

2018（平成30）年度
プログラミング授業実践事例集

発刊にあたって

東京都高等学校情報教育研究会
プログラミング教育専門委員会 能城茂雄

本研究会では平成 27 年度より専門委員会として「プログラミング教育専門委員会」を設置し、高等学校（中等教育学校後期課程を含む）情報科におけるプログラミング教育の情報収集、プログラミングに関するモデル授業の構築等を目指し、研究活動を行ってきました。その成果として、この度、授業プランの提示を目的として、委員それぞれがこれまでにやってきた授業実践を冊子としてまとめました。

プログラミング教育に関しては、以前から話題になっていましたが、ここ数年で大きな変化が見られます。平成 29 年 3 月に公開された小学校学習指導要領、中学校学習指導要領では、コンピュータ等を活用した学習活動の充実、コンピュータでの文字入力等の習得、プログラミング的思考の育成が、総則や各教科等に盛り込まれ、これまで以上に情報活用能力の育成が重要視されるようになりました。小学校段階では、コンピュータに意図した処理を行うように指示することができるということを体験させながら、将来どのような職業に就くとしても、時代を超えて普遍的に求められる力としての「プログラミング的思考」などを育成するものと示されました。自分が意図する一連の活動を実現するために、どのような動きの組合せが必要であり、一つ一つの動きに対応した記号を、どのように組み合わせたらいいのか、記号の組合せをどのように改善していけば、より意図した活動に近づくのか、といったことを論理的に考えていく力が求められています。また、中学校段階では、技術・家庭科の技術分野“D 情報の技術”において、生活や社会における問題を、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って解決する活動を通じて、情報通信ネットワークの構成と、情報を利用するための基本的な仕組みを理解し、安全・適切なプログラムの制作、動作の確認及びデバッグ等ができることが明記されるなど、プログラミングについての扱いが大きく変わりました。高等学校段階においても平成 30 年 3 月末に、高等学校の学習指導要領が告知され、全ての高校生が学ぶ必履修教科である情報 I において、コンピュータとプログラミングを学ぶことになりました。

このような状況を踏まえ、本研究会では定期的に専門委員会を開催し、情報交換を重ねてきました。高等学校で新学習指導要領が実施されるのはまだ先になりますが、委員たちは現行学習指導要領の中で次を見据え、プログラミングの授業研究を重ねてきました。この冊子にまとめられた内容は、委員がそれぞれの学校で行ってきた実践の集合です。一人でも多くの先生方の今後の参考になることを期待します。

最後になりますが、冊子作成にあたった委員たちは、文部科学省をはじめとする各資料や全国各地で行われている実践などの情報収集、自校での実践など、多忙な中にさまざまな取り組みを行ってくれました。この場を借りて御礼を申し上げます。

■ もくじ

▶ 高校実践事例

実践事例①：ブラウザで行うプログラミング基礎 (能城 茂雄)	-----	P.02
実践事例②：R を用いたプログラミング授業実践 (橘 孝博・八百幸 大)	-----	P.10
実践事例③：Alice を利用した感情表現活動 (神藤 健朗)	-----	※
実践事例④：micro:bit を用いたプログラミング授業実践 (塚本 初恵)	-----	P.20
実践事例⑤：Monaca を用いたプログラミング授業実践 (小原 格)	-----	P.26
実践事例⑥：映像教材を利用したプログラミング教育の授業実践 (飯田 秀延)	-----	P.32
実践事例⑦：プログラミングに興味を持たせる授業実践 (稲垣 俊介)	-----	P.36
実践事例⑧：ドリトルを用いたプログラミング授業実践 (山本 博之)	-----	P.44
実践事例⑨：micro:bit を用いたプログラミング授業実践 (千葉 緑)	-----	※

▶ 大学での事例

大学におけるプログラミング教育 ～理工学系学部の初年次導入科目を例に～ (赤澤 紀子)	-----	P.54
大学におけるプログラミング教育 ～情報専門学科を例に～ (角田 博保)	-----	P.58

■ 構成

- 内容説明
- 各時間の内容
- アンケート

※ 実践事例③、実践事例⑨に関しては、web 上では非公開となっています

ブラウザで行うプログラミング基礎

能城 茂雄

■ 内容説明

本単元では、問題解決の処理手順をアルゴリズムとして、論理的にまとめることの必要性を理解させるために、簡単なアルゴリズムを具体的にプログラムとして、表現させることを理解させる。なお、本内容を行う前に、プログラムに対しての苦手意識を持たせない試みとして、アルゴリズム体験ゲーム「アロゴロジック」や、教育用プログラミング言語「ドリトル」を使用した授業を1時間程度行うことで導入をスムーズにしている。本校では、アルゴリズムを利用した授業を1時間、ドリトルを利用した授業を1時間、JavaScriptを利用した授業を5時間行っている。

■ 授業の実施情報

学校名	実施科目	実施学年
	情報の科学	4年生（高校1年生相当）
時間	1授業あたりの実施人数	使用機材
50分	40人（教員は1名）	パソコン
使用言語	使用ソフト	OS
JavaScript	ブラウザ	Windows10

■ 単元の指導計画

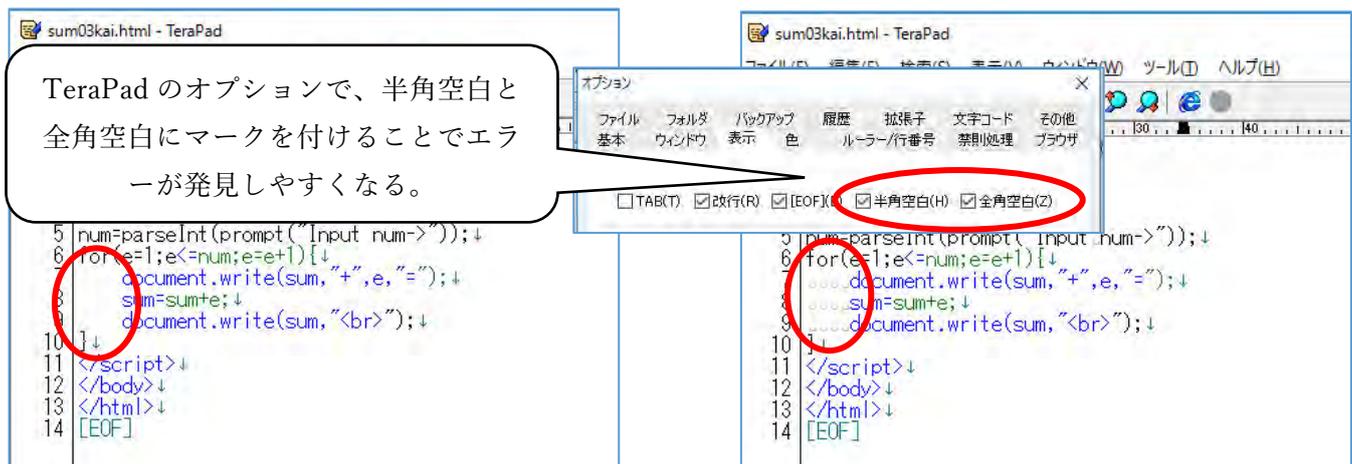
時間	学習活動・学習内容	学習目標
第1時	<p>●アルゴリズムの基礎</p> <p>プログラミングの基本的となる論理的思考（アルゴリズム）を習得させる。</p>	<ul style="list-style-type: none"> ・プログラミングの基本的な考えを理解する。 ・ブロックによる命令の意味を理解する。 ・繰り返し構造を使って命令が繰り返されることを理解する。
第2時	<p>●プログラム概論</p> <p>日本語でかけるプログラミング言語であるドリトルを使用し、コードを書く体験をさせる。コンピュータは人間が命令した内容でしか動作しないことを考えさせる。</p>	<ul style="list-style-type: none"> ・コードを書くことによって、コンピュータが動作することを理解させる ・命令によって、キャラクターの動きが変化することや、文法を間違うと、命令が意図した命令と異なる場合に動作しないことを理解する。
第3時	<p>●プログラムⅠ</p> <p>プログラムの作成方法を身に付けさせる。コンピュータに文字を表示する方法を習得させる。</p>	<ul style="list-style-type: none"> ・プログラムはコンピュータへの指示であることを理解する。 ・プログラムには文法などのルールがあることを理解する。
第4時	<p>●プログラムⅡ</p> <p>アルゴリズムの基本構造を意識し、処理手順を図で示す方法を習得させる。繰り返し命令におけるルールを確認にし、ループ処理を使ったプログラムを習得させる。</p>	<ul style="list-style-type: none"> ・フローチャートで示された内容がプログラムと一致することを理解する。 ・同じ内容である複数の命令を実現するためには、繰り返し命令を用いたほうが良い点を理解させる。

<p>第5時</p>	<p>●プログラムⅢ 繰り返し命令を用いて、総和を求めるプログラムを作成する。また作成したプログラムを参考にして、入力に対して結果が出力されるようにプログラムを改造する方法を習得させる。</p>	<ul style="list-style-type: none"> ・前時の復習と確認を兼ねて繰り返し命令を使用したプログラムを発展させる。 ・サンプルプログラムを組み合わせ、プログラムを作成する力を身に付けさせる。 ・意図したとおりに、画面表示を行う場合のルールや、方法を理解させる。
<p>第6時</p>	<p>●プログラムⅣ 条件によって、動作が変わるプログラムを作成する方法を習得させる。また大量のデータを扱うための配列の使用方法を習得させる。</p>	<ul style="list-style-type: none"> ・条件式によって、プログラムが動作を変えることを理解させる。 ・変数を大量に用意するのではなく、配列を利用し、繰り返し命令を用いてデータ処理ができることを理解させる。
<p>第7時</p>	<p>●プログラムⅤ これまで学んできた基本的なアルゴリズムを組み合わせて、二重ループやデータの入れ替え、ソートやサーチなどのアルゴリズムを作成する方法を習得させる。</p>	<ul style="list-style-type: none"> ・アルゴリズムを組み合わせることで、プログラムが作成されていることを理解させる。 ・データを整理するための考え方や、ソート・サーチなどのアルゴリズムについて考えさせる。

■ 環境設定・使用ソフトの準備

ブラウザで行うプログラミングでは、特別なソフトウェアを用意しなくても授業を行えることが魅力のひとつである。具体的には、Windows 標準のブラウザ(Internet Explorer)とテキストエディター (メモ帳) があればどのようなコンピュータ教室でも授業が行える。

標準ブラウザとメモ帳でも授業が行えるが、フリーソフトである。TeraPad を使用することで、半角スペースを全角にしてしまうエラー (Bug) や命令の間違いを発見しやすくなる。



TeraPad は、以下のサイトから無料でダウンロードでき、インストール不要版もダウンロードできる。

TeraPad

<http://www5f.biglobe.ne.jp/~t-susumu/library/tpad.html>

第1時 アルゴリズムの基礎

■ 本時の内容・目標

- ・プログラムの基本的な考え方を理解する。
- ・ブロックによる命令の意味を理解する。
- ・繰り返し構造を使って命令が繰り返されることを理解する。

■ 準備物

- ・教科書・ノート・筆記用具（生徒側に特に必要な物はない）
- ・事前に、アルゴリズム体感ゲーム・アルゴロジック（<https://home.jeita.or.jp/is/highschool/algo/>）にアクセスできることを確認しておく。

■ 授業の流れ

① 説明

1. 次回以降に学習するプログラミングの前に、アルゴリズムの重要性を体験する。

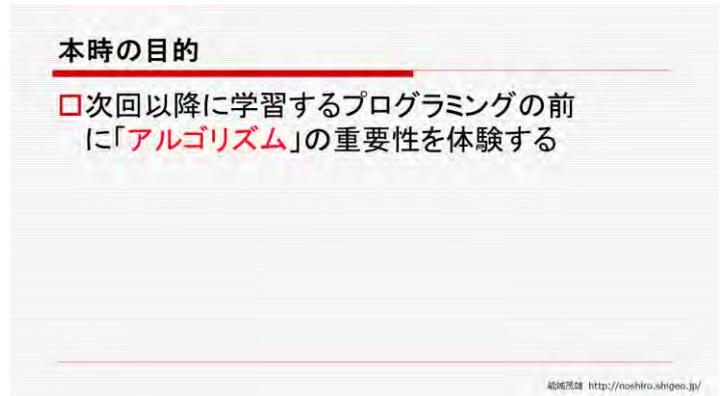


図1 「本時の目的」を明確にする

② アルゴリズムについての説明

1. アルゴリズムは、コンピュータを使って目的を達成するための処理手順であることを説明する。
2. アルゴリズムは、コンピュータで計算を行うときの計算方法や、物事を考えるときの「やり方」であることを理解させる。

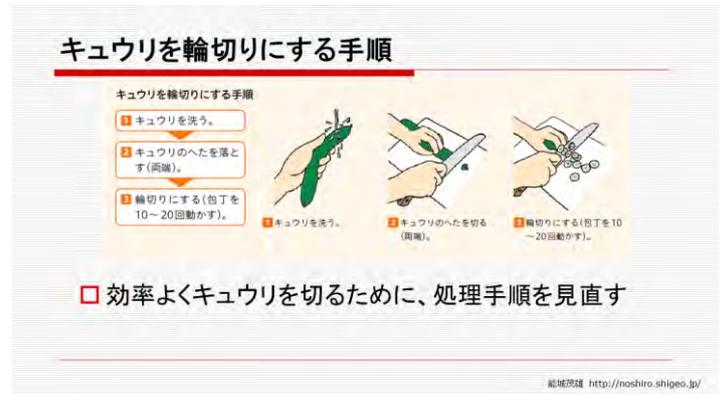


図2 キュウリを輪切りにする手順

③ 例示：キュウリを輪切りにする手順

1. キュウリを輪切りにする手順を例として、手順を工夫すること、処理手順が変わることを理解させる。生徒に身近な例としては、インスタントラーメンを作る手順なども紹介し、手順を考えて物事に取り組まないと無駄が生じることも理解させるとよい。



図3 キュウリを輪切りにする手順の改良

④ アルゴロジックの導入

1. アルゴロジックの操作方法を説明する。このとき、説明しすぎないように注意すること。
2. 結果を授業の最後に確認するので、授業終了前10分前に、チェックを受けること。

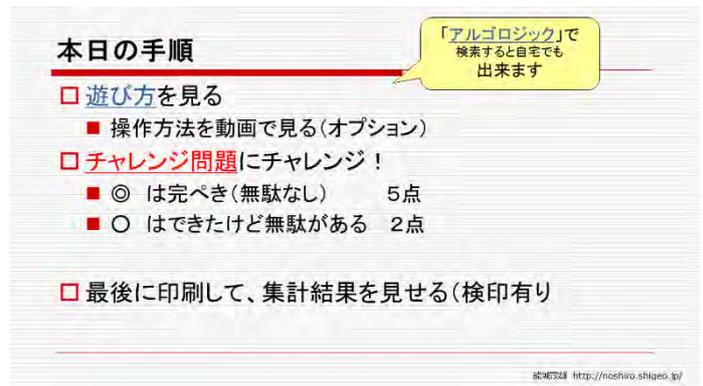


図4 評価についての具体的な指示

⑤ 本時のまとめ

1. コンピュータにとって、アルゴリズムが重要であることが理解できたか。

■ 留意点・ポイント

- ・プログラムについての学習は、苦手意識を持たせてしまうと生徒には、苦痛でしかない。そのため、本時では、プログラムの学習が楽しいと思えるための導入として、アルゴリズム体験ゲーム「アルゴロジック」を導入する。
- ・アルゴロジックは、興味を持つ生徒にとっては、非常に楽しめるサイトである。授業として導入する安倍には1単位時間(50分)の中で、説明と実際に問題に取り組む時間を確保する程度が良い。2単位時間だと、問題が高度になり徐々に生徒に飽きがみられる。
- ・アルゴロジックを通じて学ばせたい、もうひとつのポイントは、プログラムの基本構造のひとつである「繰り返し」の概念を理解させることである。アルゴロジックのブロックで、繰り返しを実現する方法を学習した生徒は、プログラミング言語による式で記述する命令も理解しやすくなるため、導入として適している。

■ 参考資料

JEITA アルゴリズム体験ゲーム・アルゴロジック
<https://home.jeita.or.jp/is/highschool/algo/>

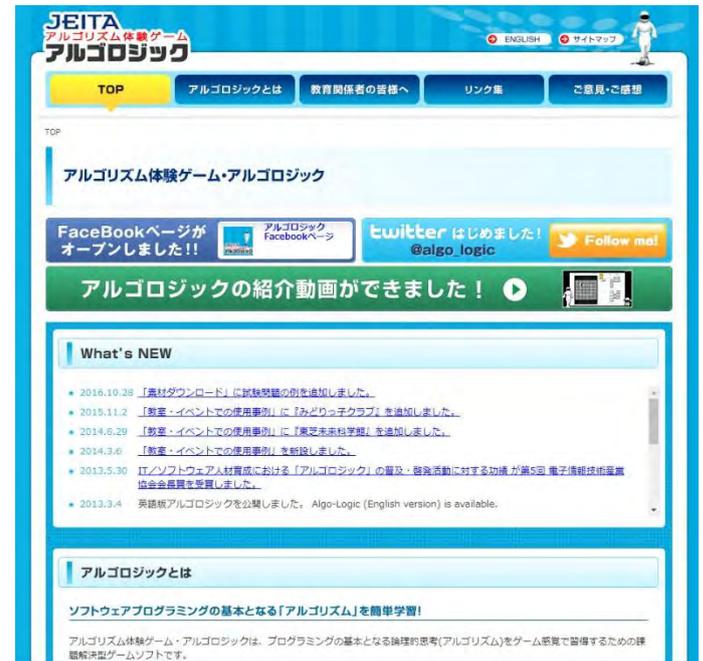
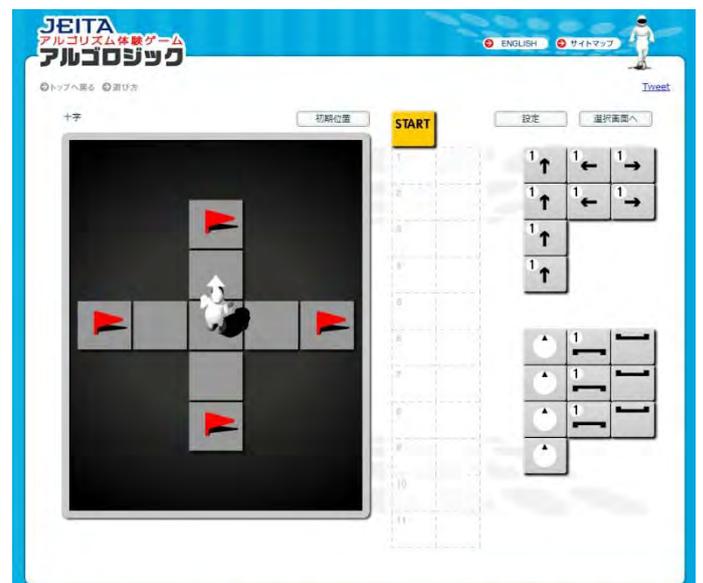


図5 アルゴロジックトップページ



第2時 プログラム概論（宝物拾いゲームの作成）

■ 本時の内容・目標

- ・プログラムの基本的な考え方を理解する。
- ・ソフトウェアがどのように作られているかを体験する。

■ 準備物

- ・事前に、教育用プログラミング言語「ドリトル」をオンライン上で実行するか、インストール版（管理者権限不要）を使用するか決定しておく。
- ・インストール版を使用する場合には、生徒個々にファイルをダウンロードさせるのではなく、事前に配布フォルダー等に、ダウンロードしておくとうよい。

■ 授業の流れ

① 宝物拾いゲームでソフトウェアの仕組みを学ぶ

1. 前回学習したアルゴリズムの経験を活かし、簡単なゲームを作成することを説明する。
2. 作成するゲームは、「かめた」というキャラクターが宝物（フラグ）をゲットするゲームであることを伝える。生徒の反応によっては、完成したゲームを見せてもよい。
3. 作成するゲームは、簡単なものであるため、自分で改造したいときは自由に改造していいことを伝える。

② ドリトルの画面

1. 配布フォルダーなどから入手（または事前にコンピュータにコピー）した ZIP ファイルを解凍させる。解凍後、dolittle.jar を実行する。（dolittle_tab.jar という実行ファイルもある。こちらは複数ウィンドウではなく、をタブで実行画面と表示画面を切り替える。）
2. ドリトルを起動したときの初期画面について説明する。プログラムを記述する画面と実行画面の違いを理解させる。

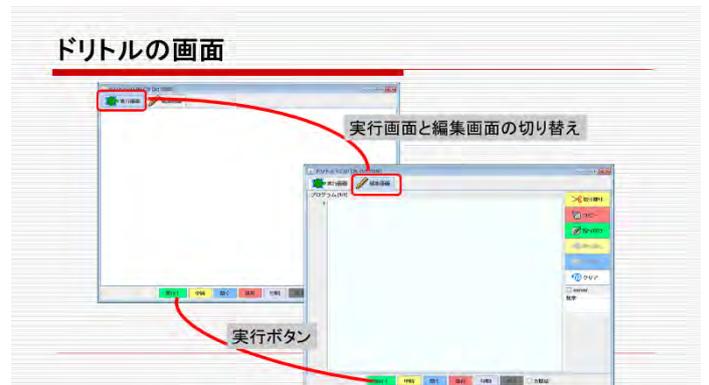


図2 ドリトルの実行画面と編集画面

③ キャラクターを登場させる。

1. まず、ドリトルの基本キャラクターである「かめた」を登場させるために、プログラムを1行記入させ実行させます。生徒によっては、「！」などの記号を入力できない場合があるので、わからないときは、隣の人と助け合うように声掛けをします。

④ 操作ボタンを作成する。

1. 2行目のプログラムとして、「かめた」を操作するボタンを作成します。このとき、実行ボタンを押して、ボタンが登場することは確認してもよいが、左ボタンは押さないように指示します。

⑤ 操作ボタンを押すと？

1. 作成したボタンを押したら「かめた」がどのような動きをするか生徒に考えさせます。多くの生徒は「かめた」が左に動くと予想します。実際には、「かめた」はまったく動きません。ここでは、左ボタンを作成したが、左ボタンを押しても、何がどのように動くかは定義していないことを理解させます。

⑥動作の定義

1. 作成したボタンに動作を定義します。
2. スライドでは、「かめた」が左周りに 30 度回るように命令を書いています。生徒に自由に改造してもいいよと念を押すことで、90 度や、360 度などを指定する生徒が出てくるので、その場合の動作などにも触れて、コンピュータは命令したとおりに動作することを体感させます。

⑥右ボタンの作成と動作の定義

1. 左ボタンを例にして、右ボタンを作成させます。動作についても忘れずに記述させます。
2. 単なるプログラムの写経とならないように、考えさせる行動も取り入れることで、自分らしさをプログラムの改造という観点で表現させます。

⑦かめたを動かす。

1. 「かめた」を歩かせるための命令を記述します。
2. プログラムの数値については、特に説明せず、生徒が自主的に数値をいじることで、「かめた」の動作が変わることに気付かせます。

⑧宝物を出現させる。

1. 「かめた」が拾う宝物（チューリップ）を出現させる。
2. プログラム中の 100 100 は数学で学習した座標の考え方が適用されていることを説明し、複数の宝物を出現させる方法を考えさせる。

⑧宝物と衝突したら宝物が消える。

1. プログラムを書かせる前に、「かめた」が宝物と衝突しても何も起こらないことを説明する。
2. なぜ、何も起こらないかを考えさせ、ボタンを押したときに動かなかったときと同じ現象であることに気付かせる。
3. 応用として、宝物に「かめた」がぶつかったときに、「かめた」が消えるにはどうすればよいか考えさせるとよい。

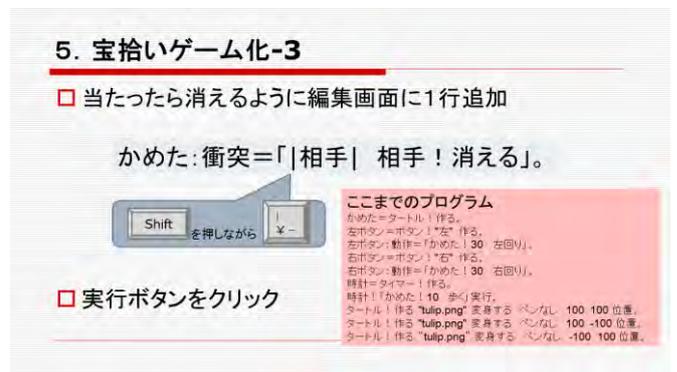


図9 衝突時の動作を定義

■ 留意点・ポイント

- ・ドリトルによるプログラム作成は、日本語で記述できるため文字入力時の半角・全角を気にしなくてよいというメリットがある。本時で作成した「宝さがしゲーム」は、参考資料にある「1時間で学ぶソフトウェア」から引用している。生徒の進捗によっては、点数を数えるようにするなど要素を増やすことも可能である。

