

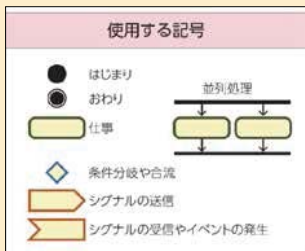
## Q & A

**Q1**：処理の手順を表すのはフローチャートがいいですか？それともアクティビティ図がいいですか？

**A1**：表現の幅が広いアクティビティ図をおすすめしますが、図法の技能としてよりも、処理の手順を考えることが大切です。

### 設計に使う図法

- ・今までは処理の手順を表すのにフローチャートを用いていました。新しい学習指導要領では、統一モデリング言語 (UML) のアクティビティ図を使用することが示されています。フローチャートでは表しにくかった表現もできます。
- ・厳密な図法を覚えさせるよりも、考えたことをまとめ、相手に伝えるように表現できることが重要です。



教科書 p.208

**Q2**：インターネットにつなげられない環境では、ネットワークはどうすればいいですか？

**A2**：ネットワークの種類は限定していないため、校内 LAN や、Bluetooth などの利用も考えられます。

**Q3**：コンテンツのデザイン性はどのくらい重視すればいいですか？

**A3**：情報デザインについて学習するため、見やすく、伝わりやすいデザインは評価します。装飾性に凝りすぎないようにしましょう。



開隆堂出版株式会社

<http://www.kairyudo.co.jp>

本社	〒113-8608 東京都文京区向丘 1-13-1	☎ 03(5684)6111
北海道支社	〒060-0061 札幌市中央区南一条西 6 丁目 11 番地 札幌北辰ビル 8 階	☎ 011(231)0403
東北支社	〒983-0852 仙台市宮城野区榴岡 4-3-10 仙台 TB ビル 4 階	☎ 022(742)1213
名古屋支社	〒464-0802 名古屋市中千種区星が丘元町 14-4 星ヶ丘プラザビル 6 階	☎ 052(789)1741
大阪支社	〒550-0013 大阪市西区新町 2-10-16	☎ 06(6531)5782
九州支社	〒810-0075 福岡市中央区港 2-1-5 FVC ビル 3 階	☎ 092(733)0174

## 技術・家庭「技術分野」

教科書を使ってできる!

ネットワークを利用した

# 双方向性のあるコンテンツ

プログラミングによる問題解決

令和3年度用  
内容解説資料

### 目次

- 「双方向性のあるコンテンツ」とは何だろう？ …… 2
- プログラミング言語は何を選べばいいのだろう？ …… 2
- どうやって指導すればいいのだろう？ …… 3
- 教科書ではどうなっているのだろう？ …… 4
- プログラミングの指導はどうすればいいのだろう？ …… 6

開隆堂

## 「双方向性のあるコンテンツ」とは何だろう？

「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による問題解決」（以下、「双方向性のあるコンテンツ」）を分かりやすくするため、三つに分けて考えてみましょう。

### ネットワークを利用した

コンピュータ同士で相互に通信できることが求められます。どのような通信方法を用いるという規定はないため、インターネットに限らず、校内LANや特定の場所に限定したネットワークでも構いません。

### 双方向性のあるコンテンツ

コンテンツとは、デジタル化された文字や画像などを人間にとって意味のある情報として表現したものです。使用者のはたらきかけ（入力）を処理し、異なる応答（出力）をすることが双方向性です。

### プログラミングによる問題解決

実現したいことや問題点を発見し、技術の見方・考え方をはたらかせながら、プログラミングによって問題を解決します。

言葉だけ見ると難しいと思ってしまうかもしれませんが、社会で利用されているWeb上のサービスの多くが双方向性のあるコンテンツのプログラムです。ショッピングサイトや乗換案内など、社会での必要性がますます高まっている技術です。

## プログラミング言語は何を選べばいいの？

プログラミング言語にはさまざまなものがありますが、この資料では、教科書で主に扱っている「Scratch」を中心に解説しています。Scratchは、マサチューセッツ工科大学が開発した、命令などのブロックを組み合わせる形式のプログラミング言語です。

またScratchには、サンプル画像や効果音などが豊富に用意されている上に、自分で画像を描く、編集する、カメラで撮影する、音を録音する機能など、メディアを統合的に扱うことができます。



