

Ruby等を用いたプログラミング教室の手引

廣瀬研究室

C1160497 亀谷千香子

目次

第1章	はじめに	7
1.1	プログラミング教育必修化の背景	7
1.2	プログラミング教室	7
1.2.1	内容	7
1.2.2	プログラミング言語	8
1.2.3	プログラミング言語の割合	10
1.2.4	使用されている機器	10
1.3	プログラミング教育の課題について	10
1.3.1	手引作成の背景	11
1.3.2	対象読者	11
1.3.3	手引の構成	12
第2章	プログラミング教室実施のための必要項目	13
2.1	教える側が必要なこと	13
2.1.1	プログラミングの知識	13
2.1.2	教える能力	14
2.1.3	発想力	14
2.1.4	時間管理能力	15
2.1.5	伝える力	15
2.2	プログラミング教室の目的と小学生に身につけてほしい力	15
2.2.1	Ruby てらこったと活動の目的	15
2.2.2	授業を通して身につけてほしい力	15
2.2.3	教える範囲	16
2.3	授業・日程	17
2.3.1	教える順番と教える時間	17
2.3.2	各小学校のとの調整	17
2.3.3	開催日程	18
第3章	運営	19
3.1	運営方法	19
3.1.1	役割関係の明確化	19
3.2	必要な道具	21
3.2.1	PC	22
3.2.2	USB メモリ	22
3.2.3	教科書	22

3.2.4	ノート	23
3.2.5	ホワイトボード	23
3.2.6	マグネットシート	23
3.2.7	アルファベット対応表	23
3.3	授業準備	24
3.3.1	内容の吟味	24
3.3.2	教える体制	24
3.4	各授業のサンプルプログラム	24
3.4.1	1回目のプログラム	24
3.4.2	2回目のプログラム	25
3.4.3	3回目のプログラム	25
3.4.4	4回目のプログラム	25
3.4.5	5回目のプログラム	26
3.4.6	スライド	27
3.5	模擬授業	27
3.6	他の機関との調整	28
第4章	授業をするための方法	31
4.1	当日の準備・確認事項	31
4.2	授業の流れ	31
4.3	プログラム発表について	32
4.4	授業後のミーティング	32
第5章	授業を受講した小学生の感想	33
5.1	授業で楽しいと感じる部分	33
5.2	授業で難しいと感じる部分	34
第6章	教えるための要点	35
6.1	教える上で注意すること	35
6.1.1	授業担当者	35
6.1.2	授業アシスタント	35
6.1.3	小学生の理解度に合わせる	35
6.2	その他	36
6.2.1	授業の内容が予定より進まなかった場合	36
6.2.2	参加者が来れなくなった場合	36
6.2.3	利用しているPCの調子が悪くなった時	36
第7章	実際に行うための一例	37
7.1	企画	37
7.2	運営	38
7.3	広報関係提案	38
7.4	ミーティング	39

7.5	申し込み方法の提案	40
7.6	授業	41
7.7	授業後のミーティング	41
7.8	6月にプログラミング教室を開催の実例	42
第8章	課題の考察	45
第9章	付録	47
9.1	コマンド集	47
9.2	アルファベット対応表	48

第1章 はじめに

2020年度に小学生のプログラミング教育が必修化になる。文部科学省の小学校プログラミング教育の手引は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。

小学生のプログラミング教育の必修化前である現在もプログラミング教室は開催されている。しかし、プログラミング教室を企画・運営のための情報が少ない。そのためRuby等を用いた小学生向けのプログラミング教室を運営するための事前の準備、教えるための道具、教える方法などのまとめを手引として提案する。

1.1 プログラミング教育必修化の背景

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を公開している。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からである。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であるためプログラミング教育が導入される。

1.2 プログラミング教室

小学校だけではなく、外部にもプログラミングを学べる教室は多くある。プログラミング教室の開催時期は1999年から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。プログラミング教室で学ぶ内容、プログラミング言語などについては1.2.1、1.2.2で説明する。

1.2.1 内容

プログラミング教室で使われているものは大きく分けるとテキスト型、グラフィック型がある[1]。まず、テキスト型、グラフィック型それぞれのメリット、デメリットについて述べる。