第3時 ブラウザを使ったプログラム（プログラムⅠ）

■ 本時の内容・目標

- ・コンピュータにとってプログラムがどのようなものか体験する。
- ・JavaScript という言語を使ってみる。

■ 準備物

- ・教科書・ノート・筆記用具（生徒側に特に必要な物はない）
- ・TeraPad（公式サイトから、インストール不要版を共有フォルダなどで提供、メモ帳でも可）

■ 授業の流れ

① 説明

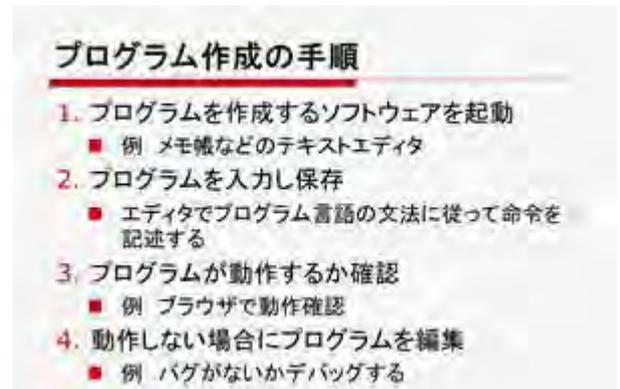
1. プログラムはコンピュータに対する指示であることを説明する。
2. プログラムを作成するための手順を確認する。

② プログラム作成上の注意と練習

1. 作成したプログラムを保存する時の注意を説明する。
（所定の場所でないと、環境復元機能によって、作成したプログラムが消えてしまうことを理解させる。）
2. フォルダ（js）を作成させる。
3. サンプルプログラムの入力

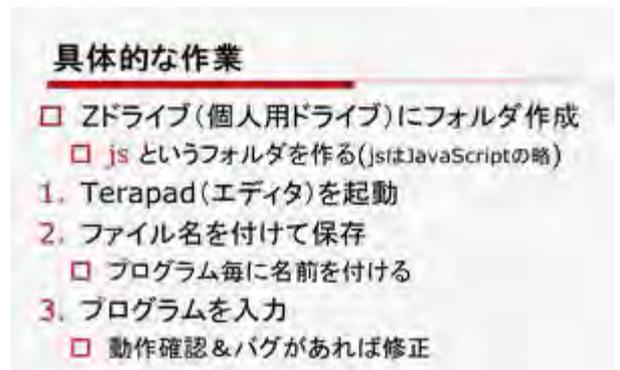


```
moji.html - TeraPad
ファイル(E) 編集(E) 検索(S) 表示(V) ウィンドウ(W) ツール(T) ^
1 <html>↓
2 <body>↓
3 <script>↓
4 document.write("MITAKA");↓
5 </script>↓
6 </body>↓
7 </html>↓
8 [EOF]
```



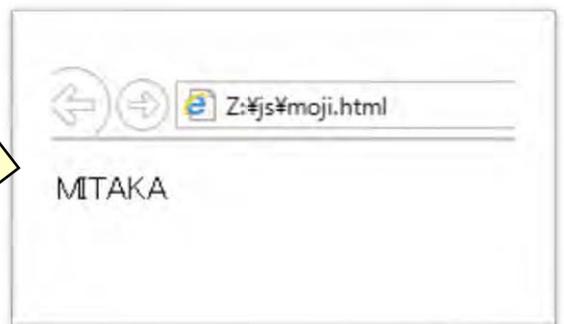
プログラム作成の手順

1. プログラムを作成するソフトウェアを起動
 - 例 メモ帳などのテキストエディタ
2. プログラムを入力し保存
 - エディタでプログラム言語の文法に従って命令を記述する
3. プログラムが動作するか確認
 - 例 ブラウザで動作確認
4. 動作しない場合にプログラムを編集
 - 例 バグがないかデバッグする



具体的な作業

- Zドライブ（個人用ドライブ）にフォルダ作成
 - js というフォルダを作る（jsはJavaScriptの略）
- 1. Terapad（エディタ）を起動
- 2. ファイル名を付けて保存
 - プログラム毎に名前を付ける
- 3. プログラムを入力
 - 動作確認&バグがあれば修正

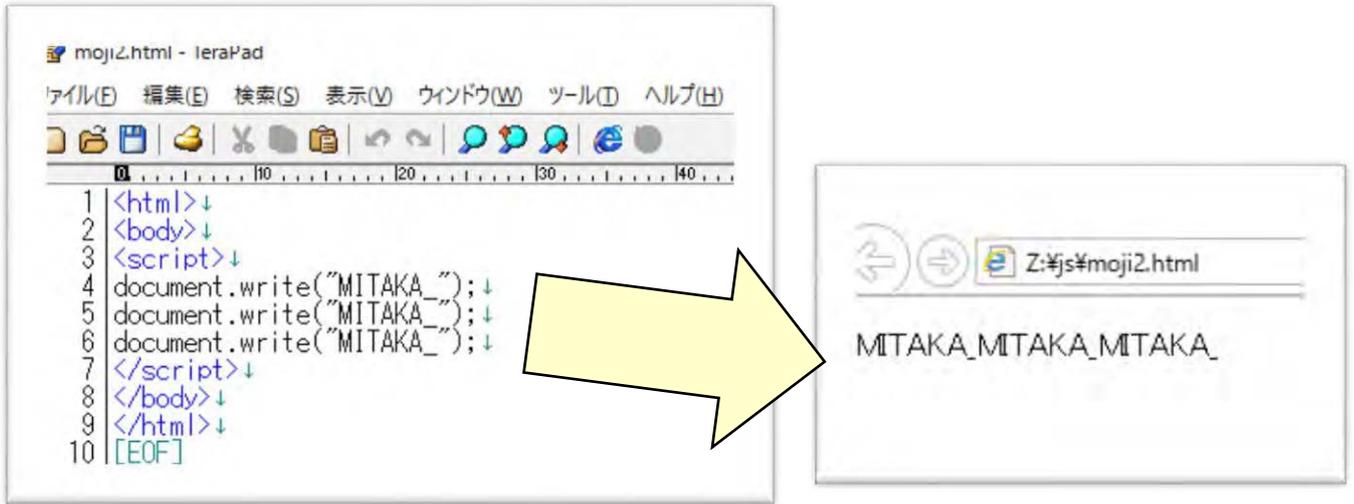


③ 実行

1. 作成したプログラムにバグがないか、意図したとおりに動作するか確認させる。
2. 作業が早い生徒には、適宜改造（表示する文字列を変更する）を進める。
3. 説明では、わざとダブルコーテーション（"）を付けないエラーなどを実演し、プログラムの誤りを修正する方法も説明する。

④ 2度目のプログラムの作成

1. 画面に名前を表示させるプログラムを発展させ、名前を3個表示するプログラムを作成させる。
2. 1度目のプログラム作成で、詳細な作成方法は指示済みなので、今回は説明をほとんど行わず、プログラムの作成方法が身についているかを確認する。

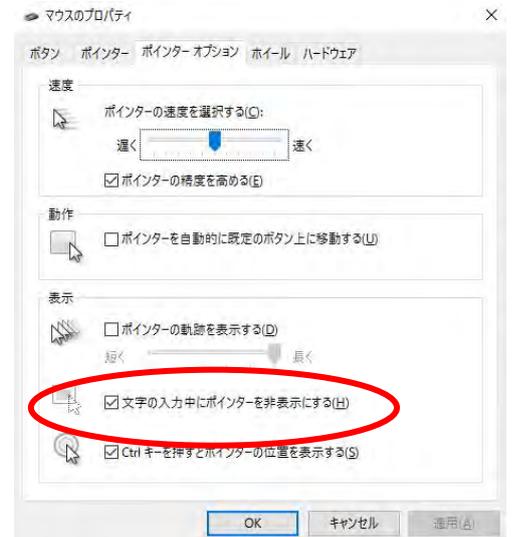


④ 本時のまとめ

1. コンピュータにとって、プログラムがどのようなものであるかを確認する。

■ 留意点・ポイント

- ・プログラムは、わかる人にとっては簡単だが、わからない人にとっては非常に難解である。そのため、本授業は、初回として誰にでもできるように極端に学習内容を簡易にしている。
- ・中間モニターなどを使用して、指導する際にマウスポインターを見失わないように、「Ctrl キーを押すとポインターの位置を表示する (S)」を有効にするとよい。
- ・プログラムの作成では、表示する文字列であっても日本語は使用させない (日本語入力のオン・オフをさせない) 特に標準テキストエディターであるメモ帳を使用している場合は、半角スペースと全角スペースを使用したことで、プログラムが動作しないバグが発生しやすいので、日本語入力をオフのまままで入力することを、明示的に説明をする。
- ・セミコロンやダブルコーテーション、かっこ (「」などを入力できない生徒がいることが想定されるので、記号の入力方法も説明したほうがよい。
- ・本時では、JavaScript によるプログラミング学習の導入を扱ったが、次時以降授業計画に基づき、学習を深めていくことが想定される。



■ 参考資料

はじめようプログラミング ～「アルゴリズム」「1時間で学ぶソフトウェアの仕組み」を経て、Java Script へ
<https://www.wakuwaku-catch.net/授業事例-バックナンバー/> 【授業事例37】 参照

Rを用いたプログラミング授業実践

橘 孝博・八百幸 大

■ 内容説明

実験や調査で収集したデータを活用するために、データの統計的な処理の手法やデータの加工、整理の技能を身に付け、その結果をもとに現象・事象を理解できるようになることが望まれる。本単元では、統計解析ソフトウェア R と、これをプログラミングする R 言語を用いて、統計処理の手法やデータの加工やシミュレーションなどを実施するためのプログラミングについて学ぶ。なお、数学 I で扱われる基本統計量やヒストグラム、散布図などのグラフについては既習とする。

■ 授業の実施情報

学校名	実施科目	実施学年
	社会と情報	高校2年生(2学期)
時間	1授業あたりの実施人数	使用機材
50分	20人(教員は1名)	パソコン
使用言語	使用ソフト	OS
R	R	Windows7

■ 単元の指導計画

時間	学習活動・学習内容	学習目標
第1時	<p>●Rの基本操作</p> <p>Rにおける入力やファイル保存について見た上で、関数電卓的な利用法を学ぶ。次に、Rの基本的な関数を学び、Rを用いた簡単な統計処理について理解する。</p>	<ul style="list-style-type: none"> ・Rの基本操作を理解する。 ・Rの基本的演算(四則演算、累乗など)を理解する。 ・Rの関数の考え方や基本的な関数を理解する。
第2時	<p>●Rでのデータ操作</p> <p>はじめにこれから使用するデータの型やデータ構造について理解する。さらにRにおけるデータの読み込み、書き出しを理解し、データの加工やアクセスする方法について学ぶ。</p>	<ul style="list-style-type: none"> ・データの型やデータ構造を理解する。 ・データの読み込み、書き出しを理解する。 ・データの抽出、並べ替え、表の結合などによるデータの再構成の方法を理解する。
第3時	<p>●Rでのプログラミング(1)</p> <p>Rでは自分で手続きを記述して関数を定義することができる。そのために必要な演算子や構文を学び、これを用いた簡単な計算を実行する関数を作成する。</p>	<ul style="list-style-type: none"> ・簡単なユーザー定義関数を作成できる。 ・比較演算子、論理演算子を理解する。 ・条件分岐の構文を理解する。
第4時	<p>●Rでのプログラミング(2)</p> <p>第3時で学習したことを踏まえ、簡単なシミュレーションを行う関数を作成する。また、Rでは統計解析した結果を表現するためのグラフ機能がある。これを用いてグラフを作成する。</p>	<ul style="list-style-type: none"> ・繰り返しの構文を理解する。 ・簡単なシミュレーションができる。 ・基本的なグラフを作成できる。

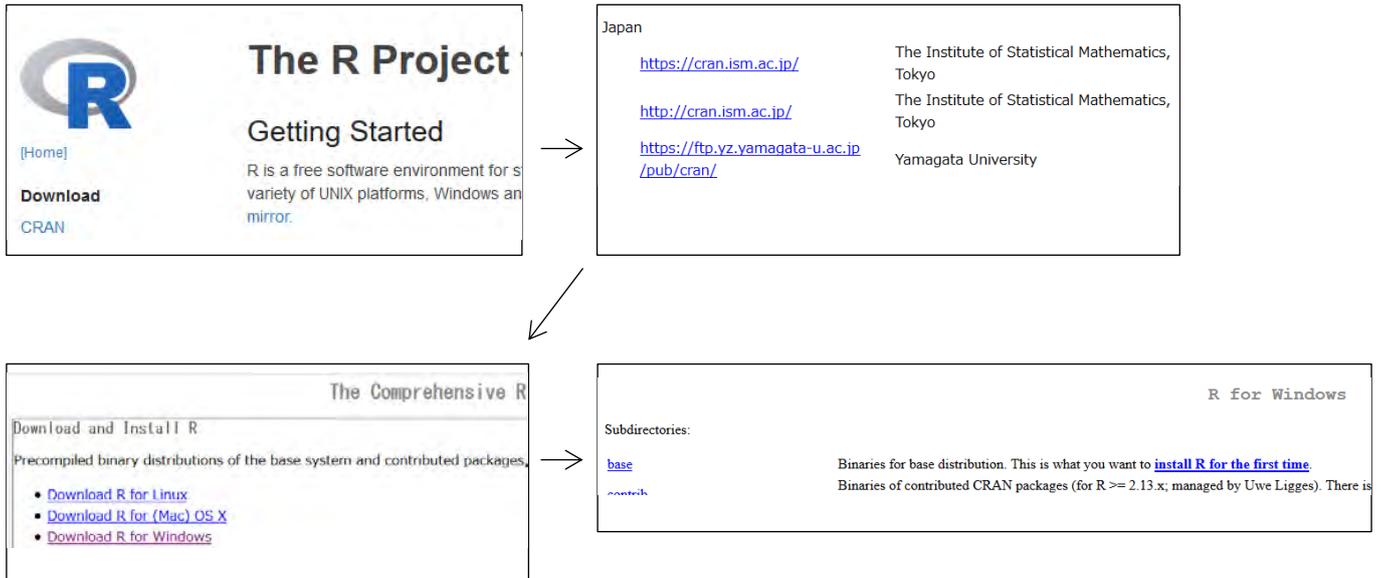
■ 環境設定・使用ソフトの準備

統計処理ソフトウェアRのプログラムは、以下のサイトから無料でダウンロードできる。

The R Project for Statistical Computing

<https://www.r-project.org/>

Rをダウンロードできるサイトは全世界にあるので、上記サイトから適切なダウンロードページへ行き、OSにあったプログラムをダウンロードする（本校では、R-3.3.2 がインストールされている）。



Windows 版を初めてインストールする場合は、基本のインストールファイルを使用する。ファイルをダウンロードしたらこれを実行し、指示に従ってインストールを進めていく。インストールが完了すると、デスクトップに実行ファイルのショートカットアイコンが現れるので、これをダブルクリックすることでソフトを起動することができる。

また、Rなどの言語をインターネット上で実行できるサイトもある。データファイルの読み込み・書き出しができないなど機能に制限がある場合がほとんどだが、簡単なシミュレーションは可能である。

ideone.com

<https://ideone.com/>

第1時 Rの基本操作

■ 本時の内容・目標

- ・ Rの基本操作や基本的演算（四則演算、累乗など）を理解する。
- ・ Rの関数の考え方や基本的な関数を理解し、簡単な統計処理が実行できるようにする。

■ 準備物

- ・ 授業ワークシート（本時に使用する数式や関数を掲載）

■ 授業の流れ

① R Consoleの起動と四則演算による入力練習

1. スタートメニューなどからR Consoleを起動する。
2. あらかじめ用意した数式を入力させる。
3. Rで算術演算子を用いて加減乗除や累乗などの計算ができることを確認する。

A screenshot of the R Console window showing several arithmetic operations. Lines connect the operations to labels in boxes on the right:

- `> 3+4` is connected to a box labeled "加算" (Addition).
- `[1] 7` is the output of the addition.
- `> 3-4` is connected to a box labeled "減算" (Subtraction).
- `[1] -1` is the output of the subtraction.
- `> 3*4` is connected to a box labeled "積算" (Multiplication).
- `[1] 12` is the output of the multiplication.
- `> 3/4` is connected to a box labeled "除算" (Division).
- `[1] 0.75` is the output of the division.
- `> 3^4` is connected to a box labeled "累乗" (Power).
- `[1] 81` is the output of the power operation.

② 簡単な組み込み関数の利用

1. Rにはさまざまな組み込み関数が用意されていることを説明する。
2. 平方根を出力する関数 `sqrt`、数列の総和を出力する関数 `sum` を例として、組み込み関数を用いたコマンドを入力させる。
3. 組み込み関数で求めた値が正しいことを確認する。

A screenshot of the R Console window showing built-in functions. Lines connect the functions to labels in boxes on the right:

- `> sqrt(2)` is connected to a box labeled "2の平方根" (Square root of 2).
- `[1] 1.414214` is the output of `sqrt(2)`.
- `> sum(1, 2, 3, 4, 5)` is connected to a box labeled "1から5までの和" (Sum from 1 to 5).
- `[1] 15` is the output of `sum(1, 2, 3, 4, 5)`.
- `> 2^(1/2)` is also connected to the "2の平方根" box.
- `[1] 1.414214` is the output of `2^(1/2)`.
- `> 1+2+3+4+5` is also connected to the "1から5までの和" box.
- `[1] 15` is the output of `1+2+3+4+5`.

③ 簡単なデータの作成

1. Rにおけるデータ構造の代表としてベクトルについて説明する。
2. 計算結果やデータを格納するための変数について説明する。
3. あらかじめ用意したコマンドを入力させる。
4. それぞれのコマンドの出力結果を確認する。

A screenshot of the R Console window showing vector creation and operations. The code and its output are as follows:

```

> c(1, 2, 3, 4, 5) #1から5までの数列からなるベクトル
[1] 1 2 3 4 5
> vec1 <- c(1, 2, 3, 4, 5) #上のベクトルを変数vec1に格納
> sum(vec1) #vec1の要素の総和
[1] 15
> vec1*2 #vec1の要素をそれぞれ2倍してできるベクトル
[1] 2 4 6 8 10
> vec1^2 #vec1の要素をそれぞれ2乗してできるベクトル
[1] 1 4 9 16 25
> |
    
```

④ Rでの簡単な統計処理

1. 統計における代表値（平均値、中央値、最頻値）について説明する。
2. あらかじめ用意したコマンドを入力させる。

```

> data1 <- c(7,6,4,8,9,6,5,7,8,5,8,7,10,6,7,7,8,9,4)
> sum(data1)
[1] 131
> mean(data1)
[1] 6.894737
> median(data1)
[1] 7
> table(data1)
data1
 4  5  6  7  8  9 10
 2  2  3  5  4  2  1
> max(data1)
[1] 10
> min(data1)
[1] 4
> length(data1)
[1] 19
> |
    
```

Callouts in the image:

- sum(data1) → 総和
- mean(data1) → 平均
- median(data1) → 中央値
- table(data1) → 度数分布表を作成する (最頻値を求めるためのもの)
- max(data1) → 最大値
- min(data1) → 最小値
- length(data1) → ベクトルの長さ (要素の個数)

3. 分散、標準偏差、相関係数について説明する。
4. あらかじめ用意したコマンドを入力させる。

```

> data1 <- c(1,2,3,4,5)
> (data1 - mean(data1))^2
[1] 4 1 0 1 4
> sum((data1 - mean(data1))^2)/length(data1)
[1] 2
> var(data1)
[1] 2.5
> sd(data1)
[1] 1.581139
> data2 <- c(1,1,2,2,3)
> cor(data1,data2)
[1] 0.9449112
> |
    
```

Callouts in the image:

- $(data1 - mean(data1))^2$ → data1 の各要素に対して偏差の2乗を求める
- $sum((data1 - mean(data1))^2)/length(data1)$ → data1 の標本分散
- var(data1) → Rで用意されている分散や標準偏差の組み込み関数は不偏分散、不偏標準偏差である
- sd(data1) → Rで用意されている分散や標準偏差の組み込み関数は不偏分散、不偏標準偏差である
- cor(data1,data2) → data1 と data2 の相関係数

⑤ R Console の終了

■ 留意点・ポイント

- ・最頻値や標本分散を求める組み込み関数は実装されていない。必要な場合は、後述するように各自で関数を定義する（ユーザー定義関数）。

第2時 Rでのデータ操作

■ 本時の内容・目標

- ・データの型やデータ構造を理解する。
- ・データの読み込み、書き出しを理解する。
- ・データの抽出、並べ替え、表の結合などによるデータの再構成の方法を理解する。

■ 準備物

- ・授業ワークシート（本時に使用する数式や関数を掲載）
- ・配布用データファイル（CSV形式）

■ 授業の流れ

① Rでのデータ形式（データ型、データ構造）

1. コンピュータではデータの値に数値、文字列などの型を持たせることを説明する。
2. あらかじめ用意したコマンドを入力させる。
3. 与えられたデータの型を確認する。

```
R Console
> vec1 <- c(1,2,3) ; vec2 <- c("1","2","3")
> vec1 ; vec2
[1] 1 2 3
[1] "1" "2" "3"
> mode(vec1) ; mode(vec2)
[1] "numeric"
[1] "character"
> sum(vec1)
[1] 6
> sum(vec2)
sum(vec2) でエラー: 引数 'type' (character) が不正です
> |
```

② Rでのデータ構造（ベクトル、行列、データフレーム）

1. コンピュータではデータの値を格納する構造を決めること説明する。
2. あらかじめ用意したコマンドを入力させる。
3. 与えられたデータの構造を確認する。

```
R Console
> vec <- c(1,2,3,4,5,6)
> vec
[1] 1 2 3 4 5 6
> mat <- matrix(c(1,2,3,4,5,6),2,3)
> mat
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> df <- data.frame(name=c("aaa","bbb","ccc","ddd"), point=c(2,3,5,7))
> df
  name point
1  aaa     2
2  bbb     3
3  ccc     5
4  ddd     7
> df$point
[1] 2 3 5 7
> |
```

③ データの読み込み

1. あらかじめ作業ディレクトリにデータファイルを保存しておく。
2. コマンドを入力し、データファイルの内容を読み込み、オブジェクト df に代入する。
3. df の内容を確認する。

```
R Console
> df <- read.csv("data1.csv")
> df
  sex region name Japanese Math English
1  1  tokyo  aoki      23    56    65
2  2  tokyo  ehara     43    45    44
3  2  saitama kato      11    97    76
4  1  chiba   kayama    56    32    23
...
26 2  tokyo  hoshino    55    48    61
27 1  tokyo  mama      43    83    63
28 1  saitama miyajima 56    81    66
29 2  chiba   yano      77    79    67
30 1  chiba   watanabe  34    89    70
> |
```

④ データの加工、抽出

1. dfの要素、行、列を抽出する。

```

> df[1,2]
[1] "tokyo"
> df[1,]
  sex region name Japanese Math English
1   1  tokyo aoki      23    56     65
> df[,2]
 [1] "tokyo"      "tokyo"      "saitama"    "chiba"      "chiba"      "kanagawa"
 [7] "saitama"    "saitama"    "USA"        "kanagawa"   "kanagawa"   "tokyo"
[13] "kanagawa"   "tokyo"      "tokyo"      "chiba"      "tokyo"      "tokyo"
[19] "USA"        "tokyo"      "tokyo"      "saitama"    "chiba"      "chiba"
[25] "saitama"    "tokyo"      "tokyo"      "saitama"    "chiba"      "chiba"
> df$region
 [1] "tokyo"      "tokyo"      "saitama"    "chiba"      "chiba"      "kanagawa"
 [7] "saitama"    "saitama"    "USA"        "kanagawa"   "kanagawa"   "tokyo"
[13] "kanagawa"   "tokyo"      "tokyo"      "chiba"      "tokyo"      "tokyo"
[19] "USA"        "tokyo"      "tokyo"      "saitama"    "chiba"      "chiba"
[25] "saitama"    "tokyo"      "tokyo"      "saitama"    "chiba"      "chiba"
> |

