみあわせで つくることができます。

3つの制御構造と構造化言語

現代のプログラミング言語の多くは、以下の3つを表現できる仕組みを持っています。(これを「制御構造」と呼びます)

「順次」(うえからした)

「反復」(くりかえす)

「分岐」(わかれる)

このような言語を「構造化言語」と呼びます。これらを組み合わせることで、見通しが良いプログラムが作れるとされています。

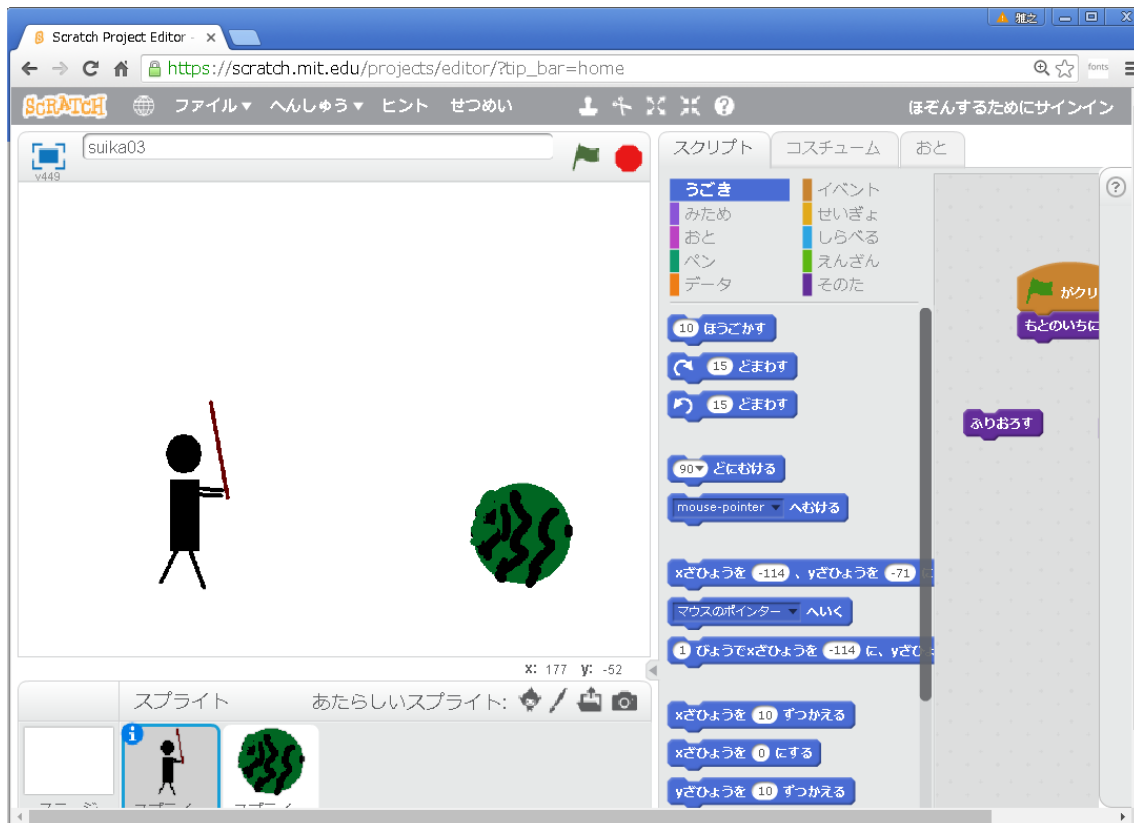
かつてのBASICなどでは、「goto文」というものがありました。プログラムの中、どこにでもジャンプできるというものです。これさえあれば、実は「反復」も「分岐」も表現することができます。「反復」はふりだしに戻るジャンプですし、「分岐」は場合ごとに飛び先を指定するジャンプだからです。けれども、「goto文」を多用したプログラムは、どこからどこまでが反復の範囲になるのか、見通しが悪いため、「スパゲッティ(のようにならまつた)プログラム」と揶揄されていました。書いた本人ですら良く分からないけど、動いている、といったプログラムになりがちでした。

スクラッチでも、この3つの構造が一目で分かるブロックで、処理の範囲やまとまりが分かるように設計されています。現代的なプログラムの入門を意識して作られています。

現実世界の制御構造？

前のページで例を上げた運動会のプログラムでは、「雨が降ったら、お昼ごはん用に体育館を開放する」とか、「雨天が続いていたら2回まで順延する」といったことが計画されています。これは「分岐」や「反復」そのものです。物事の流れや進行を考える上でも、プログラムの考え方は役に立ちます。色々なマニュアルなどに出てくる「流れ図」も、もとはプログラムの流れを表現したものです。