テキスト型

文字を並べてプログラムを作成するものである。

(例) ドリトル、JavaScript、Ruby、Python、java、Cなど

グラフィック型

画面上でプログラムを作成するものである。

(例) Scratch、Viscuit、アルゴリズムックなど

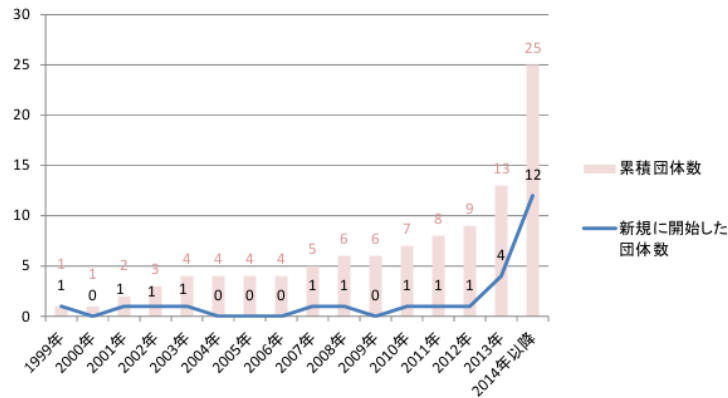


図 1.1: プログラミング教室始時期 [2] 図 5-3 より引用

表 1.1: テキスト型

○/△	理由
○	概念を表す語を追加することで表現力が大きくなる
○	入力、修正が比較的容易
○	複雑なプログラムも作成可能
△	書き方を覚える必要がある
△	すぐに見てわかりにくい

1.2.2 プログラミング言語

プログラミング言語とは、コンピュータへ指示を出すためのプログラムを作成するのに使われている言語である。プログラミングを教えるときは、教える人の対象、プログラムの難易度、参加者に作成したいプログラムに合わせてプログラミング言語を決める必要がある。主に利用されているプログラミング言語について説明する [2]。

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内の「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。

ドリトル

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。

表 1.2: グラフィック型

○ /△	理由
○	書き方、規則を覚える必要がない
○	文字を使わなくて良いため誰でも簡単に作成できる
△	修正に時間がかかる
△	複雑なプログラムを作成するのに限界がある

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。

1.2.3 プログラミング言語の割合

総務省によるとプログラミング教室で使用されているプログラミング言語の割合は Scratch が最も高い (図 1.2)。その理由は、短期間での学習に適していると考えられるためと述べられている [2]。

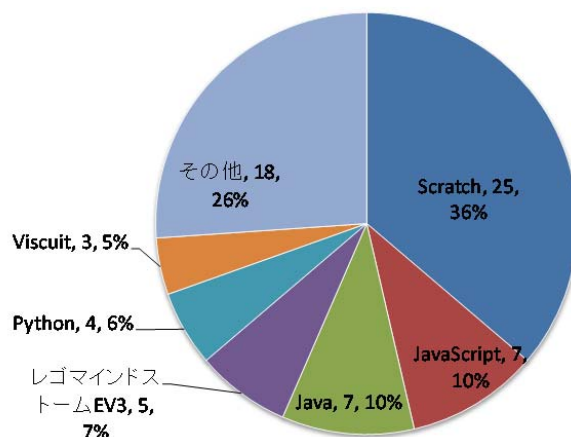


図 1.2: 教室・講座で利用されているプログラミング言語 [2] 図 5-11 より引用

1.2.4 使用されている機器

プログラミングで使用されているものの割合は図 1.3 である。多くのプログラミング教室では PC が多く使用されている。その他にもラズベリーパイという小型のボードコンピュータである (図 1.4)。Linux などの OS を搭載すれば、インターネットアクセスができるものである。

1.3 プログラミング教育の課題について

プログラミング教育では、プログラミングツールに関する課題、プログラミング環境に関する課題、プログラミング教育を行う教員の課題などが報告されている [3]。プログラムを分かりやすく指導するとなると道具、教材のだけではなく、行う場所、指導する人などの課題がでてくる。その点を踏まえて第 8 章では課題とプログラミング教室の企画、運営をした結果の考察をまとめた。

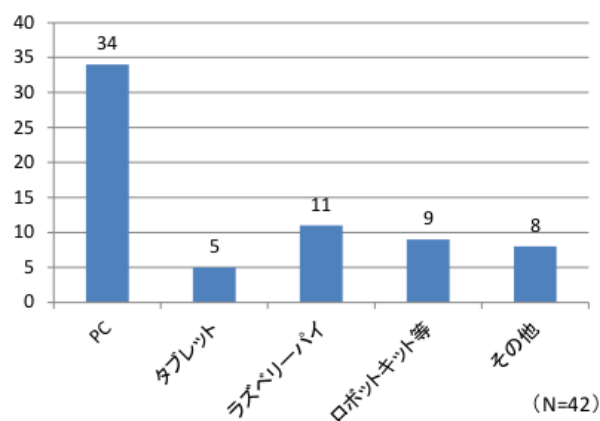


図 1.3: 教室・講座で使用されている機材 [2] 図 5-14 より引用



図 1.4: ラズベリーパイ

1.3.1 手引作成の背景

多くの Web サイトにてプログラミング教室を開催するための方法が公開されている。公開されている情報としてはとしてはプログラミング教育の視点、使用教材、所要時間などである。しかし、その情報では足りない部分が多くあると考えた。例えば、小学生にプログラミングを教える体制、教え方の方法など具体的な内容などである。そのため、実際にプログラミングを教える企画を立ててから授業を行うための手引が必要であると考えた。

