

2020 年 6 月 18 日
株式会社インプレスR&D
<https://nextpublishing.jp/>

短くシンプルな例題で、しっかり学べるプログラミングの基本

「Scratch 本格入門

命令機能詳細・プログラミング作法・デバッグがわかる」発行

Scratch3.0 対応

インプレスグループで電子出版事業を手がける株式会社インプレス R&D は、『Scratch 本格入門 命令機能詳細・プログラミング作法・デバッグがわかる』（著者：多田 憲孝、内藤 富美子）を発行いたします。

『Scratch 本格入門 命令機能詳細・プログラミング作法・デバッグがわかる』
<https://nextpublishing.jp/isbn/9784844378983>



著者：多田 憲孝、内藤 富美子
小売希望価格：電子書籍版 1600 円（税別）／印刷書籍版 1800 円（税別）
電子書籍版フォーマット：EPUB3／Kindle Format8
印刷書籍版仕様：B5 判／モノクロ／本文 224 ページ
ISBN：978-4-8443-7898-3
発行：インプレス R&D

<<発行主旨・内容紹介>>

本書は、初心者のための Scratch プログラミング入門書です。学習の要点を押さえた短くシンプルな例題を用いており、効率的に短期間で学べます。また、標準的命令を網羅しているため、読者がオリジナルのゲーム等を作りたいとき、それを実現するための情報を提供し、Scratch 命令マニュアルとしても活用できるものです。

本書の特徴は以下のとおりです。

- (1) 学習の要点を押さえた短くシンプルな例題を用い解説してあります。
- (2) 短くシンプルな例題ですが、ゲームで活用できる実践的な Tips(小技)を含んでおり、ゲーム作成の楽しさが感じられるように考慮してあります。
- (3) Scratch で用意されている標準的の命令(ブロック)をすべて解説してあります。
- (4) プログラミング技術と作法について解説してあります。例:わかりやすいプログラムを作るコツ、さまざまな処理の流れをコントロールする方法、データの正当性をチェックする方法など。
- (5) プログラムの誤りをバグといい、それを修正する作業をデバッグといいいます。
バグの事例とデバッグの基本的な方法について解説してあります。
(本書は、次世代出版メソッド「NextPublishing」を使用し、出版されています。)

第1章 初めてのプログラミング

1.2 スプライト (キャラクター) と背景を用意しよう！

1.2.1 プロジェクト

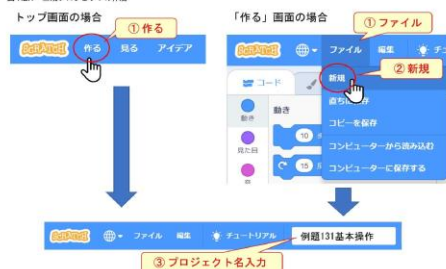
(1) プロジェクト

プロジェクトとは、ある処理 (仕事) を行うためのプログラムやデータのまとまりのことです。Scratch ではプロジェクトのことを **作品**ともいいます。

(2) 新規プロジェクトの作成

次の操作でプロジェクトを作成します。最初のプロジェクト名は「Untitled」と表示されていますが、ここではプロジェクト名を「例題131 基本操作」に変更します。
トップ画面上部の「作る」→「プロジェクト名」＝例題131 基本操作
(または「作る」画面上部の「ファイル」→「新規」→「プロジェクト名」＝例題131 基本操作)
なお、本書では上記のとおりツールバーや操作画面の選択項目は「 」で囲み表記し、「→」にて操作手順を示します。

図1.2.1 新規プロジェクトの作成



1.2.2 コードエディター

プロジェクトが起動すると、次のようなコードエディターが表示されます。

図1.2.2 コードエディター画面



コードエディターは次のパーツにより構成されています。

- (a) **ツールバー**: データの読み書き、編集などを処理できるボタンが用意されています。
- (b) **ブロックパレット**: プログラムで使用する命令を**ブロック**といいます。ブロックが用意されているエリアを**ブロックパレット**といいます。
- (c) **コードエリア**: プログラムを作るエリア。起動時、このエリアには何もありません。ブロックパレットから、必要なブロックをドラッグして、このエリアへ移動してきます。このブロックを組み合わせてプログラムを作ります。ブロックのまとまりを**コード**といいます。
- (d) **ステージ**: スプライトが動いたりするエリア。**スプライト**とはキャラクターや物のことです。ステージには**背景**を設定することができます。
- (e) **スプライトリスト**: プロジェクトで使うスプライトを管理するエリア。起動時、[スプライト1] (ネコ) のスプライトが用意されています。
- (f) **ステージリスト**: プロジェクトで使う背景を管理するエリア。起動時、透明 (白に見える) の背景が用意されています。
- (g) **バックバック**: スプライトやコードを一時保管するエリア。起動時、このエリアには何もありません。他のプロジェクトで使いたいスプライトやコードをこのエリアにコピーしておき、別のプロジェクトで取り出して使うことができます。

