

# プログラミング教室の運営と手引の作成

亀谷千香子

令和元年〇月〇日



概要



# 目次

<b>第1章</b>	<b>はじめに</b>	<b>7</b>
1.1	背景	7
1.1.1	小学生プログラミング教育の必修化	7
1.1.2	プログラミング教室の開催状況	7
1.1.3	事例	8
1.1.4	プログラミング言語	8
1.1.5	プログラミング教室の企画・運営についての課題	10
1.2	目的	10
<b>第2章</b>	<b>プログラミング教室</b>	<b>11</b>
2.1	Ruby てらこったについて	11
2.2	Ruby にした理由	11
2.2.1	学生の知識	11
2.2.2	他との差別化	12
2.3	身につけたい力	12
2.4	指導内容	13
2.4.1	指導体制	14
2.4.2	授業について	14
2.5	使用した道具	14
2.5.1	PC	15
2.5.2	USB メモリ	15
2.5.3	教科書	15
2.5.4	ノート	16
2.5.5	ホワイトボード	17
2.5.6	マグネット	17
2.5.7	アルファベット対応表	17
<b>第3章</b>	<b>仮説と判断基準</b>	<b>19</b>
3.1	仮説	19
3.2	判断基準	19
3.3	身につけたい力の判断基準	19
3.3.1	アンケートの種類	20
3.3.2	作成したプログラムより判断	20

<b>第 4 章 実施結果</b>	<b>21</b>
4.1 授業前 . . . . .	21
4.1.1 機会操作について . . . . .	21
4.1.2 プログラミングについて . . . . .	21
4.1.3 学習について . . . . .	22
4.2 毎回の授業 . . . . .	22
4.2.1 楽しかったこと . . . . .	22
4.2.2 難しかったこと . . . . .	22
4.3 最後の授業後 . . . . .	23
4.3.1 授業について . . . . .	23
4.3.2 プログラミングについて . . . . .	23
4.3.3 学習について . . . . .	24
4.3.4 その他 . . . . .	24
4.4 作成したプログラム . . . . .	24
<b>第 5 章 考察</b>	<b>25</b>
5.1 機械操作について . . . . .	25
5.2 プログラミングについて . . . . .	25
5.3 身につけたい力について . . . . .	25
5.4 論理的思考について . . . . .	26
5.5 波及効果について . . . . .	26
<b>第 6 章 結論</b>	<b>27</b>
<b>第 7 章 付録</b>	<b>29</b>

# 第1章 はじめに

はじめに、この研究を始めるに至った経緯を説明する。

## 1.1 背景

小学生プログラミング教育の必修化とプログラミング教室の開催状況、プログラミング教室を運営するための課題について述べていく。

### 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。

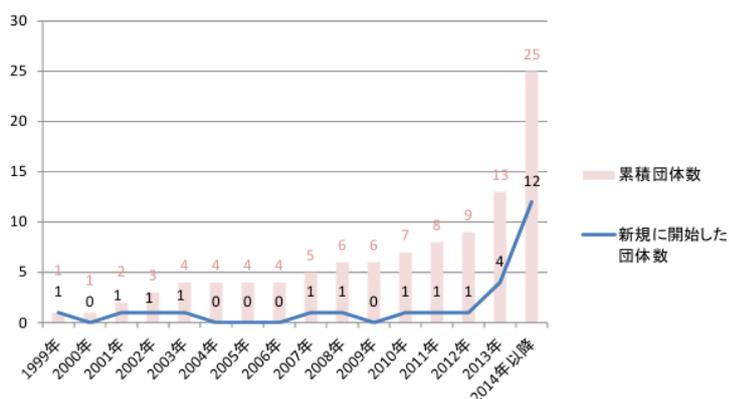


図 1.1: プログラミング教室始時期 [2] 図 5-3 より引用

### 1.1.3 事例

プログラミング教室の事例について述べる。

#### 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。

#### 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。

#### 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無い小学校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は図 1.2 のようになっている。多くのプログラミング教室で利用されているものは、Scratch である。また、その他にも JavaScript、Java などである。

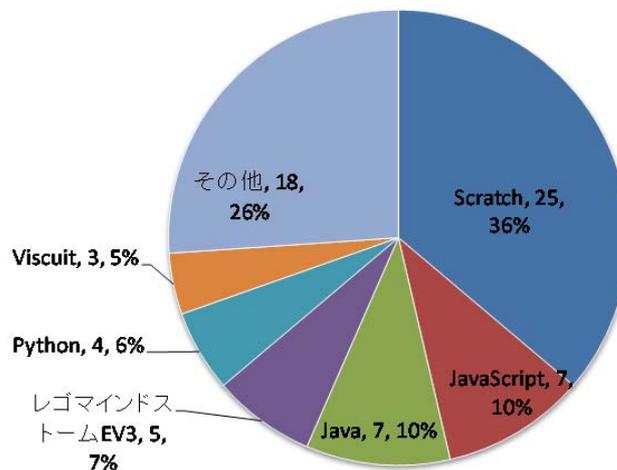


図 1.2: 教室・講座で利用されているプログラミング言語 [2] 図 5-11 より引用

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。

## Viscuit

原田康徳氏に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。

## 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。

## 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。

## 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。

## 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。

## 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。

## 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。

## Python

Guido van Rossum によって開発され、全米の大学では初心者には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。

