○ この教授用資料は、プログラミング言語「Scratch」と「ドリトル」を用 いた実習で活用するための資料です。「Scratch」はバージョン 3.0 に対 応しています。この資料は、弊社 Web サイトにも掲載しています。

弊社Webサイト「高等学校 情報」 https://www.nichibun-g.co.jp/textbooks/joho/





Scratchとドリトルではじめる ~6STEPでプログラムの基礎がわかる~

日文教	牧授用資料
令和元年	F(2019年)5月8日発行
編集・新	約天 佐々木秀樹
発行所	日本文教出版株式会社
	〒558-0041 大阪市住吉区南住吉4-7-5
	TEL:06-6692-1261

本書の無断転載・複製を禁じます。

プログラミング

CD33455

日本文教出版 株式会社 http://www.nichibun-g.co.jp/

阪本社	〒558-0041 大阪市住吉区南住吉4-7-5 TEL:06-6692-1261 FAX:06-6606-5171
京本社	〒165-0026 東京都中野区新井1-2-16 TEL:03-3389-4611 FAX:03-3389-4618
州支社	〒810-0022 福岡市中央区薬院3-11-14 TEL:092-531-7696 FAX:092-521-3938
海支社	〒461-0004 名古屋市東区葵1-13-18-7F•B TEL:052-979-7260 FAX:052-979-7261
海道出張所	〒001-0909 札幌市北区新琴似9-12-1-1 TEL:011-764-1201 FAX:011-764-0690

本資料は、一般社団法人教科書協会 「教科書発行者行動規範」に則り、 配布を許可されているものです。





本書の使い方

本書では、Scratchとドリトルというプログラミング言語を通して、プログラミングの基礎知識を学びます。 はじめに使用する言語の基本操作を学び、その後、いくつかのプログラムを作成します。プログラムを作 成・解説するページは、複数のステップに分かれています。各ステップで示された手順に従ってプログラム を作成すれば、最終的に冒頭にある「完成例」を作成することができます。



※ ScratchはMIT Media LabのLifelong Kindergarten Groupによって開発されています。 詳しくはhttps://scratch.mit.edu/を参照してください。

目次

Theme SCRATCH

Scratch編

•	Scratchに触れてみよう Scratchの基本操作	2
テーマ 1	アニメーションをつくろう 繰り返しの利用①	4
テーマ2	図形を描こう 繰り返しの利用②	6
テーマ 3	2つ以上のものを動かそう 複数スプライトの利用	8
テーマ4	音を鳴らそう 条件分岐の利用	10
テーマ 5	ゲームをつくろう① 条件分岐の応用と乱数の利用	12
テーマ6	ゲームをつくろう② メッセージと変数、数値演算の利用	14

Theme DOLITITLE

ドリトル編

ドリトルのインストール

•	ドリトルに触れてみよう ドリトルの基本操作	16
テーマ 1	はじめてのドリトルプログラミング 宝物拾いゲーム	
テーマ 2	図形を描こう 繰り返しの利用	20
テーマ 3	図形に命令しよう 図形オブジェクト・組図形オブジェクトの利用	22
テーマ 4	アニメーションをつくろう タイマーオブジェクトの利用	24
テーマ 5	ゲームをつくろう① 衝突定義の利用	
テーマ6	ゲームをつくろう② ボタンや条件分岐の利用	
Scratch 7	7カウントの作成とサインイン	30

ScratchはWebブラウザ上でプログラムを作成・実行することができます。さらにWebブラウザ上で プログラムを保存する場合は、Scratchアカウントが必要です。アカウントの作成方法についてはp.30、 31の解説を参照してください。

ドリトルはコンピュータにインストールしてプログラムを作成・実行します。インストール方法についてはp.32の解説を参照してください。

.. 32

Scratchに触れてみよう — Scratchの基本操作

Scratchは、Webブラウザ上で動作するビジュアルプログラミング環境で、あらかじめ用意された命令を 組み合わせることでプログラムを作成します。直感的に操作できるうえ、音や画像を扱えるマルチメディア 機能もあり、多彩なコンテンツを制作することができます。

まずは、プログラムを作成・実行するための基本操作と手順を学びましょう。



ブロック (命令) の一覧が表示 されます。スクロールするとす べてのカテゴリーのブロックを

確認できます。

画面右上の白い四角の 部分をステージと呼び ます。ここでプログラ ム実行時の状況を確認 することができます。 ステージとステージ上に存在す

るスプライトの一覧が表示され

ます。水色の太い枠は、ステー

ジやスプライトが現在選択中で あることをあらわします。

● 基本的な手順

ステップ1 プログラム作成を開始する

https://scratch.mit.edu/にアクセスして、ページ 左上にある「作る」をクリックします。

ステップ2 プログラムの作成

▲ カテゴリーで異なるカテゴリー(ここでは「イベント」)を選択すると、 B ブロックパレットに並ぶブロックが変わります。

ブロックを組み合わせる場合は、 C コードエリア にブロック (ここでは F vernexe) をドラッグして、右 図のようにガイドラインが出たことを確認してからド ロップします。

ステップ3 プログラムの実行

ステージの左上にある ▶ をクリックするとプログ ラムが実行されます。ここでは、ステージ上のネコが 右に少しだけ移動します。 プログラムを停止したいときは、● をクリックし ます。









● ブロックパレットや ○ コードエリア上でブロックをクリックすると、そのブロックのあらわす命令が現在選択中のステージやスプライトに対して実行され、動作の内容を確認することができます。また、ブロックと同様に、コード (p.2の ○ コードエリア参照)の動作の内容も確認することができます。
 ● コードエリアからブロックやコードを削除したい場合は、右クリックして「ブロックを削除」を選択するか、 ○ ブロックパレットへドラッグ&ドロップします。複数のブロックがあるときは、 ○ コードエリアを右クリックして [○個のブロックを削除」を選択すると、すべてのブロックを削除することができます。また、ブロックやコードを右クリッ

● 保存の方法

Scratchアカウントでサインインすると、作成したプログラムをWeb上に保存することができます。 Scratchアカウントの取得方法とサインインの方法については p.30 を参照してください。

	⊕• 274% ■	8 🛊 チュートリアル	ネコを動かす				
1 0-14	1 3273-4	41.0					
11	521						
					-		×
			10	\$	\$ L	ut:	
	RETS	() プロジェクトページEN	18 (05CMP	8		n-litera	ey -
							×
	10					LU I	*
たと書	*						
un Ca	1.0						

クしたときに「複製」を選択するとコピーできます。

画面上部にタイトルを入力します。自分やほかの人に内 容がわかるタイトルにしましょう。

作成したプログラムは自動的に保存されますが、変更 した内容をすぐに保存したい場合は、画面右上の「直ち に保存」をクリックします。また、画面左上の「ファイ ル」から「コンピューターに保存する」を選択してコン ピュータに保存することもできます。