※この他に、[コスチューム] タブが選択されている場合は、**ペイントエディター** (スプライト・背

第2章 Scratch でこんなことができる!(各種命令の機能)

2.2 スプライトの位置を決めよう！(座標の設定)

2.2.1 位置と座標

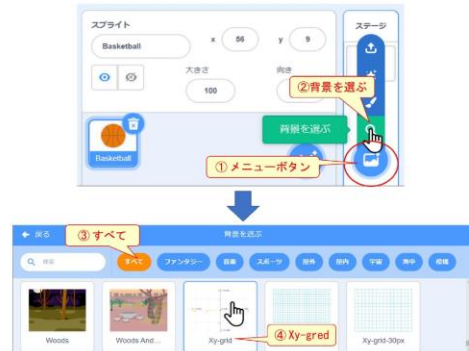
- 次のとおり、新規にプロジェクトを作成します。
- (a) プロジェクト名: 例題302座標の設定 ★1.2.1(2)
 - (b) スプライト: [スプライトリスト]の[メニューボタン]→[スプライトを選ぶ]→[スポーツ]→[Basketball] (バスケットボール)を追加→[スプライト1]を選択→[削除] ★1.2.3

図2.2.1 スプライトの設定



- (c) 背景: [ステージリスト]の[メニューボタン]→[背景を選ぶ]→[すべて]→[Xy-grid] (エクスワイグリッド、格子)を追加→[ステージリスト]の[ステージ]をクリック→[背景]タブ→[背景1]→[削除] ★1.2.4

図2.2.2 背景の設定



背景には十字の太い線が描かれています。これを座標軸といい、横(水平)の線をx軸、縦(垂直)の線をy軸といいます。ボールをドラッグして動かしてみましょう。すると、スプライトリストのxとyの欄の値が変化していることがわかります。下図のとおり、この値とボールのx軸とy軸の目盛りの値を比較してください。

第3章 記憶すること計算すること(変数と演算)

3.5 文字列を加工しよう！(文字列処理)

3.5.1 文字列処理に関するブロック

[演算] カテゴリには、2つの文字列を1つの文字列にまとめたり、文字列の長さを数えるなど、文字列処理に関するブロックが用意されています。その処理内容を次に示します。

表3.5.1 文字列処理に関するブロック

カテゴリ	ブロック	処理内容	例
演算	文字列を結合	指定した文字列や数字をつなげて1つの文字列を作る。例: 名前と年齢を結合 (年齢は12)	私は12歳です。
	文字列の長さを数える	変数、数値ブロックも使用できる。指定した文字列から指定した文字(半角)の文字を取り出す。全角、半角、空白ともに1文字と数える。例: 名前と年齢を結合 (年齢は12)	私は12歳です。の長さは14文字です。
	文字列の長さを数える	指定した文字列の長さを数える。全角、半角、空白ともに1文字と数える。例: 名前と年齢を結合 (年齢は12)	私は12歳です。の長さは14文字です。

文字列や数値を3つ以上つなげる場合は、次のようにブロックを重ねて使います。変数「年齢」の値が12であるならば、この例の処理結果は「私は12歳です。」となります。

図3.5.1 複雑な文字列の作り方



3.5.2 プログラム例(文字列処理)

(1) プロジェクトの作成

- (a) プロジェクト名: 例題351文字列処理 ★1.2.1(2)
- (b) スプライト: [人]グループの[Abby] (アビー、人名) ★1.2.3
- (c) 背景: [室内]グループの[Bedroom 1] (ベッドルーム1) ★1.2.4
- (d) 変数: [Abby] >> [名前] ※表示設定=チェックオフ (変数の左側のチェックを外す) ★3.1.1

(2) プログラムの説明

文字列処理を行うプログラムの例を次に示します。下図のとおり、名前をローマ字で入力し、そ

の先頭の文字を取り出し、イニシャルとして表示します。この例では名前を「Sakura Tanaka」と入力しています。このプログラムを作成・実行し、キーボード入力と文字列が処理される様子を観察しましょう。

図3.5.2 文字列処理



プロジェクト名が「例題351文字列処理」であることを確認してから保存します。★1.3.4

第6章 プログラムの間違い探し(デバッグ)