1.3.2 対象読者

この手引は、プログラミング教室を企画、運営をしようと考えている人である。1.3.1 で述べたようにプログラミング教室を行うための情報が不十分である。そのため、準備方法、広報、他の機関との連携についてまとめている。

1.3.3 手引の構成

東北公益文科大学(以下本学)で行ったプログラミング教室「Ruby てらこった」の活動を基準にまとめた。はじめてプログラミング教室を企画、運営する人にも分かりやすいように決める順番で述べている。そのため手引の構成としては以下のようにになっている。

- プログラミング教育の実施のための必要項目
活動する人の必要な知識や能力、プログラミング教室を行う目的、小学生に身につけてほしい力、日程の立て方について
- 運営
企画したものを運営するための役割、業務内容、授業準備について
- 授業をするための方法
授業の流れ、当日の準備について
- 参加した小学生の感想
全5回の授業に参加した小学生が楽しかったと感じた部分、難しいと感じた部分について
- 教えるための要点
小学生に教える時の授業担当者、アシスタントそれぞれの注意点について
- 行うための一例
Ruby てらこったの活動のように企画、運営をするために何にどのくらいの時間をかけて準備しているのか、円滑の活動を行うための方法について

第2章 プログラミング教室実施のための必要項目

本章では、プログラミング教室実施のために必要な教える人の知識・必要な力と計画の立て方について述べる。教える人の知識・必要な力については、組織の利点を活かしてどのようにプログラミング教室を企画するかを述べていく。計画の立て方については、プログラミング教室を行うための目的に沿って効果的に進めるための方策について述べる。

2.1 教える側が必要なこと

プログラミング教室の活動を行う時に必要で必要なこととして「知識」「教える能力」「発想力」「時間管理能力」「伝える力」5つが必要であるのではないかと考える。プログラミング教室で教える場合はただプログラミングが得意である人を優先するのではなく、小学生に合わせて楽しく学んでもらえるように工夫できる人を優先した。その理由は以下のような理由である。

2.1.1 プログラミングの知識

プログラミング教室で実際に教える時は小学生の質問に対して分かりやすく教えることができることが前提になっている。そのために、ただ内容を理解しているだけではなく、自分なりに考えて教えられる程度に理解していなければならない。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でプログラミング言語のRubyを学ぶ。1年間で学ぶ内容は以下の内容である [4]。本学の授業では13の範囲まで学ぶが、Ruby たらこっただで教える範囲は1から7までである。

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 Web 上に効果的な形で視覚化させる方法を学ぶ

2.1.2 教える能力

ただ教えるだけでは、分かりやすく教えることができない。なぜならば、小学生の一人一人理解している部分、キーボードの操作の速度などが違うため、それぞれに合わせて教える必要があるからだ。小学生とコミュニケーションを取りながら分かりやすく教えるため、Ruby てらこったのメンバー募集は小学生と関わることを、教えることが比較的得意な学生を優先的にした。例えば、教職課程を履修している学生、学習支援サークルとして小中学生に学習を教えている学生、2年次の必修の基礎プログラミングのティーチングアシスタント (以下 TA) をしている学生などである。

2.1.3 発想力

どうしたら小学生がプログラムを楽しく、分かりやすく学んでもらうことができるのかをイメージしてアイデアを出す必要がためある。例えば、誰かが授業で使う資料を見せてきた時にこうしたらもっと良くなるなど意見を1つ出すだけでも分かりやすさが変わってくる。また、小学生の質問に対してどうしたら分かりやすく伝わるかなどを考える必要がある。

2.1.4 時間管理能力

割り振られたものを完璧に締め切りまでで完成させるためには、どのくらいの準備時間がかかるのかということ意識しなければならない。締め切りで提出すればいいという意識で作業をすると良い物ができない。そのために、ただ作業をするのではなく、他のメンバーから確認してもらい改善、修正する時間を作るということが必要になってくる。作成物の修正の時間のことを考えて余裕をもって作成をしなければならなので逆算をして物事を考えてくる必要がある。また、作業以外にも他にしなければならないことがあると考えられるため時間の管理というのを重視する素養が重要である。

2.1.5 伝える力

ここで使う伝える力とは、正確に相手に伝えるということの意味している。どの業務も自分一人だけで作業できることは少ない。そのため、メンバーで協力、分担しながら作業を進めていくことのほうが多い。そのためどの作業がどこまでできているのかなど具体的にメンバーに伝えるということは大切である。そのため、伝えるというのを重視する素養が重要である。

2.2 プログラミング教室の目的と小学生に身につけてほしい力

プログラミング教室を行う目的とプログラミング教室を通して身につけてほしい力について述べていく。

2.2.1 Ruby てらこったと活動の目的

本学では2017年度私立大学研究ブランディング事業の採択を受け、その活動一つとして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「Ruby てらこった」の企画・運営を行ってきている[6]。そのプログラミング教室を行う目的は、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。また、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。Ruby てらこったの対象は小学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。