ネコがステージの端ではね返りながら歩くアニメーションを作成します。ここでは、ひと続きの命令を何 回も実行する「繰り返し」の使い方を理解しましょう。



ステップ1 ネコがずっと動き続ける



ステップ2 ネコがステージの端で反転する



1 ステージの端にネコが着いたら、反転する。

ステージの端でネコがはね返るように変更します。

ステージの左上にある ● をクリックすると、

─── ●●●● の [10] をクリックすると数字

を入力できます。正の数だけでなく、小数

や負の数も入力することができます。この例の場合 は、変更するとネコの1回あたりの移動距離が変わり、

【確認】実行すると、ネコがステージの右端へ進んで消えてし

まいます。ネコの尻尾の部分を左側へドラッグ&ドロップする

次のステップでは、ネコが消えてしまわないようにするため、

それを繰り返すことで見た目の速さが変わります。

と、ステージ内へ戻すことができます。

2 右に10歩移動することをずっと繰り返す。

【確認】ステージの端にぶつかると、ネコの上下左右が反転します。次のステップで、左右にのみ反転するようにします。

ステップ3 ネコが左右にのみ反転する



スプライトの詳細設定を行います。

- 1 スプライトの名前を「ネコ」に変更する。
- 2 向きの欄をクリックする。
- 3 🖂 をクリックする (左右のみに反転する)。
- 4 ネコの進む方向を変更する。



ステップ4 ネコが歩いているように見せる



スプライトのコスチュームを変更して、
 0.1秒動作を止める。

Scratchではスプライトの見た目のことを コスチュームといい、各スプライトには複 数のコスチュームを設定することができます。ここで は 2012/31-ACT という [見た目] カテゴリーのブロッ クによって10歩移動するごとにコスチュームを切り 替え、歩いている様子を表現しています。

「制御」カテゴリーのブロック 0 000 の[1] をク リックして [0.1] を入力すると、2 の 0 000 をつ くることができます。ここではネコの動作を0.1 秒間 停止させることで、ネコの進む速さやコスチュームを 切り替える速さを制御します。

-0月1 これでプログラムは完成です。

をクリックして実行すると、ネコがステージの端ではね返りながら歩き続けるアニメーションになります。
「繰り返し」を利用すると、ひと続きの命令を、同じ順番で何回も実行することができます。この繰り返しの部分は、上の命令から順番に繰り返し実行されます。実行したい命令は何かを考えるだけでなく、どのような順番で命令を実行すればよいのかを考えてプログラムをつくることが重要です。



🔵 やってみよう

- 2. 100歩歩くごとに3秒立ち止まるように、プログラムを変更してみよう。



ネコが歩きながら正方形を描くアニメーションを作成します。ここでは、「繰り返し」の回数を指定して命令 を実行する方法を学びます。



ステップ1 ネコが正方形の1辺を歩く



2 10歩移動することを10回だけ繰り返す。

1 ステージの左上にある ▶ をクリックすると、



のブロックです。「ずっと」ブロック と同じように、取り囲んだブロック を繰り返し実行しますが、このブロックは命令を繰り 返す回数(「10」の部分)を指定することができます。

【確認】正方形の1辺だけをネコが歩くようになります。ステップ2では直角に曲がり、次の辺を歩く準備をします。

ステップ2 ネコが向いている方向を反時計回りに90度回転する



1 反時計回りに90度回転する。

2 2 2 1 は 「動き」 カテゴリーのブロック です。ネコが向いている方向を、矢印の方 向 (反時計回り) に 90 度回転させます。

【確認】歩いたあとにネコの向きが変わります。次のステップ では、ここまでの処理を4回繰り返すことで、正方形の4辺を 歩くようにします。

ステップ3<mark>〉ネコが正方形を描くように歩く</mark>



1 右を向く。

2 「10歩移動することを10回繰り返したのち、反時計回りに90度回転する」という動作を4回繰り返す。



このコードではネコが反時計回りに進んで 正方形を描くように歩きますが、 つ (20 xiet) のかわり に で (20 xiet) を使うと、時計回りに進むようになりま す。

【確認】ネコが正方形を描くように歩きます。次のステップ4では、ネコが歩いた跡を描画するようにコードを変更します。

ステップ4 ネコが歩いた跡を描画する



線を描画するために、拡張機能を利用します。画面 左下のごをクリックして「ペン」を選択してくださ い。カテゴリーに「ペン」が追加され、新しいブロッ クを利用できるようになります。

 ステージ上にペンで描かれているものがあれば、 すべて消す。

2 動く跡を描画するために、ペンを下ろす。
 3 ステップ3の動作を実行する。

4 ペンを上げて、描画をやめる。

レインをTot 、 インをTot 、 インをLife は、 「ペン」カテゴリーのブロックです。このカ テゴリーにあるブロックは、スプライトで線などを描 画するときに使います。ペンの色や太さを変更するブ ロックもあります。

60AL! これでプログラムは完成です。

▶ をクリックして実行すると、ネコが歩きながら正方形を描くアニメーションになります。 ここでは「○回繰り返す」 ブロックを利用し、回数を指定して同じ命令を繰り返す方法を学びました。

た。「ずっと」ブロックとの違いを理解し、場面に応じて利用できるようになりましょう。

🔍 やってみよう

- 繰り返しの回数や回転する角度を変更して、正三角形や正五角形を描くようにプログ ラムを変更してみよう。
- 右の図のように正三角形と正方形を組み合わせた図形を描くプログラムを作成してみよう。



テーマ1、2のプログラムにはネコのみが登場しましたが、Scratchでは複数のスプライトを扱うことができます。ここではネコがマウスのポインターに向かって歩き、2つの目玉がネコを見るアニメーションをつくります。ここからは、各ステップで使うブロックのカテゴリーは命令の内容をヒントに探してみてください。



ステップ1 背景を設定する



1 画面右下の ご をクリックする。 2 [Xy-grid] をクリックする。

② で選択した [Xy-grid] はステージの座標 をあらわす背景です。ステージ全体は中央 を原点(0,0)とし、×座標が-240から240、×座標 が-180から180の座標であらわされます。この背景を 利用するとスプライトの座標を確認しやすくなります。

ステップ2 **ネコがマウスのポインターに向かって歩く**



テーマ1のステップ3 (p.5) で変更したように、詳細 設定でスプライトの名前を「ネコ」に変更します。 1 ネコをステージ左下 (-240, -180) に移動し、 2 回転方向を「左右のみ」にする。

3 ネコを「マウスのポインター」 へ向けながら動かす。



【確認】実行してみると、ネコがマウスのポインターに向かって 歩くようになります。次に目玉をつくります。

ステップ3<mark>> 目玉のスプライトを</mark>描く





イヌがネコに出会ったら、「♪ワン」と吠えるアニメーションをつくります。ここでは、「イヌがネコに出会った ら」のような条件を使って、条件を満たす場合にのみ特定の処理を実行する方法を学びます。