```

2. dfから与えられた条件を満たす行を抽出する。

```

> df[df$sex == 1, ]
  sex region name Japanese Math English
1   1  tokyo aoki      23    56     65
4   1  chiba kayama    56    32     23

27  1  tokyo  mama     43    83     63
28  1  saitama miyajima 56    81     66
30  1  chiba watanabe  34    89     70
> df[df$Math >= 90, ]
  sex region name Japanese Math English
3   2  saitama kato      11    97     76
23  1  chiba hiruma     87    91     87
>

```

3. dfの中の3教科合計点のベクトルを作成し、オブジェクト total に代入する。

4. dfとtotalを結合したデータフレーム df2 を作成する。

```

> total <- df[,4]+df[,5]+df[,6]
> total
 [1] 144 132 184 111 184 215 125 131 176 108 169 181 253 154$
[19] 182 178 145 188 265 167 188 164 189 203 223 193
> df2 <- data.frame(df,total)
> df2
  sex region name Japanese Math English total
1   1  tokyo aoki      23    56     65    144
2   2  tokyo ehara     43    45     44    132
3   2  saitama kato     11    97     76    184
4   1  chiba kayama    56    32     23    111
5   2  chiba kinoshita  89    41     54    184

```

■ 留意点・ポイント

- ・作業ディレクトリは「getwd()」コマンドで取得することができる。
- ・データ構造において、行列は値がすべて同じデータ型であるのに対して、データフレームは異なるデータ型の値を混在させることができる。
- ・データフレームの列を抽出するときは列番号を指定する方法と、列のラベルを用いて指定する方法がある。

第3時 Rでのプログラミング（1）

■ 本時の内容・目標

- ・簡単なユーザー定義関数を作成できる。
- ・比較演算子、論理演算子を理解する。
- ・条件分岐の構文を理解する。

■ 準備物

- ・授業ワークシート（本時に使用する数式や関数を掲載）
- ・配布用プログラムファイル

■ 授業の流れ

① 簡単なユーザー定義関数の作成

1. ユーザー定義関数の書式を説明する。
2. ワークシートを用いて、入力した値を2倍する関数 `twice` を作成する。
3. ワークシートを用いて、標本分散を求める関数 `var2` を作成する。

```
関数名 <- function(引数・入力値){
  < 計算処理の1行目 >
  < 計算処理の2行目 >
  < . . . . . >
  return(返り値・出力値)
}
```

```
R Console
> twice <- function(x){
+   y <- x*2
+   return(y)
+ }
> twice(2)
[1] 4
> twice(-3)
[1] -6
> twice(0)
[1] 0
> |
```

```
R Console
> var2 <- function(vec){
+   y <- mean((vec - mean(vec))^2)
+   return(y)
+ }
> var2(c(1,2,3,4,5))
[1] 2
> var(c(1,2,3,4,5))*(5-1)/5
[1] 2
> |
```

関数 `var2` は標本分散(各々のデータの値と平均値の差を2乗したものの平均)

② 比較演算子、論理演算子

1. 比較演算子、論理演算子について説明する。
2. ワークシートを用いて、比較演算子、論理演算子の戻り値を確認する。

比較演算子				論理演算子			
記号	意味	記号	意味	記号	意味	記号	意味
<code>==</code>	等しい	<code>!=</code>	等しくない	<code>&</code>	かつ	<code>&&</code>	かつ
<code>></code>	より大きい	<code>>=</code>	以上	<code> </code>	または	<code> </code>	または
<code><</code>	より小さい	<code><=</code>	以下	<code>!</code>	否定	<code>xor</code>	排他的論理和

```

R Console
> x <- 3
> (x > 2) & (x > 4)
[1] FALSE
> (x > 2) | (x > 4)
[1] TRUE
> !(x == x)
[1] FALSE
> vec1 <- c(TRUE, TRUE, FALSE, FALSE)
> vec2 <- c(TRUE, FALSE, TRUE, FALSE)
> vec1 & vec2
[1] TRUE FALSE FALSE FALSE
> vec1 && vec2
[1] TRUE
> |
    
```

```

R Console
> a <- 2 ; b <- 3
> a == a
[1] TRUE
> a == b
[1] FALSE
> a != b
[1] TRUE
> a < b
[1] TRUE
> a > b
[1] FALSE
> vec1 <- c(2,3,4) ; vec2 <- c(5,3,1)
> vec1 == vec2
[1] FALSE TRUE FALSE
    
```

③ 条件分岐 (if 文、else 文)

1. 条件分岐の構文の書式を説明する。
2. ワークシートを用いて、あらかじめ用意した関数を作成する。
3. 与えられたデータの値を確認する。

```

if(条件){
  < 条件が「真」の場合の計算処理の 1 行目 >
  < ..... >
}
else{
  < 条件が「偽」の場合の計算処理の 1 行目 >
  < ..... >
}
    
```

```

R Console
> colatz <- function(x){
+   if (x %% 2 == 0){
+     y <- x/2
+   }
+   else{
+     y <- 3*x+1
+   }
+   return(y)
+ }
> |
    
```

a %% b は a を b で割った余りを表す

関数 colatz は入力値が偶数であれば 2 で割った値を、奇数であれば 3 倍して 1 加えた値を出力

```

R Console
> colatz(5)
[1] 16
> colatz(16)
[1] 8
> colatz(8)
[1] 4
> colatz(4)
[1] 2
> colatz(2)
[1] 1
    
```

■ 留意点・ポイント

- ・関数はデータ型や、データ構造、演算子などを用いて、一連の処理をひとまとめにしたものと考えることができる。
- ・比較演算子や論理演算子は、その結果が真の場合は TRUE を、偽の場合は FALSE を値として返す。
- ・論理演算子の論理積（かつ）や論理和（または）について、短い形式（&, |）は他の演算子と同様にベクトル中の要素ごとの比較を行い、長い形式（&&, ||）はベクトルの最初の要素を比較する。
- ・すべての整数において、関数 colatz にその値を入力し、出力される値をさらに関数 colatz に入力することを繰り返すと、最終的に 1 になることが予想されている。

第4時 Rでのプログラミング（2）

■ 本時の内容・目標

- ・ 繰り返しの構文を理解する。
- ・ 簡単なシミュレーションができる。
- ・ 基本的なグラフを作成できる。

■ 準備物

- ・ 授業ワークシート（本時に使用する数式や関数を掲載）
- ・ 配布用プログラムファイル

■ 授業の流れ

① 繰り返し（while 文、for 文）

1. while 文の処理と構文の書式を説明する。
2. 前回扱った関数 colatz のプログラムファイルからソースをコピーし、コンソールに張り付ける。
3. ワークシートを用いて、関数 colatz2 を作成する。
4. いくつかの入力値に対して、関数 colatz2 から出力される値を確認する。

```
while(条件){  
  < 条件が「真」の場合の計算処理の1行目 >  
  < 条件が「真」の場合の計算処理の2行目 >  
  < …………… >  
}
```

```
R Console  
> colatz2 <- function(x){  
+   count <- 0  
+   while(x>1){  
+     x <- colatz(x)  
+     count <- count + 1  
+     print(x)  
+   }  
+   cat("repeat count\n")  
+   return(count)  
+ }  
> |
```

```
R Console  
> colatz2(40)  
[1] 20  
[1] 10  
[1] 5  
[1] 16  
[1] 8  
[1] 4  
[1] 2  
[1] 1  
repeat count  
[1] 8  
> |
```

② 繰り返し（for 文）

1. for 文の処理と構文の書式を説明する。
2. ワークシートを用いて、関数 colatz3 を作成する。
3. 関数 colatz3 から出力される値を確認する。

```
for(ループ変数 in ベクトル){  
  < 繰り返す計算処理の1行目 >  
  < 繰り返す計算処理の2行目 >  
  < …………… >  
}
```

② グラフの作成

1. 扱えるグラフの種類（棒グラフ、折れ線グラフ、散布図、箱ひげ図など）について説明する。
2. ②で出力された結果を用いて、グラフを描画する。

■ プログラム

```
R Console
> colatz3 <- function(n) {
+   c_list <- NULL
+   for(x in 1:n){
+     count <- 0
+     while(x>1){
+       x <- colatz(x)
+       count <- count + 1
+     }
+     c_list <- c(c_list, count)
+   }
+   return(c_list)
+ }
> |
```

1 から入力した自然数まで、それぞれの数値に対して初めて 1 が出力されるまで繰り返し Colatz 関数に入力する回数のベクトルを求める関数。

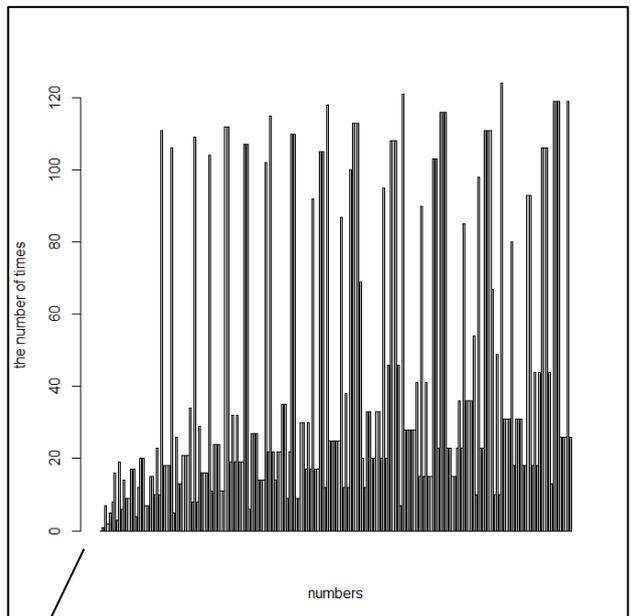
回数を格納するベクトルは空にしておく。

Colatz 関数に入力した回数のカウンター。

前回作成した Colatz 関数を呼び出す。

求めた回数を、格納するためのベクトルに結合させていく。

```
R Console
> colatz3(200)
 [1] 0 1 7 2 5 8 16 3 19 6 14 9
[13] 9 17 17 4 12 20 20 7 7 15 15 10
[25] 23 10 111 18 18 18 106 5 26 13 13 21
[37] 21 21 34 8 109 8 29 16 16 16 104 11
[49] 24 24 24 11 11 112 112 19 32 19 32 19
[61] 19 107 107 6 27 27 27 14 14 14 102 22
[73] 115 22 14 22 22 35 35 9 22 110 110 9
[85] 9 30 30 17 30 17 92 17 17 105 105 12
[97] 118 25 25 25 25 25 87 12 38 12 100 113
[109] 113 113 69 20 12 33 33 20 20 33 33 20
[121] 95 20 46 108 108 46 7 121 28 28 28
[133] 28 28 41 15 90 15 41 15 15 103 103 23
[145] 116 116 116 23 23 15 15 23 36 23 85 36
[157] 36 36 54 10 98 23 23 111 111 111 67 10
[169] 49 10 124 31 31 31 80 18 31 31 31 18
[181] 18 93 93 18 44 18 44 106 106 106 44 13
[193] 119 119 119 26 26 26 119 26
> |
```



```
R Console
> barplot(colatz3(200), ylim=c(0,125),
+ xlab='natural numbers',
+ ylab='the number of times')
> |
```

1 から 200 までの自然数で Colatz3 を適用した結果と、これを棒グラフで描画したもの。縦軸の範囲を 0 から 125 としている。

■ 留意点・ポイント

- ・ R では、for 関数はベクトルに設定されている要素を順番にループ変数に設定して処理をしている。
- ・ Colatz3 関数内の「1:n」は 1 から n までの自然数列のベクトルを意味する。
- ・ Colatz2 関数内で用いた cat 関数は値や文字列を表示させるだけのもので、関数内での処理の途中経過を表示させるのに利用される。なお「\n」は改行を意味する（コンソール内では「¥」はバックスラッシュとして表示される）。
- ・ ユーザー定義関数の中で別のユーザー定義関数を呼び出すことができる。

micro:bit を用いたプログラミング授業実践

塚本 初恵

■ 内容説明

本単元では、シミュレータを利用して初めてプログラミングすることを体験する。情報社会のサービスを作る側の視点について学ぶと共に、自らも簡単なプログラミングをすることで意識を高めることが狙いである。情報システムを理解する第1歩として、教育用プログラミング言語「micro:bit」を用いれば、シミュレーションを容易に体験できる。本校では、3学期に初めて、プログラミングの単元としては1回目となる。

■ 授業の実施情報

学校名	実施科目	実施学年
	社会と情報	高校2年生
時間	1授業あたりの実施人数	使用機材
65分	40人(教員は2名)	パソコン
使用言語	使用ソフト	OS
micro:bit	ブラウザ	Windows 7

■ 単元の指導計画

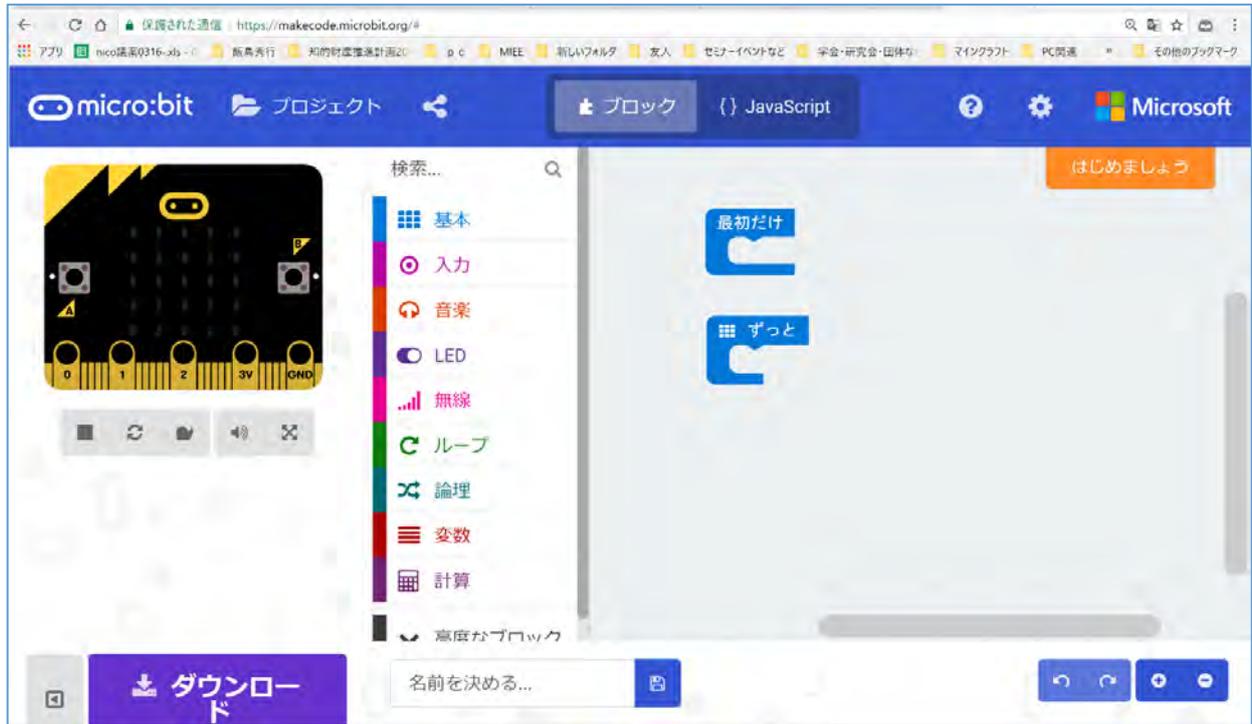
時間	学習活動・学習内容	学習目標
第1時	<p>●micro:bit シミュレータを用いたプログラミングの基礎(1)</p> <p>GUI部品を用いたプログラミングについて学習する。命令によって表示されるLEDの動きが変化することや、繰り返し構造を用いることで同じ操作が用意に繰り返せることを確認する。LEDの表示の仕方、変化のさせ方を考える。音の長さや音程を用いてメロディを作成する。</p>	<ul style="list-style-type: none"> ・GUIによるプログラミングを理解する。 ・さまざまな命令の意味を理解する。 ・繰り返し構造を使って命令が繰り返されることを理解する。 ・視覚的に動きを理解する。 ・縦横5x5LED表示板の点灯表示の変化をドット絵の組み合わせによって工夫することに気づき、画像の自動化を体験する。 ・一つ一つの音階と長さによりメロディを表現できることを理解し、自分なりのメロディを作成することで、音の自動化もできることを体験する。 ・画像の変化とメロディを聞くことが同時にできることを体験する。
第2時	<p>●micro:bit シミュレータを用いたプログラミングの基礎(2)</p> <p>変数の使い方や、条件分岐によって処理が切り分けられることを学習する。ボタンを用いて、操作を切り分けることができることを確認する。他の人のプログラムにリミックスしてみる。</p>	<ul style="list-style-type: none"> ・条件分岐を使って処理が切り分けられることを理解する。 ・変数の使い方について理解する。 ・ボタンの役割について理解する。 ・友達の作品に触れ、リミックスする方法を理解する。

■ 環境設定・使用ソフトの準備

教育用プログラミング言語「micro:bit」のプログラムは、以下のサイトからブラウザで体験できる。

プログラミング言語「micro:bit」

<https://makecode.microbit.org>



上記サイトから直接、ブラウザを介して「micro:bit」を使うことができる。使用できる OS は、Mac、Windows 7以降の PC、Chromebook、Linux（ラズベリーパイも含む）をサポートしている。

（本校では、「IE11」を利用したが、音が聞こえないなどのトラブルがでやすい）

※ 推奨は、トラブルがあまりないので「Google Chrome」をお勧めする。

ブラウザ上の自分で作成した「hex」ファイルは、ダウンロードのボタンを選択すると PC のデフォルトで設定された（Window7 標準だとダウンロード）フォルダに保存される。ダウンロードしたファイルは共有フォルダなどにコピーしておけば、全員でのプログラムを使うことができるため、それぞれの端末や個人ドライブなどにコピーして、自分のブラウザに表示することができる。また、他の人のプログラムを表示できるので、ほかの人がどんなプログラムを書いているかを閲覧できる。自分のコンピュータに実行環境を作れるので、シミュレート画面もすぐに確認できる。

■ 準備した物

今回は、micro:bit の実機を使わない方法で、授業をしたが、実機を使用するなら下記のようなものも用意するとよい。

- ・ micro:bit 本体
- ・ USB ケーブル（一方は、マイクロ USB のインターフェース）
- ・ ワニ口クリップ 2 本
- ・ アンプ付きスピーカー
- ・ 電池ボックス（または Android 用充電電池）
- ・ 電池

※ アンプ無しスピーカーは、micro:bit に電流が流れすぎて負担がかかる。

第1時 micro:bit シミュレータを用いたプログラミングの基礎（1）

■ 本時の内容・目標

- ・ GUI 部品を用いたプログラミングについて学習する。命令によって表示される LED の動きが変化することや、繰り返し構造を用いることで同じ操作が用意に繰り返せることを確認する。
- ・ LED の表示の仕方、変化のさせ方を考える。
- ・ 音の長さや音程を用いてメロディを作成してみる。

■ 準備物

- ・ PC（ブラウザは、chrome 推奨）・インターネット環境

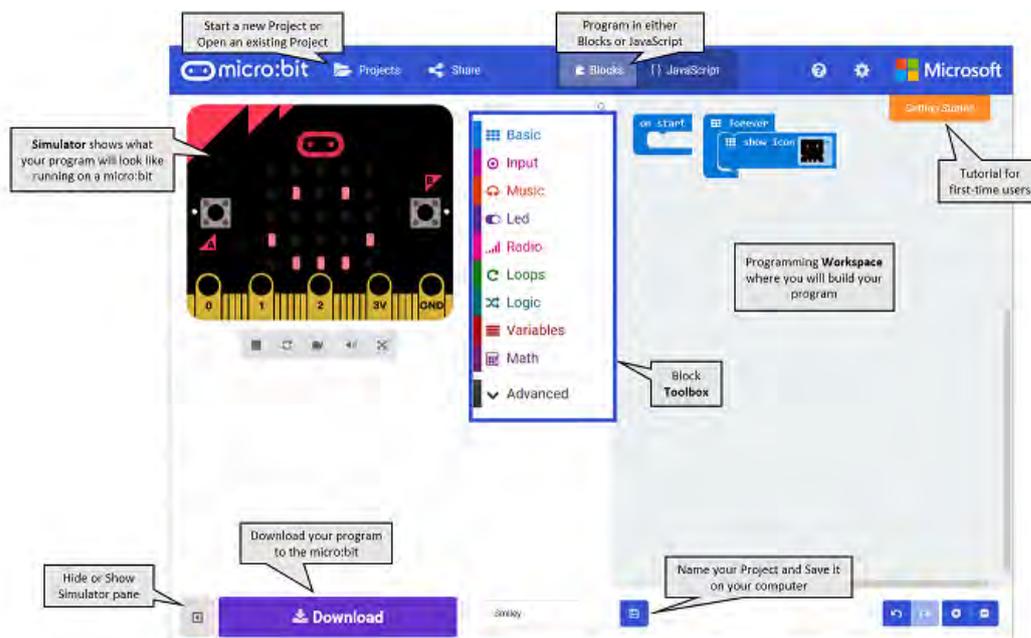
■ 授業の流れ

① 本日の実習についての説明

1. プログラムされた実機をみせて micro:bit で LED 表示される様子を見せ、どんなことができるかを気付かせる。
2. 実機を順番にクラスに回す。

② 機能・操作を知り実習する

1. <https://makecode.microbit.org> を起動する。
2. シミュレータ、ブロックツールバー、ワークスペースの 3 つの枠から構成されていることを説明する。



<http://microbit.org/assets/posts/2018-01-19-train-the-trainer/topic-1.pdf> 参照

3. microbit.org のサイト（web サーバ）よりローカル（自分の）PC に作成したファイルのダウンロードの仕方を参照する。作品ができたら、共有フォルダに提出して他人の作品も閲覧する。



※ 一般的にファイルは「ダウンロード」フォルダに保存される。

4. 機能 <http://microbit.org/ja/guide/features/>などのページを参照する。

- 個々にプログラムできる 25 個の LED
- プログラム可能な 2 個のボタン
- 物理的に接続するための端子
- 光と 温度のセンサー
- 動きセンサー（加速度計とコンパス）
- 無線機と Bluetooth を通した無線通信
- USB インターフェース

5. 基本命令（Basic commands）LED 表示

http://microbit.org/assets/posts/2018-01-19-train_the_trainer/topic_2.pdf を参照して操作する。

Basic
Use basic micro:bit functions and actions.

- showNumber**
Scroll a number on the screen.
- showIcon**
Draws the selected icon on the LED screen.
- showLeds**
Draws an image on the LED screen.
- showString**
Display text on the display, one character at a time.
- clearScreen**
Turn off all LEDs.
- forever**
Repeats the code forever in the background.
- pause**
Pause for the specified time in milliseconds.
- showArrow**
Shows an arrow on screen.