「すいかわり」で「うえからした」とスクラッチのきそ



このがめんが スクラッチです。ここで プログラムをつくります。

スクラッチと仲間たち

スクラッチは、MIT(マサチューセッツ工科大学)メディア・ラボの Lifelong Kindergarten グループによって開発された教育用プログラミング環境です。玩具の「ブロック」になぞらえてプログラムを作ることができます。

スクラッチのようにブロックに例えるスタイルは、まとまりが一目で分かるため、非常に有効で、他にも多くのプログラミング環境が作られています。スクラッチの次に学ぶのに好適なものや、スマートフォンのアプリを開発できるもの、また、教育用ロボットの制御を行えるものもまであります。オープンソースの開発環境で無償で利用できるものも多くあります。以下に簡単に紹介します。

alice 3D(<http://www.alice.org>)

カーネギー・メロン大学で開発された環境で、3Dのキャラクターを動かすことができます。例えば、「たま」という名前の猫を作り、「たまのしっぽを 20 度、後ろから前に動かす」といった指示を記述することができます。スクラッチでは扱っていない「オブジェクト志向」という考え方も取り入れており、Java と互換性のある言語の表現にすることもできます。スクラッチの次のステップとして優れた環境と言えましょう。

app inventor (<http://appinventor.mit.edu/>)

google が開発し、MIT が運営している環境です。アンドロイドのスマートフォンやタブレットで動作するアプリを作ることができます。スマートフォンのカメラ、GPS、電話などの機能をフルに活用できるため、非常に幅の広い本格的なアプリが、ブロック形式で作れるのが特徴です。

LEGO Mindstorm (<http://www.lego.com/>)

ブロック玩具で有名な LEGO から、プログラム可能なロボットを作ることのできるキットが販売されています。教育機関などで導入されているところも多くあります。実物のモーターやセンサーを使ったプログラムを作ることができます。スクラッチや app inventor から制御できる仕組みがあります。

プログラミン(<http://www.mext.go.jp/programin/>)

文部科学省が公開している、初等教育用のプログラミング環境で、スクラッチ同様ブロック形式を採用しています。

動きのあるアニメーションが簡単に作れるように機能や用語を選んでいます。「順次」「反復」「分岐」といった構造も意識せずに学べるようになっています。解説や用語の選択も小学生向けに丁寧に作られているので、最初の入門用に良い環境です。

はじめの きそは「うえからした」です。

コンピュータは、ひとつひとつ じゅんばんに しごとをします。

その じゅんばんを ひと が きめなければなりません。

みなさんは「すいかわり」をしたことありますか？

じゅんばんを かんがえて みましよう。

スクラッチで、ブロックのように めいれいを ならべて つくります。

まうすを つかって いっしょに やってみます。

「はた」のブロックは とくべつな ブロックです。「かいし」するときの う
ごきを つなげます。

「すいかわり」は、順次構造の流れを理解させるために作ったものです。

スクラッチには「ブロックを作る」機能があり、一般的なプログラミング言語での「関数やプロシージャの作成」を行なうことができます。その機能を使って「まえにすすむ」「ふりおろす」のブロックを作っています。

また、スクラッチでは、画像を「スプライト」と呼びます。単なる画像ではなく、複数の画像やプログラム(「スクリプト」と呼んでいます)が含まれています。すいかの「スプライト」を選択し、「コスチューム」や「スクリプト」の内容を覗いてみてください。割れるための仕掛けが確認できるはずです。これを少し改変するだけで、逃げるかわりに巨大化したり、すいかが桃に変わったり、桃が割れて桃太郎が出てきたり、と様々なプログラムを作ることができます。



教材にはこちら↑からアクセス

「ダンス」で、「くりかえし」、じぶんでつくるスクラッチ

はじめに、みぎがわを みてみましょう。

これから つくる プログラムの おてほんがあります。

「くりかえし」のブロックは、ほかの ブロックを はさめるように なっています。

このブロックを いっしょに つくって みましょう。

たくさんつくるのは、めんどうです。

スタンプの かたちの ボタンで おなじものをつくることができます。

おと や いろ も かえることができます、

キーボードで うごかす やりかたも ためして みましょう。

「ダンス」は、スクラッチ入門用のチュートリアル付属のプログラムを改変したものです。

プログラムが勝手に自動的に動いている部分と、人間の操作に反応して動く部分を組み合わせたもので、どちらもプログラムにとって大事であることが理解できるように作られたものです。

スクラッチのチュートリアルはとても優れているのですが、翻訳ができていない部分も若干あり、また画像でのブロックが英語表記になっているため、そのままでは小学生にお勧めしにくいところがあります。本スクール終了後に、ぜひ、保護者の方も一緒に、一通り進めてみられて下さい。スクールで触れることのできなかつた、色々な技を身に付けることができます。



