プログラミング教室の運営と手引の作成

廣瀬研究室 4 年
C1160497 亀谷千香子

概要

プログラミング教室が年々増加している。しかし、プログラミング教室を企画・運営するための方法が詳しく書いてあるものがあまり公開されていない。そこでプログラミング教室の手引を作成することにした。東北公益文科大学 (以下本学) では、小学 5、6 年生向けプログラミング教室「Ruby てらこった」を学生が企画し運営を行った。Ruby てらこったでは、参加する小学生に対して分かりやすく、楽しくプログラミングを教えるために、教える内容、授業準備、道具の準備を行った。また、プログラミングを通して論理的思考力以外の力を身につけたいと考えた。文部科学省がプログラミング教育で身につけたい力として挙げている「知識及び技能」、「思考力、判断力、表現力等」、「学びに向かう力、人間性等」以外にも「工夫する力」、「伝える力」である。はじめてプログラミング教室を運営する人が Ruby てらこったのようにプログラミング教室を行うための方法、授業を通して小学生に先に述べた力が身につくにかなどをまとめた手引を作成した。

目 次

第1章　はじめに

小学生のプログラミング教室の運営と手引作成の研究に至った背景を説明する。

1.1　背景

2020 年度に小学生のプログラミング教育が必修化になる。プログラミング教育で身につけたい力は、「知識及び技能」、「思考力、判断力、表現力等」、「学びに向かう力、人間性等」である。また、小学校内だけのプログラミングだけではなく、外部のプログラミング教室の開催も年々増加している。そのことから、小学生のプログラミング教育の必修化、プログラミング教室の開催状況から述べていく。

1.1.1　小学生プログラミング教育の必修化

2020 年度からの小学生プログラミングの必修化に向けて文部科学省では小学校プログラミング教育の手引など公開している。プログラミング教育を導入する理由として、普段の生活の中で身近な家電、自動車にはコンピュータが内蔵されており、生活を便利にしている。コンピュータはプログラムで動いているため、仕組みを理解し情報を適切に活用、選択し問題を解決していくことが重要だからである。小学生のプログラミング教育で身につけたい力として「知識及び技能」、「思考力、判断力、表現力等」、「学びに向かう力、人間性等」が述べられている [1]。

1.1.2　プログラミング教室の開催状況

プログラミング教室は 1999 年から 2014 年以降までを比べてみると、年々増加していることが分かる図 1.1。2013 年以降からはプログラミング教室を開催している団体が急増している。

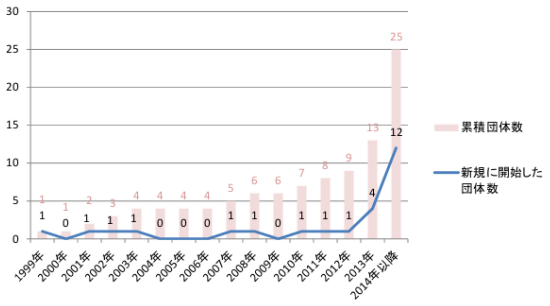


図 1.1: プログラミング教室始時期 [2] 図 5-3 より引用

1.1.3　事例

プログラミング教室の事例ついて述べる。

松本大学キッズプログラミング教室 [3]
まつもと広域ものづくりフェアで開催した。講習対象を小学校高学年以上である。Scratch を使用。

宇城市プログラミング教室 [4]
KidsVenture 、さくらインターネット株式会社、宇城市と崇城大学が連携し教室を開催した。対象は小学 4〜6 年生である。内容は、電子工作で自作したコンピュータ上にテキスト型言語によるプログラミングである。

JavaScript による 1 日プログラミング教室 [5]
まつもと広域ものづくりフェアで開催した。対象は中高生である。ゲームを簡単に作成できるように enchantjs を使用。enchantjs とは HTML5 と JavaScript を利用することにし、ゲームを簡単に作れるライブラリーである。

子ども向けプログラミング教室 [6]
対象は小学生である。低学年、中学年、高学年の 3 つのクラスに分けて実施。プログラミンを使用。プログラミンは 1.1.4 で詳しく説明する。

小学生向けプログラミング体験講座 [7]
対象はプログラミングを体験したことがない小学校 4 年生から 6 年生である。Scratch 使用。

1.1.4　プログラミング言語

他のプログラミング教室で使われているプログラミング言語で高い比率を占めるのが Scratch についで JavaScript、Java である (図 1.2)。

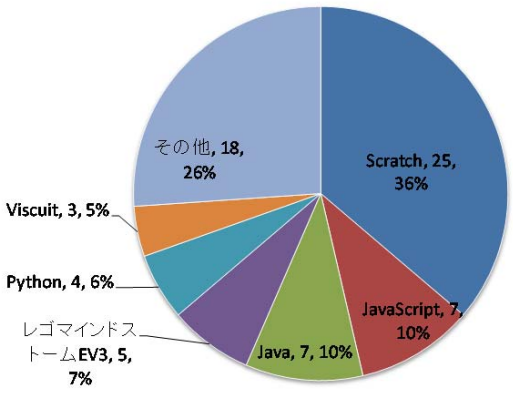


図 1.2: 教室・講座で利用されているプログラミング言語 [2] 図 5-11 より引用

主なプログラミング言語について説明する。

LOGO
Seymour A. Papert が児童の思考能力向上を目的として 1960 年代に開発されたもので、命令文によって画面上の「タートル」を動かし、タートルの軌跡で線画を描くようになっている。

ドリル
筑波大学久野、大阪電気通信大学兼宗が開発したものであり、LOGO 同様にタートルを動かし、図形などを描く機能がある。https://dolittle.eplang.jp/download

Viscuit
原田康徳氏よって開発された手書きイラストを用いたアニメーション作成機能に特化したプログラミング言語であり、タブレットで利用可能である。https://www.viscuit.com/