ステップ1 イヌを登場させる





ライトライブラリーが表示される。
2 [Dog2] をクリックする。
3 スプライトリスト右下の ☑ を押して [Blue Sky]
をクリックする。

1 スプライトリスト右下の 5 をクリックする。スプ

4 ネコとイヌのそれぞれのコードを左図のように作 成する。また、詳細設定で、スプライトの名前を 「ネコ」「イヌ」に変更する。



【確認】ステージ上のネコとイヌがそれぞれ別の動作をします。 次は、イヌがネコに触れたかどうかを調べる準備をします。

ステップ2 イヌがネコに触れたかどうかを調べるブロックを用意する



「調べる」カテゴリーのブロックパレット内にある
 CLART で、色の部分 (●)をクリックする。
 2 2 をクリックする。

3 ネコのオレンジ色の部分をクリックする。

【確認】次は実際にイヌがネコに触れたかどうかを調べます。

ステップ3 イヌがネコに触れたかどうかを調べる





をクリックして実行すると、ネコはステージの端ではね返りながら歩き、イヌはマウスのポインターの位置に従って歩きます。イヌがネコに触れたときの動作と触れていないときの動作の違いを確認してください。

ここで利用した「もし~なら」ブロックのように、条件によって異 なる処理を行うことを「条件分岐」といいます。プログラムをつく るうえでとても重要な考え方なので、しっかり理解して使いこな せるようになりましょう。



• やってみよう

- 1. ネコがイヌに出会ったときに「♪ニャー」と鳴くプログラムを作成してみよう。
- スペースキーが押されたときに、ネコが「♪ニャー」と鳴いて、イヌが「♪ワン」と吠えるプログラムを 作成してみよう。

《ヒント》 スペース・ キーが Went を使うと、キーが 押されたかどうかを調べることができます。



落下するリンゴをネコがキャッチするゲームをつくります。リンゴはランダムな位置から落下し、地面に 落ちるかネコに当たると消えて、再びランダムな位置に現れます。前回(テーマ4)に引き続き、条件分岐を 利用しますが、ここでは条件を満たさない場合にも特定の処理を実行する方法を学びます。



ステップ1 ゲーム開始時のネコの位置を設定する



1 ステージの左上にある ■ をクリックすると、
 2 ネコが (0, -110) に移動する。

3 回転方向を「左右のみ」にする。

【確認】ゲームを開始したときのネコの位置を設定できました。 次はネコを矢印キーで操作できるようにします。

ステップ2 ネコを左右の矢印キーで操作する



1 「右向き矢印」 キーが押されたら、

2 ネコの進む方向を90度 (右向き) にして10歩動か し、コスチュームを変更する。

3 同様に「左向き矢印」 キーが押された場合の動作を 設定する(進む方向は-90度にします)。

20-2+1=20000240 を利用すると、キーの入 力があった場合の動作を設定することがで きます。キーの名前の部分をクリックすると、操作す るキーを選択できます。

【確認】 左右の矢印キーでネコを操作できるようになります。 次のステップでは、リンゴのコードを作成します。

ステップ3 ゲーム開始時のリンゴをランダムな位置に設定する



[Apple] というスプライトを追加して、「リンゴ」という名前に変更する。背景は「Blue Sky」にする。
 リンゴの×座標を-220から220のランダムな数値(乱数)、y座標を150にする。



【確認】実行するごとに、リンゴの出現する位置が変わります。 ステップ4では、リンゴを落下させます。

ステップ4 リンゴが空から地面まで繰り返し落下するようにする



ステップ5 リンゴがネコに当たった場合、音を鳴らしてステージ上部に戻る





● やってみよう

1. リンゴが出現する位置のy座標もランダムになるように、プログラムを変更してみよう。

2. 2つめのリンゴをつくり、1つめのリンゴと異なる速さで落下するようにプログラムを変更してみよう。



テーマ5のプログラムを改良して、ネコがキャッチしたリンゴの個数を数えるようにします。また、キャッ チしたリンゴの個数によってゲームを終了する条件も設定します。



ステップ6 ネコがキャッチしたリンゴの個数をあらわす変数を用意する



「変数」カテゴリーで「変数を作る」をクリック。
 新しい変数名の欄に「個数」と入力して「OK」をクリック。

変数とは、数値などのデータを入れておく 箱のようなものです。ここではキャッチし たリンゴの個数を覚えておくために用意します。

【確認】ステージ左上に 📾 💴 と表示されるようになりま す。次は個数を数える準備をします。

ステップ7 リンゴがネコに当たったことを、ネコに知らせるための準備をする



「イベント」カテゴリーの左図のブロックで、「メッセージ1▼」をクリックして「新しいメッセージ」を選択。
 新しいメッセージ名の欄に「当たった」と入力して「OK」をクリックする。

3 「~を送る」 ブロックで 「当たった」 を選択。

【確認】変数とメッセージのブロックの準備ができました。次のステップでメッセージのブロックを利用します。

ステップ8 リンゴはメッセージを送り、ネコはメッセージを受け取る



リンゴはネコに当たったら、
 「当たった」メッセージを各スプライトに送る。
 ネコは「当たった」メッセージを受け取ったら、
 「おいしい」と0.5秒言う。

メッセージを使うと、あるスプライトにイ
 ベントが発生したことを、ほかのスプライ
 トやステージに通知することができます。

【確認】リンゴに当たることで、ネコが「おいしい」と言うよう になりました。次は、キャッチしたリンゴの個数を数えます。

ステップ9 ネコがキャッチしたリンゴの個数を数える



 がクリックされたら、変数「個数」を0に設定 (p.12のステップ1のコードに追加します)。

2 「当たった」メッセージを受け取ったら、変数「個数」を1ずつ増やす(ステップ8の3と4の間にブロックを入れます)。

【確認】キャッチしたリンゴの個数を数えられるようになりました。最後のステップ10では、この個数を使って、ゲームを終了する条件をつくります。

ステップ10 ネコがキャッチしたリンゴが5個になったら終了する



2 コードの実行を終了する。

「演算」カテゴリーには、大小の比較などを 行うブロックが用意されています。各ブロッ ク内の空欄部分には、「5」のような数値だけではなく、 「個数」のようなブロックを入れることもできます。



🔵 やってみよう

- 1. リンゴをキャッチするごとに、リンゴの落下速度が上がるようにプログラムを変更してみよう。
- 2. 虫 (Ladybug2) のスプライトを追加し、ネコに虫が当たるとネコが「虫だ!」と1秒言って、ゲームを 終了するようにプログラムを変更してみよう (虫の動かし方は自由に決めよう)。

ドリトルに触れてみよう — ドリトルの基本操作

ドリトルは教育用言語のひとつです。オブジェクトと呼ばれるモノをつくり、それに命令を送ることでプロ グラムを動作させます。主人公はカメ (タートルオブジェクト) で、そのカメに命令を送ることで、絵を描い たり、ゲームをつくったりすることができます。