「microbit.org/ja/teach」検索→教育リソースの下方にあるカリキュラムの「From Getting Started to Games - starting Computer Science with the BBC micro:bit」を選択→「Lessons」→2 Lesson 「Basic commands」を参照

6. サウンド（音楽）についても同様に操作をする。試させた後、どのブロックが使いやすいかを問う。

検索... 🔍

- 音を鳴らす 高さ 真ん中のド 長さ 1 拍
- 音を鳴らす (Hz) 高さ 真ん中のド
- 休符 (ミリ秒) 1 拍
- メロディを開始する ダダダム くり返し 一度だけ

生徒にどのブロックが適切か試行錯誤しながら、適切なものを選択させ、気付かせる。友達と相談しながら試し、生徒間でファイルを閲覧しあう。

第2時 micro:bit シミュレータを用いたプログラミングの基礎（2）

■ 本時の内容・目標

- ・変数の使い方や、条件分岐によって処理が切り分けられることを学習する。
- ・ボタンを用いて、操作を切り分けることができることを確認する。
- ・他の人のプログラムにリミックスしてみる。

■ 準備物

- ・PC（ブラウザは、chrom 推奨）・インターネット環境・共有フォルダ（サイトの共有リンクでも代用できる）

■ 授業の流れ

① 本日の実習についての説明

1. 変数の使い方やボタンを使った条件分岐によって処理が切り替わることを学習する。
2. 他の人のプログラムを取り込みリミックスしてみる。

② 機能・操作を知り実習する（生徒に試行錯誤させて確かめさせる。）

1. <https://makecode.microbit.org> を起動する。
2. ボタン A、ボタン B、ボタン A+B の使い方を学ぶ。
3. 繰り返しや一時停止ブロックの使い方を学ぶ。
4. 変数を使ってみる。変数に任意の名前をつけ、値が変わることを確かめる。
5. 例をみてプログラミングしてみる。
6. プログラムを自分で変更する。また、新たに考えて作成してみる。
7. 作成したファイルをローカル PC に保存する。【microbit.org のサイト（web サーバ）よりローカル（自分の）PC に作成したファイルのダウンロードをする。】
8. 作品をサイトの「共有」ボタン→プロジェクト「共有プロジェクトを公開する」を選択、URL のリンクをコピーして、作品をサイトにアップしてみる。



9. 8で取得したURLにアクセスし、他人の作品も閲覧する。

ダウンロードボタンから自分のPCのフォルダにhexファイルを保存できる。

10. 他人の hex ファイルを取り込み、編集してみる。

共有フォルダに置かれたファイルを自分の PC にコピーしてそのファイルを開く方法



- ・サイトの左上「プロジェクト」のボタンを選択し、次の画面の「自分のプロジェクト」から「ファイルを読み込む」を選択して、目的の hex ファイルを選択する。

11. 「makecode.microbit.org」のサイトに表示された画面で、再プログラミングする。

12. 他人と閲覧しあって、工夫した点や改良した点などを話し合う。

■ プログラム



■ 留意点・ポイント

- ・操作をすべて教師側が教えるのではなく、生徒自身が操作を調べさせることにより、試行錯誤して手段を見つけ出す手助けをする。机間巡視をして進んでいない生徒に目的達成の糸口を見つける言葉がけをする。

Monaca を用いたプログラミング授業実践

小原 格

■ 内容説明

「問題の解決と処理手順の自動化」を学習する単元として、「数当てゲーム」の作成を行う。ここで言う「数当てゲーム」とは、予め決められた1から100までの「当たり」の整数1つを、数を挙げて当てていくゲームであるが、数を言うごとに、その数が当たりの数よりも大きいか、小さいかの情報を与えられることを通してゲーム性を高めているものである。このプログラムは10行ほどで書け、また、繰り返し、判断分岐も含まれているため、学習用の教材としては扱いやすいものの1つと考えられる。

このプログラム作成にあたり、クラウドでの開発環境である Monaca を利用した。Monaca はスマートフォンのアプリ作成プラットフォームであり、入力をアシストする機能や共同制作の機能のほか、作成したプログラムをスマートフォンで実行させることもでき、さらに興味関心を高めさせることができる。

■ 授業の実施情報

学校名	実施科目	実施学年
	情報の科学	高校1年生
時間	1授業あたりの実施人数	使用機材
45分	40人(教員は1名)	パソコン
使用言語	使用ソフト	OS
Javascript	ブラウザ(crome) + Monaca	Windows7

■ 単元の指導計画

時間	学習活動・学習内容	学習目標
第1時	<p>●アルゴリズムの基礎</p> <p>順次・分岐・反復の3つの制御構造を復習し、学校に行くまでの過程など、身近な内容をアルゴリズムを用いて表してみる。</p>	<ul style="list-style-type: none"> ・コンピュータの特徴を理解する。 ・3つの制御構造を理解する。 ・3つの制御構造を組み合わせ、身のまわりにあるアルゴリズムについて考える。
第2時	<p>●Javascriptの基本</p> <p>HTMLやJavascriptの基本を理解し、3つのグループエキスパート活動として、「画像の表示」「判断分岐」「繰り返し」をそれぞれ学習する。</p>	<ul style="list-style-type: none"> ・HTMLの構造について簡単に理解する。 ・Javascriptでの変数や、入力画面の表示方法などの基本的な内容を理解する。 ・画像表示、分岐、反復の表現を理解する。
第3時	<p>●数当てゲームの作成(1)</p> <p>数当てゲームの内容を理解した後、別々のエキスパートグループから3人が1グループでチームを作り、ワークシートを参考に、ゲームのプログラムを作成する。</p>	<ul style="list-style-type: none"> ・数当てゲームのアルゴリズムを理解する。 ・3人で知識を集め、協力し、数当てゲームのアルゴリズムを実現するJavascriptプログラムを作成する。
第4時	<p>●数当てゲームの作成(2)</p> <p>数当てゲームプログラムを完成させる。完成したチームは、答えの自動生成や入力項目のチェックなどに取り組む。</p>	<ul style="list-style-type: none"> ・Javascriptを用いての問題解決を理解する。 ・Javascriptを用いた問題解決に興味を持つ。 ・Javascriptを効果的に用いて、答えの自動生成や入力条件など評価・改善を行う。

■ 環境設定・使用ソフトの準備

クラウドでの統合環境「Monaca」については、以下のサイトから利用することができる。なお、ブラウザは Chrome が推奨されている。

HTML5 マルチプラットフォームアプリ専用開発環境「Monaca」 <https://ja.monaca.io/>

上記サイトからアカウントを作成しログオンする。なお、無料プランでも十分に活用できるが、Education 版では共同作業がより効果的にできたり、保存できるプロジェクト数が増えたりと、より効果的な活用が見込まれる。



新規にプロジェクトをゼロから立ち上げることも可能であるが、Monaca はインポート機能が充実しており、インポート用パッケージを作成し共有フォルダー等から生徒にインポートさせることにより、あらかじめ教員が準備した内容から簡単に進めることも可能である。

プロジェクトのインポート

プロジェクト名 *

説明

インポート方法

- URLを指定してインポート
- プロジェクトのパッケージをアップロード
 選択されていません
- Gitのレポジトリからインポート

■ 謝辞

この授業実践は、文部科学省委託事業平成 27 年度情報教育指導力向上支援事業プログラミン教育実証授業とともに、アシアル株式会社に多大なるご配慮・御協力をいただいた。深く感謝の意を表する。

第2時 Javascript の基本

■ 本時の内容・目標

- ・ HTML の構造について簡単に理解する。
- ・ Javascript での変数や、入力画面の表示方法などの基本的な内容を理解する。
- ・ 画像表示、分岐、反復の表現を理解する。

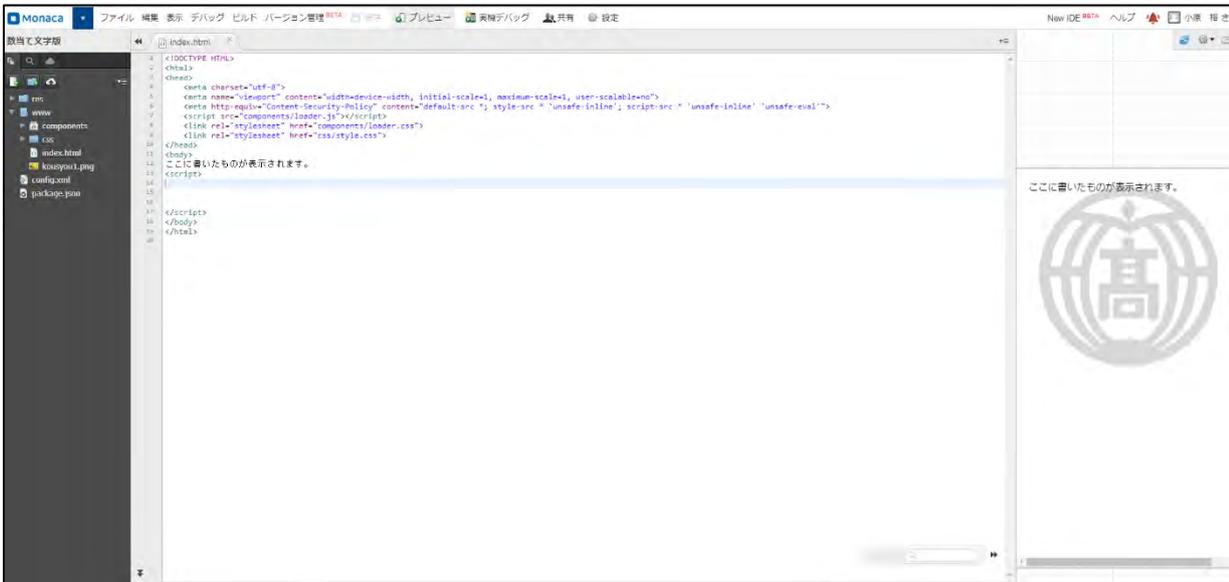
■ 準備物

- ・ 授業ワークシート（本時に使用するプログラムコードと解説を掲載）

■ 授業の流れ

① ワークシートの配布と Monaca の設定

1. あらかじめ用意したワークシートを入手させる。（共有フォルダから個人フォルダにコピーさせる）
2. Monaca のサイトを説明し、ログオンさせる。
3. 教員が用意したパッケージをインポートさせる。



② Javascript の変数や入力画面の説明

1. prompt のサンプルコードを用いて利用法を理解させる。
2. parseInt を使う意味と利用方法について理解させる。

例1) プロンプト(手入力)の表示:prompt(表示)と、画面への表示:document write(文字列)

```

<script>
suki=prompt("好きな言葉は");
alert(suki);
document.write(suki);
</script>

```

解説:
「好きな言葉は」という表示のもとに「入力窓」を出し、入力されたものをsukiという変数に代入
変数 suki に入っている文字列を警告窓(alert)に出す
変数 suki に入っている文字列を画面に直接表示(write)する

※ 数学では、変数は1文字(例えば、y)ですが、Javascriptの場合は、複数文字(例えばsuki)で一つの変数とできる。
※ "suki"という風に" "で両端をくくると、suki という文字列そのものが出力されてしまいますので注意!

例2) プロンプト(手入力)に数字を入れる場合:parseInt(prompt(表示))

```

<script>
suki=parseInt(prompt("好きな数字は"));
alert(suki);
document.write(suki);
</script>

```

解説:
「好きな数字は」という表示のもとに「入力窓」を出し、入力された数字をsukiという変数に代入
変数 suki に入っている数字を警告窓(alert)に出す
変数 suki に入っている数字を画面に表示(write)する

※ prompt は文字列として扱われるので、数字を入れる場合はparseIntを付け加える。(二重括弧になるので注意! :)
(parseInt がついていないと、例えば 2 + 3 が 23 になってしまったりすることがある)

③ エキスパートグループに分かれ、各自の演習に取り組む

1. 出席番号を3で割り、「1余る」「2余る」「割り切れる」でエキスパートグループを作る。
2. 「1余る」グループは判断分岐、「2余る」グループは反復、「割り切れる」グループは画像表示およびアラートを出すプログラムを作成させる

実習2 簡単なプログラムを作ってみよう(1余る人向け)

課題1 (必須) 2つの数値を手入力し、それらの数が等しかったら「等しい」と「アラート」を出すプログラムを作れ。
(等しくない時は、特に何もしないで終了する)

課題2 (任意) 2つの数値を手入力し、それらの数を比較して大きい方の数値を「アラート」に出すプログラムを作れ。

※monaca上で行うこと。先の「練習」プロジェクトを修正して良い。
※教えあっても良いが、必ず意味を理解すること。このあと、別の組のメンバーと1チームになるので、頼れるのは今だけ。
キミの知識があつてこそ、最終課題(数当てゲーム)が完成する！
※「参考」シートにヒントがあります。

実習2 簡単なプログラムを作ってみよう(2余る人向け)

課題1 (必須) 『入力窓(プロンプト)に「好きな言葉」を入力すると、その言葉を画面に表示する作業』を、5回繰り返すプログラムを作れ。

課題2 (必須) 入力窓(プロンプト)に「好きな言葉」を入力すると、その言葉を画面に5つ表示するプログラムを作れ。

※monaca上で行うこと。先の「練習」プロジェクトを修正して良い。
※教えあっても良いが、必ず意味を理解すること。このあと、別の組のメンバーと1チームになるので、頼れるのは今だけ。
キミの知識があつてこそ、最終課題(数当てゲーム)が完成する！
※「参考」のシートや教科書113ページにもヒントがあります。

実習2 簡単なプログラムを作ってみよう(割り切れる人向け)

課題1 (必須) 画面の文字列「おはよう！」を「こんばんは！」に、画面の下の太陽の絵「morning.png」を月の絵「evening.png」に差し替えるプログラムを作れ。

課題2 (必須) 2つの数を入力すると、その2つの数の和を「和は〇ですね」とアラートを出すプログラムを作れ。
(答えだけでなく、「和は〇ですね」と日本語のセリフもつけること)

課題3 実行させると、1から10までの数値をランダムに「アラート」で表示するプログラムを作れ。
(よりゲーム性を高めたい人向け:「発展」のシートを見よう)

※monaca上で行うこと。先の「練習」プロジェクトを修正して良い。
※教えあっても良いが、必ず意味を理解すること。このあと、別の組のメンバーと1チームになるので、頼れるのは今だけ。
キミの知識があつてこそ、最終課題(数当てゲーム)が完成する！
※「発展」のシートや教科書113ページにもヒントがあります。

■ 留意点・ポイント

- ・漢字やひらがなの入力が終わったら、すぐに日本語入力を OFF にすることを強調する。これにより、全角文字の入力によるエラーを減らすことが見込まれる。
- ・parseInt の役割について説明するとともに、数値を入力したい場合には、これを入れないと 2+3 が 23 と表示されてしまったりすることについて触れておく。
- ・エキスパート活動ではグループ内で自由に教えあいをして構わないことを強調する。
- ・Javascript のサンプルコードをワークシートに載せておくとともに、生徒に自由に参照させるように確認させる。
- ・今後の流れについても簡単に触れるとともに、次回からはそれぞれのエキスパートグループから1名ずつからなる小グループを作ることになるため、自分の役割をしっかりと認識させる。
- ・サンプルプログラムを丸写ししてもできないため、命令の意味をしっかりと理解するように指導する。

第3・4時 数当てゲームの作成

■ 本時の内容・目標

- ・ 目的の動作をさせるために、順次、分岐、反復を用いたアルゴリズムを考えさせる
- ・ Javascript を用いて、アルゴリズムからプログラムを完成させる
- ・ グループで協力し知識やアイデアを出し合い、協同して効果的に作業を行おうとする。

■ 準備物

- ・ 授業ワークシート（本時に使用するプログラムコードと解説を掲載）

■ 授業の流れ

① グループ作成とゲーム内容の確認

1. 前回のエキスパートグループからそれぞれ一人ずつ、3人（または4人）グループになる。
（出席番号を3で割った余りでエキスパートを作っているのので、連続する3人がグループとなる）
2. 3人で「数当てゲーム」を実際に行い、内容を確認する。

※「数当てゲーム」とは

1から100までの整数のなかから、「当たり」となる数を1つ選び、その数を別の人が当てるゲーム。外れた場合、ヒントとして、「当たり」の数が「もっと大きい」か「もっと小さい」という情報が与えられる。これを手がかりに、決められた回数以内で当てられるように数を選んでいく。

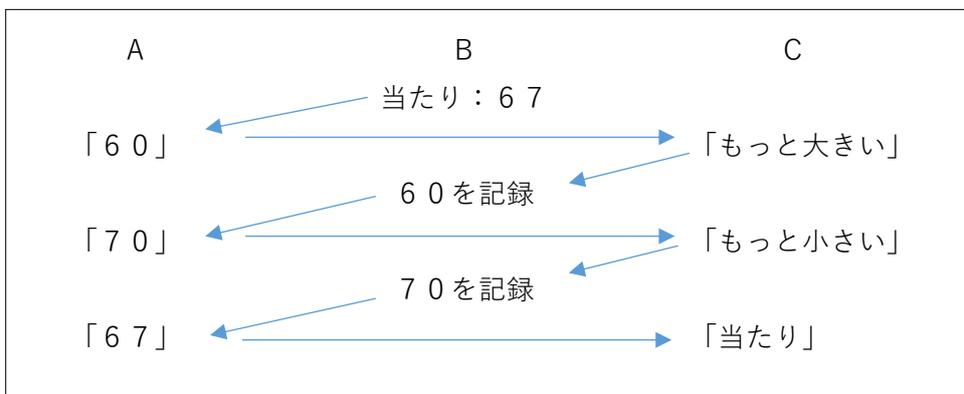
グループ3名 (A,B,C) の役割は適宜選べば良いが、例えば、

A：当たりの数を予想し、当てる生徒

B：「当たり」の数を考え、「予想した数」を記録していく生徒

C：「予想した数」が「当たりの数」と等しいか判断し、「もっと大きい」「もっと小さい」等と言う生徒

などに分担すると良い。流れは、例えば以下のとおり



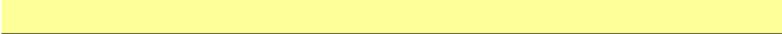
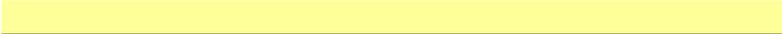
② 数当てゲームのアルゴリズム確認とプログラムの作成

1. 実際のゲーム体験を通して、ゲームのアルゴリズムを考える。
当てるチャンスは「5回」とし、当たったら宝石入りの宝箱に変わり、外れたら空の宝箱となるように表示させることとする。

2. 前回の各自の知識を集結させ、Monaca を利用してプログラムを作成させる。

実習 数当てゲームを作ってみよう
 ※まずは「流れ」を考えて日本語で埋めた後、それに対応する命令をmonacaに打ち込んでみよう！ 1行が、Javascriptの1行分に対応しています。

例) 数当てゲーム(1から100まで:チャンス5回)の流れ(概要)

- ① あたりの数を決めて入力する。
- ② 5回繰り返し(繰り返し作業ここから)
- ③ 
- ④ もし、「予想した数」が「当たりとする数」と同じだったら、(条件作業ここから)
 「タイトル」の文字を「当たり」に変え、
 「アイコン」の絵を「あたり用」の宝箱(box_treasure.png)に変更し、
 繰り返しから強制的に抜ける。(条件作業ここまで)
- ⑤ 
- ⑥ 
- ⑦ (繰り返し作業ここまで)
- ⑧ もし、5回答ええていたら、(条件作業ここから)
 「タイトル」の文字を「はずれ」に変え、
 「アイコン」の絵を「はずれ用」の宝箱(box_empty.png)に変更する。(条件作業ここまで)
- ⑨ 「メッセージ」の文字を、「正解は・・・」に変更する。

④ 発展課題に取り組みさせる

1. セリフの変更、条件を満たさない入力への対応、乱数による自動での当たりの数の生成などを行う。
2. 利用される可能性が高い内容を予め示し、参照させやすくしておく

☆よく使われる命令

<code>document.write("●●"+abc);</code>	ブラウザ内に、●●という文字列と[abcの中身の数]を表示する
<code>confirm("△△")</code>	「OK」と「キャンセル」がついた窓を△△というメッセージとともに表示する
<code>x=prompt("▲▲")</code>	ポップアップで、▲▲という文字列を表示し、入力を促してその値をxに当てはめる(代入する)
<code>window.alert("○○"+abc)</code>	ポップアップで、○○という文字列と[abcの中身の数]を表示する(「window」は省略可)
<code>parseInt(abc)</code>	abcの中身の文字列を数値に変換します。
<code>break</code>	繰り返しなどの{}の中から、作業を中止して強制的に抜け出します。
<code>x=Math.random()</code>	xに乱数(0以上1未満)の乱数(不規則な数)を当てはめる(代入する)
<code>x=Math.floor(abc)</code>	xにabcの中身の数を切り捨てた数を当てはめる(代入する)
<code>x=x+1</code>	xの値を1増やす(新しいxの値として、今までのxに1加えた数を代入する)
<code>document.getElementById("abc").innerHTML="●●"</code>	"abc"というIDの文字列を●●に変更する
<code>document.getElementById("def").src="ファイル名"</code>	"def"というIDの画像ファイルを「ファイル名」のものに変更する

■ 留意点・ポイント

- ・前回各エキスパートグループが行った内容も全員に周知させ、わからなかった生徒にも対応できるようにさせる。
- ・アルゴリズムでつまづいているグループは、コードを入力し試行錯誤しながら考えることも視野に入れて指導を行う。
- ・早く終わったグループは、台詞を変える、チャンスの回数を変える、今までに入力した数を画面に表示させる、乱数を使い自動で当たりを決める、等の工夫改善を促す。
- ・Monacaでのプロジェクトパッケージインポート機能を使うことにより、教員が予め途中まで作成したファイルを読み込ませて続きを作業させるという方法も考えられる。これにより、本来的なプログラミングの部分以外のHTMLでのつまづき等を減らすことができる。

映像教材を利用したプログラミング教育の授業実践

飯田 秀延

■ 内容説明

Life is Tech!社の映像教材とテキストを利用したプログラミング教育の実践である。本実践では、生徒は10時間の授業で、各自で映像教材とテキストを観ながら、オリジナルのwebサイトを作成する。授業をアクティブ・ラーニング形式にし、教員は極力「教えない」ようにしたり、過去や他の生徒の作品をアーカイブにしていつでも閲覧できるようにしたりするなどの工夫をすることにより、生徒の思考力・判断力・表現力や、主体的に学ぶ力を向上させられることが期待できる。

■ 授業の実施情報

学校名	実施科目	実施学年
	情報の科学	高校1年生
時間	1授業あたりの実施人数	使用機材
50分×5～10(適宜変更可)	40人(教員は1名)	パソコン
使用言語	使用ソフト	OS
HTML	Webブラウザ、テキストエディタ	Windows7

■ 単元の指導計画

Lesson1～10の主な内容は以下の通りである。すべてのレッスンに、指導案(PDF)と授業スライド(ppt, pdf)、生徒教科書(PDF)が用意されている。なお詳細は「TECH for TEACHERS」のサイトで確認されたい。

Webデザインコース		授業内容	Lesson名	内容・機能
Lesson 1	オリエンテーション	1 オリエンテーション	ようこそ、ITの世界へ	ITの世界への第一歩となります。最新のIT業界の動画を観た後、「検索クイズ」を通して、楽しみながら、WEBの世界に触れてみる。
		2 Webサイトをブラウザで書き換える	WEBを書き換えよう!	Google Chromeの機能を使って、Webを書き換える。
		3 検索ゲーム	検索を使いこなそう!	GoogleChromeでの画像検索の方法を学習する。
Lesson 2	ソースコード	1 これからやる全体像	これからつくるWEBデザイン	これからWebサイトを作成するにあたり、どのような機能を実装するのか、完成図を見る。
		2 htmlの文字を変える	HTMLの編集をしよう!	「HTML」を使ったWebの編集を行う。実際に「エディタ」を使って、HTMLファイルを編集、保存し、ブラウザを更新する手順を学ぶ。
		3 画像を変えてみる	画像を変更しよう!	Webサイトに表示する画像の変更方法を学ぶ。
Lesson 3	HTML	1 画像を検索して置き換える	検索画像に変更しよう!	GoogleChromeから好きな画像を検索し、webサイトに表示させる方法を学ぶ。
		2 リンク	リンクを設定しよう!	「リンク」について学ぶ。「HTML」を使って、リンクの設定を行う。
		3 見出し	見出しを理解しよう!	インターネットの中身について学ぶ。「見出しタグ」についての講義を行う。
Lesson 4	CSS①	1 文字色の変更	CSSで文字の色を変更しよう!	「CSS」について学ぶ。スタイルシートによる色の変更を行う。
		2 文字サイズの変更	CSSで文字のメリハリをつけよう!	「CSS」について学ぶ。スタイルシートによる文字サイズの変更方法を行い、その効果について学習する。
		3 画像の変更	CSSで背景画像を変えよう!	「CSS」による背景の画像の設定について学ぶ。実際にスタイルシートを書き換え、背景画像を変更する。
Lesson 5	CSS②	1 ボックス幅の変更、ボーダー	CSSで枠をデザインしよう!	文字の背景のサイズ変更と、枠の追加方法について学ぶ。また、枠を付けた際の効果について学ぶ。
		2 CSSのクラスの説明	CSSの魔法「クラス」を学ぼう!	「CSS」の「クラス」について学ぶ。
		3 いろいろなCSSを使ってみよう	CSSで応用デザイン!	「CSS」を使いこなす。枠のデザインの変更方法、文字を透明にする方法など、これまでの発展編を学ぶ。
Lesson 6	bootstrap①	1 bootstrapの基本	bootstrapの使い方について学ぼう!	「bootstrap」について学ぶ。実際に、「index.html」に記入してみる。
		2 便利なクラスを使おう	bootstrapでボタンをつくらう!	「bootstrap」でボタンを作成する。ボタンを作成し、リンクを設定する方法を学ぶ。
		3 便利なクラスを使おう②	bootstrapで画像エリアをデザインしよう!	写真エリアのデザインを変更する。画像を丸く切り取る方法と、配置する列の変更方法について学ぶ。
Lesson 7	bootstrap②	1 レスポンシブカラム	レスポンシブサイトについて学ぼう!	「レスポンシブサイト」について学ぶ。レスポンシブ対応による効果について学ぶ。
		2 ナビゲーション	ナビゲーションをつくらう!	「ナビゲーション」について学ぶ。「ナビゲーション」があることによる効果について学ぶ。
		3 ナビの色を変える	ナビゲーションの色を変えよう!	「ナビゲーション」の色を変更する方法について学ぶ。
Lesson 8	UI UX	1 UI、アイコンの設定	UI(ユーザーインターフェース)を学ぼう!	「UI」について学ぶ。使いやすさを意識することによってwebのデザインが変わってくることに気づく。「bootstrap」をつかって簡単にデザインを変更する方法を学ぶ。
		2 UX、スクロールの設定	UX(ユーザーエクスペリエンス)を知ろう!	「UX」について学ぶ。使いごちを意識することによってwebのデザインが変わってくることに気づく。心地よいスクロールができるようにwebページに工夫を施す方法について学ぶ。
		3 UI、UXサイト紹介	実際のサイトからUI、UXを感じよう!	実際のwebページを、「UI」「UX」を意識して閲覧してみる。
Lesson 9	外部サービス連携	1 youtubeの埋め込み	動画を埋め込んでみよう!	外部サービスを埋め込んでみる。どのようなサービスを埋め込むことができるのか学ぶ。この回では、動画(YouTube)を埋め込む方法について学ぶ。
		2 mapの埋め込み	地図を埋めこもう!	外部サービスを埋め込んでみる。この回では、地図(Google map)を埋め込む方法について学ぶ。
		3 外部サービスの連携説明	実際の便利サービスの埋め込みを知ろう!	便利なサービスが埋め込まれているサイトを閲覧してみる。
Lesson 10	ITの未来	1 総復習と今後の学習、ITの重要性	ITの重要性	これからのITの可能性について考える。これまでのレッスンを振り返る。プログラミングを続けることで得られるものについて学ぶ。
		2 ITのリスクと可能性	ITのリスクと可能性	ITのリスクと可能性について学ぶ。
		3 中高生とITの未来	中高生とITの未来	自分の手で作り上げたものが、誰かの人生を変える可能性を秘めていることに気づく。世界を変えられるチャンスは、特別なものではなく、誰にでもあるということについて学ぶ。