Scratch
MIT メディアラボが開発したプログラミング言語学習環境であり、ブロックの組み合わせによってプログラミングするオブジェクト指向言語である。https://scratch.mit.edu/

Blockly
Google が作成したビジュアルプログラミング開発ライブラリであり、ブラウザ上で動作するオープンソースのブロック型言語である。タブレットで利用可能である。https://developers.google.com/blockly/

Smalruby
高尾宏治 (ネットワーク応用通信研究所) によって開発された Ruby をもとにしたビジュアル言語である。https://smalruby.jp/blog/2017/01/14/smalruby-editor-0-4-1-has-been-released.html

プログラミン
文部科学省が Scratch を参考に開発したものでブラウザで動作するプログラミング学習用サービスである。 http://www.mext.go.jp/programin/

JavaScript
Netscape Communications 社によって開発されプログラミング教室・講座では WEB アプリケーションの作成に用いられることが多い言語である。http://www.ecmascript.org/

Java
Sun Microsystems によって開発され、プログラミング教室・講座では Android アプリ作成に用いられるケースが多い言語である。https://www.oracle.com/technetwork/java/javase/overview/index.html

Python
Guido van Rossum によって開発され、全米の大学では初心者にプログラミングを教育する教材として最もカリキュラムに取り入れられている言語である。https://www.python.org/

C

ブライアン・カーニハンとデニス・リッチーによって開発され、現在もっとも普及しているプログラミング言語である。国際標準化機構 (ISO) や日本工業規格 (JIS) にも標準として採用されている。

HTML

CERN（欧州合同素粒子原子核研究機構）の研究者であったティム・バーナーズ・リーによって開発・公開され OS X や iOS 向けのアプリケーションの開発に利用できる言語である。現在は改定版である HTML5 が多く利用されており、HTML5 では高度な Web アプリも作成可能である。`https://www.w3.org/html/`

1.1.5　プログラミング教室の企画・運営についての課題

本学では Ruby を用いたプログラミング教室を行った。詳しくは 2.3 で述べる。実際に大学生がプログラミング教室を企画・運営をする時に、どのような準備、指導する内容、指導体制など決める基準が明確になかったため準備する期間が長かった。また、プログラミング教室の企画の立て方、運営の方法、準備する道具、準備期間について詳しくまとめてあるものが Web 上で公開されているものがない。そのためプログラミング教室を行うための手引が必要となるのではないかと考えた。

1.2　目的

プログラミング教室を運営し、参加した小学生は 1.1.1 で述べた身につけたい力が身につけることができたのか、指導内容及び指導順番は適切であったかなどを分析する。また、はじめてプログラミング教室を企画・運営をする人にとって分かりやすい手引を作成する。

第2章　プログラミング教室

東北公益文科大学 (以下本学) では平成 30 年度から私立大学ブランディング事業の一環として、小学 5、6 年生向けの Ruby を用いたプログラミング教室を行っている。プログラミング教室の内容については以下の通りである。

2.1　Ruby てらこったの概要

Ruby てらこったとは小学 5、6 年生を対象にしたプログラミング教室である。庄内地方を中心とした地域の若者達に情報技術を教え、情報社会を生き抜くために必要な力を身につけていくこと目的としている。本学の私立大学ブランディング事業の一環として行った。本学のブランディング事業については次の 2.2 で述べる。

2.2　私立大学ブランディング事業

昨年度から私立大学ブランディング事業で Ruby てらこったを行った。私立大学ブランディング事業とは大学の特色ある研究を基軸として全学的な独自色を大きく打ち出す取組をここなう私立大学の機能強化を促進することである [8]。本学では、5 つの軸がある [9]。

1. 地域資源の掘り起こし研究
2. 庄内地域の文化の保存・集積（アーカイブ化）
3. 庄内地域の無形文化財を IT 技術により伝承する研究
4. 地域文化を IT 技術により他の地域の人にもわかりやすく伝え、庄内地域の魅力を発信するための手法の研究
5. 地域資源を産業振興や IT 技術により活用できる人材の育成

その中の 5. 地域資源を産業振興や IT 技術により活用できる人材の育成としてプログラミング言語の Ruby を用いたプログラミング教室の企画・運営を行ってきている。詳しくは 2.3 で説明する。

2.3　Ruby てらこったの内容

Ruby てらこったの対象は小学 5、6 年生である。その理由としては、キーボード入力、操作の関係からアルファベットを習い終わっている方が良いと考えたためである。小学生のうちからプログラミング言語 Ruby を用いた本格的なプログラミングを学ぶ機会を提供したいと考えているためである。授業は、全 5 回である。授業内容については、2.6 で説明する。

2.4　Ruby にした理由

プログラミング言語 Ruby にした理由は本学の学生の知識と他との差別化という点からである。

2.4.1　学生が持っている Ruby 知識

本学では 2 年次に 1 年間必修科目の基礎プログラミングで Ruby を学ぶ。学ぶ内容は以下の通りである [10]。

1. 計算システムの基本操作と概念

   なぜプログラミングが必要であるのかを知り、基本的なコマンド (コンピュータに特定の機能の実行を指示する命令) 操作、Unix(OS の一つ) について、ファイル、ディレクトリの概念などを学ぶ

2. プログラミングの基礎

   インタプリタ、プログラミングについて学ぶ

3. 変数と値

   変数の概念と値処理の方法を学ぶ

4. 演算子

   プログラム上での四則演算を方法を学ぶ

5. 制御構造

   条件分岐や場合分けの方法を学ぶ

6. 出力処理を行うメソッド

   画面出力の基本的な内容を学ぶ

7. 配列

   番号をつけて一度に多くのデータを処理する方法を学ぶ

8. パターンマッチング (正規表現)