まずは、プログラムを作成・実行するための基本操作と手順を学びましょう。

● メイン画面





● 基本的な手順

ステップ1 ドリトルの起動

ドリトルをインストール (p.32) したときにできた フォルダ内にある起動ファイル (dolittle.bat) をダブ ルクリックして、ドリトルを起動します。

ステップ2 プログラムの作成

「編集画面」の1行日に

タートル!作る。

と入力します。

ステップ3 プログラムの実行

画面下にある [実行!] ボタンをクリックします。 すると「実行画面」に切り替わり、入力したプログラ ムが実行されます。ここでは、画面の中央にカメが表 示されます。

プログラムを追加したり変更したりする場合は、画 面上の「編集画面」 ボタンをクリックしてプログラム の画面に戻ります。

文 プログラムを入力するとき、日本語入力のオン (全角) とオフ (半角) は、キーボードの [半角/全角] キー で切り替えます。英数字や記号は本来「半角」で入力しますが、ドリトルでは「全角」で入力することもで きます。だたし、マイナス記号「-」は、使用するコンピュータの環境によっては「全角」で入力すると正しく認識さ れない場合があるので、注意が必要です。

- X = 🛛 -

▶·**1**1 赤-ム

* クイック アクセス

= デスクトップ

🍒 ダウンロード

B ドキッイント

B. ドクチャ

■ ビデオ

1 21-590

共有 表示

← → → ↑ 📜 > PC > ダウンロード > dolittle238wir □ 名前

● 保存と終了の方法

確認

? ドリトルを終了しますか?

lはい(Y) いいえ(N)

作成したプログラムを保存し、ドリトルを終了します。



1 ドリトルの [保存] ボタンをクリックします。 2 次に表示される 「保存」 画面で、 保存先を確認(または変更)し、 • プログラムに名前 (ファイル名) をつけて、 • [保存] ボタンをクリックします。 「.dtll という拡張子がつけられたプログラムが保存されます。 保存したファイルを利用するときは、かならずドリトルの「開く」 ボタンから開きます。ファイルのダブルクリックでは開けないの で注意してください。

 ドリトルの「終了」ボタンをクリックします。 2 表示される 「確認」 画面で 「はい」 をクリックすると終了します。

ConeDrive arduino.ini 2016/03/11 16:14 book.pdf PC dolittle.bat 2016/03/11 16:15 ● ネットワーク dolittle exe 2016/03/11 16:19 dolittle jar 2016/02/11 16:1

アプリケーション ツール dolittle238win

ダブルクリック

更新日時

2016/03/11 16:14

2016/03/11 16:14

2016/03/11 16:14

2016/03/11 16:14

2016/03/11 16:14

2015/04/06 18:54

2016/03/11 16:14

使理

arduino_dolittle

l ire

III III

11b64

sample

studuino

studuino setu

※最新バージョンV3 (2019年4月時点) の場合は [dolittle tab. batl をダブルクリックします。





ドリトルでのプログラミングに慣れるために、ここでは簡単なゲームをつくります。細かい文法はテーマ2 以降で学びます。まずは各ステップに従い、プログラムを入力してください。そして、1行入力するたびに実 行し、実行結果にどのような変化があるか確認してみてください。

ステップ1 → 画面にゲームの主役をつくろう

かめた=タートル!作る。 <------- 1

11 「かめた」という名前でカメ (タートル) をつくりま す。



【確認】画面の中央にカメが表示されます。

ステップ2 「かめた」を操作するボタンをつくろう

左ボタン=ボタン!"左"■作る。 <------ 2

2 「左ボタン」をつくります。



【確認】実行すると画面左上に「左」というボタンが表示されま す。しかし、クリックしても何も起こりません。次のステップ で、押したときの動作をコンピュータに教えます。

※ステップ2~7のプログラム内の■は全角スペースをあら わしています。スペースキーで空白を入力してください。

ステップ3 ボタンを押した (クリックした) ときの動作を定義しよう

左ボタン:動作=「かめた!30 左回り。」。 <------ 3

3 ボタンを押したときに、「かめた」が左に30度回転



するようにします。 【確認】実行すると、「左」 ボタンを押すごとに 「かめた」 が左に 回転するようになります。

ステップ4 右に回転するボタンもつくろう

右



4 「右ボタン」をつくり、ボタンを押したときに「か めた」が右に30度回転するようにします。 【確認】「左」「右」のボタンで「かめた」がそれぞれ左と右に回転 します。

ステップ5 ∫ 「かめた」にエンジンをつけよう

時計=タイマー!作る。 時計! 「かめた!10 歩く。」実行。



【確認】実行すると「かめた」が走り出します。そのままにして おくと画面の外へ逃げていくので、ステップ2~4で作成し た「左」「右」ボタンで「かめた」を操作してみましょう。

ステップ6 宝物 (チューリップ) をつくろう

タートル!作る "tulip.png" 変身する ペンなし 100 100 位置。



6 宝物をつくります。tulip.pngは半角で入力します。

ドリトルの画面は、画面中央を原点(0,0) とする xv 座標になっています。画面下にあ る 「方眼紙」 にチェックマークを入れると、実行画面 で座標を確認できます。

【確認】 カメの右上に宝物 (チューリップ) が表示されます。次 のステップでは、宝物を増やします。

宝物を増やそう ステップ7

タートル!作る "tulip.png" 変身する ペンなし 100 -100 位置。 7 タートル!作る "tulip.png" 変身する ペンなし -100 100 位置。

> 7 ステップ6の6を「コピー」し、その下の行に2回 [貼り付け]ます。数値の部分のみ変更します。 【確認】次は「かめた」が宝物を拾えるようにします。

ステップ8 「かめた」が宝物を拾えるようにしよう

かめた:衝突=「│相手│相手!消える。」。◀…… 🛽

8 「かめた」と宝物が重なったときに相手を消して、 拾ったように見せます。

※|(縦線、パイプ)は、「5htt]+(ド)キーで入力できます。



GOAL! 、たった11行のプログラムですが、「左」 ボタンと 「右」 ボタンでカメを操作して宝物を拾う簡単な ゲームが完成しました。プログラムとはどういうものか、わかりましたか? 今回の実習を通し て気づいたことを確認してみましょう。 • ゲームなどのソフトウェアは、 書かれていないことは実行されない。

- プログラムでつくられている。
- プログラムは人間が書いている。
- プログラムは特別な「言語」で書く。
- 文法が違うとエラーになる。
- まちがって書くと、まちがって動く。

- 上から順に実行される。
- ある状態になったとき(ボタンが押されたとき、 衝突したときなど) に実行される命令もある。
- ソフトウェアは自分たちでつくることができる。