プログラミングは、大学生が授業を行い教える。小学生1人に対して大学生が1人でサポートできるような体制にし、小学生一人一人に合わせて教えることができるようにした。授業内容は比較的簡単な内容であり、授業を通して簡単なゲームプログラムを作成できるように内容を組み立てた。

2.2.2 授業を通して身につけてほしい力

小学生の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」が述べられている[5]。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

身近な生活でコンピュータが活用されていることや問題の解決には必要な手順があることに気付くこと。

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 ならではの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作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。

2.2.3 教える範囲

教える範囲としては小学生が比較的的理解しやすいところであり、例えばクイズ、くじ引きのゲームを自分の力で考えられるようにするために必要なものを教える範囲とした。

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である。`print` と `puts` も同じように画面出力のためのものであるが、`puts` は末尾に改行した形で出力される。`printf` は書式付の `print` 文であるため数字などを出力させるために利用した。

文字列入力処理に必要なメソッド (`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なある。

制御構造 (`while`, `if`, `elsif`, `else`)

制御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り、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やクイズの結果の判定や条件で繰り返しの処理を行うために必要である。繰り返しの処理を行うために `while`、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
  hoge1
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 つを組み合わせることで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である。

sleep 関数 (sleep)

プログラムを時間指定して一時停止することができるため大学生の課題でも反応が良いため小学生も楽しむことができると考えたので必要である。

2.3 授業・日程

小学生に教える順番、授業時間の設定、開催日時について述べる。

2.3.1 教える順番と教える時間

基本的な操作から徐々に難しい内容になるよう教える順番を設定している。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。

教える時間は 1 回の授業で 2 時間としている。小学生が集中して授業ができるように概ね 45 分から 50 分ぐらい授業をしたら約 15 分位の長めの休憩を入れるようにした。

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

2.3.2 各小学校のとの調整

地域の小学生に参加してもらうために各小学校に資料を配布する。そこには、何をするのか、日程、問い合わせ先、申し込み方法などの内容をまとめ、資料配布では教育委員会、小学校に開催するプログラミング教室の内容日程など説明する。

2.3.3 開催日程

プログラミング教室の全5回の授業に参加し学んでほしいことから、小学生がプログラミング教室に参加しやすくするために学校が基本休みである土曜日、日曜日や学校・地区の行事と重ならないように開催日程を設定した。

第3章 運営

本章では、プログラミング教室を運営するために必要な組織の構築、授業準備について述べる。組織の構築では、役割、業務内容などの運営するための方法について、授業準備では、小学生に楽しくプログラミングを学んでもらうための必要な準備について述べる。

3.1 運営方法

開催に先立ち、それを運営する組織を作る必要がある。ここでは、Ruby てらこったでの構築例をもとに、実際の活動を円滑に進めるための事前準備、主な役割、業務内容、ミーティング方法について述べる。

3.1.1 役割関係の明確化

活動をする時に、スムーズに作業ができるように役割を明確にする必要がある。Ruby てらこったは、図 3.1 に示した体制を構築して運営を進めた。役割は、表 3.1 のように決めた。

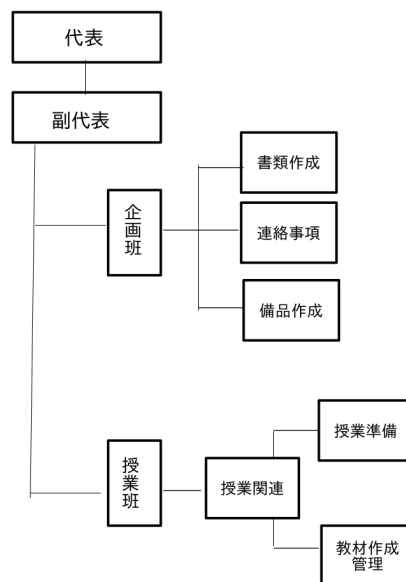


図 3.1: 組織図

リーダー、授業担当、各授業のアシスタント、タイムキーパー、連絡係について詳しく述べる。

表 3.1: 役割

役割	担当	内容
リーダー	固定	活動全体をまとめる
授業担当	交代制	プログラミングの授業を行う
アシスタント	固定	参加者のサポートをする
タイムキーパー	交代制	タイムテーブルに合せ担当者に時間を伝える

リーダー

リーダーは、全体的の計画を立てる、作業の進行状況の把握、ミーティングの進行などの業務がある。内容については、以下の通りである。

- 全体的の計画を立てる

計画では、いつまでに何をするのかを明確にする。明確にしたらグループ全員で内容を吟味する時間、内容修正する時間などを含め逆算し、作業の締め切りを決める。

- 作業の進行状況の把握

締め切りを決めてそのままにするのではなく作成を頼んだ人に対して、締め切りの何日か前に中間成果物を見せてもらうようにするなどして作業の進み具合を把握する。

- ミーティングの進行

何を話し合うのか決めておき、あらかじめメンバーに伝える。作成した成果物に対して全員からどうしたらもっと良くなるのかを意見をもらう時間にする。修正したものをいつまでに共有するか決める。例えば、授業用の教材の場合は、小学生でも分かりやすいか、もっと分かりやすくできないかなど意見をもらう。その後、次のミーティングまで修正した教材を作成するというをした。また、業務の割り振りなどもする。業務の割り振りでは、メンバーの話を聞き、それをふまえて的確に割り振る必要がある。ミーティングの流れは図 3.2 で説明する。