   データに含まれる文字列のパターンを指定し、そこから必要な部分を取り出す方法を学ぶ

9. ファイルの入出力

   直接ファイルに入出力をする方法を学ぶ

10. 計算機の内部構造

    デジタルで数字や文字などを表現する方法について学ぶ

11. ハッシュ

    順番、個数にとらわれず任意の属性と値を結びつけ管理する方法をぶ学ぶ

12. 再帰

    大きな問題を類似した小さな問題に分割して考える方法を学ぶ

13. CGI

    自分が作成したシステムを効果的な形で視覚化させる方法を学ぶ

2.4.2 他のプログラミング教室との差別化

1.1.4 で述べたように、プログラミング教室で Scratch を使用する場合が多い。しかし、Scratch のようなグラフィック型ではなく、小学生のうちからテキスト型の本格的なプログラミングを学ぶ機会を提供したいと思い Ruby を教えることにした。

2.5 プログラミング教育で小学生に身につけたい力

小学生にプログラミング教育を通して身につけたい力として「知識及び技能」、「思考力、判断力、表現力等」、「学びに向かう力、人間性等」が述べられている [1]。その 3 つの力に追加して Ruby てらこったで身につけてほしい力についても述べる。

知識及び技能
: 生活でコンピュータが活用されていくことや、問題の解決には必要な手順があること

思考力、判断力、表現力等
: 発達の段階に即してプログラミング的思考力を育成すること

学びに向かう力、人間性等
: 発達段階に即して、コンピュータの働き、よりよい人生や社会づくりに生かそうとする態度を涵養すること

この項目に加え Ruby てらこったでプログラミングを学んだ小学生に身につけたい力は以下の通りである。

工夫する力
: 授業で使うサンプルのプログラムを自分で考えて、工夫をして、他の人とは違うプログラムをつくれるようになることである。

伝える力
: 伝える力としては、2 つの意味がある。1 つ目は、自分の作成したプログラムの発表を通して工夫し点、頑張った点など伝えられるようになれることである。

  2 つ目は、周りの人が作成したプログラムの発表を聞いて感想など伝えられるようになることである。

2.6　指導内容

指導内容は、全 5 回の限られた回数の中で小学生が理解できる比較的に簡単な内容にした。また、プログラムを作成、実行する時に複雑にならないようにした。教える順番が基本的な操作から徐々に難しい内容になるよう設定している。また、5 回目で自由に作成し、発表できるようなプログラムを作成できるように組み立てた表 (2.1)。

表 2.1: 授業内容

回数	内容	作成プログラム
1 回目	基本操作	画面出力 自己紹介プログラム
2 回目	ループ	繰り返し処理のプログラム
3 回目	配列	データ処理のプログラム
4 回目	条件分岐	条件のあるプログラム
5 回目	まとめ	習ったことを応用してプログラムを作成

出力メソッド (print, printf, puts)

アスキーアートのように入力したものをそのまま出力させるために必要と判断した。print は画面出力である。puts も同じように画面出力のためのものであるが、 末尾に改行した形で出力される。purintf は書式付の print 文であるため数字などを出力させるのに便利である。

文章処理メソッド (gets, chomp)

キーボードに打ち込んだ値を文字列として取得するゲームプログラムを作るときに必要と判断した。

制御構造 (while, if, elsif, else)

レジスタープログラムなどの入力したものを繰り返しの処理を行うときに必要と判断した。また、クイズの結果の判定や条件で繰り返しを行うために使うため必要であると判断した。盛業構造とはプログラムで実行される流れを定めたものである。while は繰り返しの処理を行うために使うものである。条件分岐には if, elsif, else を使用した。使い方の例は以下の通りである。

```
if a
   hogehoge1
elsif b
  hogehoge2
else
  hogehoge3
end
```

条件が a であれば if の hogehoge1 の部分が実行される。条件が a ではなく 2 つ目の b であれば hogehoge2 の部分が実行される。条件が a でも b でもない時は else の hogehoge3 の部分を実行する。

配列及び乱数 (srand, rand)

2 つを組み合わせてクイズ問題を用意して乱数で選ばせたり、ジャンケンの手の内をランダムに出したりするために使うので必要と判断した。

sleep 関数 (sleep)

プログラムを時間を指定して一時停止することができるので小学生が楽しむことができるのではないかと判断した。

2.6.1 指導体制

授業の担当者 1 人、タイムキーパーが 1 人、授業アシスタントは参加する小学生の人数と同じようになるような体制にした。1 対 1 で教える体制にしたのは、授業に追いつけない部分をサポートできるようにするためである。授業が進む速度に対して、小学生が理解する速度に差が発生することがあった。その際には、随時授業の合間に時間をとり、小学生が理解する速度に合わせて授業を展開していく形を心がけた。

2.6.2 授業について

小学生でも内容が分かりやすく、楽しく学ぶことができるように意識した。1 回の授業時間を 2 時間に設定した。50 分ほど授業をしたら約 15 分ほどの長めの休憩を取り、小学生が集中しやすいようにした。授業の流れは以下の通りである。

1. 前回の授業の復習

   前回の授業で学んだ内容のポイントを復習する。

2. タイピング練習

   タイピング練習では、授業で使う英単語を 3 回ずつ練習する時間をとる。

3. 内容の説明

   内容の説明では、授業で新しく学ぶことを説明する。重要なポイントは、メモをとる時間を作る。

4. サンプルプログラム

   サンプルプログラムは、新しく学ぶ内容を使う。最初にどのようなプログラムなのか実行させる時間にする。

5. サンプルプログラムの説明

   4 のサンプルプログラムがどうしてそのような動きをするのか説明する時間をつくる。

6. 休憩

   長めの休憩をとる。

7. プログラム作成

   プログラムの作成では、学んだ内容を使い自分で考えて作成してもらう。作成の時は、ホワイトボードにマグネットを貼ったり、ペンで書いたりしてからパソコンに入力する。イメージ通りにプログラムが作れない時は、アシスタントがサポート、説明をする。サポート、説明で重要なのは、どうしたらプログラムがうまく動くのか考えてもらい理解してもらうということだ。