教材にはこちら↑からアクセス

「つかまえて!」で、「くりかえし」「わかれる」

「ねこふんじやった」を えんそうするプログラムが あります。

「ねこふんじやった」を 2 かい くりかえしていることが わかりますか？

このように でんしがつきを えんそうさせるのも プログラムです。

「かえるのうた」を いっしょに つくってみましょう。

つぎに、おてほんの プログラムを うごかしてみましょう。

マウスをつかって ねこを さわって みましょう。

「つかまえて」は、大学での教材として監修者と学生が作ったものです。

マウスの操作による「分岐」、「ねこふんじゃった」の演奏で、回数指定の「反復」と無限「反復」、などを盛り込んだ例題です。

角度や進む距離を変更することで、様々な動きを見ることができるようになっています。

スクラッチでは、猫の動きは、猫そのもの視点から記述できるようになっています。「座標どどこから、座標どどこへ移動せよ」という形式ではなく、「前に進む」「90度右を向く」といった形式で記述します。今、どの方向を向いているか、という情報を保持し、それを元に、進む方向と距離を指定する形式です。

これは「タートルグラフィックス」と呼ばれるものです。画面を動いている「亀に指示する」。「亀の上に自分が乗っている」ように感じられるところから名付けられたもので、古くから教育用の言語で使われている手法です。「はねかえる」などの命令を使うと、昔懐かしいピンポンゲームが驚くほど簡単に作ることができます。スクラッチの公式サイトで公開している事例を試してみてください。

(<https://scratch.mit.edu/projects/10128515/>)



教材にはこちら↑からアクセス

ここで「わかる」が どうしようします。

「マウス」が ねこ に ふれているとき

「マウス」が ねこ に ふれていないとき

の ふたつで、うごきが わかれます。

それぞれ、どんな めいれいが かいていると おもいますか？

スクラッチのブロックでも、一般のプログラミング言語でも、「分岐」の記述は少々複雑でややこしいです。〈1〉どちらに分岐するか判断する条件(「もし○○ならば」の「○○」にあたる部分)、〈2〉条件が成立した場合の処理、〈3〉条件が成立しなかった場合の処理 の3つを書かなければならないからです。

また、雑誌などでよくある運勢判断や性格判断のチャートでは、左右に分岐するのが一般的ですが、スクラッチでも一般のプログラム言語でも、上と下に分岐のブロックをまとめます。処理のまとまりが多くなっても対応できるようにするためですが、直感的に分かりにくいようです。

スクラッチでは、アルファベットのEのような形のブロックで、上と下に分けています。確認して慣れるようにして下さい。

「じどううんてん」で、「いろ」をしらべて「わかる」

こんどは、やくにたつ プログラムをつくってみます。

じどうで くるまの うんてんをする プログラムです。

どんな めいれいを すればよいとおもいますか？

プログラミングでは、コンピュータが わかるように、めいれいする ひつよ
うが あります

いっしょに かんがえて みましょう



教材にはこちら↑からアクセス

「じどうんてん」はスクラッチの前身とも言える「squeak」のサンプルプログラムを改変したものです。

色を検知して道路から外れたことを確認し、逆方向にハンドルを切る、というシンプルなプログラムですが、道路の形や、進む速度、ハンドルを切る形によっては、上手く進まないことを体験できるように作ってあります。「バグ(欠陥)」というよりはむしろ、「頭で考えれば上手く動くように思えるけど、実際には上手く動かないことがある」ことは、プログラム開発でよくあります。そのようなプロセスを楽しんでいただければと思います。

「よむ」と「つくる」

プログラミングが じょうたつするための こつが ふたつあります。

ほかのひとが つくった プログラムを よむ こと

じぶんで、いろいろと つくってみること

かんたんなプログラムをつかって、ほぞんする ほうほうを まなびます。

また、ほかのひとが つくった プログラムを よんだり、かいぞうしたり する ほうほうを まなびます。

「読んで」「作る」ために

スクラッチが優れているのは、誰でも公式サイトでプログラムを公開できること、公開されたプログラムは誰でも中身を見て、改変を試すことができることの二つにあります。「読む」と「作る」ことを一緒に試すことができるようになっています。

まず、「例を見る」の game をプレイして中身を読んでみましょう。

(https://scratch.mit.edu/starter_projects/)

特に、pong starter、 maze starter、hide and seek は、是非見ていただきたい非常に優れた例題です。

保存と開くときに注意！

保存する際に以下に留意して下さい。ワープロや表計算では「開く」「保存する」にあたるメニューが「てもとのコンピュータからアップロードする」「てもとのコンピュータにダウンロードする」になっています。これは、スクラッチが MIT のコンピュータで動いているためです。我々は web 経由でアクセスしているため、作ったプログラムを、自分のコンピュータから MIT のコンピュータに移したり(アップロード)、MIT のコンピュータから自分のコンピュータに移したり(ダウンロード)することになります。