教科書 p.241

命令がブロックという見える形で表示されているので、何ができるのかがわかりやすい。

文法が分からなくても形状で何がつながるのかが把握できる。

命令語を覚える必要がなく、また、キーの打ち間違いなどによるミスが発生しない。

### プログラミング言語を選ぶ視点

プログラミング言語は目的によってさまざまなものが開発されており、総合的な優劣は判断できません。学校現場においては、以下のような観点を考えます。

- 使用できる環境 … コンピュータの性能、ネットワーク回線、関連する教材（計測・制御など）など
- 生徒の状況 … 小学校での学習状況、生徒の興味など

開隆堂出版の教科書でScratchを主にした理由としては、**扱いやすく小学校でも取り扱う可能性が高い**ため接続性のよさが期待できること、教材でブロックを用いた形式をサポートしていることが多いことなどから、**処理の手順を工夫するなど、課題の解決に取り組みやすくなる**ことです。

## どうやって指導すればいいの？

世の中で使われている情報の技術の「見方・考え方」の基本となる内容（ネットワーク、情報デザイン、セキュリティ、メディア等）を学習した上で、問題解決を行います。そして、学習を通して情報の技術を工夫し創造していこうとする態度を育成します。

身近な技術を調べることを通して、双方向性のあるコンテンツの技術の「見方・考え方」を身につけます。

教科書 p.198

教科書 p.238

「見方・考え方」をはたらかせて、身近な生活や社会の中から、双方向性のあるコンテンツを利用した問題解決に取り組みます。

学習をふり振り返りながら、「見方・考え方」を定着させます。また、最新の話題と共に、技術を工夫し創造する態度を育成します。

教科書 p.263

### 問題解決の重要性

【技術・家庭編】中学校学習指導要領解説では、内容の取扱いにおいて、「プログラムの命令の意味を覚えさせるよりも、課題の解決のために**処理の手順（アルゴリズム）を考えさせる**ことに重点を置くなど、情報の技術によって課題を解決する力の育成を意識した実習となるよう配慮する」と記されています。使用する言語や、コーディングの技能にこだわりすぎず、問題解決を意識した授業が求められます。プログラミング教育における技術分野の役割は大きいですが、技術分野の目標を達成するための要素として、**生徒が問題解決を適切に行えるためのプログラミングを考える**ことが大切です。

# 教科書ではどうなっているのだろう？

## 2-1. 双方向性のあるコンテンツによる問題解決

### 社会とのつながり

発見した問題が、社会のどのような要求とつながっているかがわかります。

### 制作工程表

問題解決に必要な要素がひと目でわかるようになっています。ここでは、

- 発見した問題
  - 設定した課題
  - 全体の構想
  - (「入力-処理-出力」で分ける考えやすいです)
  - 使用するメディア
  - アクティビティ図
  - UI
- などの要素をまとめています。

### プログラムの例

プログラムの中心となる部分を掲載しています。どの機能なのかを示すことで、内容を理解しながらプログラミングができます。

## 実習例 1 安心して使えるメッセージ交換アプリ

### 問題の発見と課題の設定

#### ①問題の発見

ほかの人が勝手にメッセージをやり取りできてしまうため、やりとりした本人が不安になる。

#### 【社会とのつながり】

- 本人認証【セキュリティ】

#### ②課題の設定

使用開始時に本人認証をする機能をもつメッセージ交換アプリを作成することにした。

### 設計・制作

#### ①全体の構想

○メッセージを打ち込む前に、パスワードによる簡易的な本人認証を行う。

【入力】パスワードの入力、文字入力

【処理】入力されたパスワードが正しいかの判断、通信

【出力】成功/失敗かを表示、文字の表示、文字の送信

\*ここで用いられている認証システムは簡易的なもので、社会で利用されているものは異なります。

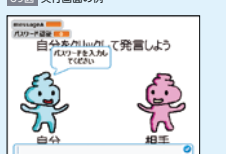
### 3表 安心して使えるメッセージ交換アプリの制作工程表

発見した問題	ほかの人が勝手にメッセージをやり取りでき、なりすましができてしまう。		
設定した課題	プログラムを起動したらパスワードによる本人認証を行う。		
課題を解決するために必要な仕様(全体の構想)	<b>入力</b> ・パスワードの入力 ・文字の入力	<b>処理</b> ・正しいパスワードと照合 ・通信(入力した文字、入力完了の合図、文字の送信)	<b>出力</b> ・パスワードが違ふことを表示 ・受信した文字を表示 ・文字の送信
使用するメディア	○画像:自分と相手のイラスト(png)		
課題を解決するためのアクティビティ図とUI(詳細の設計)	○画像:自分と相手のイラスト(png)    ○音声:警告音(mp3) アクティビティ図 (詳細の設計)		