8. 作成したプログラム発表

   作成したプログラムは、発表する時間をつくる。発表方法はみんなの前で作成したプログラムを実行し発表する方法と他の人が作成したプログラムを実行してみる方法の 2 つである。小学生が作成したプログラムについては 4.3.4 で詳しく述べる。

2.7　使用した道具

授業で使う道具は以下のようなものを用意した。小学生に分かりやすく教えるために教科書、マグネットなどを作成した。

表 2.2: 必要な道具一覧

アイテム	用途
PC	小学生の操作用、授業用、スライド用のため使用
USB メモリ	作成したプログラムを記録するため使用また、教室閉講後にも利用できよう 無料で配布可能なシステムをインストールしておく
教科書	授業の内容を確認すため使用
ノート	必要なことを書くために使用
ホワイトボード	プログラムを PC に入力する前に作成したいプログラムを 書くために使用
マグネット	プログラムを作成するときに何を組み合わせると良いのか分かりやすくするために使用
アルファベット対応表	文字を入力するときに分かりやすくするため使用

2.7.1　PC

PC は小学生用、授業用、スライド用の 3 種類使用した。小学生用と授業用の OS は NetBSD である。NetBSD とはオープンソースのオペレーティングシステムであり、本学の教育計算機と同じ OS だと学生が教えやすいと考えたからである。また、授業以外でも使ってほしいため無料で配布できるオープンソースが良いと考えたからである。

- 小学生用

授業中のプログラムを作成、Web ページからプログラムをダウンロードするために使用行った。2 つプロジェクターがなかった場合はテレビに映して授業をした。

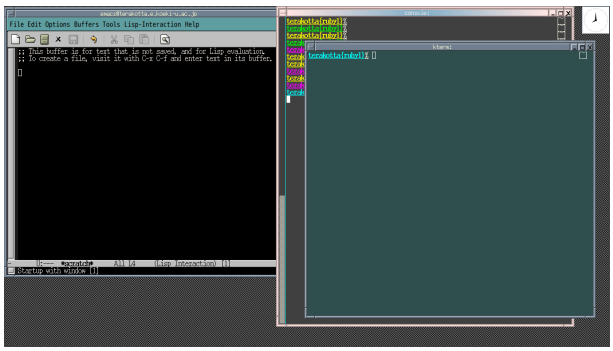


図 2.1: PC 画面

- 講師用

  小学生の使用している計算機と同じであるためどのように操作するのか小学生に分かりやすくするため使用

- スライド用

  スライド用を用意することで画面の切り替えすることができるためスムーズに内容の説明ができる

使用した教室に 2 つ独立したプロジェクタがあったので授業用、スライド用両方を映して授業を行った。2 つプロジェクターがなかった場合はテレビに映して授業をした。

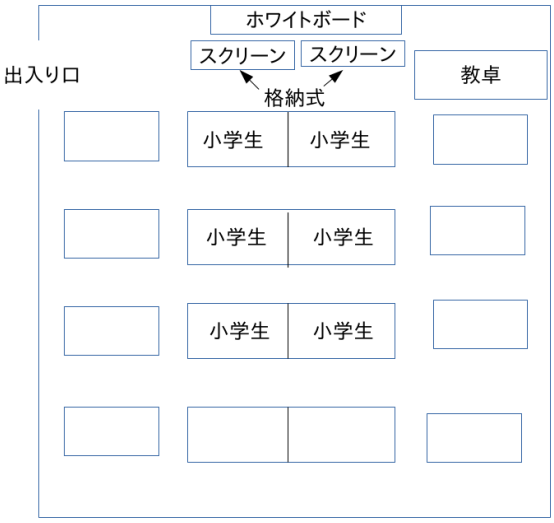


図 2.2: 教室見取り図

2.7.2　USB メモリ

今回は 16GB 以上の容量を持つ高速な USB フラッシュメモリに OS と教材に使用するソフトウェア一式をインストールし、USB メモリから起動して授業用環境として使用できるようにした。行った。2 つプロジェクターがなかった場合はテレビに映して授業をした。

2.7.3　教科書

授業内容に合わせて、学生らで作成した。オリジナルのイラストや分かりやすい図などを取り入れ、興味が湧くような内容にすることを心がけた。また、語句の説明なども小学生でも分かるようにした。内容は以下のようなものである。

プログラミングとは
　プログラミングとは何か、Ruby とは何かを説明する

ファイル、ディレクトリ
　ファイルとディレクトリの説明をする

print, puts, gets について
　画面出力の方法、キーボードから入力させる方法をまとめた

while について
　ループについて学ぶ

配列、乱数
　配列を使ってランダムに取り出す方法をまとめた

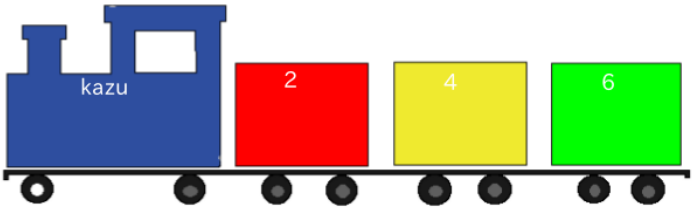


図 2.3: 配列のイメージ図

if, elsif, else について
　条件分岐をさせる方法、if, elsif, else の違いについて

プログラムを工夫するためのメソッドについて
　プログラムを出力させるときに文字が変化したり、時間をおいて表示させたりするものをまとめた

コマンド集
: プログラムの作成時などに覚えておくと便利なコマンドをまとめて書いておき、分からない時にすぐに確認できるようにした

2.7.4 ノート

初回の授業で配布し、小学生が学んだことや気づいたこと、ポイントなどをこまめに書き込めるように用意した。授業内では、ノートに書く場面があれば先生が指示を促し小学生がノートを書く時間を設けた。