授業担当

各5回の授業を一人で行うのは、大変であるため、各授業で担当者を変えた。授業担当の業務は以下のようなものがある。

- 教材作成

担当の授業で使用する教材を作成する。教材としては、スライド、サンプルプログラムなどである。詳しくは 3.3 で述べる。

- 模擬授業

実際の授業を想定し、タイムテーブルを確認しながら行う。自分では分かりやすく教えているつもりでも相手には分かりにくい部分などは気づきにくいので、多くの人から意見をもらう機会を作る。

- 授業

小学生の様子(操作はできているか、指示が伝わっているか)を見ながら分かりやすく丁寧に意識して授業を行う。

各授業のアシスタント

小学生一人に対して一人のアシスタントをつけ、参加した小学生が学びやすいようにした。また、小学生が声をかけやすいように、なるべく毎回同じアシスタントがつけるようにした。アシスタントの業務は以下の内容である。

- 小学生が分からない所の説明
授業の内容が難しかったり、分からなかった場合にサポートしている小学生に合わせて教えることをする。
- 操作の補助
キーボード操作が遅い場合がある。その時に、キーボードの位置を教えたり、コマンドを教えたりする。

タイムキーパー

タイムキーパーは、各授業のタイムテーブルに合わせて授業担当者に時間を伝える。そのため、授業の進み具合など確認して授業が円滑に進むようにサポートする。時間に余裕がある時は、プログラム作成の時間や発表時間などにするなど休憩中に打ち合わせする。反対に、時間が想定よりも足りない時は、変更する部分など提案してなるべく時間内で授業が終わるように考えて動く必要がある。

連絡係

連絡係は参加する保護者の方にメールで連絡を取ることをする。はじめに送ったメールに対して返信してもらい連絡が取れることを確認する。メールで連絡をするので誰が読んでも正確に伝わる内容を早くお知らせすることを意識して業務をする必要がある。また、メールで質問が来た時も対応をする。連続して連絡をとるため、連絡係を固定する。

3.2 必要な道具

授業で使う道具は以下のようなものを用意した。

表 3.2: 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る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
マグネットシート	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に使用

3.2.1 PC

小学生用、授業用のPC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、本学の計算機と同じOSだと学生が教えやすいからである。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。PCは、小学生用、授業用の他にスライド用のPCも用意した。

- 小学生用 PC

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した。

- 講師用 PC

小学生に分かりやすくするため小学生の使用している計算機と同じものを使用した。

- スライド用 PC

教室では2つのプロジェクタが利用できるため、教卓PCから実際の操作画面と同時に、教材のスライドを映すためのPCを別途用意した。教室に2つプロジェクターが無かった場合は大型液晶ディスプレイに映して授業をした。

3.2.2 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(24ページ図(a))。USBメモリは、OSのインストールした後に、参加者一人のアカウントを作成し、名前を記入して配布をした。

3.2.3 教科書

授業内容に合わせて、学生らで作成した(24ページ図(b))。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

小学生でも分かるようにファイルは、絵を書く時の紙でディレクトリは紙をしまっておく場所というようにした

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

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(9.1を参照)

3.2.4 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(24ページ図(c))。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。

3.2.5 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(24ページ図(d))。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。

3.2.6 マグネットシート

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(24ページ図(e))。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。

3.2.7 アルファベット対応表

小学5年生は、ローマ字の大文字だけ習っているため、小文字で書かれたプログラムリストの小文字はわからない。特に、大文字と小文字が違う「l(エル)」と「i(アイ)」や「h(エイチ)」と「n(エヌ)」と「r(アール)」などが小学生にとって分かりにくい。そのため、大文字

小文字を理解するために作成し配布した(24ページ図(g))。また、アルファベット対応表の配布だけではなく、小文字で書かれたRubyの単語の文字ごとに大文字のルビを振るようにした。

3.3 授業準備

授業の準備では授業を行うまでの流れを述べていく。

3.3.1 内容の吟味

授業での内容をミーティングなどで話し合い内容を確認する。教える内容では、小学生でも分かりやすく学ぶことができるものを中心にしている。そのため、小学生にとって内容が難しすぎないか、変数などのイメージ図はどうするかなど話し合う。1回分の授業で教える内容が多くなり過ぎないか、教えるプログラムは分かりやすいか、プログラム作成の時間が十分に取れるかなどを考える。教える内容が多くなり過ぎないかという基準については模擬授業を試みたり、各授業での小学生の理解度を見たりして内容を決めた。プログラムについては3.4で述べる。