### 68表 プログラムの例



### 69表 実行画面の例



開隆堂の教科書では、制作の目的や問題解決のための視点、プログラムの例などを掲載しているため、見直しをもって双方向性のあるコンテンツのプログラミングに取り組むことができます。

### 他言語での例示

他の言語でも例示をしているため、Scratchを選択しない学校でも安心して取り組みます。

ほかのプログラミング言語で同じようなものが作れないかな？

```

プログラム例:
Python: "password" 変数,
if password == "password":
    print("パスワードが正しく入力されています。")
else:
    print("パスワードが正しくありません。")
print("パスワードを入力してください。")
password = input("パスワードを入力してください。")

```

スマートメディア



### 共通の流れ

問題解決の流れはどの実習例でも共通しているため、取り組みやすくなっています。

### 評価・改善

①評価  
制作したアプリの画面や処理手順などを発表し、アイデアの有効性、機能性などを相互評価した。

### ②改善

- 次のような課題が見つかった。
- ・パスワードを打ち間違えても終了してしまい不便。
- ・ユーザー名も入れるようにできないだろうか？
- ・パスワード以外の本人認証はできないだろうか？
- ・この方法で本当に安全なのだろうか？

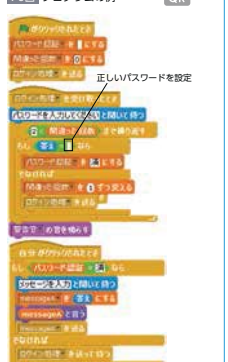
### 改善例 パスワードの入力回数を決められるチャットアプリ

パスワードで認証するとき、3回間違えると警告音を出すようにプログラムを改良しよう。

### 4表 パスワードの入力回数を決められるチャットアプリの制作工程表

発見した問題	一度間違えると使用できなくなるのが不便だが、何度も再入力できるというログインされそうに心配。		
設定した課題	パスワードを間違え上限回数を決めて、それを越えたと警告音を出す機能をもたせる。		
課題を解決するために必要な仕様(全体の構想)	<b>入力</b> ・パスワードの入力 ・文字の入力	<b>処理</b> ・パスワードを照合 ・間違いの回数を数える ・通信	<b>出力</b> ・文字の送信 ・文字の表示 ・警告メッセージ
使用するメディア	○画像:自分と相手のイラスト(png)    ○音声:警告音(mp3)		
課題を解決するためのアクティビティ図とUI(詳細の設計)	アクティビティ図 (詳細の設計)		