■ 環境設定・使用ソフトの準備

Life is Tech!社の TECH for TEACHERS の映像教材とテキストを利用するためには、以下のサイトで登録をする必要がある。なお登録は無料である（平成 30 年 4 月現在）。

TECH for TEACHERS

<https://tech4teachers.jp/>

Web ブラウザとテキストエディタは、映像教材の中で使用している関係上、それぞれ Google Chrome と Sublime Text 3 を推奨する。なお Sublime Text 3 はシェアウェアであり、特に大きな制限なく無料で試用できるが、継続的な利用にはライセンス（80 米ドル）の購入が必要となる。

Google Chrome

<https://www.google.co.jp/chrome/index.html>

Sublime Text 3

<https://www.sublimetext.com/>

第1時～第10時 映像教材を利用したプログラミング教育の授業実践

■ 本時の内容・目標

- ・HTMLによるwebプログラミングを通じ、思考力・判断力・表現力や、主体的に学ぶ力を習得する。

■ 準備物

- ・映像教材とテキスト（電子データにより、生徒がそれぞれ自由に閲覧できるような環境を構築する）。
- ・テキストエディタ（SublimeText3を推奨）。
- ・webブラウザ（Google Chromeを推奨）。
- ・他の生徒の作品をアーカイブとして自由に見ることができる環境。
- ・Lesson1～10のすべての教材の一覧が見られるようなページの作成。

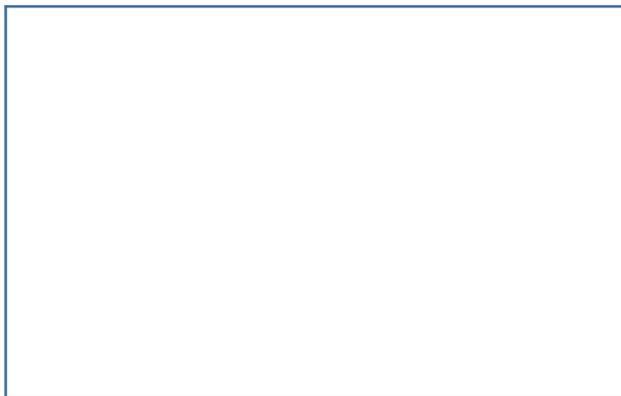
■ 授業の流れ

① 教材一覧の参照

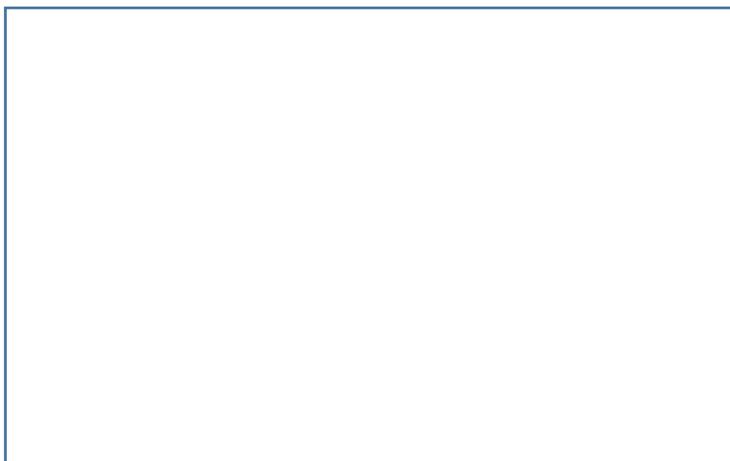
1. 教材一覧のページを開かせる。
2. 授業の進度に応じて、取り組むレッスンを指示する。
3. 映像の指示に従って、テキストエディタによりソースコードを編集するように指示する。
4. 適宜机間指導を行い、生徒からの質問に応じる。



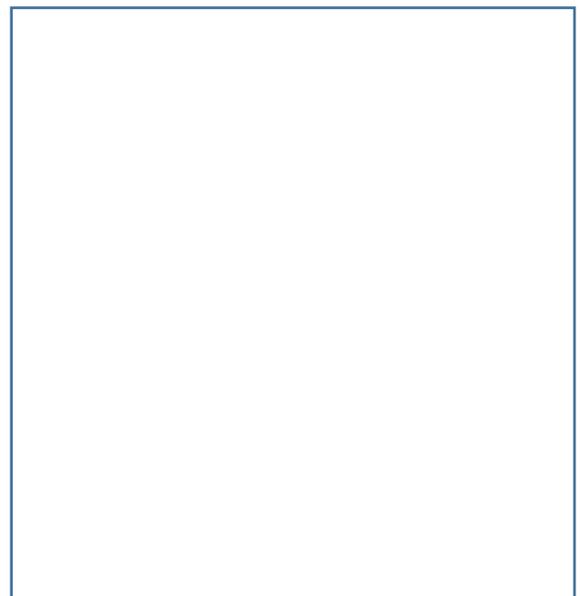
教材一覧ページの例



編集画面の例



映像教材の例



テキストの例

② 他の生徒の作品の参照

1. 授業中に適宜他の生徒の作品を参照するように促す。
2. アイデアに詰まっている生徒には、web デザインのサイトなどを参照させる。

③ 相互評価

1. すべての制作が終わった後で、相互評価を行わせる。
2. 5段階による評価の他に、コメントもその場でフィードバックさせるようにする。

平成28年度 Web作品

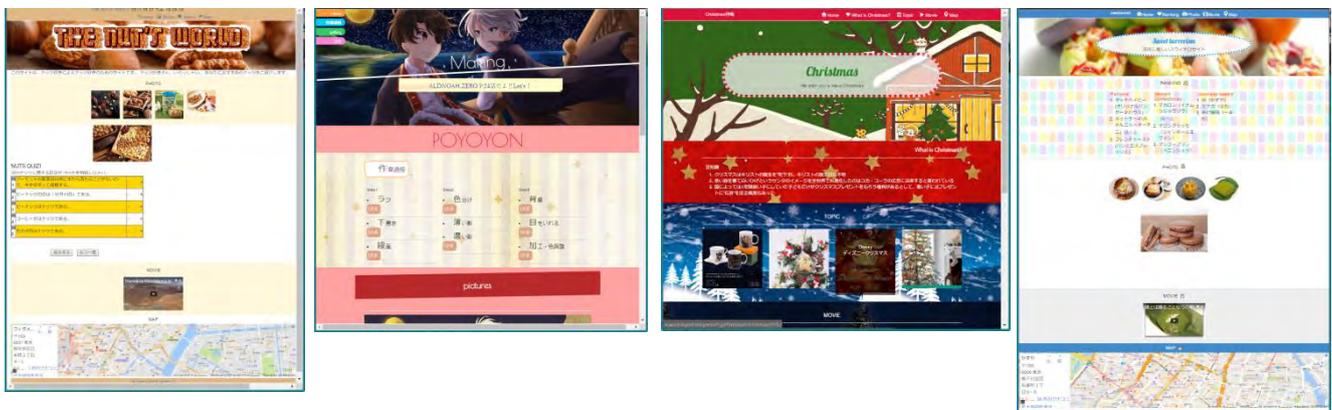
A組										B組										C組									
1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20	11	12	13	14	15	16	17	18	19	20	11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30	21	22	23	24	25	26	27	28	29	30	21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40	31	32	33	34	35	36	37	38	39	40	31	32	33	34	35	36	37	38	39	40

D組										E組										F組									
1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20	11	12	13	14	15	16	17	18	19	20	11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30	21	22	23	24	25	26	27	28	29	30	21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40	31	32	33	34	35	36	37	38	39	40	31	32	33	34	35	36	37	38	39	40

生徒作品一覧の例

■ 作品例

多様で独自性の高い作品が制作された。



■ 留意点・ポイント

● 授業の進め方

- ・映像教材とテキストをみながら「自分で作業を進める」ように促す。
- ・教員は「ファシリテーター」に徹し、できるだけ「教えない」ようにする。

● 作品制作のポイント

- ・テーマ設定は自由にする。
- ・いつでも過去の作品を参照できるようにする。
- ・いつでも他人の作品（進行状況）を参照できるようにする。
- ・放課後や休み時間、自宅での作業も許可する。
- ・映像教材にない技術なども歓迎する。
- ・最後に相互評価をさせ、結果をすぐにフィードバックする。

● 雰囲気作りのために

- ・立ち歩いて他人の作品を見たり、相談したり、教え合ったりすることを促す。
- ・BGMを流し、楽しい雰囲気で作業を行えるようにする。
- ・よい作品や進んでいる生徒の作品を随時紹介する。

プログラミングに興味を持たせる授業実践

稲垣 俊介

■ 内容説明

「情報の授業以前にプログラミングをしたことがありますか」という質問に対し、「ある」と答えたものが、10.9%にとどまり、「ない」と答えたものが89.1%であった。また、「あなたはプログラミングに興味がありますか」という質問に対し、「あまり興味がない (35.7%)」「興味がない (14.0%)」と答える生徒が約半数である。よって、実習をメインとする本単元では、約半数の「興味がない」とする層を、ますます興味がないとは思わせない工夫が必要であると考えます。つまり、プログラミングが難解で自分には関係のないものだと思わせるのではなく、とても興味深いと感じるこのとのできる機会となる単元となることを目指した。また、すべてブラウザ上の Web サービスの組み合わせで実施した単元計画であり、手軽に実施できることも特徴である。

■ 授業の実施情報

学校名	実施科目	実施学年
	情報の科学	高校3年生
時間	1授業あたりの実施人数	使用機材
50分×2 (2時間連続)	40人前後 (教員は1名)	PC
使用言語	使用ソフト (Web サービス)	OS
ドリトル, VBA ※VBAは教員の提示のみ	アルゴリズム, ドリトル, VBE ※VBEは教員の提示のみ	Windows8

■ 単元の指導計画

時間	学習活動・学習内容	学習目標
第1,2時	<p>●アルゴリズムとは (アルゴリズム演習)</p> <p>プログラムの基本となるアルゴリズムの基礎を身近な例で説明し、実習としてアルゴリズムを用いて理解を深める。</p>	<ul style="list-style-type: none"> ・アルゴリズムとは何かを理解し、簡単に説明できる。 ・順次処理と繰り返し処理を体験する。
第3,4時	<p>●プログラミング演習 (ドリトルの演習)</p> <p>実際のプログラミングの体験を「ドリトル」を利用して行う。実際のソフトウェアもこのようなプログラムで動いていることを確認する。</p>	<ul style="list-style-type: none"> ・ドリトルを通じて、プログラムとは何かを理解し、簡単に説明できる
第5,6時	<p>●順次・繰り返し・分岐の理解 (アルゴリズム2演習)</p> <p>プログラムの基本となるアルゴリズムの基礎を身近な例で説明し、実習としてアルゴリズムを用いて理解を深める。また、他の言語ではどのように表現するのかを確認する。</p>	<ul style="list-style-type: none"> ・プログラムの基本構造として、順次、繰り返し、分岐の処理があることを理解し、簡単に説明できる ・他の言語では、「分岐」をどのように表現されるかを確認し、共通していることを認識する。

■ 利用した主な Web サービス

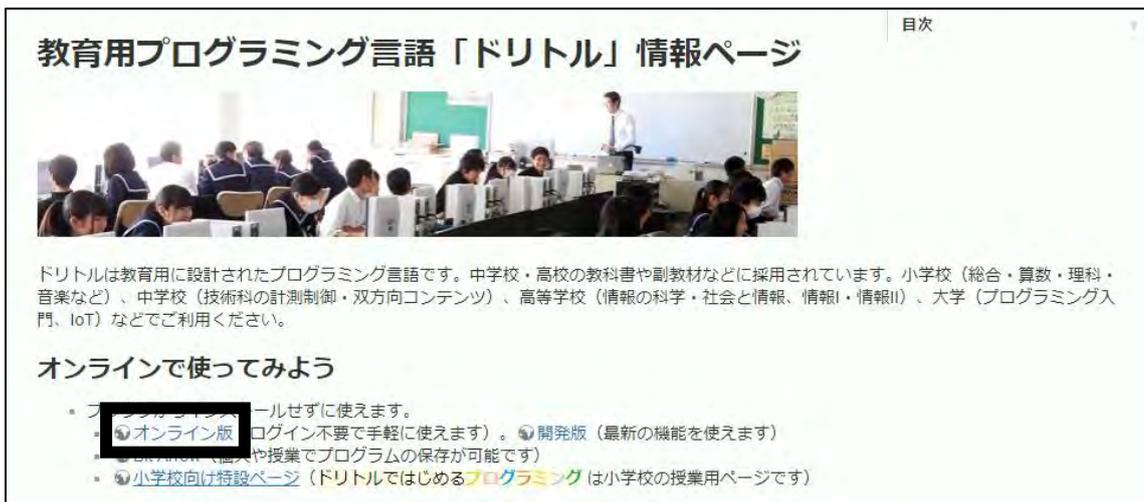
● アルゴリズム体験ゲーム・アルゴロジック <https://home.jeita.or.jp/is/highschool/algo/>

「アルゴロジック」とは、アルゴロジックは一般社団法人 電子情報技術産業協会（JEITA）が開発したアルゴリズムを学ぶソフトウェアである。「アルゴロジック」は、上記のサイトで利用できる。オンラインであれば、すべてブラウザ上で動作ができるので手軽に利用できる。



● 教育用プログラミング言語「ドリトル」情報ページ <http://dolittle.eplang.jp/>

「ドリトル」とは、大阪電気通信大学教授の兼宗進先生により開発された教育用プログラミング言語である。「ドリトル」は、上記のサイトで利用できる。オンライン版とインストール版が用意されているが、オンライン版ではブラウザ上で動作ができるので手軽に利用できる。オンライン版はインストール版より機能は少なくなるが、本稿の実践の内容はオンライン版ですべて実行可能である。



■ 利用したソフトウェア

● Excel VBE

本単元では VBA の演習を生徒にはさせておらず、あくまで教員から提示をするのみにとどめている。しかし、生徒の習得状況や授業時間数によって、本単元の後に生徒に VBA の演習をする単元を設けることもある。

第1, 2時 アルゴリズムとは (アルゴリズム演習)

■ 本時の内容・目標

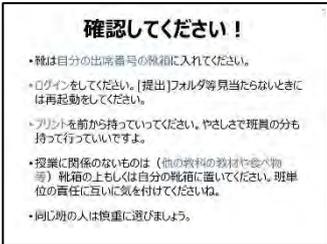
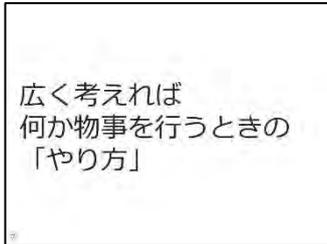
- ・アルゴリズムに興味を持つとともに、アルゴリズムとは何かを理解し、簡単に説明できる。
- ・アルゴリズムの順次処理と反復処理を体験し、理解を深める。

■ 準備物

- ・授業ワークシート
- ・提示用のスライドデータ

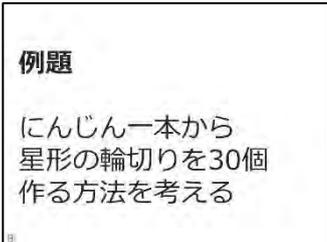
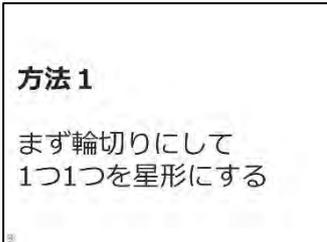
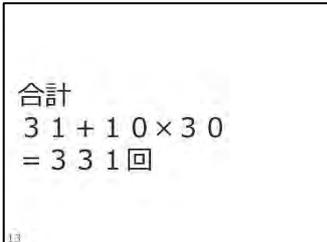
■ 授業の流れ (提示した主なスライドと解説内容)

① 導入

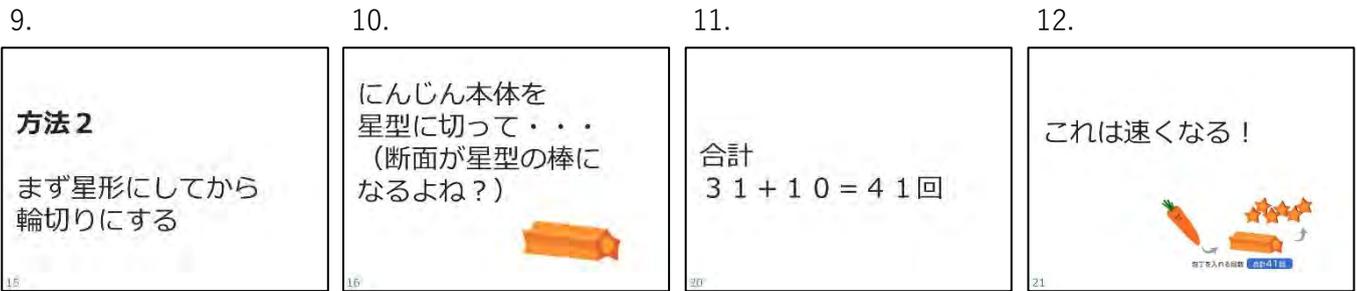
<p>1.</p> 	<p>2.</p> 	<p>3.</p> 	<p>4.</p> 
---	---	--	---

1. 授業開始前において、生徒がPC室に入室した際に確認する事項を提示しておく。
2. 本日はアルゴリズムについて学ぶことを伝える。聴いたことがあるかをたずねるなどして、生徒の授業への参加を促す。
3. 生徒の発表の後に、今回学ぶアルゴリズムとは「問題を解くための処理手順のことである」と伝える。
4. 料理の例を出す。料理は一つ一つの動作をきちんと行っても、処理手順が違えば出来上がりが違ったり、効率が変わったりすることを伝える。

② アルゴリズムとは何かを考える例題

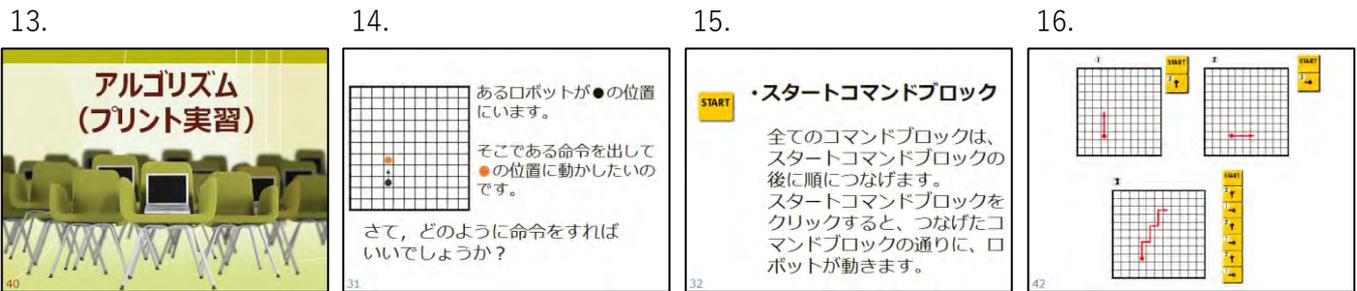
<p>5.</p> 	<p>6.</p> 	<p>7.</p> 	<p>8.</p> 
---	---	--	---

5. アルゴリズムを考える例題として提示する。
6. 方法の1つ目として、「すべてを輪切りした後に、それぞれ1つ1つを星型にする」やり方を考えさせる。このやり方であると、包丁を入れる回数は全部で何回になるかを考えさせる。また、条件として端は落とすとした。生徒には挙手を促し、正解した生徒にはどのようにして計算をしたかを答えさせる。
7. 輪切りを作るのに31回、それぞれを星型にするために10回包丁を入れることを説明する。
8. このやり方では時間がかかってしまうので、他のやり方を検討することを伝える。また、他のやり方を思い出す生徒には、こういったやり方について、考えることができたかを発表させる。

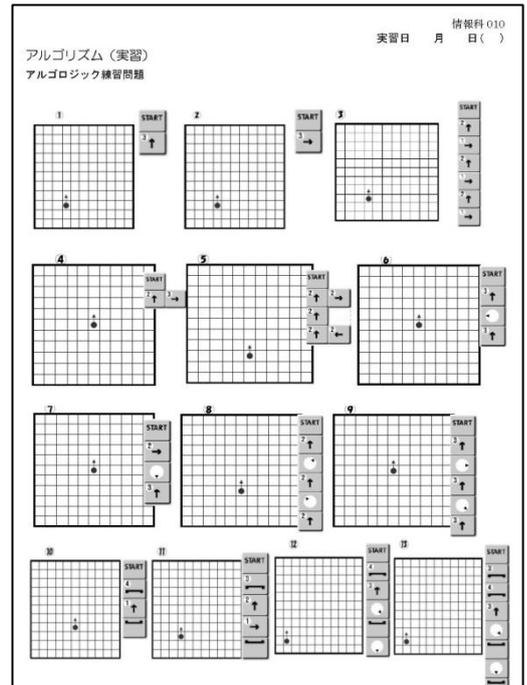


9. 生徒からの発表があれば、それに伴い方法2を提示する。ない場合も一定時間後に提示する。
10. にんじん本体を星形に切ってから、輪切りにする方法が考えられることを伝える。そして、この方法であると、包丁を入れる回数は何回になるかを計算させ、発表を促す。
11. 本体を星形にするのに10回、輪切りにするのに31回となる。
12. やり方を変えることで回数を減らせると解説する。また、「ソフトウェアのプログラムとは『やり方＝アルゴリズム』をプログラミング言語を用いて具体的に記述したものであると述べる。さらに、『やり方』を工夫してより良いアルゴリズムを考えることが、良いプログラミングと言える」と伝え、同じ計算でも良い方法であれば、より早く計算できることと同じであると解説する。

③ アルゴロジックの解説と演習



13. 今日学ぶ教材である「アルゴロジック」はアルゴリズムの考えかたを理解する「ゲーム」とであると強調する。
14. 例題として示す。やり方として以下を伝える。
15. 命令として「スタート」「前進」「左進」「右進」「回転」「繰り返し」がある。それぞれのコマンドブロックを使って、その命令をロボットに伝えることができると述べる。
※「スタート」以外も説明した。本稿では省略します。
16. 右図のプリントを配布する。1問ずつ演習をさせて、発表をさせる。その繰り返しで①～⑬まで演習をする。



④ アルゴロジック Web の問題演習

アルゴロジックのサイトを各個人で見せて演習を行う。◎の評価を出すことができれば、その方法をプリントにメモして後ほど提出をして、評価をする。

■ 出典

- ・アルゴリズム体験ゲーム・アルゴロジック | JEITA ソフトウェアで未来をつくる

<https://home.jeita.or.jp/is/highschool/algo/>

第3, 4時 プログラミング演習（ドリトルの演習）

■ 本時の内容・目標

- ・ 「ドリトル」を通じてプログラミングの体験をする。
- ・ プログラムとは何かを理解し、簡単に説明ができるようになる。

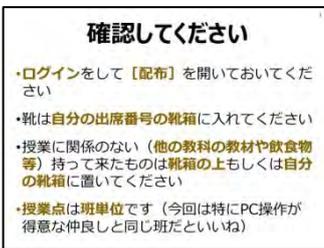
■ 準備物

- ・ 授業ワークシート
- ・ 提示用のスライドデータ

■ 授業の流れ（提示した主なスライドと解説内容）

① 導入

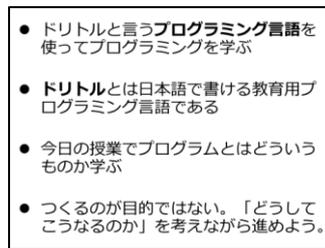
1.