3.3.2 教える体制

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。

3.4 各授業のサンプルプログラム

授業用のサンプルプログラムは授業の内容のポイントをおさえたものにした。授業中に説目を加え理解を深めてもらうようにした。

3.4.1 1回目のプログラム

画面出力のprint,putsを使ったプログラムを作成している。このプログラムでは、printとputsの違いを理解してもらうために両方使用した。画面に出力される””部分の文字を変更する。速く操作ができる小学生にはprintまたはputを増やして好きな文字を書き込ませて変えるようにした。

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

print "名前\n"
print "学校名\n"
puts "好きな食べ物"
puts "好きな動物"
```


3.4.2 2回目のプログラム

変数、whileのループを使ったプログラムを作成している。gets.chompを最初に追加して所持金を入力させたり、最後の合計に1.08をかけて消費税込みの金額を出したりなど変えることができる。

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

puts "スーパーのレジ買ったものの値段を入れてね。(100 円→100 と入力しよう)"
memo = 0

while true
  print "値段は? (終わりたい時は q を押してね)"
  nedan = gets.chomp

  if nedan == "q"
    break
  end

  memo += nedan.to_i
  printf("今の合計は %d 円だよ \n",memo)
end
printf("今回の合計は %d 円です。お買い上げありがとうございました! \n",memo)
```

3.4.3 3回目のプログラム

配列と乱数を組み込んだプログラムをサンプルにした。この配列の部分をグー、チョキ、パーに変えるとジャンケンプログラムに変えることができる。

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

kuji = ["1等","2等","3等","4等","ハズレ"]
print("なにができるかな～?")

puts("....")

sleep(1)
srand()
nani = rand(5)
printf("あなたは%sでした!\n",kuji[nani])
```

3.4.4 4回目のプログラム

条件分岐のifを使いクイズプログラムをサンプルにした。クイズの問題文を変えるだけでなく選択肢を増やしたり、回答によって違うものを画面出力できるように変えることができる。

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

puts "今の元号はなんですか?"
```

```
puts "1:平成"
puts "2:令和"
puts "3:昭和"

print "答えを入力:"
kotae = gets.to_i

sleep(1)

if kotae == 1 then
  puts "惜しい！ハズレだよ！"
elsif kotae == 2 then
  puts "正解だよ！読み方はわかるかな？？"
elsif kotae == 3 then
  puts "ザンネン！ハズレだよ！"
else
  puts "その番号はないよ！！"
end
```

3.4.5 5回目のプログラム

既習の内容を含めたプログラムをサンプルにした。

```
#!/usr/bin/env ruby

puts "コンピュータとじゃんけんをしよう!"
puts "5回中3回勝てればクリアだよ!"
sleep(3)
puts ""
print "それじゃ GAME START\n"
puts ""

win = 0
kaisuu = 1
janken = ["だしてないよー", "グー", "チョキ", "パー"]

while kaisuu <= 5
  sleep(2)
  puts ""
  printf("%d回目!(あなたの勝利数:%d)\n", kaisuu, win)
  puts ""
  print "じゃんけん!(グーなら「1」 チョキなら「2」 パーなら「3」を押してね): "
  while true
    you = gets.to_i
    if you >= 4
      you = 0
    end
    com = rand(3) + 1
    printf("ぽん!(あなたは「%s」でコンピュータは「%s」)\n",
           janken[you], janken[com])
    sleep(1)
    if (you==1&&com==2) || (you==2&&com==3) || (you==3&&com==1)
      puts "あなたの勝ち!"
      win += 1
      kaisuu += 1
    end
  end
end
```

```
        break
    elsif (you==1&&com==3) || (you==2&&com==1) || (you==3&&com==2)
        puts "あなたの負け!"
        kaisu += 1
        break
    elsif you == com
        print "あいこで(グーなら「1」 チョキなら「2」 パーなら「3」を押してね): "
        redo
    else
        puts "なにもだしてないからあなたの負け!"
        kaisu += 1
        break
    end
end
end
end

sleep(3)
printf("あなたの勝利数は「%d 回」\n", win)
sleep(3)

if win >= 3
    puts "おめでとう!"
    sleep(2)
    puts "+-----+"
    system 'banner YOU WIN!!'
    puts "+-----+"
else
    puts "残念..."
    sleep(2)
    puts "+-----+\n"
    system 'banner GAME OVER'
    puts "+-----+\n"
end
```

3.4.6 スライド

授業で使うスライドはあくまでも授業の副教材であることを意識する。スライドに書く内容は図、重要な部分だけにする。するとポイントなどが小学生にも分かりやすく伝えることができる。