6.2.3 ブレークポイントを利用する

バグが疑われる処理部分にて、デバッグのために一時的に処理を止めて、そのときの変数や値ブロックの値を観望したいことがあります。この一時停止する箇所を**ブレークポイント**といいます。ブレークポイントを実現するには、下図左側のように、バグが疑われる処理部分の直前にブロック【0 まで待つ】を追加します。すると、このブロックにより処理が一旦停止します。つまり、このブロックを置いた箇所がブレークポイントになります。この例では、スペースキーを押すと次へ進むようにしています。繰り返し処理の中で使用する場合は、連続してキーが押されないように【(0.1) 秒待つ】を追加した方がよいでしょう。また、下図右側のように【もし】ブロックを使って、条件付きブレークポイントを実現することもできます。

図6.2.6 ブレークポイントを設定する機能の実現

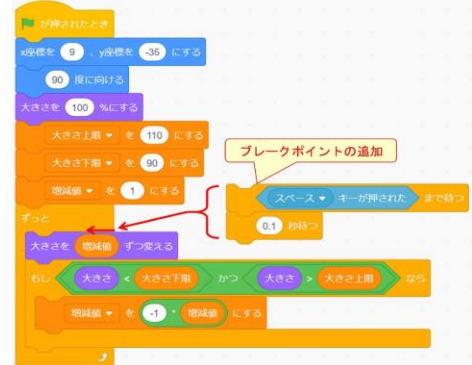


それでは、実際にブレークポイントを使ってデバッグしてみましょう。次のとおり、新規にプロジェクトを作成します。

- (a) プロジェクト名: 例題623ブレークポイントの利用 ★12.10
- (b) スプライト: [動物] グループの [Pufferfish] (パフアフィッシュ、フグ) ★12.3
- (c) 背景: [海中] グループの [Underwater 1] (アンダーウォーター、海中) ★12.4
- (d) 変数: [Pufferfish] >> [大きさ上限、大きさ下限、増減値] ※表示設定=チェックオン ★3.1.1

ここでは、フグが膨らんだりしぼんだりする様子を示すプログラムを例題とします。まず下図のとおり、プログラムを作成しましょう。この段階では、ブレークポイントは追加しないでください。このプログラムを実行すると、フグは膨らみ続けて、しぼみません。バグがあるのです。

図6.2.7 フグが膨らんだりしぼんだりするプログラム (バグあり)



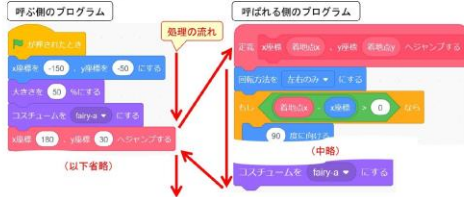
このバグの原因を探しましょう。まず下図のとおり、ブロックパレットの【変数】カテゴリにある変数 [増減値]、[大きさ下限]、[大きさ上限]、および【見た目】カテゴリにある値ブロック【大きさ】のチェックボックスをチェックオンし、ステージモニターで値を観察できるようにします。ステージモニターが重なった場合は、ドラッグして移動・調整します。

第7章 命令をまとめて使いやすくしよう!(ブロックの定義)

7.1.2 定義したブロックの処理の流れと引数

先のプログラムを例に、ブロックの処理の流れについて説明します。下図のとおり、ブロック【x座標(180)、y座標(30)へジャンプする】が実行されると、それが定義されているブロックが呼ばれます。すると、処理の流れは、**呼ぶ側のプログラム**から**呼ばれる側のプログラム**(定義したブロック)の先頭へ移ります。そして、その定義したブロックの処理が終了すると、呼ぶ側のプログラムの次のブロックへ処理の流れが戻ります。

図7.1.6 ブロックの処理の流れ



次に、ブロックのデータの流について説明します。下図のとおり、呼ぶ側のプログラムのブロックに設定した具体的なデータ(x座標=180、y座標=30)を**実引数**(じつひんすう)といいます。実引数の値は、呼ばれる側のプログラム(定義したブロック)の**着地点x**、**着地点y**に引き渡され、**着地点x**=180、**着地点y**=30と記憶されます。このように呼ばれる側のプログラムにあるデータを受け取る記憶場所を**仮引数**(かりひんすう)といいます。仮引数が複数ある場合は、実引数の記述順に対応してデータを受け取ります。仮引数は呼ばれる側のプログラムの中では、値ブロックのように扱い、計算などに利用します。なお、本書では仮引数を**着地点x**のように枠で囲み表記します。

図7.1.7 実引数と仮引数



【演習711】多角形の描画

(1) プロジェクトの作成

- (a) プロジェクト名: 演習711多角形の描画 ★12.10
- (b) スプライト: デフォルトの [スプライト1] (ネコ) をそのままとし、変更しない。
- (c) 背景: デフォルトの [背景1] (何も描かれていない背景) をそのままとし、変更しない。
- (d) 変数: [スプライト1] >> [外角] ※表示設定=チェックオフ ★3.1.1