2.7.5 ホワイトボード

ホワイトボードにプログラムを書いてから PC にプログラムを打ち込んだ方が間違うことが少なくなり、分かりやすいと考えたため用意した。また、小学生がホワイトボードに書きながらプログラムを組んで行くことで理解を深める目的で使った。

2.7.6 マグネットシート (小学生用)

その授業の内容で勉強する新しいことや大切なものをマグネットシートにした。それらを貼るだけでよいようにし、記憶しやすくするために作った。これらを授業内でホワイトボードに貼りながら作業を進めていった。例を挙げると、「 while 」と「 end 」などのセットで扱うものに関しては、セットで先にマグネットをホワイトボードに貼り付けてからプログラムを組み立てるなどの工夫をした。あらかじめ用意されているものを使用したため、間違いの減少や作業時間の削減にも繋がった。

2.7.7 マグネット (黒板用)

小学生にメソッドの部分を分かりやすくするために、黒板に貼るようのマグネットを用意した。

2.7.8 アルファベット対応表

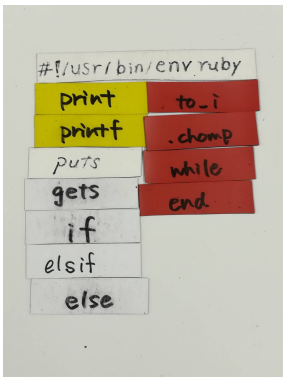
小学 5、6 年生はアルファベットを習い終えているが大文字小文字を理解するために作成し配布した。キーボードでは大文字表記になっているため「 l(エル) 」と「 i(アイ) 」や「 h(エイチ) 」と「 n(エヌ) 」「 r(アール) 」などが間違いやすかった。

(a) USB メモリ

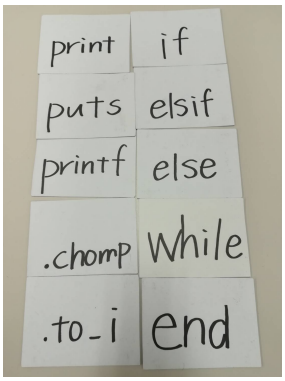
(b) 教科書

(c) ノート

(d) ホワイトボード



(e) マグネットシート



(f) マグネットシート (黒板用)

ー アルファベット表 ー

A a	B b	C c .chomp	D d
E e elsif , else , end	F f	G g gets	H h
I i if	J j	K k	L l ls
N n \n	M m	O o	P p print , puts
Q q	R r ruby , rand	S s sleep	T t .to_i
U u	V v	W w while	X x
Y y	Z z		

(g) アルファベット対応表

第3章　プログラミングで身につく力の仮説

この章では参加した小学生がプログラミングプログラミングの授業を通してどのような変化があるか仮説を立る。仮説を判断するため基準を述べる。

3.1　プログラミング教室実施の仮説

参加する小学生はある程度プログラミングに興味があり、難しいという印象よりも楽しそうという印象があると考える。授業を通してプログラムが難しいというイメージが減少するのではないかと考えた。また、授業を通して波及効果として学習が好きな方に変化し、別のことにも挑戦してみようという意欲的になると仮説を立てる。

3.2　身につく力の判断基準

実際に 3.1 の仮説を立てたものを検証をする方法としてアンケートを作成した。アンケートは 3 種類作成をした。アンケートの種類と結果については以下の通りである。また、知識などについては、小学生が作成したプログラミングを参考に判断する。また、アンケートで判断できないものは小学生が作成したプログラムより判断する。

3.3　身につけたい力の判断基準

2.5 で述べた力を Ruby てらこったでの具体的に判断する基準を述べる。

- 知識及び技能

  プログラムを作成できる

  様々なメソットを覚え、使うことができる

- 思考力、判断力、表現力等

  自分が作成したいプログラムに対してどのメソットを使うとできるか考えることができる

- 学びに向かう力、人間性等

  サンプルプログラムをどのよう工夫できるか自分で教科書を見てたり、大学生に聞きたりして取り組むことができる

- 工夫する力

  サンプルプログラムを自分が作りたいものに作り変えることができる

- 伝える力
  作成したものに対して発表ができる
  他の人の発表を聞いた後に感想を伝えることができる

3.3.1　アンケートの種類

アンケートは 1 回目の授業の開始前、毎回の授業後、最後の授業後に行った。

授業開始前アンケート

- 普段の機械操作について
- プログラミング教室参加理由
- プログラミングの経験
- プログラミングのイメージ
- 作成したいプログラム
- 学習について

毎回の授業後アンケート

- 授業で楽しかったこと (何でも可)
- 授業中難しかったこと (何でも可)

最後の授業後アンケート

- 授業について
- プログラミングについて
- 論理的思考力について
- 学習について

3.3.2　作成したプログラムより判断

知識などは、小学生が授業をした後にサンプルプログラムを改造、5 回目の授業では学んだ内容をまとめたオリジナルプログラムを作成する。そのプログラムより、知識が身についたか判断する。

第4章　プログラミング教室の実施結果

2019年度のRubyてらこったは、6月開始の教室には6名、7月開始の教室には6名、合計で12名参加した。12名のアンケートの結果と作成したのプログラムは以下の通りである。アンケート結果については () の中の数字が回答人数となっている。

4.1　授業前

普段の機械操作とプログラミングのイメージ、経験、学校の学習についてのアンケートを実施した。

4.1.1　機械操作について

- パソコンは使いますか

  週3回以上 (0) 週2回以上 (2) 週1回以下 (6) 使わない (4)

- スマホは使いますか

  週3回以上 (9) 週2回以上 (0) 週1回以下 (1) 使わない (2)