3.5 模擬授業

模擬授業では授業担当者が実際の授業を想定して行う。主に見るポイントとして以下のようなものを確認する。

- 教え方が分かりやすいか (プログラムの説明など)
- ノートを取る部分の確認
- スライドが見やすいか (スライドの配色、文字が多すぎないか)
- 時間配分 (プログラムの作成時間があるか、時間が無くなったらどこを省いて説明するか)

3.6 他の機関との調整

プログラミング教室を運営をするときは単独で行うことが難しいと考えるための他の機関との協力が必要である。本学が執り行った場合について記す。

プログラミング教室を行う前に、広報資料(活動内容、プログラミング教室活動日時、募集条件、締め切り、問い合わせ先などの情報)配布である。資料配布のための募集期間、締め切りなどの時間を考えた。日程が決まったら本学から酒田市教育委員会に連絡をとり、教育委員会から各小学校に資料を配布して、その後各小学校の先生から小学生に資料を配布して参加したい小学生から連絡をもらった。

プログラミング教室を開催すると決めて参加者を募集する時に、プログラミング教室に参加したいと考える小学生がどのくらいなのかが分からず参加人数予想ができなかった。そのため、酒田市内の全ての小学校に募集をかけるのではなく、本学の近隣の小学校、小学生数が多い小学校という条件のもと小学校を4校に絞って募集をすることを決めた。その後小学校の先生にプログラミング教室の説明を行い許可を得て資料配布をした。参加募集期間の締め切りを資料配布から約2週間ほどにした。するとある程度の小学生が実際にプログラミング教室に参加申し込みをしてくれた。

7、8月に2019年度第2回目のプログラミング教室をおこなった。最初の4校の募集により各小学校からの参加人数が大体予想がついたのと、小学校が夏休み中であるため、参加可能人数を増やしたいと考えた。そのため酒田市の全部の小学校に募集をかけた。その時に全部の小学校を訪問、プログラミング教室の説明、資料を配布するには時間がかかるため、酒田市内の小学校に配布するのは、酒田市の教育委員に活動の説明を行い協力を得て配布してもらった。

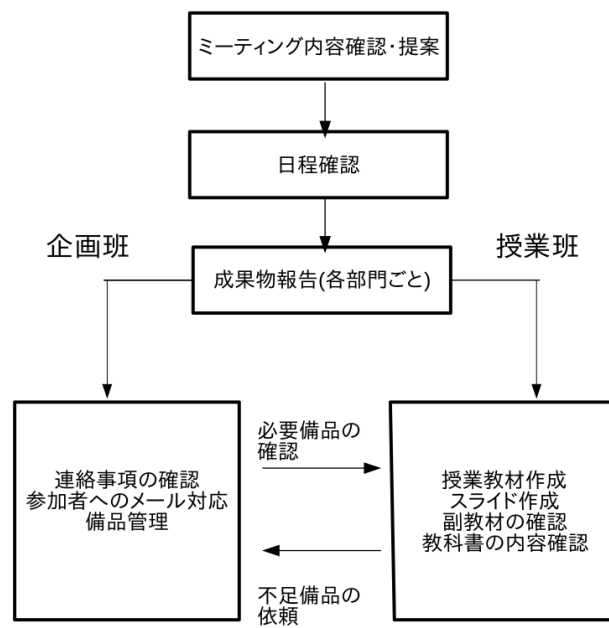


図 3.2: ミーティング方法

if 文 は、場合分けをしたいときに使う

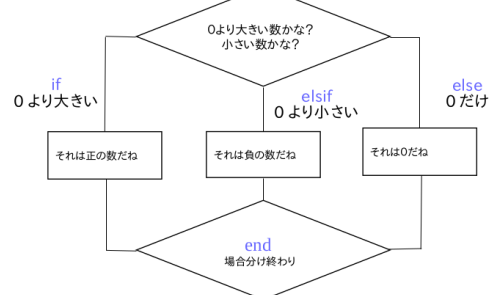


図 3.3: 条件分岐イメージ図

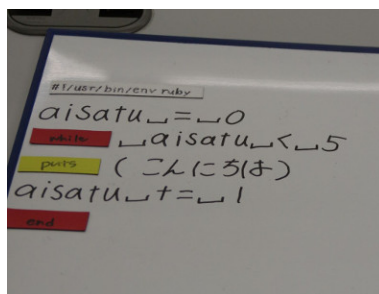
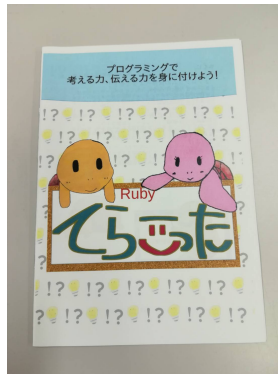