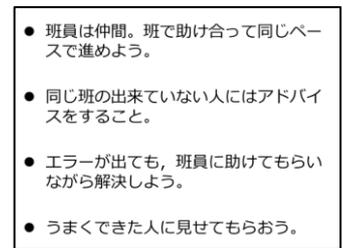
2.



3.



4.



1. 授業開始前において、生徒がPC室に入室した際に確認する事項を提示しておく
2. 本日はドリトルというプログラム言語を利用してプログラミングを行うことを伝える。ドリトルの画面などを見せて、中学時代等に体験したことがあるかを聴き、生徒の授業への参加を促す。
3. 今日の授業目標とは「プログラムがどういうものを学ぶ」ものであるかを述べて、作りあげることが目的ではなく「どうしてこうなるのか」を考えながら実習をすることを伝える。
4. 実習であるため、教員一人ではサポートしきれないと考え、仲間同士で助けあえることを伝える。

② ドリトルと用語の説明

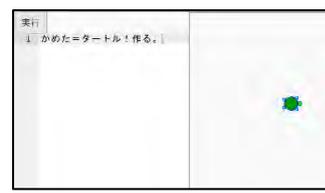
5.



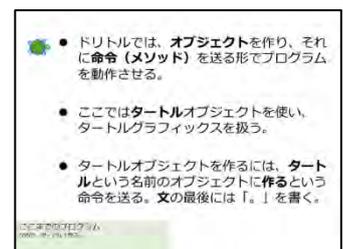
6.



7.



8.



5. 今日は「ドリトル」を使って、宝物拾いゲームの作成をすることを伝える。
6. ドリトル (<http://dolittle.eplang.jp/>) のサイトを開き、オンライン版をクリックするように指示する。
7. 全員が開くことができたことを確認した後に、「かめた=タートル!作る。」（1行名）と入力させ、「実行」をクリックさせる。するとカメラが出てきたことを確認させる。「=」「!」「。」の入力方法がわからない生徒もいることもあるので、互いにその確認をさせる。
8. 「かめた=タートル!作る。」の意味を話し合わせる。その後「オブジェクト」「メソッド（命令）」の説明をし、今回は「タートル!作る。」というオブジェクトに「かめた」という名前を付けたことを説明する。「名前」は「変数」のことであり、数学における変数の意味と関連させて考えさせる。

③ ドリトル実習

```

実行
1 かめた=タートル！作る。
2 左ボタン=ボタン！"左" 作る。
3 左ボタン：動作=「かめた！30 左回り」。
4 右ボタン=ボタン！"右" 作る。
5 右ボタン：動作=「かめた！30 右回り」。
6 時計=タイマー！作る。
7 時計！「かめた！10 歩く」実行。
8 タートル！作る "tulip.png" 変身する ベンなし 100 100 位置。
9 かめた：衝突=「|相手| 相手！消える」。
10 タートル！作る "tulip.png" 変身する ベンなし (乱数(600)-300) (乱数(400)-200) 位置。|
    
```

教員が1行1行入力し（もしくは用意したコピーを貼り）ながら授業を進める。入力が極端に遅い生徒にはサポートをするが、原則は生徒同士で助け合いながら、互いに理解しながら実習を進めていく。

- 2行目～入力後に出来上がったボタンをクリックさせが何も反応がない。それはなぜか考えさせる。
- 3行目～この行の意味は何かを考えさせる。ドリトルはオブジェクトにメソッドを送ることでプログラムが実行することを伝え、「！」によって、命令を送るオブジェクトを指定していることを理解させる。また、「30」をパラメータであると伝え、30°の意味があることを気づかせる。
- 4行目～右ボタンも作ることを伝えるが、入力内容は細かくは指示をせずに考えさせる。多くの生徒がコピーを使わずに入力するため、少し入力したあたりでコピーが使えることを伝える。
- 6行目～入力をさせて、質問が出た場合には詳細に以下の説明をする。タイマーは、あらかじめ指定された間隔で、指定された回数または時間だけ、ブロックを繰り返して実行する。標準では、間隔が0.1秒、回数が100回に設定されている。繰り返す間隔は、「時計！0.2 間隔。」のように「間隔」メソッドで指定する。単位は秒である。繰り返す回数は、「時計！10 回数。」のように「回数」メソッドで指定する。
- 8行目～入力ができると宝物（チューリップ）が表示される。「100 100」は位置を示していることを伝え、XY座標を黒板に書くと多くの生徒は気づく。X、Y座標軸と線対称、原点と点対称にも宝物を配置することを伝え、生徒はこの1行をコピーして配置しようと試みる。また、何名かの生徒は、他の場所にも宝物を配置し始める。気づかない生徒にはヒントを適宜出していく。
- 9行目～宝物を拾ったように見せるために、かめたと宝物が重なったら相手が消えるようにしてみよう、と伝える。また、入力の際には「|」の入力の仕方を教える。
- 10行目～乱数の使い方や、「tulip.png」以外にも「kuno.png」「star.png」「crab.png」「fish.png」が用意されていることを伝え、プログラミングの自由さを体験させる。

■ 授業者からのコメント

- ・ ただ入力をするための実習ではないことを繰り返し生徒に伝え、「どうしてこうなるのか」ということを意識させる発問を入れると良いです。
- ・ 高3文系クラスでの実施など、理系の雰囲気があると嫌がる生徒が多くいる場面での指導もあると思います。本授業ではそういった生徒にもプログラミングを体験させ、以前の単元で学習している情報機器の裏側では、このようにプログラムが動作をしていることを感じさせる実習となると考えます。

■ 出典

- ・ 教育用プログラミング言語「ドリトル」情報ページ <http://dolittle.eplang.jp/>

第5, 6時 順次・繰り返し・分岐 (アルゴリズム2演習とVBA提示)

■ 本時の内容・目標

- ・ プログラムの基本構造として、順次、繰り返し、分岐の処理があることを理解し、簡単に説明できる。
- ・ 他のプログラム言語では「分岐」をどのように示すのかを確認し、共通していることを認識する。

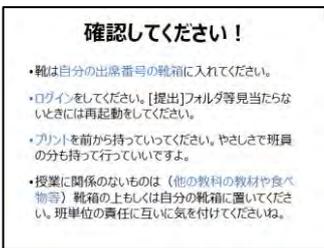
■ 準備物

- ・ 授業ワークシート
- ・ 提示用のスライドデータ

■ 授業の流れ (提示した主なスライドと解説内容)

① 導入

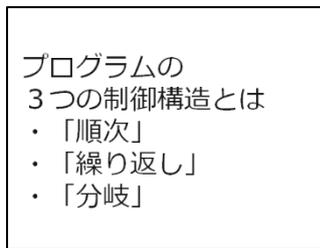
1.



2.



3.



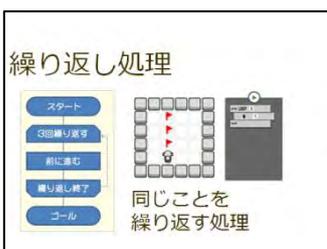
1. 授業開始前において、生徒がPC室に入室した際に確認する事項を提示しておく
2. 本日は前回、前々回に学んだ、アルゴリズム、ドリトルを踏まえて学習をすることを伝える。
3. プログラムには3つの制御構造「順次」、「繰り返し」、「分岐」があり、この3つを組み合わせることで表現されていることを伝える。これらの3つの処理を理解し、説明できるようになることが本時の目的であること伝える。

② 順次・繰り返し・分岐 処理

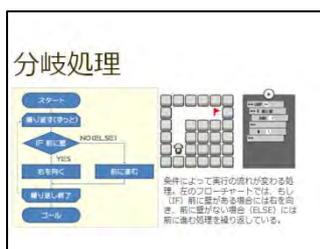
4.



5.



6.



4. 「順次」処理とは順番に命令を実行する処理のことであると説明し、フローチャートとロボットに旗を取らせる動作を見せながら説明をする。
5. 「繰り返し」処理とは同じことを繰り返す処理であると説明し、フローチャートとロボットを前に進む処理を繰り返し実施している動作を見せながら説明をする。
6. 「分岐」処理とは、条件によって実行の流れが変わる処理であると説明する。そして、フローチャートと、もし (IF) 前に壁がある場合には右を向き、前に壁がない場合 (ELSE) には前に進む処理を繰り返す動作を確認させ、その動作を見せながら説明する。

③ アルゴリズム 2 の操作方法の説明と実習

<p>7. ①前進コマンドブロック</p> <p>ロボットが向いている方向に前進するコマンドブロック。カウンタで指定した数だけロボットを前進させる。</p>	<p>8. ④IFコマンドブロック</p> <p>IFは「もし」と言う意味でアルゴリズム2では、ロボットのいるひとつ前のマス調べ、「壁がある場合」にIFコマンドブロックとエンドコマンドブロックにはさまれたコマンドブロックの動作を実行する。</p>	<p>9. ⑤ELSEコマンドブロック</p> <p>ELSEは「そうでなければ」と言う意味で、アルゴリズム2では、ロボットのいるひとつ前のマス調べ、「壁がない場合」にELSEコマンドブロックとエンドコマンドブロックにはさまれたコマンドブロックの動作を実行する。IFコマンドブロックと一緒に使用する。</p>	<p>10. アルゴリズム 2 演習</p> 
---	--	---	---

7.~9. 「前進」「方向転換」「ループとエンド」「IF」「ELSE」コマンドブロックについて説明する。実際に動かす画面を見せながら説明すると分かりやすい。特に IF と ELSE は初めて学ぶ処理（分岐処理）であるため、丁寧に解説する。※紙面の都合上、全部はお示しできません。

10. Web での実習を実施する。配布したワークシートにそれぞれ「◎」の結果となる方法を記録させる。この授業では班単位の実習とし、多くの「◎」を取ることができたチームから高得点とした。

④ 他のプログラムでの「分岐」の表現方法（VBA プログラムの提示）

図 1

```

Sub bunki()
Dim kazu As String
kazu = InputBox("数字を入力してください")
If kazu Mod 2 = 0 Then
MsgBox ("even number")
Else
MsgBox ("odd number")
End If
End Sub
    
```

図 2

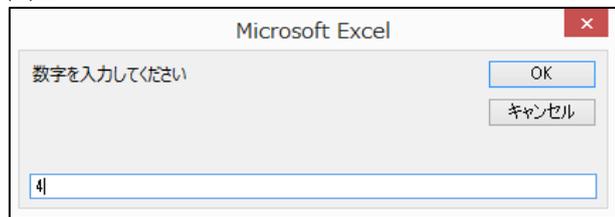
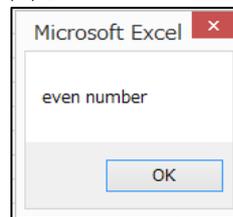


図 3



1. 図 1 の VBA プログラムを見せ、「これは何を表しているのかを考えてみよう」と発問する。
2. もし生徒の挙手があれば、ノーヒントでもいいが、出ないならば、実際にこのプログラムを動かして確認をする。動かすと、図 2 のダイアログボックスが表示され、「2」と入力すると「even number」（図 3）と表示される。次のヒントとしては、「Mod」という命令は割り算の余りを出すものと教える。
3. 「If」「Else」はアルゴリズム 2 で学んでおり、「If～Then～」は英語での意味と同じように考えることを伝える。細かな意味は解らなくても、どういうプログラムであるのかを理解することができる。
4. この問題では、「順次」「繰り返し」「分岐」のどれを示しているのが判断できれば良いとした。

■ 授業者からのコメント

- ・ 生徒による VBA の実習を本授業後に設定すると理解が早いので、時間があれば入れてみてください。

■ 出典

- ・ アルゴリズム体験ゲーム・アルゴリズム <https://home.jeita.or.jp/is/highschool/algo/>

ドリトルを用いたプログラミング授業実践

山本 博之

■ 内容説明

本単元では、ネットワークを利用した情報システムを自らの力で作れることを体験する。情報社会のサービスを作る側の視点について学ぶと共に、情報セキュリティへの意識を高めることが狙いである。情報システムを作るのはハードルが高いように感じるが、教育用プログラミング言語「ドリトル」を用いれば、情報システムの作成を容易に体験できる。本校では、1学期に Javascript を8時間扱っているため、本単元はプログラミングの単元としては2回目となる。

■ 授業の実施情報

学校名	実施科目	実施学年
	情報の科学	高校1年生
時間	1授業あたりの実施人数	使用機材
50分	40人(教員は1名)	パソコン
使用言語	使用ソフト	OS
ドリトル	ドリトル	Windows8.1

■ 単元の指導計画

時間	学習活動・学習内容	学習目標
第1時	● ドリトルを用いたプログラミングの基礎(1) GUI部品を用いたプログラミングについて学習する。命令によってキャラクターの動きが変化することや、繰り返し構造を用いることで同じ操作が用意に繰り返せることを確認する。	<ul style="list-style-type: none"> ・GUIによるプログラミングを理解する。 ・さまざまな命令の意味を理解する。 ・繰り返し構造を使って命令が繰り返されることを理解する。
第2時	● ドリトルを用いたプログラミングの基礎(2) 変数の使い方や、条件分岐によって処理が切り分けられることを学習する。タイマーオブジェクトを用いて、一定間隔で命令を繰り返すことを確認する。	<ul style="list-style-type: none"> ・条件分岐を使って処理が切り分けられることを理解する。 ・変数の使い方について理解する。 ・タイマーの役割について理解する。
第3時	● ネットワークを利用したプログラミング(1) ネットワークを用いたプログラミングを学習する。キャラクターの位置情報を送受信し、マルチゲームを作成する。	<ul style="list-style-type: none"> ・IPアドレスによって通信が行われていることを理解する。 ・サーバーとのデータのやりとりを意識しながらプログラミングできる。
第4時	● ネットワークを利用したプログラミング(2) ネットワークを用いたプログラミングを学習する。文字情報を送受信し、チャットシステムを作成する。1対1だけでなく、複数人でのデータのやり取りができることを確認する。情報セキュリティに関して考える。	<ul style="list-style-type: none"> ・サーバーの役割からセキュリティの必要性を考察できる。 ・自分たちが作成したシステムと、実際の通信サービス(SNS、掲示板など)とを関連付けて考えることができる。

■ 環境設定・使用ソフトの準備

教育用プログラミング言語「ドリトル」のプログラムは、以下のサイトから無料でダウンロードできる。

プログラミング言語「ドリトル」

<http://dolittle.eplang.jp/>

上記サイトからダウンロードページへ行き、OSのバージョンにあったプログラムをダウンロードする。
(本校では、「dorittle32win.zip」を利用した)

ダウンロード	
Windows用 (JRE8同梱版) (約130MB) dolittle32win.zip	<ul style="list-style-type: none"> ・WindowsOSで利用する場合は、通常、こちらをダウンロードしてください。 ・本パッケージのドリトルでは、音楽の演奏をサポートしません。音楽演奏が必要な場合のみ、下記のJRE6同梱版を利用してください。 ・dolittle.batで起動します。 ・Java8(JRE8)が同梱されています。ハードディスクやUSBメモリなどに展開するだけでドリトルを使用できます。 ・旧バージョンと同じ編集画面と実行画面をタブで切り替える画面構成を利用したい場合は、dolittle_tab.batで起動してください。
Windows用 (JRE6同梱版) (約104MB) dolittle32win-jre6.zip	<ul style="list-style-type: none"> ・ドリトルで音楽演奏プログラムを作る場合のみ、こちらのパッケージをダウンロードしてください。 ・Java6(JRE6)が同梱されています。 ・JRE6の影響により、編集画面上の「実行」ボタンを押した時に、実行画面が編集画面の後ろに隠れてしまう場合があります。ご注意ください。 ・基本操作等はJRE8同梱版と同様です。
Macintosh用(約130MB)	<ul style="list-style-type: none"> ・Java8が同梱されています。dmgに含まれる「Dolittle」を「アプリケーション」にコピーするだけでドリトルを使用できます。 ・「アプリケーション」の「Dolittle」をDock(起動メニュー)に登録しておくとう便利です。 ・初めてドリトルを起動したときに、ユーザーごとの「書類」フォルダの下に「Dolittle」というフォルダが作られます。必要に応じて、startup.iniなどのini初期化ファイルや、プ

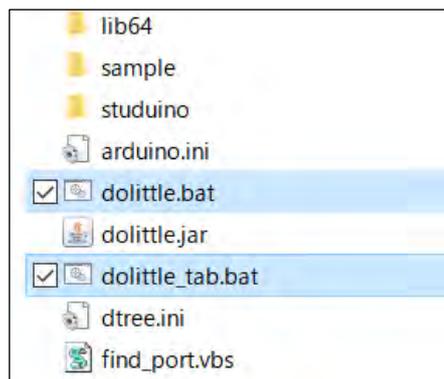
ダウンロードしたファイルは共有フォルダなどに展開しておけば、全員でひとつのプログラムを使うことができるため、それぞれの端末や個人ドライブなどにコピーする必要はない。容量が約130MBあるので、共有フォルダを作成することをおすすめする。

展開されたファイルの、

「dolittle.bat (実行画面とプログラム編集画面が分かれるタイプ)」

「dolittle_tab.bat (実行画面とプログラム編集画面をタブで切り替えるタイプ)」

のどちらかをダブルクリックすることでソフトを起動できる。



第1時 ドリトルを用いたプログラミングの基礎（1）

■ 本時の内容・目標

- ・ GUI 部品の作り方や、ドリトルのさまざまな命令の意味を理解する。
- ・ キャラクターの動きの変化や、繰り返し構造により同じ操作が繰り返されることを確認する。

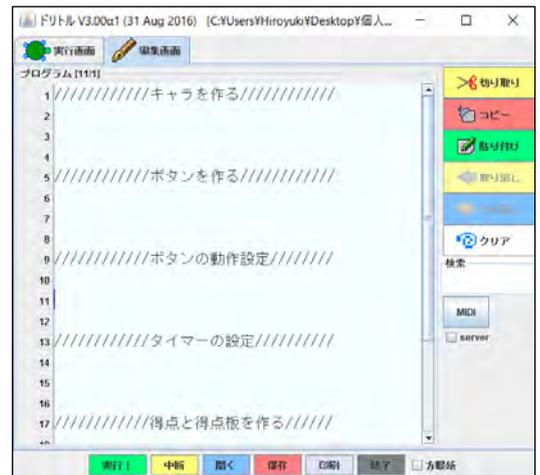
■ 準備物

- ・ 授業ワークシート（本時に使用するオブジェクトに関する説明や、本日作成するプログラムコードを掲載）
- ・ 配布用プログラムファイル

■ 授業の流れ

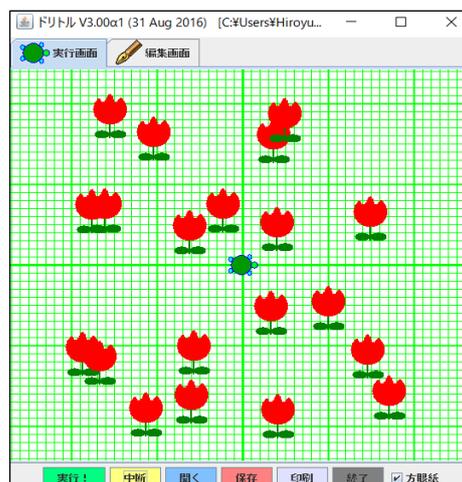
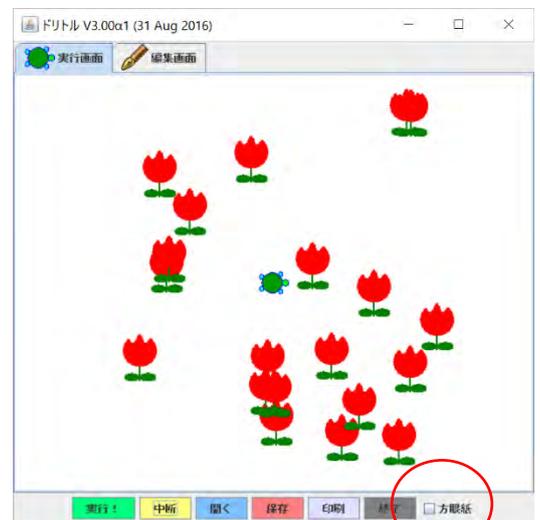
① プログラムファイルの配布とドリトルの起動

1. あらかじめ用意したプログラムファイルを配布する。
(個人フォルダなどにコピーさせる)
2. ドリトルを起動させ、プログラムファイルを読み込ませる。
3. プログラムの編集画面と実行画面について説明する
4. 日本語でプログラムを作成できることを説明する。



② タートルオブジェクトの作成

1. 「かめた」というタートルオブジェクトを作成させる。
2. 「かめた」と同じように「かめこ」の作成を指示する。
3. 「かめこ」の位置をずらし、軌跡を消させる。
4. 方眼紙機能を使い、座標に関する説明する。
5. 「かめこ」を tulip.png に変身させる。
(画像の代わりに@twitter アイコンに変身させても良い)
6. 「かめこ」の位置情報に乱数を用いて、さまざまな方向に「かめこ」を出現させる。
7. 繰り返し文を使い、「かめこ」を 20 匹出現させる。



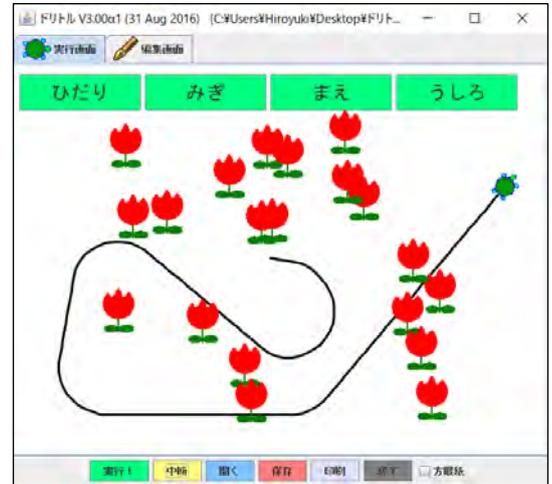
③ ボタンオブジェクトの作成

1. 「左」という名前のボタンを作成させる。
2. 「右」「前」「後」という名前のボタンを作成させる。



④ ボタンオブジェクトへの動作設定

1. 「左」という名前のボタンの動作に、「かめた」が10度左方向に回る命令を登録させる。
2. 「右」に右周り、「前」に10歩く、「後」に-10歩く命令を登録するように指示する。
3. プログラムを実行させ、「かめた」が予想通りの動きを確認させる。
4. 左ボタンに LEFT を定義し、キーボードの「←」キーで操作できるようにさせる。
5. 「右」に RIGHT、「前」に UP、「後」に DOWN を定義するよう指示する。
6. キーボードの上下左右キーで、「かめた」が操作できることを確認させる。



⑤ プログラムファイルの保存とドリトルの終了



■ プログラム

```

1 ///////////////////////////////////////////////////キャラを作る//////////////////////////////////////
2 かめた=タートル!作る。
3
4 「
5 かめこ=タートル!作る ペンなし (乱数(400)-200) (乱数(400)-200) 位置 "tulip.png" 変身する。
6 」!20 繰り返す。
7
8 ///////////////////////////////////////////////////ボタンを作る//////////////////////////////////////
9 左=ボタン!"ひだり" LEFT"作る。
10 右=ボタン!"みぎ" RIGHT"作る。
11 前=ボタン!"まえ" UP"作る。
12 後=ボタン!"うしろ" DOWN"作る。
13
14 ///////////////////////////////////////////////////ボタンの動作設定//////////////////////////////////////
15 左:動作="かめた!10 左回り"。
16 右:動作="かめた!10 右回り"。
17 前:動作="かめた!10 歩く"。
18 後:動作="かめた!-10 歩く"。
    
```

Annotations in the image:

- Red box around lines 2-6 points to circled number 2.
- Red box around lines 9-12 points to circled number 3.
- Red box around lines 15-18 points to circled number 4.