同じ理由で、プログラムのファイル(拡張子が sb2 になっています)をダブルクリックしただけでは開きません。

あらかじめ scrach のサイト(<http://scratch.mit.edu>)に接続して、編集画面に移り(「つくる」をクリック)、その画面で「てもとのコンピュータからアップロード」を選ぶ必要があります。

おわりに

プログラムが さいしょに おもったよりも むずかしいものでなくて、たのしいものであると坎じていただけたらうれしいです

みなさんがこれから いろいろなプログラムを じょうずに つかったり、すばらしい プログラムをつくるようになったらいいなあとおもっています。

これからも たのしまれて、コンピュータやネットワークと つきあっていてください。

短いスクールでしたが、お楽しみいただけましたでしょうか。

プログラミングで、世界に存在しているものを作ることができます。また、世界に存在していないものも作ることができます。コンピュータがタイプライターになったり、電話機になったり、車の運転手になったり、様々な形を取っていくのは、プログラムの力によるものです。

世界にあるものを作るには、対象のものを深く理解しなければなりません。世界に存在していないものを作るには、イメージや構想をプログラムの論理や制御文で書けるまで練り上げてゆく必要があります。いずれも、コンピュータやネットワークの理解だけでは不十分です。作り手の総合力が試されます。ボールゲームを作るには、物理の法則が役に立ちます。家計簿のプログラムには簿記の知識が必要になるかもしれません。RPGを作るには、世界中の神話や物語がヒントになることでしょう。

ぜひ親子で一緒に、プログラミングを通じて様々なことを考え、話し、楽しまれて下さい。

ローマ字表

くんれいしき
訓令式・()内はヘボン式

	だん 段		あ	い	う	え	お			
ぎょう 行	大文字		A	I	U	E	O			
	小文字		a	i	u	e	o			
あ			あ a	い i	う u	え e	お o			
か	K k	か ka	き ki	く ku	け ke	こ ko	きゃ kya	きゅ kyu	きょ kyo	
さ	S s	さ sa	し si (shi)	す su	せ se	そ so	しゃ sya (sha)	しゅ syu (shu)	しょ syo (sho)	
た	T t	た ta	ち ti (chi)	つ tu (tsu)	て te	と to	ちゃ tya (cha)	ちゅ tyu (chu)	ちょ tyo (cho)	
な	N n	な na	に ni	ぬ nu	ね ne	の no	にゃ nya	にゅ nyu	にょ nyo	
は	H h	は ha	ひ hi	ふ hu (fu)	へ he	ほ ho	ひゃ hya	ひゅ hyu	ひょ hyo	
ま	M m	ま ma	み mi	む mu	め me	も mo	みゃ mya	みゅ myu	みょ myo	
や	Y y	や ya		ゆ yu		よ yo				
ら	R r	ら ra	り ri	る ru	れ re	ろ ro	りゃ rya	りゅ ryu	りょ ryo	
わ	W w	わ wa								
ん		ん n								
が	G g	が ga	ぎ gi	ぐ gu	げ ge	ご go	ぎゃ gya	ぎゅ gyu	ぎょ gyo	
ざ	Z z	ざ za	じ zi (ji)	ず zu	ぜ ze	ぞ zo	じゃ zya (jo)	じゅ zyu (ju)	じょ zyo (jo)	
だ	D d	だ da	ぢ zi (ji)	づ zu	で de	ど do	ぢゃ zya (jo)	ぢゅ zyu (ju)	ぢょ zyo (jo)	
ば	B b	ば ba	び bi	ぶ bu	べ be	ぼ bo	びゃ bya	びゅ byu	びょ byo	
ぱ	P p	ぱ pa	ぴ pi	ぷ pu	ぺ pe	ぽ po	ぴゃ pya	ぴゅ pyu	ぴょ pyo	

スクラッチで学ぶプログラミング思考

ぐんまジュニア I C T クラブ プログラミングスクール入門編テキスト

第二版

監修 築雅之（高崎商科大学）

執筆

築雅之（1章、2章、9章）

木暮紀樹（3章、4章、5章、およびプログラム）

夏目智明（6章、7章、8章、およびプログラム）

発行 平成 30 年 8 月 4 日

主催 ぐんまプログラミング教育推進協議会

協力 高崎商科大学

なお、本テキストは「親子でスクラッチ体験 キッズ・プログラミンスクール高崎校 入門編テキスト第二版」(平成 28 年)を一部改定したものです