カメ(タートル)を歩かせると、動いた跡に線が残ります。これを利用して、ここではカメを画面上で動か して正方形を描きます。また、ひと続きの命令を何度も実行する「繰り返し処理」を利用して、プログラムを シンプルに記述する方法も学びます。



ステップ1 線を引く

かめた=タートル!作る。 < …… 1 かめた!100 歩く。 ◀ ◀ 2 かめた!90 左回り。 ◀ ◀ 3



1 「かめた」という名前でタートルをつくる。 2 「かめた」が100歩前進する。 3 「かめた」が左に90度回転する。



【確認】正方形の1辺が描けます。3 で直角に曲げておくこと で、次の1辺を描く準備をしています。

ステップ2 動きをまとめる (続けて命令する)

かめた=タートル!作る。	< 1
かめた!100 歩く。	< 2
かめた!90 左回り。	∢ 3
かめた!100 歩く 90	左回り。 < 🗝 🔤 👍

4 「かめた」が100歩前進して左に90度回転する。



【確認】23の2行と4は同じ意味の命令です。4のように一 連の動作をひとまとまりにして書くと、わかりやすいプログラ ムになります。

ステップ3 正方形にする



5 ステップ2の4をあと2回繰り返し正方形にする

(23の2行も1行にまとめておきましょう)。

【確認】これで正方形は完成です。しかし、このプログラムに は同じ命令が4つあり、むだが多いことに気づきます。このよ うに同じ命令を何回も記述する場合は、「繰り返し処理」の利用 により、プログラムをシンプルに記述することができます。次 のステップで、実際に「繰り返し処理」を記述する方法を学び ます。

ステップ4 「繰り返し処理」を利用してプログラムをまとめる

かめた=タートル!作る。 「かめた!100 歩く 90 左回り。」!4 繰り返す。 ステップ3の5の部分を、1行だけ残して削除します。 6 残った1行を[]で囲む。 7 [| 内の命令を4回繰り返す。

「」で囲まれた部分は、ブロックオブジェク トという、オブジェクトの一種になります。 オブジェクトなので「! | で命令を送ることができます。 7の「繰り返す」は文字通り処理を繰り返し行うため の命令で、引数で指定された回数だけ、「」内の命令を 繰り返し実行します。 《繰り返し命令の基本形》

「繰り返したい命令」!繰り返し回数 繰り返す。

GOAL!

これでプログラムは完成です。ステップ3、4のプログラムはどちらも正方形を描くプログラムです が、書き方を工夫することで、プログラムがシンプルになりました。「繰り返し処理」 はプログラムを つくるうえでとても重要な考え方なので、記述方法をしっかり理解しておきましょう。

かめた=タートル!作る。 「かめた!100 歩く 90 左回り。」!4 繰り返す。 ― 繰り返しの部分

🜔 やってみよう

- 1.1辺の長さが200の正三角形を描くように、プログラムを変更してみよう。
- 2. 正五角形や正六角形、星形を描くプログラムを作成してみよう。
- 3. 「回転角度|や「繰り返し回数」(必要なら「歩数」も)を変更して、いろいろな図形を描いてみよう。





ここではロボットの顔を描画しながら、図形オブジェクトや、複数の図形をまとめた組図形オブジェクト の扱い方を学びます。これらのオブジェクトの扱い方を覚えると、自由にイラストを描けるようになります。



゙゚プログラム゚

正三角形を描き、

3 「右目」 を青で塗りつぶす。

に命令できるようになります。

かめた=タートル!作る。 右目=「かめた!30 歩く 120 左回り。|!3 繰り返す (青)図形を作る。 左日=右日!作る 50 0 移動する。 □=かめた!15 円 (赤) 図形を作る 40 -15 位置。 かめた!90 向き。 輪郭=かめた!50 歩く 30 右回り 50 歩く 60 右回り 70 歩く 60 右回り 50 歩く 30 右回り 50 歩く 閉じる 図形を作る -20 -50 位置。 かめた!消える。 ロボット=図形!(右目)(左目)(口)(輪郭)結合する。 ロボット!20 左回り。

2 「右目」という名前で図形オブジェクトにする。

「かめた」の描いた図形は、「かめた」の一部

▶ です。「図形を作る| 命令を使うことで、独

立したオブジェクト (図形オブジェクト) になります。

塗りつぶしなどに使用する色は () で指定します。黒・

赤・緑・青・黄色・紫・水色・白の8色を使うことが

【確認】図形オブジェクトにしたことで、3のように直接図形

ステップ1 描画した図形をオブジェクトにして操作する

かめた=タートル!作る。 「かめた!30 歩く 120 右目=かめた!図形を作る 右目!(青) 塗る。	左回り。」!3 。 <2 、 3	繰り返す。
2 でつけた名前で命令を	2~4行目の命	令を
伝えることができます。	まとめることも	っできる。

右目=「かめた!30 歩く 120 左回り。」!3 繰り返す (青) 図形を作る。



ステップ2 オブジェクトを複製する



4 「右目」の複製として「左目」をつくる。 5 [左目]を右に50移動する。

でき、自分で色をつくることもできます。



【確認】「作る」命令で「右目」を複製して、「左目」をつくること ができました。次は口を描きます。



10

10 「輪郭」を描いて図形オブジェクトにする。 11 描画が完了したので [かめた] を隠す。



【確認】ロボットの絵が完成しましたが、このままでは単なる 部品の寄せ集めです。次のステップで、すべての部品をまと めて1つのオブジェクト (組図形) にします。

ステップ5 複数の図形を組図形にし、まとめて操作する

ロボット=図形!(右目)(左目)(口)(輪郭)結合する。 🛶 12 ロボット!20 左回り。 <!!!!

60 右回り 70 歩く 60 右回り 50 歩く

30 右回り 50 歩く 閉じる 図形を作る

-20 -50 位置。

9

かめた!消える。 <



12 「右目」 「左目」 「口」 「輪郭」 をまとめて、 「ロボット」 という名前で組図形オブジェクトにする。 13 [ロボット]を左に20度回転する。



、これでロボットの顔は完成です。組図形オブジェクトにすることで、ロボットの顔全体に対する 操作ができるようになりました。

描いた図形をオブジェクトとして動かすことができるのは、ドリトルの特徴のひとつです。いろい ろなイラストを描いて、図形オブジェクトや組図形オブジェクトを扱う感覚を身につけましょう。

🔘 やってみよう

1. プログラムを変更して、右の図のようにロボットに耳をつけてみよう。

2. 車や船などのイラストを考えて、それを描くプログラムを作成してみよう。





タイマーオブジェクトを利用して、気球を動かすアニメーションをつくります。タイマーオブジェクトは一 定の間隔で命令を繰り返し実行するもので、アニメーションやゲームを作成する場合に利用します。