### 70表 プログラムの例



### 改善例

問題解決の「改善」で発見した課題について、実際にどのように取り組めるのかを例示しています。

教科書 p.242, 243

## 実習の流れの例(8時間)

### 問題の発見と課題の設定(1時間)

- ①問題の発見  
問題を発見し、社会とのつながりを考える。
- ②課題の設定  
問題を解決するための課題を決める。

### 設計(3時間)

- ①全体の構想  
基本的な機能を「入力-処理-出力」で考える。
- ②詳細設計とまとめ  
処理の流れや画面などの設計をし、まとめる。

### 制作(3時間)

- ・「自分」と「相手」のイラストを作成する。
- ・スプライト(キャラクター)を画面に配置する。
- ・それぞれの動作をプログラミングする。

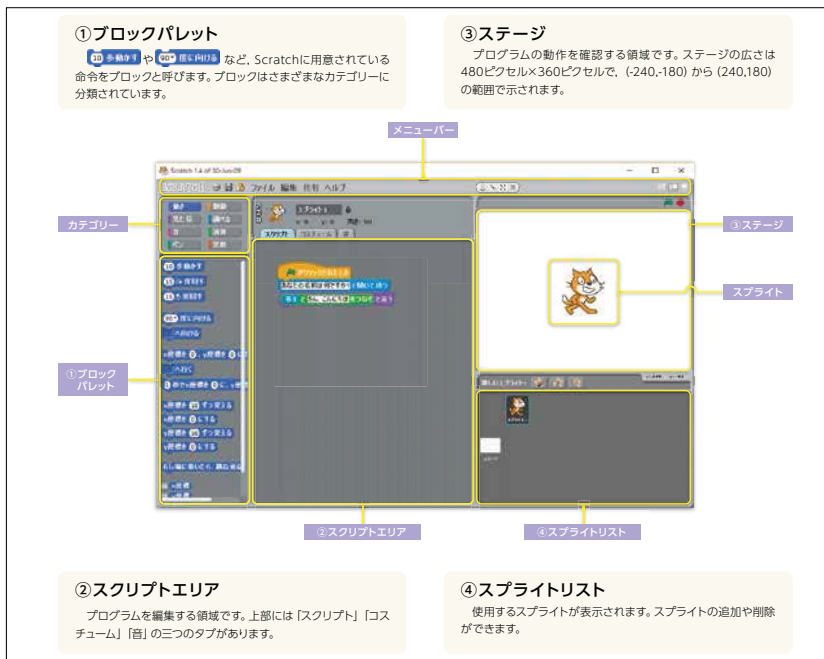
### 評価・改善(1時間)

- ①評価  
実際に使用し、相互評価を行う。
- ②改善  
課題点や改善点を探す。

# プログラミングの指導はどうすればいいのだろう？

## 巻末資料

プログラミングはすべてをマスターしてから問題解決を行うよりは、**問題解決を行いながら足りない部分を調べる**という方法が、社会でもよく使われています。巻末資料を活用することで、限られた授業時数を有効に使うことができます。



教科書 p.281

## QRコンテンツ

QRコードにアクセスすることで、プログラムの解説や、全体の例文を見ることができます。



QRマークのある右下のページのQRコードからコンテンツのページにアクセスできます

見たいタイトルを選択するとコンテンツが表示されます

開隆堂の教科書では、プログラミングに行き詰まっても、巻末資料やQRコンテンツで手厚くフォロー。問題解決に取り組みながらも、無理なく安心してプログラミングについての理解を深めることができます。

「変数」や「条件分岐」なども、実際にプログラムを作成しながらだと理解しやすくなります。また、基本的な考え方なので、他のプログラムに応用することも可能です。

## Scratchのプログラミング2(変数と条件分岐)

ここでは、ネコをクリックすると、クリックされた総回数をネコが言い、スペースキーを押すと回数がリセットされて0に戻るプログラムを作りましょう。

クリックされると回数が一つずつ増える単純なプログラムですが、コンピュータは前の数字を記憶させないと、1を足した次の数字を求めることができません。このようなときに、一時的に情報を記憶させておくものを「変数」といいます。

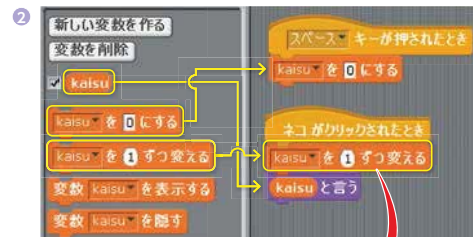
- 最初に、スプライトリストにあるネコをクリックして選択します。  
変数を作るには、オレンジの「変数」カテゴリから「新しい変数を作る」を押します。変数は、いくつでも作ることができますが、名前をつけて識別できるようにします。今回は「kaisu」にしましょう。  
通常は「すべてのスプライト用」として作成します。



変数はプログラムを作るうえで必要となる概念です。Scratchでは変数に日本語で名前をつけることもできますが、使用する機能によっては日本語が正しく表示されないため、半角の英数字で名前をつけるようにしましょう。

- 次に、目的の動きができるようにブロックをつなげていきましょう。  
紫の「～と言う」というブロックの中に「kaisu」と名づけられた丸いオレンジの変数ブロックを入れることで、変数の値をいわせることができます。

**課題** 偶数のときにネコが鳴くようにしましょう。下図のように、「～もし～なら」ブロックの中に三つのブロックを入れて条件を指定します。



「～を～ずつ変える」というブロックは、もともと「1」とかかれています。Scratchでは、白い窓の部分は、自分でかき換え、変数や計算するブロックを入れることができます。

教科書 p.284