(2) プログラムの説明

下図左側のプログラムは三角形を描くプログラムです。これを修正して多角形を描画するプログラムを作成しましょう。なお、下図右側のプログラムのとおり、多角形を描くブロックを定義して使用するものとします。

- (a) スプライトの初期の位置、向きは元のプログラムと同じ値とします。
- (b) 描画部分を全部消す処理とペンの色(黒)を設定する処理は、定義するブロックには含まないものとします(下図右側のプログラムを参照)。★2.13.1.2.13.2
- (c) 多角形を描画するブロックの名前および仮引数は次のとおりです。

【多角形 (頂点の数) 角形、一辺(辺の長さ)】を描画する ★7.1.1

<<目次>>

第1章 初めてのプログラミング

第2章 Scratch でこんなことができる！(各種命令の機能)

第3章 記憶すること計算すること(変数と演算)

第4章 場合分けと繰り返し(条件分岐と繰り返し)

第5章 出来事に対処すること同時にやること(イベント処理と並行処理)

第6章 プログラムの間違い探し(デバッグ)

第7章 命令をまとめて使いやすくしよう！(ブロックの定義)

<<著者紹介>>

多田 憲孝(ただ のりたか)

新潟工業短期大学教授、大阪国際大学教授を経て、現在プログラミングスクール「Wonder Processor」代表。

大阪国際大学名誉教授。1972 年より Fortran 言語でプログラミングを始める。振動解析、エキスパートシステム、スポーツ工学分野の運動解析などの研究に従事。スキーマの回転運動の数値解析を基に、VR を利用したスキーシミュレーターや AR を利用したスキー指導システムを開発。また、Scratch から C#言語に移行するプログラミング学習者を支援するための学習環境(Scratch シミュレーター)を開発。大学では、情報関連の講義および演習を担当。

主な著書に、「日本語版 Unity 2019 C#プログラミング入門」(インプレス R&D)、「アルゴリズム設計の基礎」(日本理工出版会)、「コンピューターと情報システム」(日本理工出版会)などがある。

内藤 富美子(ないとう ふみこ)

関西学院大学非常勤講師、大阪国際大学非常勤講師、大阪YMCA 国際専門学校講師兼情報授業コーディネーター、尼崎市中小企業センターパソコン教室メイン講師。大学専門学校では情報リテラシーやプログラミング演習を担当。パソコン教室では、ビジネス講座をはじめ小中高生やシニアへのリテラシー講座などを担当。幅広い世代への情報教育に携わっている。

著書に「コンピューターリテラシーMicrosoft Office Excel 編」(日本理工出版会)がある。

※執筆担当：第1～3章および第5～7章 多田憲孝、第4章 内藤富美子

<<販売ストア>>

電子書籍：

Amazon Kindle ストア、楽天 kobo イーブックストア、Apple Books、紀伊國屋書店 Kinoppy、Google Play Store、honto 電子書籍ストア、Sony Reader Store、BookLive!、BOOK☆WALKER

印刷書籍：

Amazon.co.jp、三省堂書店オンデマンド、honto ネットストア、楽天ブックス

※ 各ストアでの販売は準備が整いしだい開始されます。

※ 全国の一般書店からもご注文いただけます。

【インプレス R&D】 <https://nextpublishing.jp/>

株式会社インプレス R&D(本社：東京都千代田区、代表取締役社長：井芹昌信)は、デジタルファーストの次世代型電子出版プラットフォーム「NextPublishing」を運営する企業です。また自らも、NextPublishing を使った「インターネット白書」の出版など IT 関連メディア事業を展開しています。

※NextPublishing は、インプレス R&D が開発した電子出版プラットフォーム(またはメソッド)の名称です。電子書籍と印刷書籍の同時制作、プリント・オンデマンド(POD)による品切れ解消などの伝統的出版の課題を解決しています。これにより、伝統的出版では経済的に困難な多品種少部数の出版を可能にし、優秀な個人や組織が持つ多様な知の流通を目指しています。

【インプレスグループ】 <https://www.impressholdings.com/>

株式会社インプレスホールディングス(本社:東京都千代田区、代表取締役:唐島夏生、証券コード:東証1部9479)を
持株会社とするメディアグループ。「IT」「音楽」「デザイン」「山岳・自然」「旅・鉄道」「学術・理工学」を主要テーマに専門
性の高いメディア&サービスおよびソリューション事業を展開しています。さらに、コンテンツビジネスのプラットフォー
ム開発・運営も手がけています。

【お問い合わせ先】

株式会社インプレス R&D NextPublishing センター

TEL 03-6837-4820

電子メール: np-info@impress.co.jp