プログラム

かめた=タートル!作る。
求=「かめた!50 歩く 45 右回り。」!8 繰り返す
(紫) 図形を作る。
ケーブル=「かめた!50 歩く 90 右回り。」!4 繰り返す
図形を作る 0 -120 位置。
かめた!(緑) 線の色 40 線の太さ。
ゴンドラ=かめた!80 歩く 図形を作る –15 –190 位置。
気球=図形!(球)(ケーブル)(ゴンドラ) 結合する。
時計=タイマー!作る。
時計!0.02 間隔 2 時間。
かめた!"akazukin.gif" 変身する ペンなし 23 –170 位置。
時計!「気球!0 5 移動する。かめた!0 5 移動する。」実行。
時計!「気球!0 -5 移動する。かめた!0 -5 移動する。」実行。

ステップ1 気球を描く





「球」を正八角形として描き、紫色で塗りつぶす。
 「ケーブル」を正方形として描き、「球」の下に並べる。
 「ゴンドラ」を緑色の太い直線として描き、「ケーブル」の下に並べる。
 部品を結合して、「気球」という組図形をつくる。

ステップ2 タイマーを利用して気球を動かす





5 「時計」という名前でタイマーオブジェクトをつくる。 6 「気球」を上へ5移動する命令を「時計」で繰り返し 実行する。

【確認】実行してみると、気球がゆっくり上へ移動します。これは5つつくったタイマーオブジェクト(名前は「時計」)が6つで「気球を上へ5ずつ移動する」命令を繰り返し実行するためです。次のステップで、タイマーの設定を変更します。

ステップ3 タイマーの設定を変更する

時計=タイマー!作る。 < 🖛 🧧	
時計!0.02 間隔 2 時間。	⊲ ······ 7
時計!「気球!0 5 移動する。」実行。	4 6

5 と 7 をまとめることもできる。

時計=タイマー!作る 0.02 間隔 2 時間。

 7 タイマーの設定を変更する (ステップ2の5と 6 の間に追加します)。

「間隔」 命令は繰り返しの間隔、「時間」 命令 は動かし続ける時間を設定します(単位: 秒)。「時間」 命令のかわりに、「回数」 命令で繰り返し 回数を設定することもできます。

【確認】次は、気球を下へ移動します。

ステップ4 **タイマー**を続けて実行する

時計!「気球!0 –5 移動する。」実行。 < 🚥 🛽 🛚





ステップ5 気球に女の子を乗せて一緒に動かす

<u>かめた!"akazukin.gif" 変身する</u> 9	ペンなし 23 -170 位置。 10	タートルオブジェクトは組図形にすることができない ので、気球とは別に動かす必要があります。
時計!「気球!0 5 移動する。 <u>かめ</u> 7	<u>と!0 5 移動する。</u> 」実行。 -	< <u>6</u>
時計!「気球!0 -5 移動する。かめ		8



- 9 「かめた」の姿を赤ずきん (akazukin.gif) に変えて、
 10 線を描かないように設定してからゴンドラに位置 を合わせる (ステップ2の 6 の前に追加します)。
- 「気球」に合わせて「かめた」を上へ移動する(ス テップ2の6に追加します)。
- 12 「気球」に合わせて「かめた」を下へ移動する(ス テップ4の8に追加します)。



問題なくプログラムを作成できていれば、気球に女の子が乗った状態で上下に移動します。アニ メーションやゲームをつくるには、タイマーオブジェクトが欠かせません。タイマーの設定方法 や使い方を十分に理解し、いろいろなオブジェクトを動かしてみましょう。

● やってみよう

- 1. 気球が上下に繰り返し移動するようにプログラムを変更してみよう。
- 2. 気球が3秒間上昇したあと、左上に進むようにプログラムを変更してみよう。



図形オブジェクトやタイマーオブジェクトを応用して、「ブロックくずしゲーム」をつくります。



プログラム

// タートルをつくる ボール=タートル!作る ペンなし。 ボール!90 向き 0 -100 位置。 かめた=タートル!作る。 かめた!(青) 線の色 20 線の太さ。

// 部品をつくる

ブロック=かめた!80 歩く 図形を作る -40 0 位置。
床=かめた!(赤) 線の色 380 歩く 図形を作る
-190 一30 位置。
壁=かめた!90 向き (緑) 線の色 400 歩く
90 右回り 400 歩く 90 右回り 400 歩く
0形を作る -200 -200 位置。
ブロック!-180 160 位置。
ブロック2=ブロック!作る (黄色) 塗る 90 0 移動する。
ブロック3=ブロック2!作る (紫) 塗る 90 0 移動する。
ブロック4=ブロック3!作る (水色) 塗る 90 0 移動する。
パドル=かめた!0 向き 100 歩く (青) 図形を作る
-50 -160 位置。

// タイマーでボールを動かす 時計=タイマー!作る 0.02 間隔 8 時間。 時計!「ボール!10 歩く。」実行。

// 衝突の定義

ボール:衝突=タートル:跳ね返る。 ブロック:衝突=「自分!消える。」。 床:衝突=「|相手|相手!消える。」。

ステップ1 ボール用と部品作成用のタートルをつくる

 // タートルをつくる (ボール=タートル!作る ペンなし。 ボール!90 向き 0 -100 位置。)
 かめた=タートル!作る。 (ボー・3)
 かめた!(青) 線の色 20 線の太さ。 (ボー・4)