- テレビゲームはしますか (iPad, Switchも含めて)

  週3回以上 (7) 週2回以上 (2) 週1回以下 (1) しない (2)

- パソコンゲームはしますか

  週3回以上 (1) 週2回以上 (1) 週1回以下 (1) しない (9)

- 携帯ゲームはしますか

  週3回以上 (0) 週2回以上 (5) 週1回以下 (2) しない (5)

4.1.2　プログラミングについて

- なぜプログラミング教室に参加しましたか

  親のすすめ (4) 友達が一緒だから (0) プログラミングに興味があるから (7) その他 (1)

- プログラミングをしたことがありますか

  したことがある (5) したことがない (7)

- プログラミングは楽しそう

  とても楽しそう (6) 楽しそう (6) 楽しそうではない (0) 全く楽しそうではない (0)

- プログラミングは難しそう

  とても難しそう (2) 難しそう (7) 難しそうではない (3) 全く難しそうではない (0)

- 作ってみたいプログラムはありますか (自由記述)

  ゲームプログラム、計算させるプログラム、学習のプログラム、字を書くプログラム、占いプログラム

4.1.3　学習について

- 学校の勉強が好き

  とても好き (4) 好き (6) 好きではない (2) 全く好きではない (0)

- 学校の勉強が難しい

  とても難しい (0) 難しい (2) 難しくはない (6) 全く難しくない (4)

4.2　毎回の授業

授業の振り返りとして授業中に楽しかったこと、難しかったことを書いてもらった。

4.2.1　楽しかったこと

1 回目
コマンドを入力すること、文字を打つこと、自己紹介プログラムを発表すること、プログラムを作ること、大学生と話したこと
2 回目
キーボード入力、プログラムを改造すること、他の人の作成したプログラムを実行すること、大学生と話したこと
3 回目
プログラムを改造したこと、他の人が作成したプログラムを実行すること、乱数と配列を使ってプログラムがつくれたこと、大学生と話したこと
4 回目
くじ引きのプログラムを作成したこと、クイズプログラムを作成したこと、大学生と話したこと
5 回目
オリジナルのプログラムを作成したこと、発表したこと

4.2.2 難しかったこと

1 回目
自己紹介プログラムを発表すること、キーボードの位置が分からなかったこと、文字を打つこと、print と puts の違い、キーボード入力、
2 回目
break とは何か理解すること、ループの考え方、プログラムを改造すること、文字を打つこと
3 回目
乱数、プログラムを改造すること
4 回目
プログラムを改造すること、if, elsif, else の違い、文字を多く打つこと
5 回目
プログラムを考えること

4.3 最後の授業後

5 回目の授業後に 1 回目から 5 回目までの授業について、プログラミングを実際にしてみてどうだったか、学校の学習について、今後挑戦したいことについてアンケートをとった。

4.3.1 授業について

- 授業時間はどうでしたか

  長い (1) ちょうどよい (11) 短い (0)

- 内容は分かりやすかった

  とても分かりやすい (7) 分かりやすい (5) 分かりにくい (0) とても分かりにくい (0)

- 大学生はやさしかったですか

  とても優しい (12) 優しい (0) 優しくない (0) 全く優しくない (0)

- 授業に参加してみて楽しかったことを書いてください (自由記述)

  全てが楽しかった、プログラムを工夫すること、クイズプログラムを作成したこと、大学生と話したこと、長いプログラムが作成できたこと

- 授業に参加してみて難しかったことを書いてください (自由記述)

  覚えることが多いこと、変数、たくさん入力すること

4.3.2　プログラミングについて

- プログラミングは楽しかった

  とても楽しかった (8) 楽しかった (4) 楽しくなかった (0) 全く楽しくなかった (0)

- プログラミングは難しかった

  とても難しかった (1) 難しかった (7) 難しくなかった (2) 全く難しくなかった (2)

- プログラミングをまたしたい

  とてもしたいと思う (5) したいと思う (7) したくないと思う (0) 全然したくないと思う (0)

- 筋道をたてて考える力が身についたと思う

  とても思う (3) 思う (9) 思わない (0) 全く思わない (0)

4.3.3　学習について

- 学校の勉強が好き

  とても好き (5) 好き (4) 好きではない (3) 全く好きではない (0)

- 学校の勉強が難しい

  とても難しい (0) 難しい (3) 難しくはない (4) 全く難しくない (5)

4.3.4　その他

- 挑戦してみたいことを書いてください (何でも可)