■ 留意点・ポイント

- ・「かめた」、「かめこ」といった変数は自由に決めさせても良いが、あとで変数を呼び出す際に自ら決めた変数を使用するよう注意が必要。
- ・基本的に全角入力で問題はないが、画像や twitter アイコンの指定、キーボードの割り当てなどは半角にする必要がある。
- ・繰り返し回数を10000回などと指定する生徒がいて、ドリトルがフリーズする場合がある。

第2時 ドリトルを用いたプログラミングの基礎（2）

■ 本時の内容・目標

- ・変数の使い方や、条件分岐によって処理が切り分けられることを学習する。
- ・タイマーオブジェクトを用いて命令を一定間隔で繰り返すことを確認する。
- ・プログラムを自ら改変し、実行結果の変化を確認できる。

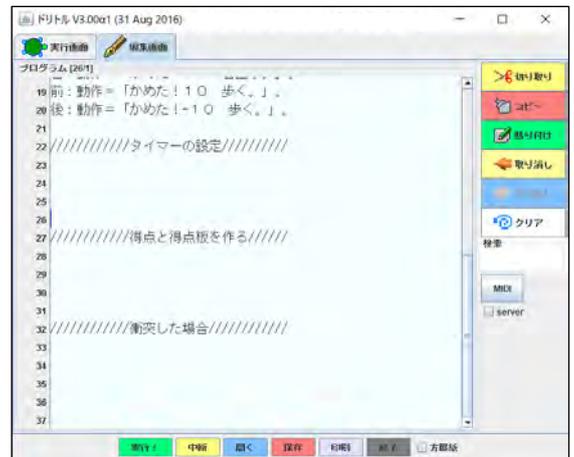
■ 準備物

- ・前回の授業ワークシート
- ・前回のプログラムファイル / 欠席者用のプログラムファイル

■ 授業の流れ

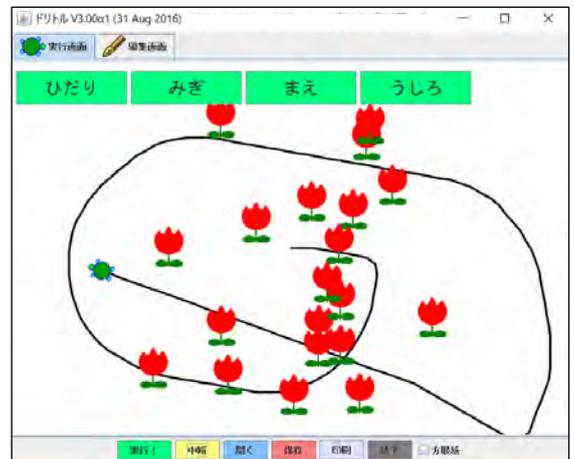
① ドリトルの起動とプログラムファイルの読み込み

1. ドリトルを起動させ、前回のプログラムファイルを読み込むように指示する。
2. 起動やプログラムファイルの読み込み方法を忘れた生徒のために中央モニタで説明する。



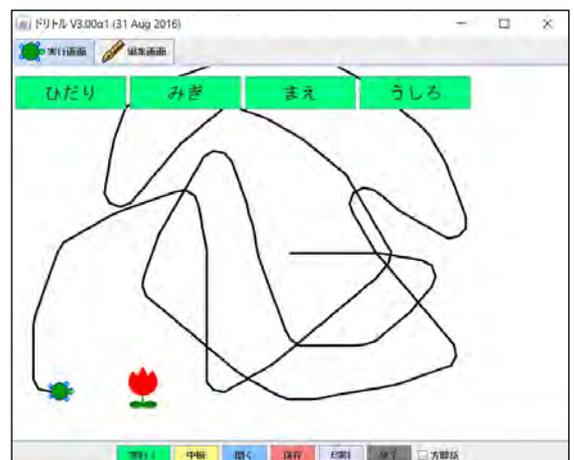
② タイマーオブジェクトの作成

1. 「時計」という名前のタイマーを作成させる。
2. 「時計」の設定をさせ、何分間動作するかを考えさせる。
3. 「時計」で動かす命令を書かせる。
(「かめたが前方向に8歩く命令」)
4. プログラムを実行させ、「かめた」が自動で動くことを確認させ、実行後すぐにキーボードの「←」「→」でコントロールするよう指示する。



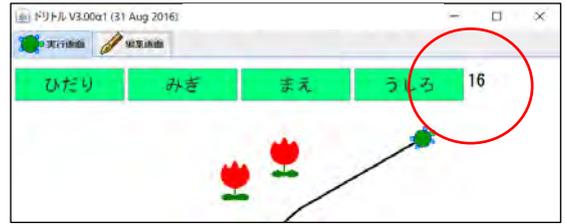
③ 衝突した場合の処理

1. ぶつかったオブジェクトを消す命令を書かせる。
(「|相手|については深く触れない」)
2. プログラムを実行させ、「かめた」にぶつかったオブジェクト（「かめこ」）が消えることを確認させる。



④ ラベルオブジェクトの作成

1. 「得点」という名前の変数を作成し、0を代入させる。
2. 「得点板」という名前のラベルオブジェクトを作成し、「得点」を表示させる。

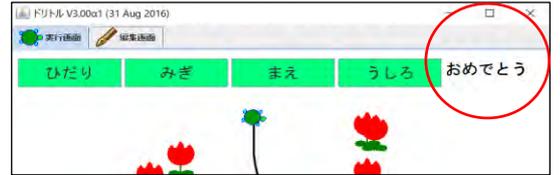


⑤ 得点のカウントアップするプログラムを作成

1. 衝突判定の中に、「得点」をカウントアップする命令を書かせる。
2. 「得点板」に「得点」を表示させる。
3. プログラムを実行させ、衝突した場合に「得点」がカウントアップされることを確認する。

⑥ 条件分岐によるプログラムを作成

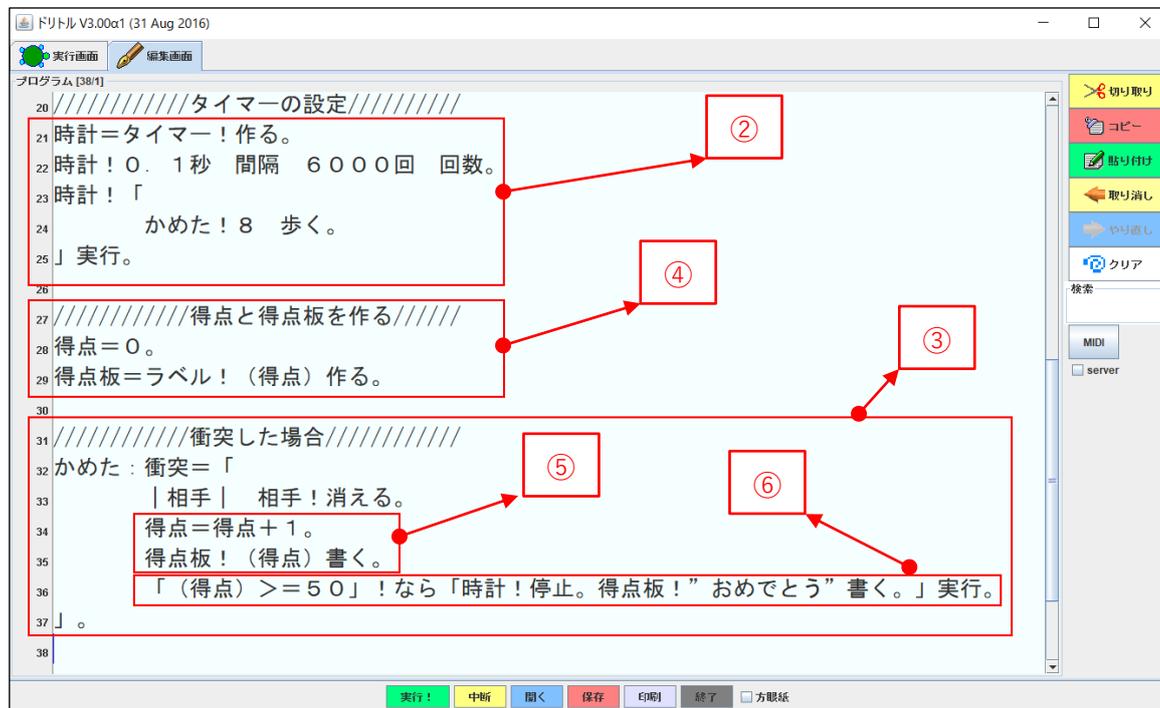
1. 条件分岐の文法を確認させる。（「」！なら「」実行。）
2. 「得点」が10以上を条件にさせる。
3. 条件が成り立った場合、「時計」を停止させ、「得点板」に おめでとう を表示させる。
4. プログラムを実行させ、条件分岐通りに動くかを確認させる。



⑦ 今までのプログラムのアレンジ

1. キャラクターのスピードを変えたり、画像を変更したり、今まで記述したプログラムを改変させる。

■ プログラム（前回の続きなので、前回のプログラムの後半部分）



■ 授業者からのコメント

- ・前回、保存ができていない生徒や、欠席した生徒用のプログラムファイルを作っておくと良い。
- ・プログラムをただ写すだけでなく、最後にはオリジナルを考えさせる時間を必ず取るようにする。
- ・プログラムのアレンジが思い付かない生徒には、スピードを変えるなどのヒントを与えると良い。

第3時 ●ネットワークを利用したプログラミング（1）

■ 概要

- ・ネットワークを用いたプログラミングを学習する。
- ・キャラクターの位置情報を送受信し、マルチゲームを作成する。
- ・IP アドレスによって通信が行われていることを体験しながら理解する。

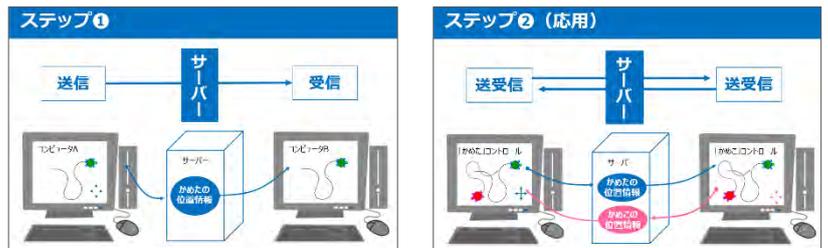
■ 準備物

- ・授業ワークシート（作成するプログラムコードを掲載）
- ・配布用プログラムファイル

■ 授業の流れ

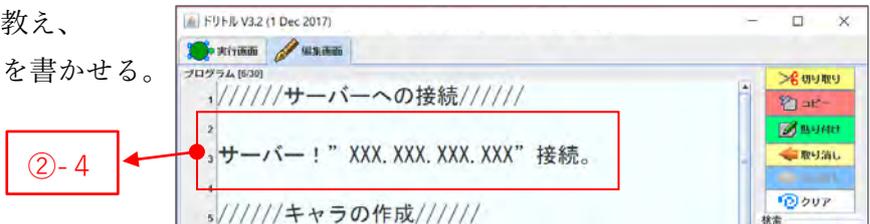
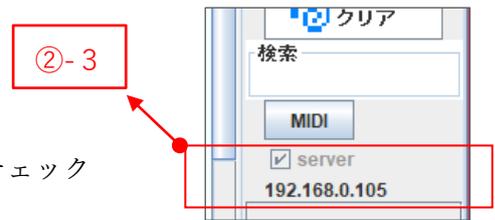
① 本日の実習についての説明

1. ステップ①の実習について説明する。
2. ステップ②の実習について説明する。
3. サーバーの役割を説明し、サーバー役を決めさせる。



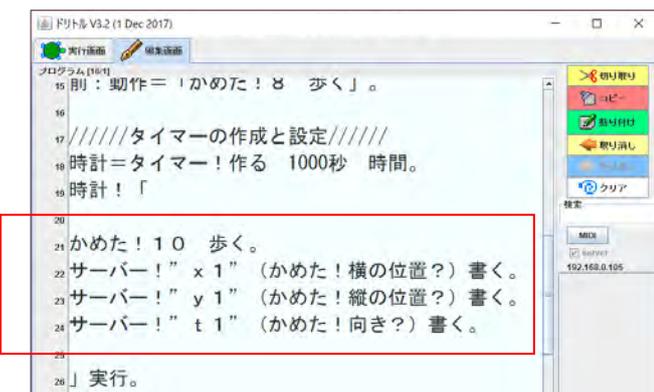
② ドリトルの起動とプログラムファイルの読み込み

1. あらかじめ教員が用意したプログラムファイルを配布する。
2. ドリトルを起動させ、プログラムファイルを読み込ませる。
3. サーバー役の生徒に、ドリトルをもう一つ起動させ、server にチェックを入れさせる
4. 表示された IP アドレスをペアに教え、お互いプログラムに IP アドレスを書かせる。

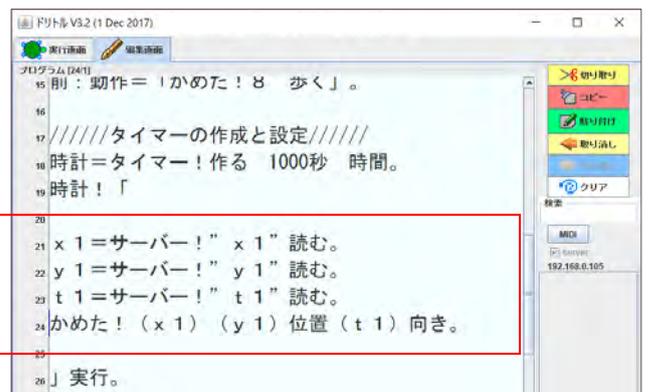


③ ステップ①の実習を行わせる

1. 送信側と受信側のどちらを作成するかを決めさせ、プリントのプログラムを写させる。
2. 写しながら、プログラムの意味を考えるように指示する。
3. プログラムを書き終えたペアは、同時にプログラムを実行し、自分のキャラクターがペアの画面でも動いていることを確認させる。



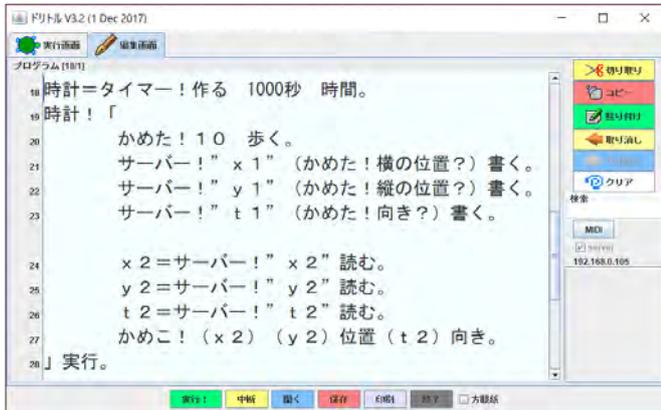
送信側のプログラム



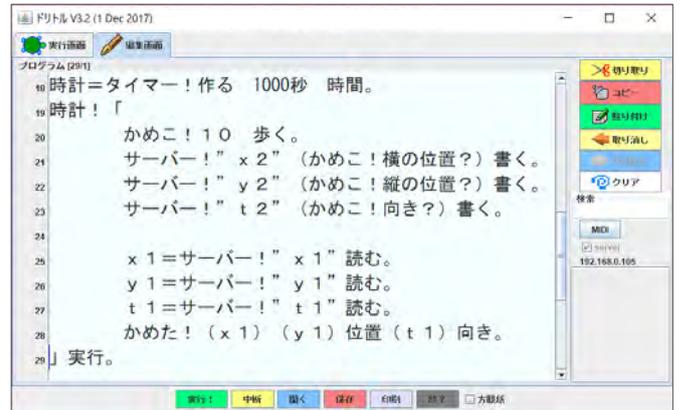
受信側のプログラム

④ ステップ①が完成したペアに、ステップ②の実習を行わせる

1. お互いのキャラクターを相手画面でも動かすにはどうしたら良いかを考えるよう指示する。
2. x 2、y 2、t 2 という新しい変数を作るようにヒントを出す。



「かめた」側のプログラム

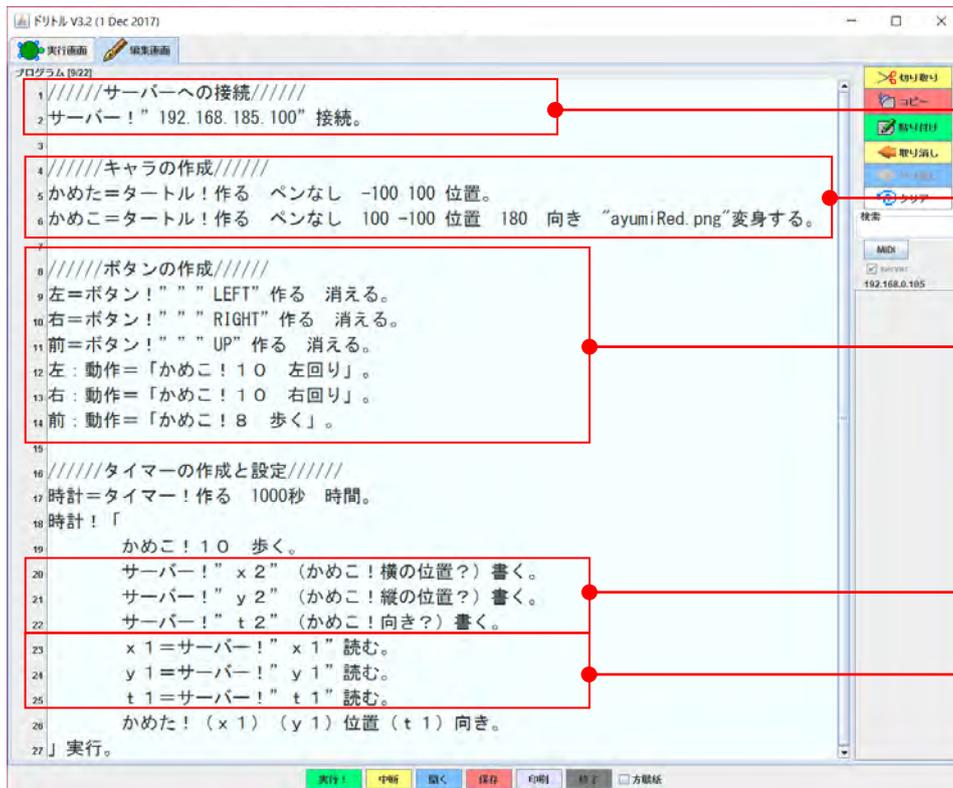


「かめこ」側のプログラム

⑤ 今までのプログラムをペアでアレンジする

1. キャラクター同士がぶつかった場合にキャラクターが消えるようにプログラムを改変させる。
2. 得点表示や時間制限などの機能を追加するなどのアレンジを考えさせる。

■ プログラム（ステップ②の完成プログラム<かめこ側>）



サーバーへの接続のため、IP アドレスを指定する。

「かめこ」「かめた」を作成する。

キャラクターの動作をキーボードに登録する。

自分キャラクターの位置情報をサーバーに送信する。

相手キャラクターの位置情報をサーバーから読み込み、相手キャラクターに反映させる。

■ 授業者からのコメント

- ・使用する変数「x 1」「y 1」「t 1」は全角で統一するよう指示をすると良い。
- ・「時計」を二つ作ってしまうペアに、「時計」は増やさないように指示をする。
- ・机間指導をしながら、うまくいかないペアをサポートする。

第4時 ●ネットワークを利用したプログラミング（2）

■ 概要

- ・ネットワークを用いたプログラミングを学習する。
- ・文字情報をサーバーに送信し、チャットプログラムを作成する。
- ・IP アドレスによって通信が行われていることを体験しながら理解する。
- ・通信プログラムを自ら作成することで、情報セキュリティへの意識を高める。

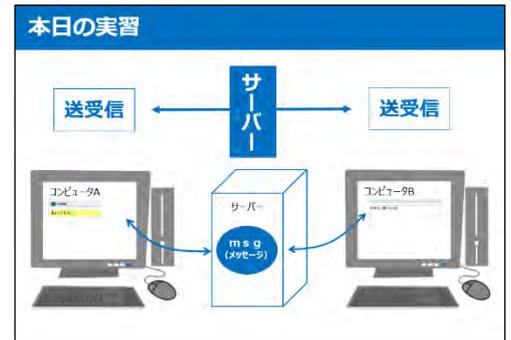
■ 準備物

- ・授業ワークシート（作成するプログラムコードを掲載）
- ・配布用プログラムファイル

■ 授業の流れ

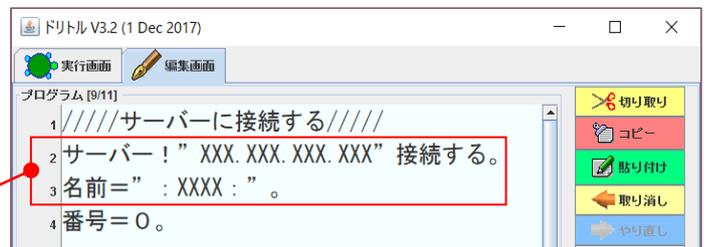
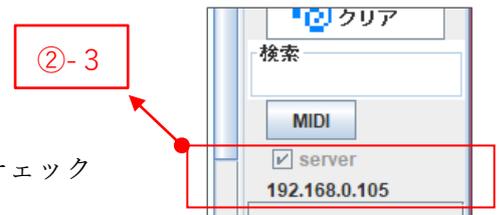
① 本日の実習についての説明

1. チャットプログラムの作成について説明する
2. 2人1組を作らせ、サーバー役を決めさせる。



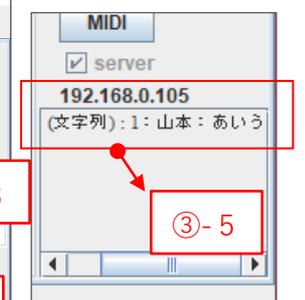
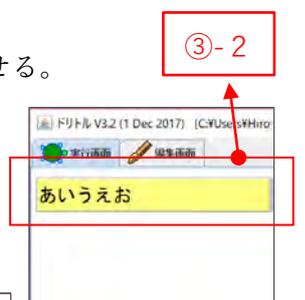
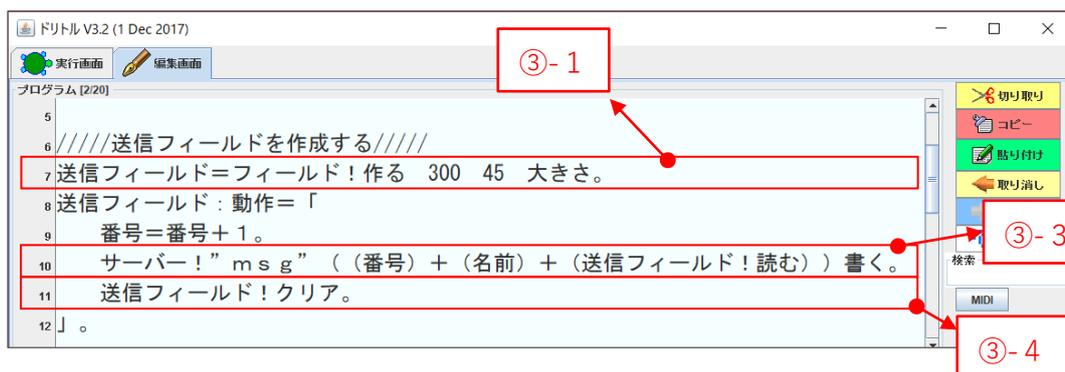
② ドリトルの起動とプログラムファイルの読み込み・設定

1. あらかじめ教員が用意したプログラムファイルを配布する。
2. ドリトルを起動させ、プログラムファイルを読み込ませる。
3. サーバー役の生徒に、ドリトルをもう一つ起動させ、server にチェックを入れさせる
4. 表示された IP アドレスをペアに教え、お互いプログラムに IP アドレスを書かせる。
変数「名前」に自分の名前を書かせる。
(ニックネームでも可とする)