2 ボール用のタートル

1 次の4行の内容の説明を書く。

「//」の右側は実行時に無視される部分で、 「コメント」といいます。一時的に実行しない 場合やプログラムの説明などを書く場合に利用します。

2 ボール用のタートルとして「ボール」をつくり、向きと位置を設定する。

3 部品作成用のタートルとして「かめた」をつくる。

4 「かめた」の描く線の色を青、太さを20にする。

【確認】実行すると画面にはカメが2匹表示されます。次のス テップでは、「かめた」を使ってブロックと床を描きます。

人 <u>テッノ2</u> 部品をつくる // 部品をつくる	
ブロック=かめた!80 歩く 図形を作る -40 0 位置。 末=かめた!(赤) 線の色 380 歩く 図形を作る -190	
	5 「ブロック」を1個つくる。 6 「床」をつくる。
ステップ3 ボールを動かす	
// タイマーでボールを動かす 持計=タイマー!作る 0.02 間隔 8 時間。 ◀ 7 持計!「ボール!10 歩く。」実行。 ◀ 8	 7 [時計]という名前でタイマーオブジェクトをつくる。 8 [時計]を使って「ボール」を動かす。 【確認】実行すると、ボールがブロックをすり抜けます。次の ステップで衝突したときの動きを設定します。
ステップ4 衝突したときの動きを設定	言する
// 衝突の定義 ボール:衝突=タートル:跳ね返る。 ◀───── 9 ブロック:衝突=「自分!消える。」。 ◀───── 10 床:衝突=「 相手 相手!消える。」。 ◀───── 11	9 「ボール」 は衝突すると自然にはね返る。 10 「ブロック」 は衝突されると消える。 11 「床」 に衝突したものは消える。
 ● はね返る ● はね返る 	衝突時に実行する命令を設定できます。ブ ロックオブジェクトの先頭に 相手 と書 くと、衝突した 「相手」に命令を送ることができます。 《衝突定義の基本形》 オブジェクト:衝突=「衝突時に実行する命令」。
11 消える	【確認】 実行すると、ボールがブロックにはね返り、床で消え る様子を確認できます。次に、部品を追加します。
ステップ5 部品を追加する (ステップ	2の 6の下に追加する)
壁=かめた!90 向き (緑) 線の色 400 歩く 90 右回り 400 歩く 90 右回り 400 歩く 図形を作る -200 -200 位置。 ブロック!-180 160 位置。 ◀ 13 ブロック2=ブロック!作る (黄色) 塗る 90 0 移動する。 ブロック3=ブロック2!作る (紫) 塗る	 12 外側の「壁」をつくる。 13 ステップ2の5 でつくった「ブロック」の位置を変更する。 14 ブロックを複製し、色を変えて配置する。 15 「パドル」をつくる。
90 0 移動する。 ブロック4=ブロック3!作る (水色) 塗る 90 0 移動する。 パドル=かめた!0 向き 100 歩く (青) 図形を作る -50 -160 位置。	ステップ3の B の行の先頭に 「//」を入れ てボールを停止させると、部品の位置が確 認しやすくなります。
パドル=かめた!0 向き 100 歩く (青) 図形を作る -50 -160 位置。	認しやすくなります。 記成です。ここでは衝突定義を利用することで、ブロッ いった衝突時の動作を設定しました。ステップ1の2の ボール」の速さを変更して実行してみましょう。



「ゲームをつくろう①」のプログラムを改良します。パドルを操作するボタンを追加し、得点を数える機能 と勝敗を判定する機能を加えて、ブロックくずしゲームを完成させます。



プログラム

// タートルをつくる

ボール=タートル!作る ペンなし。 ボール!45 向き 0 -100 位置。 ここにp.26の 「ゲームをつくろう①」の4行目から「// 衝突の定 義」の手前までのプログラムが入ります。

// 衝突の定義

得点=フィールド!0 作る。 ボール:衝突=タートル:跳ね返る。 ブロック:衝突=「自分!消える。得点!10 増やす。」。 床:衝突=「|相手|相手!消える。時計!中断。」。

// パドルを操作する

左ボタン=ボタン!"左" "LEFT" 作る。 左ボタン:動作=「パドル!-40 0 移動する。」。 右ボタン=ボタン!"右" "RIGHT" 作る。 右ボタン:動作=「パドル!40 0 移動する。」。

// 勝敗の判定

時計!待つ。 「(得点!読む)>=40]!なら「 ラベル!"ゲームクリア" 作る。 」そうでなければ「

ラベル!"ゲームオーバー" 作る。

」実行。

ステップ6 パドルを操作するボタンを追加する(①のプログラムに続けて書く)

// パドルを操作する

ボタンオブジェクト:動作=「クリックしたときに実行する命令」。

16 「左ボタン」という名前でボタンオブジェクトをつ くる。ボタンの表示は「左」にする。

「左」をクリックしたときに、「パドル」を左へ移動。
 「右ボタン」をつくり、ボタンの表示を「右」にする。クリックしたときに「パドル」を右へ移動。

*LEFT"や"RIGHT"は、画面上のボタンだけ でなく、左右の矢印キーでも操作できるようにするために指定します。また、左の吹き出しのように、作成したボタンオブジェクトには、ボタンをク リックしたときに実行する命令を設定できます。

【確認】画面左上に「左」「右」というボタンが表示されます。ボ タンや矢印キーでパドルが移動することを確認しましょう。ボ タンが壁に重なる場合は、画面を大きくして実行し直します。

ステップ7 ボールの向きを変更する(①のステップ1の 2を変更する)

ボール! <u>90</u> 向き 0 –100 位置。	19 「ボール」の向きを45度に変更する。
19	【確認】 実行すると、ボタンや矢印キーでパドルを操作して遊
ボール!45 向き 0 -100 位置。	べます。次はブロックを消すときに得点を数えます。

ステップ8 衝突の定義を変更して得点を数える(①のステップ4のプログラムを変更する)

a。 得点!10 増やす。」。 21 3。時計!中断。」。 22 右	 20 「得点」という名前でフィールドオブジェクトをつくり、最初の値として0を設定する。 21 「ブロック」を消すときに得点を10点増やす。 22 「床」にボールが衝突したら「時計」を止める。
	フィールドは、数値や文字を入力・表示で きるオブジェクトです。得点の表示や時間 のカウントダウン表示などに利用できます。 22の変更はまだプログラムの見た目に影響しません が、次のステップで利用します。
*	【確認】画面左上に得点が表示され、ボールをブロックに当て ることで得点が増えるようになります。次に、この得点を読み 取って勝敗を判定します。
	$\circ \land \neg \neg \neg \rightarrow \leftarrow \leftarrow \land \land \rightarrow \land \land \land \rightarrow \land \land \land \land \land \land \land \rightarrow \land \land$

ステップ9 勝敗を判定する (ステップ6のプログラムに続けて書く)

// 勝敗の判定
時計!待つ。 ▲ 23
「(得点!読む) >=40]!なら「
ラベル!"ゲームクリア" 作る。
」そうでなければ「
ラベル!"ゲームオーバー" 作る。
」実行。

23 「時計」の実行が終わってから(またはステップ8の22が実行されたら)以降のプログラムに進む。
 24 得点が40点以上の場合(すべてのブロックを消した場合)は「ゲームクリア」、40点未満の場合は「ゲームオーバー」と画面に表示する。

「待つ」命令はタイマーオブジェクトの実行の終了を待ちます。 23 の命令がないと、「時計」を実行している間に 24 の処理が実行 されてしまいます。

24は、条件によって異なる処理を行う「条件分岐」を利用しています。ここでは「得点が40点以上か」という条件によって、「ラベル」をつくる命令で 画面に表示する文字(""で囲まれた部分)を変更します。 (条件分岐の基本形》
 「条件式」!なら「
 条件が成り立つ場合の処理
 」そうでなければ「
 条件が成り立たない場合の処理
 」実行。



// 衝突の定義

10

得点=フィールド!0 作る。 <

ボール:衝突=タートル:跳ね返る

ブロック:衝突=「自分!消える。

床:衝突=「|相手|相手!消える

これでブロックくずしゲームは完成です。ここではシンプルなプログラムにすることを重視しました が、「乱数」や「配列」などを学ぶと、ボールの開始位置をランダムにしたり、ブロックをまとめて扱っ たりすることができます。ゲームとしての完成度を高めることにも挑戦してみましょう。今後はぜひ、 オリジナルのゲームやアニメーションを考えてプログラムで表現してみてください。

• やってみよう

1. p.18、19のテーマ1で作成した「宝物拾いゲーム」で、宝物を拾うときに得点を数えて、すべての宝物 を拾ったらゲームを終了するようにプログラムを変更してみよう。

ゲームをつくろう② ―ボタンや条件分岐の利用― 29

Scratchアカウントの作成とサインイ

.....