  もっと難しいプログラムを作る、家でもプログラミングをする、自分でゲームを作りたい、みんなにプログラミングを教える、50m クロールで泳ぎたい

参加した小学生が 5 回目のオリジナルプログラムで作成たものは以下のようなものがある。

このプログラムはクイズプログラムで学校の授業で習ったことをクイズにし、回答時間を出力したものである。ポイントは、条件分岐である。それぞれの問題に 4 つの選択肢があり、回答をキーボード入力から受け取り、選択したものによって出力するものを変えている。ただ、出力させるのではなく、sleep を使い工夫をしている。

```
# coding: utf-8
1#!/uer/bin/env ruby
# coding: utf-8

puts("(9*7)/(2+1)の答えはいくつでしょう")

puts("1:32")
puts("2:24")
puts("3:21")
puts("4:31")
print("答えを入力:")
 answer = gets.to_i
```

```
sleep(1)
puts("結果は……")
sleep(1)


if answer == 1 then
  puts("ザンネン！ハズレ!もう1度やってみよう!")
elsif answer == 2 then
  puts("ハズレだよ!おしい!")
elsif answer == 3 then
  puts("正解だよ!すごいね!!")
elsif answer == 4 then
  puts("残念!はずれだよ!")
else
  puts("その番号はないよ！！")
end

puts("")
puts("4と9の最小公倍数はいくつでしょう")

puts("1:24")
puts("2:27")
puts("3:32")
puts("4:36")
print("答えを入力:")
 answer = gets.to_i

sleep(1)
puts("結果は……")
sleep(1)


if answer == 1 then
  puts("ザンネン！ハズレ!もう1度やってみよう!")
elsif answer == 2 then
  puts("ハズレだよ!おしい!")
elsif answer == 3 then
  puts("残念!次がんばろう!")
elsif answer == 4 then
  puts("正解だよ!すばらしい!!!")
else
  puts("その番号はないよ！！")
end

puts("")
puts("日本列島の春の季節のだいたいの雲の動きは")

puts("1:南から北")
puts("2:西から東")
puts("3:東から西")
puts("4:北から南")
print("答えを入力:")
 answer = gets.to_i

sleep(1)
puts("結果は……")
sleep(1)
```

```
if answer == 1 then
  puts("ザンネン！ハズレ!もう1度やってみよう!")
elsif answer == 2 then
  puts("正解!絶好調!!!!")
elsif answer == 3 then
  puts("残念!次がんばろう!")
elsif answer == 4 then
  puts("ザンネン!ちがうよ!")
else
  puts("その番号はないよ！！")
end

puts("")
puts("日本列島は何個の島が集まって出来ているでしょうか")

puts("1:4こ")
puts("2:6こ")
puts("3:7こ")
puts("4:5こ")
print("答えを入力:")
 answer = gets.to_i

sleep(1)
puts("結果は……")
sleep(1)


if answer == 1 then
  puts("正解だよ!完璧!!!")
elsif answer == 2 then
  puts("ザンネン!ちがうよ!")
elsif answer == 3 then
  puts("もう1度確認しよう!!")
elsif answer == 4 then
  puts("おしい!!!!もう1度やってみよう!")
else
  puts("その番号はないよ！！")
end
```

このプログラムはlatterという配列の中に入っている”グラードン”,”カイオーガ”,”レシラム”,”キュレム”の4つをどのくらいの時間で入力できるかというプログラムである。ポイントは、繰り返し処理で条件があったときに、配列の中にある問題を順番に入力させるものが変わるようになっている。最後にかかった時間を出力できることである。

```
#!/usr/koeki/bin/ruby
# -*- coding:utf-8 -*-

letter = ["グラードン","カイオーガ","レシラム","キュレム"]

print("どれだけ早くポケモンを打てるかな\n")
printf("%s\n", letter)

start = Time.now.to_i

i = 0
```

```
while i < letter.length
  printf("\v%s:", letter[i])
  input = gets.chomp
  if input == letter[i]
    stop = Time.now.to_i
    STDERR.print("\t正解\n")
    i += 1
  end
end

printf("かかった時間 %d 秒!\n", stop - start)
```

第5章　プログラミング教育の考察

4 章のアンケート結果から考察を述べる。

5.1　機械操作について

機械操作については普段はパソコンよりもスマートフォンの方が使用していると結果が出た。普段パソコンのキーボード操作をしていない小学生にとっては入力・コマンド操作は難しいため授業の感想の難しかった部分の回答となったのではないかと考える。

5.2　プログラミングについて

プログラミングのイメージは楽しそうというイメージがある。しかし、難しそうというイメージもあった。授業をしてまたプログラミングをしてみたいという回答が多かった。実際にプログラミングを学んでみて難しかったという回答があった。ピアジェの認知発達理論では、図 5.1 のようになっている。そのために難しく感じた小学生がいたのだと考えられる。また、キーボード操作も難しく感じた要因ではないかと考える。

表 5.1: ピアジェが仮説化する各発達段階での子どもの思考特徴 [12] より引用

発達段階	年齢の範囲	達成可能な典型と限界
具体的操作段階	7〜12 歳	具体物を扱う限りにおいては論理的操作が可能になる。ものや事象の静的な状態的な状態だけでなく変換の状態をも表象可能、外見的な見えに左右されず保存問題や系列化やクラス化の問題解決が可能、だが科学的な問題や論理的変換のようにあらゆる可能な組み合わせを考えねばならぬ問題には困難を示す。
形式的操作段階	12 歳〜	経験事実に基づくだけではなく、仮説による論理的操作や、命題間の論理的関係の理解が可能である。より抽象的で複雑な世界についての理解が進み、たとえば、エネルギーの保存や化学的い合成に関するような抽象的概念や知識が獲得される。

5.3　身につけたい力について

知識及び技能

授業で学んだことを理解し、工夫してプログラムを作成していた小学生が多かった。

思考力、判断力、思考力等

自分が作成したいプログラムを考え何を使えばできるかを考え、大学生がサポートしながら完成されることができた。

学びに向かう力、人間性等

プログラムを作成したり、授業中に分からない所があった時に周りの大学生に積極的に質問をして解決をしていた小学生が多かった。

工夫する力

毎回の授業でサンプルのプログラムを自分で考えて変えていた。5 回目の授業のオリジナルのプログラムを作成する時に前に作成したものを改良して作成ができていたので身についたと考える。

伝える力

自分が作成したプログラムの工夫した点や頑張ったところなど発表することができていた。また、他の人が作成したプログラムに対しての感想やどうやったらこの動きになるのかなど質問をお互いにしていたので力が身についたと考える。

5.4　論理的思考について

論理的思考については 4.3.2 での結果では身についたと思う実感している小学生が多いという結果になった。5 回目の授業で今まで学んだものを使い作成するときにもどのようにしたら動くのかを考えて作成している小学生が多かった。しかし、5.2 で述べたようにプログラミングが難しいという結果も出た。

5.5　波及効果について

波及効果では授業の最初と 5 回目の授業のあとに同じ質問項目の学習 (勉強が好き、勉強が難しい) の部分についての部分に変化があるのを調べた。調べる方法として T 検定を用いた。結果は以下の通りである。