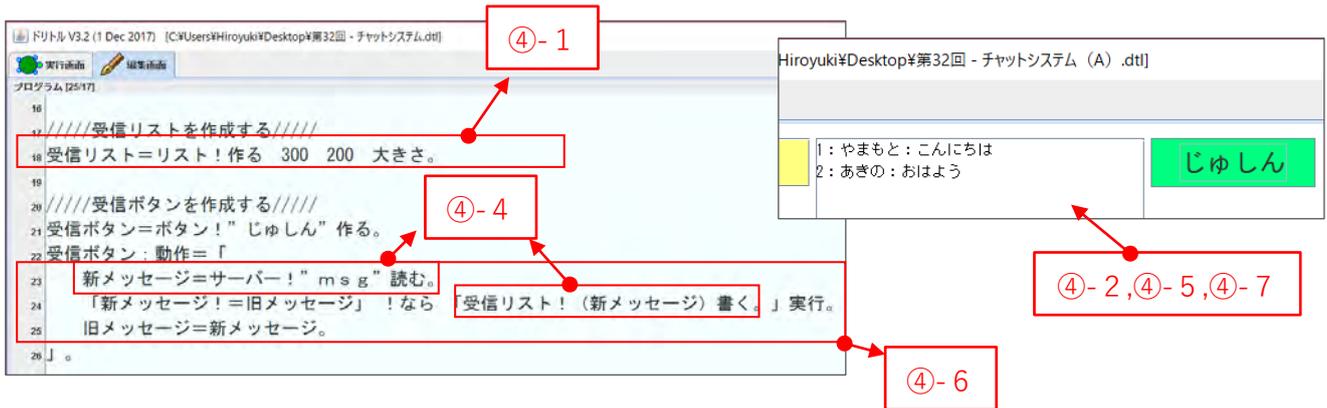
③ 送信画面の作成

1. 送信フィールドという名前のフィールドオブジェクトを作成させる。
2. プログラムを実行させ、フィールドオブジェクトがどのようなものかを確認させる。
3. 送信フィールドに対し、サーバーへメッセージを書きこむ動作を登録させる。
4. 送信後に、送信フィールドをクリアする命令を書かせる。
5. プログラムを実行させ、サーバーにメッセージが書かれているかを確認させる。



④ 受信画面の作成

1. 受信リストという名前のリストオブジェクトを作成させる。
2. プログラムを実行させ、リストオブジェクトがどのようなものかを確認させる。
3. 受信ボタンという名前のボタンオブジェクトを作成させる。
4. 受信ボタンにサーバーからのメッセージをリストに表示させる動作を登録させる。
5. 隣同士でメッセージを送受信させ、正しくプログラムが動作するかを確認させる。
6. 新着メッセージだけを受信するプログラムを追記させる。
7. プログラムを実行させ、新着メッセージだけが受信されることを確認する。



⑤ 3人以上での通信とメッセージの自動受信

1. 受信ボタンを時計オブジェクトに切り替えさせる。
2. 3人以上でグループチャットができるかを確認させる。

■ プログラム (完成プログラム<自動受信バージョン>)

The screenshot shows the completed code with the following annotations:

- サーバーへの接続のため、IP アドレスを指定する。送信者の名前を登録する。** (Server connection for IP address specification and sender name registration.)
- 送信するためのオブジェクトを作成する。送信する情報は、番号、名前、メッセージを1セットとする。** (Create objects for sending. Information to be sent is a set of number, name, and message.)
- 受信するためのオブジェクトを作成する。** (Create objects for receiving.)
- 自動受信のための動作を作成する。新着メッセージ以外は受信しないように、分岐構造を使う。** (Create actions for automatic reception. Use branching structure so that only new messages are received.)

■ 授業者からのコメント

- ・サーバーの役割がわかっていないペアがそれぞれの IP アドレスを書くので注意が必要。
- ・最後のまとめでは、実習の場面から情報セキュリティに関連する話をするが良い。

大学におけるプログラミング教育 ～理工学系学部の初年次導入科目を例に～

赤澤紀子

■ はじめに

電気通信大学では、2010年度より、情報基礎教育として「コンピュタリテラシ」と「基礎プログラミングおよび演習」を開講している。プログラミングに関しては主に「基礎プログラミングおよび演習」で学習する。学生は、情報基礎教育を礎として、一年次後学期以降には、類ごとのコースに対応して様々なプログラミングを扱う科目を履修する。本稿では、「基礎プログラミングおよび演習」を例に大学におけるプログラミング教育について紹介する。また、大学の専門によらない一般情報教育におけるプログラミング教育についても紹介する。

■ 本学における情報基礎教育

電気通信大学は、「Ⅰ類（情報系）」「Ⅱ類（融合系）」「Ⅲ類（理工系）」という3つの「類」を持つ理工学系の大学である。そのなかで「基礎プログラミングおよび演習」は、必修の初年次導入科目として、2010年度から、全学共通に開講されている。2017年度は、従来の講義と比べて、学生が講義に取り組みやすい学習環境の整備や課題の出題方法、各回の学生の取り組みに対する教員のフィードバックを行うなどの運用を改善した。

また、本学で取得可能な数学科、理科、情報科のどの教員免許を取得する場合も、「基礎プログラミングおよび演習」は、教科に関する科目として必修の科目である。

■ 講義と進め方

学習目標は、必要な基礎知識を Ruby および C 言語を用いて習得し、簡単なプログラムの作成と読解ができるようになること、および、基礎的なアルゴリズムの理解や、ソフトウェアの開発方法を理解し、問題解決の基盤となる思考能力を身に付けることである。さらに、「プログラムを自力で書く」ことが学習の中心となるように設計されている。

この講義は、一回 90 分の授業で、15 回の構成になっている。各回の内容を表 1 に示す。第 1 回から 10 回までの学習には Ruby を用いる。Ruby

でプログラミングの入門要素から始め、ある程度プログラムを作ることに慣れ、第 11 回から 15 回で C 言語に触れる。

15 回の授業のうち、Ruby、C 言語それぞれ 1 回ずつ、総合演習を用意した。総合演習では、自分が設計した仕様を実装する体験を行い、システム開発の基礎力を養うことを目標としている。Ruby（第 6 回）では、自身で作る画像を設計し、直線、多角形、円などを作り、画像を実装する。C 言語（第 15 回）では、チームによる開発を行う。これにより単にプログラミングを行うだけでなく、チームでソフトウェアを開発する際に注意すべきことを知り、プロジェクト開発の流れに触れる。

学生はそれぞれの回において、(1)予習、(2)授業、(3)課題提出の3つを行い、その回のテーマを扱うプログラムを自力で作成する。

(1) 予習：学生は、LMS(Learning Management System)に用意されている予習用の動画[1]とテキスト[2]を使い予習を行う。予習用の動画は、図 1 のように、説明スライド、教員説明、コンソールの 3 パートから構成されている。説明スライドに沿って、教員が説明を行い、必要に応じて、コンソールを使った実演を行う。テキストは、図 2 のようになっている。冊子も販売されており、学生は必要に応じて購入する。



図 1 予習用動画

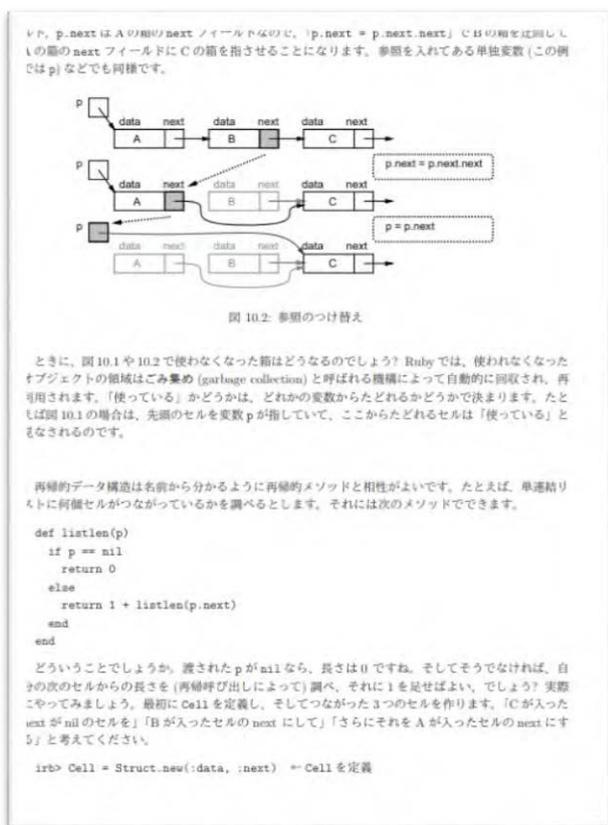


図 2 テキストの例

(2) 授業：授業の初めの 10 分程度で LMS に用意された前回の復習確認問題に取り組む(図 3)。その後、教員からの予習の補足の説明など、その回の課題に取り組むための説明を受ける。学生は予習で、わからなかったことなどを教員やティーチングアシスタントに確認しながら、課題に取り組む。授業終わりに、授業の取り組みの報告を LMS にて行う。

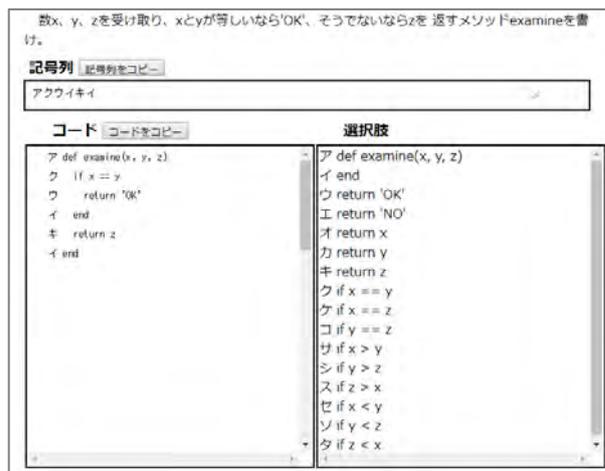


図 3 予習確認問題の出題例

(3) 課題提出：授業後、各回に設定されている複数の課題からいくつかの課題を選び、課題に沿ったプログラムを作成実行し、レポートにまとめ、LMS にて提出する。様々な熟達度の学生も、適切な課題に挑戦できるように、課題は、易しいものから難しいものまで多く用意している。学生は、自分のレベルに合った課題を選択し取り組む。このレポートでは、単にプログラム記載するだけでなく、学生自身が考えたことを記述する。

表 1 講義内容

回	カテゴリ
1	プログラミングとは、演算と式
2	枝分かれと繰り返し
3	制御構造の組合せ、配列とその利用
4	手続き/関数
5	2次元配列
6	総合実習(画像の生成)
7	整列アルゴリズム
8	乱数とランダム性
9	オブジェクト指向
10	動的データ構造
11	型と宣言
12	アドレスとポインタ型
13	文字列の扱い
14	構造体とデータ構造
15	総合実習、ソフトウェア開発

■ 学習環境

授業は、各クラス 60 名前後の学生で構成している。すべてのクラスで、本学教員が 1 名ないし 2 名で担当し、大学院学生のティーチングアシスタントを 2 名配置している。LMS として、学内の仮想マシン上に構築されている Moodle を使用している。また、Ruby、C 言語の実装および実行は、学内の教育系サーバで提供している Linux 環境を用いて行う。これらの環境は、学生は、学内外から利用可能なため、学生は場所時間を問わず、課題に取り組むことができる。

LMS には、「各回のテキスト」「予習用の動画」「予習用確認問題」「課題提出」などを提供している。Ruby では、irb(Interactive Ruby)を用いて、対話的に、手軽に実行できるようにした。

■ 評価

成績評価には、各回のレポートと学期末に行う試験を用いた。各回のレポートは、担当の教員が成績評価を行うだけでなく、必要に応じて、クラス全体へのフィードバックを行う。学期末試験は、授業でも活用している LMS を利用して、コンピュータ上で実施する。

出題は、プログラム作成能力を評価することを目標として開発された「短冊方式」(図 4)を使用している。この方式は、単純な選択肢式では評価しづらい「問われたものを組み立てる」力が測れることが期待される。出題例を図 3 に示す。回答者は右の選択肢フィールドから左のコードフィールドに選択肢をドラッグ&ドロップして解答を組み立てるようになっている。この出題形式は、前回の復習の確認問題でも使用している。

出題する問題は、第 1 回から第 14 回までの内容に対応した問題を難易度の易しい問題と難しい問題の 2 種類(14×2 題)を出題した。易しい問題と難しい問題の分類の中でも、難易度はカテゴリによって異なるが、プログラミングが得意でないと思っている学生にもある程度の点数が取れるようにし、プログラミングが得意な学生でも早く終わって時間を持て余すことのない配慮した。

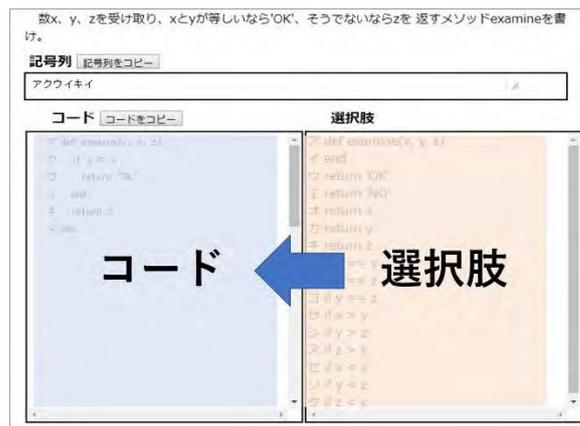


図 4 短冊方式の出題

優しい問題と難しい問題の出題概要の例を示す。

- # 1 プログラミングとは、演算と式
 - 易しい問題) 引数 x を受け取り、作業変数 y を併用して $x^2 + C$ 程度の計算を行う
 - 難しい問題) 引数 x を受け取り、作業変数 y を併用して $x^m + x^n$ 程度の計算を行う
- # 6 図形の生成
 - 易しい問題) 円や長方形を塗るルーチン呼び重なりのない配列の図を再現する
 - 難しい問題) 円や長方形を塗るルーチン呼び重なりのある配列の図を再現する
- # 10 動的データ構造
 - 易しい問題) 単連結リストの指定位置への挿入や削除などを行う
 - 難しい問題) 再帰や相互再帰により数値の単連結リストから奇数番の合計などの値を得る

各問題の解答率をみると、全体的に易しい問題の解答率が高い結果となった。また、正解率を見ると、初歩的な内容を意図した問題に加えて、ループやロジックを構成するなどある程度プログラミングの力を要求する問題の解答率が高い傾向にあった。

■ 一般情報教育でよく使われるプログラミング言語

本稿の主題は、理工系学部の情報基礎教育についてであるが、ここで、学部によらない大学1年次の一般の情報教育でのプログラミング学習についても紹介する。

一般社団法人情報処理学会事業実施統括委員会は、一般情報教育についての調査を行っている[3]。ここでは、大学単位や学部、キャンパス単位での回答を得ている。これによれば、回答のあった739組織のうち、118組織が一般情報教育にプログラミングを取り入れていると回答している。

また、教育用言語として表2の結果が得られている。単に授業で紹介しているだけでなく、学生が、その言語で書かれた簡単なプログラムを理解できるレベル以上に達しているようなプログラミング言語について、順位をつけて回答することを求めた。回答の1位を5点、2位を4点というように重みづけをして集計した結果は以下の通りとなっている。言語は、以下からの選択となっている。

C, C++, Java, Visual Basic/VBA, Fortran, Ruby, JavaScript, SQL, PHP, Scheme, Lisp, Prolog, Python, アセンブリ言語, 機械語。

同調査研究では、情報専門学科の教育用プログラミング言語についても調査を行っている。一般情報教育との比較として、情報専門学科で使用している言語を表3に示す。言語の選択肢が異なるため、表2の選択肢にない言語も入っている。

同調査研究では、情報専門学科において、C言語やJavaは定番言語として採用事例が多いと考えられ、RubyやPython等は比較的新しい言語であるが、単純性やソフトウェア生産性の高さからプログラミング教育への導入事例も比較的多いとしている。この傾向は、一般情報教育でも同様であるといえる。

表2 一般情報教育で扱うプログラミング言語

言語	総合得点
1: C	243
2: Visual Basic/VBA	209
3: Java	178
4: JavaScript	126
5: C++	55
6: Ruby	48
7: Fortran	35
8: SQL	33
9: Python	25
10: PHP	24

表3 情報専門学科で使用されている言語

言語	
1: C	11: Prolog
2: Java	12: Lisp
3: C++	13: Scheme
4: JavaScript	14: 機械語
5: アセンブリ言語	15: Processing
6: SQL	16: Matlab
7: Python	17: Fortran
8: Visual Basic/VBA	18: C#
9: PHP	19: Haskell
10: Ruby	20: Ocaml

■ 参考文献

- [1] 電気通信大学共通教育部情報部会：基礎プログラミングおよび演習2017 動画, <<https://joho.g-edu.uec.ac.jp/joho/fp2017/video.html>> (参照 2018-04-07).
- [2] 電気通信大学共通教育部情報部会：基礎プログラミングおよび演習2017 テキスト, <<https://joho.g-edu.uec.ac.jp/joho/fp2017/fp2017-180203.pdf>> (参照 2018-04-07).
- [3] 一般社団法人情報処理学会事業実施統括委員会：超スマート社会における情報教育の在り方に関する調査研究[文部科学省先導的・大学改革推進委託事業]平成28年度報告書, pp.1-80, 一般社団法人情報処理学会(2017.03)

大学におけるプログラミング教育 ～情報専門学科を例に～

角田博保

はじめに

電気通信大学の情報専門学科において 2016 年度まで 25 年以上に渡って担当していたプログラミング通論について概説する。この科目は 2 年生向けのプログラミング教育科目（必修）であり、その前にプログラミング入門科目（必修）が開講されている。プログラミング入門に続く専門学科のプログラミング教育科目の一例として紹介する。

講義概要

プログラミング通論は 25 年前から講義内容はほとんど変わっていない。初期は Pascal 言語を使っていたが、2002 年度から C 言語に変わった。それは後続の実験科目で C 言語を使ったシステムソフトウェア開発などを行うという要望もあったからである。

プログラミング通論は 90 人程度の履修生を相手に開講される必修科目である。最新の講義内容は以下のとおりである（1 回あたり 90 分、半年の講義）。

回	内容
1	C 言語の基本機能の復習
2	基本的データ型(1)ポインタ
3	基本的データ型(2)スタック、キュー、デク
4	再帰呼出し(1)関数と引数渡しの復習、再帰の概念
5	再帰呼出し(2)分割統治法、実行の仕方(スタック・配列表現)
6	再帰呼出し(3)再帰呼出しの除去
7	復習および中間試験
8	リスト構造(1)リストの定義、基本操作
9	リスト構造(2)リストの応用
10	リスト構造(3)抽象データ型としてのリスト
11	整列(1)基本整列法（選択ソート、挿入ソート、バブルソート）
12	整列(2)高速手法（クイックソート、ヒープソート）
13	整列(3)基数整列法、マージソート
14	探索 線型探索、2 分探索
15	進んだ話題
16	期末試験

主題は、プログラミングの初歩は学習したという前提で、再帰の手続き、データ構造の初歩、および、基本的アルゴリズムについて学習することである。

達成目標は、再帰の手続き、基本的なデータ構造、基本的なアルゴリズムを理解し、それらを用いた C 言語のプログラムを読むこと、書くことができることである。知っているというレベルではなく、使えるというレベルを目指した。

参考書としては、R. セジウィック著、野下ら訳：アルゴリズム C 第 1 巻＝基礎・整列、近代科学社（1996）を用いた。

講義環境

プログラミング通論には対応した演習科目がある。したがって、通論の講義中に演習をする必要はないが、形成的評価のため小テストをほぼ毎回実施した。採点結果（添削付き）は次回返却した。授業時間内に収まらないことが明らかな場合は次の日までの宿題とすることもあった。

講義は年度によって、教室にて板書やプロジェクタを使う場合も、計算機室にてオンラインで実施する場合もあった。

教室の方が、気を散らさずに、丁寧に教えることができる。黒板は広いので、板書を工夫することで、いくつかの項目を同時に示し、関連付けて講義できるという利点がある。しかし、遠くに座った学生は板書が読みづらいし、プログラムなど長いものは板書する時間がロスタイムとなりうる。プロジェクタならその点は改良されるが、黒板の大きさほどのスクリ

ーンはないので、提示の量はかなわない。また、集団での講義なので、できの良い学生にとっては講義の進度が遅く、退屈な学生もでてくる。

計算機室にて各自に端末を使わせ、講義資料を各自の端末に表示、教師がその表示資料にメモ書きが可能で、そのメモ書きが学生端末にも反映できるというシステム（自前で構築したSHoes[1]）を利用した年度もあった。席が遠くで黒板が見えないということもなく、学生ごとに資料をさかのぼって見ることのできる。小テストを個別に実施することもできるので、非常に効率的であった。それでも、端末で別の作業をこっそりする学生とか、後からでもアクセスできるので講義に集中しない学生なども出てきた。モチベーションの落ちた学生に対しては、講義に集中させることが難しく、やはり教師による机間巡視にあたる効果を出せるようなシステム改良が必要であったが、実現はできなかった。最適なシステムというのはなかなか作れないものである。

■ 評価

成績評価は小テスト、中間試験、期末試験の得点を1：1：4の比率で案分して付けた。小テストは不正解の場合は誤り箇所を指摘し再提出させた。中間試験の結果については、成績が基準に達していない場合はレポートを課した。期末試験後に総合点を付け、合格水準に達していない場合は希望により再試験を課した。

期末試験はペーパーテストのため、90分の試験時間内で本人の力がうまく発揮できない場合もある。プログラミング能力を測るには、やはりコンピュータを使わせて、時間もたっぷり与えての試験が良いと思われるが、実際にそれを行うのは難しい。現状では、ペーパーテスト以外の平常点などを用いて評価している。ただし、ペーパーテストでもプログラミングの能力を測れるような問題の可能性もある。たとえば整列アルゴリズムを正しく理解しているかを評価するには、トレース問題がかなり効果的だと思う。本講義では、ヒープソートで7個のデータを整列するトレース問題を毎年期末試験で出題していた。試験に出ることは周知し、整列する数値列は個人ごと

に別々に与えた（学籍番号を元にあるルールによって生成させる）。適当な箇所にprintfを挿入したヒープソートプログラムを問題用紙に載せ、実行時に出力される値を解答させた。この問題では、プログラムがどう動くかを理解し、実際にその通りに追跡できる能力が測れる。過去10年の結果では、問題がまあ解けた（60点以上の）学生が平均で全体の75%であった（受験者数は平均で76人）。本講義の意義はそれなりにあったと思う。

■ 講義内容について

プログラミング通論の講義内容が今のままで妥当かどうかを考えるには、前提となる入門教育でどこまで扱うかを考える必要がある。プログラミング入門として90分15回の講義で進められる範囲を考えると、高校の情報科でのプログラミング教育を前提とできない現状では、プログラムとは何かから始め、制御構造の話、変数の話、それらを理解し、プログラムを自ら記述できるところまで行き、さらに、関数の定義、配列の導入あたりまで行くくらいが現実的である。すると、プログラミング通論では、制御構造としては再帰呼出し、データ構造としては、情報専門学科であることを考慮して、その構築の元となるポインタを扱うことになる。その応用として、簡単なデータ構造、つまり、スタック、キュー、リスト、さらに、基礎的なアルゴリズムの整列法といったところまでが対応範囲となろう。ということで、現在の講義内容は、情報専門学科での2つ目のプログラミングの講義としては、妥当であると考えている。次期学習指導要領による授業を受けた高校生が最初に大学に入る2025年度からは、大幅な変更が必要となろう。

■ 参考文献

[1]菅原典子、織田恵太、赤池英夫、角田博保：集合教育に用いる即応型e-ラーニングシステムSHoesにおける組織学習支援、情報処理学会論文誌、Vol.48、No.8、p.2791-2801（2007.8）

平成 29 年度 東京都高等学校情報教育研究会 プログラミング教育専門委員会 委員一覧

委員長	能城 茂雄
委員	角田 博保
委員	赤澤 紀子
委員	橘 孝博
委員	八百幸 大
委員	神藤 健朗
委員	塚本 初恵
委員	小原 格
委員	飯田 秀延
委員	稲垣 俊介
委員	山本 博之
委員	千葉 緑