Scratchアカウントを作成し、サインインする手順を解説します。

※ Scratchには Web ブラウザ上で利用するオンライン版と、コンピュータにインストールして利用するオフライン版 (Scratch デスク トップ) があります。オンライン版では、インストールなどの作業をすることなく Scratchの最新機能を利用することができます。

ScratchのWebページにアクセスする



1 Webブラウザを起動して、URL欄に [https:// scratch.mit.edu/」と入力します。 Enterキーを押して、ページを開きます。 3 「参加する」 をクリックします。

ステップ2 ユーザー名とパスワードを入力する



- 1 自分で決めたユーザー名 (3~20文字)を入力し ます。ユーザー名には、半角英数字、ハイフン、 アンダーバーを利用できます。
- 入力した名前がすでに使われている場合は警告が 出るので、別の名前を入力します。
- 2 パスワードを入力します。確認のため2回入力し ます(一致しない場合は警告が出ます)。入力した パスワードはしっかり覚えておきます。 3 「次へ」をクリックします。

誕生年・月、性別、国を入力する ステップ3



1 誕生年・月を選択します。 2 性別を選択します。 3 国を選択します。日本の場合は [Japan] です。 4 [次へ]をクリックします。

> メールアドレスを入力する



1 メールアドレスを入力します。確認のため2回入 力します(一致しない場合は警告が出ます)。この メールアドレスには、確認用のメールが送信され ます。

2 [次へ] をクリックします。

ステップ5 登録を終了する



ステップ6メールを確認する



1 ステップ4で入力したメールアドレスのメールを 確認します。Scratchから確認用のメールが送信 されているはずです。このメールの本文中にある 「電子メールアドレスの認証」をクリックすると、 メールアドレスの確認が完了します。

了します。

これでScratchアカウントを作成し、サインインすることができました。 サインアウトするときは、画面右上のユーザー名をクリックして 「サインアウト」を選択します。

ステップ7<mark>→サインインする</mark>



- 1 サインアウトした状態からサインインする場合は、 画面右上の「サインイン」をクリックします。
- 2 ステップ2で入力したユーザー名とパスワードを 入力します。
- 3 [サインイン]をクリックします。

ScratchのWebページでは、サインインしていない状態でも、プログラムを作成・実行したり、ほかの人 がつくったプログラムを見ることができますが、サインインしていれば、作成したプログラムを保存・公開 することができます。また、公開された作品にコメントをつけたり、ほかの人が作成したプログラムをもと にして新しい作品をつくったり (リミックスといいます) することもできます。

作品に対するコメントやリミックスなどの交流を通じて、作品づくりを楽しみましょう。

ドリトルのインストール

ドリトルをインストールする手順を解説します。ここでは Windows にインストールしています。Mac な どにインストールする場合は、ステップ1の 「ダウンロード」 ページの説明を参照してください。

ステップ1 「ドリトルのダウンロード」ページにアクセスする



 Webブラウザを起動して、URL欄に [https:// dolittle.eplang.jp/download] と入力します。
 Enterキーを押して、ページを開きます。

※ドリトルにはWebブラウザ上で利用するオンライン版と、 コンピュータにインストールして利用するローカル版があり ます。

※ドリトルはJavaというソフトウェアを利用しますが、インストール版にはJavaが同梱されているので、ステップ1~3の手順を進めるだけでドリトルを利用することができます。

ステップ2 パッケージを選択する

 [Windows用]のパッケージをクリックして、ダ ウンロードします。

- 監修 兼宗進 大阪電気通信大学教授
 辰己丈夫 放送大学教授
 中野由章 神戸市立科学技術高等学校教頭、大阪電気通信大学客員准教授
 伊藤一成 青山学院大学准教授
- 著者 吉田葵 青山学院大学助教 (Scratch編)佐々木寛 北海道小樽潮陵高等学校教諭 (ドリトル編)

編集	株式会社トップスタジオ	(増子萌)
デザイン	株式会社トップスタジオ	(轟木亜紀子)
印刷・製本	東洋美術印刷株式会社	

- ・Scratch編のプログラムはScratch 3.0で動作を確認しています。公式にサポートされているブラウザは次の通りです(2019年 4月時点。最新の情報についてはhttps://scratch.mit.edu/を確認してください)。
- ・デスクトップ:Google Chrome 63以上、Microsoft Edge 15以上、Mozilla Firefox 57以上、Safari 11以上(Internet Explorerはサポートされていません)
- ・タブレット:Google Chrome Mobile 62以上、Mobile Safari 11以上
- ・ドリトル編のプログラムはドリトルV2.38 (Windows用/Java同梱版、Mac用/Java同梱版) で動作を確認しています。 動作環境についてはhttps://dolittle.eplang.jp/のダウンロードページを参照してください。

ステップ3 ダウンロードしたファイルを展開して、「dolittle.bat」をダブルクリックする

📕 🛛 🛃 🖛 🛛 do	little32wi	n		
ファイル ホーム	共有	表示		
← → • ↑ 📘	> PC :	ダウンロード > dolittle32wi	n	
		名前	更新日時	種類
- 7137 77EA		arduino_dolittle	2019/02/20 16:31	ファイル フォルダー
TX9F97	×	Doc	2019/02/20 16:31	ファイル フォルダー
🕹 ダウンロード 🛛 🦻	A	dolittle_lib	2019/02/20 16:31	ファイル フォルダー
🔮 ドキュメント	*	🔜 jre	2019/02/20 16:32	ファイル フォルダー
■ ピクチャ	*	lib	2019/02/20 16:32	ファイル フォルダー
a OneDrive		lib64	2019/02/20 16:32	ファイル フォルダー
		sample	2019/02/20 16:32	ファイル フォルダー
PC		studuino	2019/02/20 16:35	ファイル フォルダー
		arduino.ini	2019/02/20 16:31	構成設定
イットリーク		lolittle.bat	2019/02/20 16:31	Windows パッチ
		🔬 dolittle.jar	2019/02/20 16:31	Executable Jar Fi
		dolittle tab.bat	2019/02/20 16:31	Windows Kyf

ステップ2でダウンロードしたファイルを展開します。

2 展開したフォルダ内にある [dolittle.bat] (Windows バッチ ファイル) をダブルクリックすると、ドリ トルが起動します。

**最新バージョンV3 (2019年4月時点)の場合は、「dolittle_ tab.bat」をダブルクリックして起動すると、本書で記述した V2.38と同様に「編集画面」と「実行画面」が同一ウインドウ (タブによる切り替え)で表示されます。「dolittle.bat」で起 動した場合は、「編集画面」と「実行画面」は別々のウインドウ で表示されます。