- 勉強が好き

```
== Mean & S.D. ( SD=sqrt(Vtotal/N) ) ==

------------------------------------------------
          N       Mean       S.D.
------------------------------------------------
水準1     12       3.42       0.76
水準2     12          3       0.58
------------------------------------------------
 差       12       0.42       1.26
------------------------------------------------
 t(11)= 1.1055 ,    ns (.10<p)
------------------------------------------------
```

- 勉強が難しい

```
 == Mean & S.D. ( SD=sqrt(Vtotal/N) ) ==

------------------------------------------------
           N       Mean       S.D.
------------------------------------------------
水準1     12        1.5        0.5
水準2     12       2.17        0.8
------------------------------------------------
 差        12      -0.67       1.03
------------------------------------------------
 t(11)= 2.1574 ,     +  (.05<p<.10)
------------------------------------------------
```

仮説では変化があるのではないかと考えたが実際には変化がなかった。変化がなかった原因としては以下 3 つの原因があると考える。

1. 回数が少ない

   プログラミング教室開催が全 5 回と回数が少ないため変化が現れなかったと考えられる。

2. 対象人数が少ない

   今回の対象人数は 12 人であった。勉強が好きの項目で数値が変化しなかったのは 7 人、数値があがった人は 2 人、数値が下がった人は 3 人であった。勉強が嫌いの項目では、変化しなかったのは 10 人、数値が上がったのは 1 人、数値が下がったのは 1 人であった。変化しない参加者が多く数値に変化が現れなかった。

3. 質問方法に問題があった

   学習の部分だけでは、変化が現れなかった。しかし、自由記述の部分では、

   - もっと難しいプログラムを作る
   - 自分でゲームを作りたい、みんなにプログラミングを教える
   - 50m クロールで泳ぎたい

などの回答があった。そのような回答から波及効果があったのではないかと考える。そのため、学習の項目だけに変化がなかっただけで質問方法に問題があったのではないかと考える。

第6章　結論

6.1　結論

参加した小学生のプログラミングに対するイメージを知り、実際に企画、運営した Ruby を用いたプログラム教えることができた。プログラミング教室を通して身につけたいと考えていた力を身につけることができたのではないかと考える。しかし、プログラミングが難しいと感じていた参加者もいた。そのため、どのようにして難しい内容でも分かりやすく教えるのかというのが課題である。

Ruby てらこったの企画・運営の点から初めてプログラミング教室を運営する人を対象にした手引を作成した。

6.2　今後の展望

今後の展望については、以下の通りである。

はじめてプログラミング教室を運営をする人に手引を利用してもらう
: 実際に手引を利用してもらいプログラミング教室を運営してもらう。そこで情報が不足していたり、改善をしたほうがよい場合は手引を修正する。

アンケートの改善
: 今回はアンケートと作成したプログラムにより身についた力の判断をした。プログラムができてもそれぞれのメソッドについてしっかり理解しているのか分からない部分がある。そこでアンケートに加え、学習した用語を実際にテストのように書いてもらう方法で判断し検証をする。

謝辞

本研究は、平成 30 年度私立大学ブランディング事業「地域資源を産業振興や IT 技術により活用できる人材の育成」の助成を受けた成果である。酒田市教育委員会、鶴岡教育委員会、参加してくれた方など Ruby てらこったの活動に協力をしていただいた方に感謝を申し上げる。

参考文献

[1] 文部科学省. ” 小学校プログラミング教育の手引 (第二版)”. http://www.mext.go.jp/component/a_menu/education/micro_detail/__icsFiles/afieldfile/2018/11/06/1403162_02_1.pdf, (参照日 2019-5-24).

[2] 総務学省. ” プログラミング人材育成の在り方に関する調査研究報告書”. http://www.soumu.go.jp/main_content/000361430.pdf, (参照日 2019-8-1).

[3] 室谷心 .” 小学生にプログラミングを教える”. 松本大学研究紀要（2013)p269-281.

[4] 田口雄太 杉浦忠男 中島厚秀 .” プログラミング教室を通した相互成長―宇城市プログラミング教室―”. 崇城大学紀要（2019)p115-122.

[5] 室谷心 矢野口聡 浅見（林）大輔 .” 楽しさで引っ張るプログラミング入門講座：JavaScript を使った 1 日プログラミング教室用教材の開発と試用”. 松本大学研究紀要（2018)p75-82.

[6] 廣田千明 寺田裕樹 橋浦康一郎 渡邉貫治.” 地方大学における学生主体の子ども向けプログラミング教室　一秋田県における IT 教育の推進一”. 秋田県立大学ウェブジャーナル A（地域貢献部門）(2017)p71-80.

[7] 河原和好 .” 小学生を対象にしたプログラミング教育について”. 新潟国際情報大学情報文化学部紀要（2017)p27-35.

[8] 文部科学省.” 私立大学研究ブランディング事業”. http://www.mext.go.jp/a_menu/koutou/shinkou/07021403/002/002/1379674.htm, (参照日 2019-9-18).

[9] 東北公益文科大学. https://www.koeki-u.ac.jp/news_topics/branding-saitaku_2017.html, (参照日 2019-9-18).

[10] 広瀬雄二.「Ruby プログラミング基礎講座」, 技術評論社,2006,p312.

[11] 大石桃菜 佐々木大器 山口円馨 .” 東北公益文科大学における小学生向けプログラミング教室「Ruby てらこった」の取り組み”. 文部科学省 私立大学研究ブランディング事業 日本遺産を誇る山形県庄内地方を基盤とした地域文化とＩＴ技術の融合による伝承環境研究の展開（平成 29 年度～平成 33 年度)p50-54.

[12] サトウタツヤ 渡邊芳之.「心理学・入門一心理学はこんなに面白い」, 株式会社有斐閣,2011,p268.

付録