## 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。

### 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 2.1 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

本学では平成30年度から小学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 てらこったについて

昨年度から大学のブランディング事業の一つの地域資源を活用する人材育成の研究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・運営を行ってきた [9]。目的は、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。小学生でも対象は小学5、6年生にした。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。

### 2.2 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。

#### 2.2.1 学生の知識

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という必修科目がある。その科目でプログラミング言語のRubyを学ぶ。学ぶ内容は以下の内容である [8]。

1. 計算システムの基本操作と概念

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

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の基礎

3. 変数と値

変数と値処理の方法

4. 演算子

プログラミングの加減乗除の手法

5. 制御構造

プログラムで命令が実行される流れを定める手法

## 6. 出力処理を行うメソッド

決めた複雑な処理を行う手法

## 7. 配列

多くのデータをまとめて保持する手法

## 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する手法

## 9. ファイルの入出力

数にファイルを指定した場合はそのファイルから1行ずつ内容を読み込む手法

## 10. 計算機の内部構造

2進数、16進数、文字コードについて

## 11. ハッシュ

任意の値に任意の値を結び付ける手法

## 12. 再帰

ハノイの塔のような再帰処理の仕組みと手法

## 13. CGI

HTML 文書内にあるデータ入力窓とそれらを受け渡すスクリプト名を書いておき、ボタンを押すとそのスクリプトに入力値が渡るような手法

### 2.2.2 他との差別化

1.1.4でのべたように、多くの場合 Scratch を使用する場合が多い。また、小学生のうちから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。

## 2.3 身につけたい力

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]。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

#### 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ること

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

## 2.4 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回目ですべて自由で自由に作成し、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。

#### 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。

#### 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。

#### 配列及び乱数 (`srand`, `rand`)

2つを組み合わせると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。

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

### 2.4.1 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。

### 2.4.2 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るようにして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、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。次の??で詳しく述べる。

## 2.5 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。

表 2.2: 必要な道具一覧

アイテム	用途
PC	小学生の操作、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るため使用
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

### 2.5.1 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、本学のパソコンと同じ 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
- 授業用  
小学生の使用しているパソコン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。

### 2.5.2 USB メモリ

今回は 16GB 以上の容量を持つ高速なメモリに 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。

### 2.5.3 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**

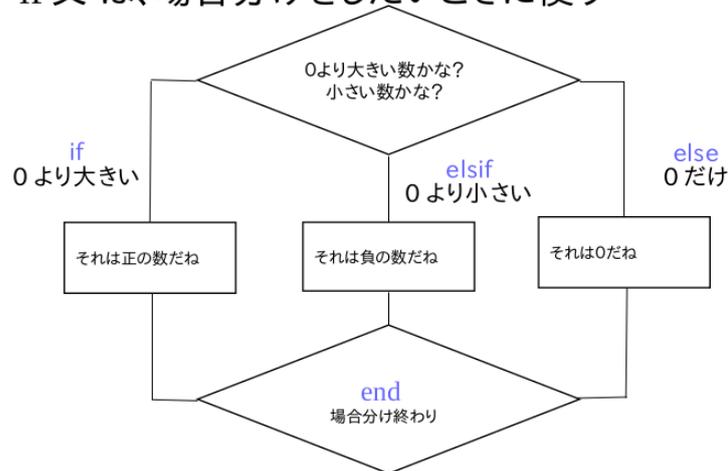
ループについて学ぶ

**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

**if, elsif, else について**

条件分岐をさせる方法、if, elsif, else の違いについて図を用いて分かりやすくした

**if 文 は、場合分けをしたいときに使う****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

**2.5.4 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。

### 2.5.5 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わかりやすいと考え、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。

### 2.5.6 マグネット

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。

### 2.5.7 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)」などが間違いやすかった。



## 第3章 仮説と判断基準

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てる。仮説を立てたのを判断するため基準を述べる。

### 3.1 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た。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 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

2.3 で述べた力を Ruby までで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  
様々なメソッドを覚え、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ド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、大学生に聞い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、知識が身につけたか判断する。

## 第4章 実施結果

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 機会操作について

- 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) 友達と一緒にだから(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、プログラムをすること、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できたこと、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試してみようだったか、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

## 4.4 作成したプログラム

- 2、3個プログラミングを載せる予定

## 第5章 考察

4章の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ホ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。しかし、難しかったという回答に対しては内容が小学生にとって少し難しかったのではないかと考える。4.2の結果にあるようにキーボード入力だけではなく内容について難しいという感想があったのでそのように考えた。

### 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すればできるかを考え、大学生がサポートしながら完成させることができた。

#### 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ミング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たい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行っ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。

## 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についての部分に変化があるのを調べた。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。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。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



## 第7章 付録



## 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](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](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じた相互成長一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 渡邊貫治.” 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] 広瀬雄二. 「Ruby プログラミング基礎講座」, 技術評論社,2006,p312
- [9] 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.