図 3.4: while と end のマグネットシートを使用



(a) USB メモリ



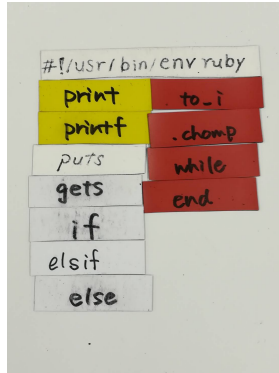
(b) 教科書



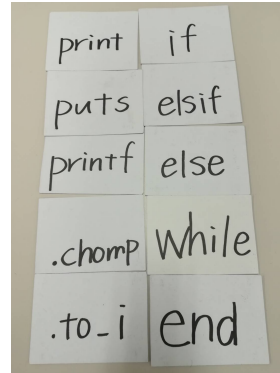
(c) ノート



(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)

— アルファベット表 —

A a	B b	C c	D d
E e	F f	G g	H h
I i	J j	K k	L l
N n	M m	O o	P p
Q q	R r	S s	T t
U u	V v	W w	X x
Y y	Z z		

(g) アルファベット対応表

第4章 授業をするための方法

本章では、少し難しいプログラムの内容でも小学生に分かりやすく、楽しく学んでもらうための当日の準備作業、授業の流れ、プログラムの発表方法、授業後のミーティングについて述べていく。

4.1 当日の準備・確認事項

当日は、授業がスムーズに進めることができるように授業が始まる1時間程度早く集合し準備する。まず、備品のホワイトボード、ペン、マグネット、PCをの設置などをする。PCの設置では、ネットワークが繋がるかを試す。その後、授業のおおまかな流れを確認する。

4.2 授業の流れ

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。例えば、1回目の授業であれば、print, puts などである。速く入力が終わった人には、5回など入力する回数を増やすようにした。

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

長めの休憩を取る。休憩の間には、小学校の話し、習い事、趣味など小学生とコミュニケーションを取る。

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シートを貼ったり、ペンで書いたりしてからPC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である。

8. 作成したプログラム発表

作成したプログラムは、発表する時間を確保する。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4.3で詳しく述べる。

4.3 プログラム発表について

自分の作成したものを発表し伝える力を身につけること、他の人の発表、プログラムの良さなどを考えてもらうため、授業で作成したプログラムを発表する機会を設けた。自分の作成したものを発表し伝える力を身につけること、他の人の発表、プログラムの良さなどを考えてもらうためである。授業で行った発表方法は2つある。その方法は以下の通りである。

みんなの前で発表する

最初に作成したプログラムの難しかった点、頑張った所など話してもらった後に、プロジェクターに映し、作成プログラム実行をして発表する方法である。発表する時に何を話せばいいのか困ってしまう小学生もいるのであらかじめ話してほしい項目を提示すると困った様子が見られなかった。

他の人のプログラムを実行する

参加者全員で座席を順に移動し、他の人の作成したプログラムをお互いに体験させる。この方法では、実際に実行できるので工夫している部分など気づくことが多い。実行してみた感想などお互いに伝えることができる。

4.4 授業後のミーティング

授業に参加したメンバーは、授業後に約30分ほどのミーティングをする。まずはじめに、授業の担当者から良かった点、悪かった点を先に話してもらう。悪かった点についてはメンバーから改善策、意見をもらう。そのあとに、授業のアシスタントから参加した小学生の様子、理解度についての意見をもらう。その後次回までの授業の改善点、対応策、準備しなければならないことなどを話し合う。

第5章 授業を受講した小学生の感想

1回目から5回目まで授業後にアンケート(授業で楽しかったこと、難しかったこと)をとった。以下の表はそのまとめである。

表 5.1: 楽しかったこと

回数	内容
1回目	PCの操作、コマンド操作、自己紹介プログラムを発表したこと、大学生と話しをしたこと
2回目	キーボードの操作、みんなが作成したプログラムを実行すること、大学生と話をしたこと
3回目	みんなが作成したプログラムを実行すること
4回目	プログラムの作成、大学生と話しをしたこと
5回目	オリジナルのプログラムを作成したこと

表 5.2: 難しかったこと

回数	内容
1回目	PCの操作、自己紹介
2回目	break、ループが難しかった、プログラムを改造すること
3回目	乱数でランダムに取り出すこと、自由にプログラムを作ること
4回目	if,elsif,elseの使い方が難しかった、プログラムを改造すること
5回目	変数が多くなりすぎて難しかった、オリジナルのプログラムを作成すること

5.1 授業で楽しいと感じる部分

楽しいと感じた部分として発表が楽しかったという意見が多かった。発表を聞き、他の参加者のプログラムを動かしてみるのが新たな発見があり楽しかったのだと考える。

また、大学生と話しをするのが楽しかったという意見があった。プログラムを作成する時間や休憩の時間など多くの場面で話す機会があったのは良かったと考える。

5.2 授業で難しいと感じる部分

操作の部分ではキーボードの操作が難しいと感想に書いていた。PCをあまり使わない小学生にとってキーボードを使い入力するには、どこに入力したい文字、数字があるのか探しながらかつので思ったよりも時間がかかるため難しく感じたのだと考える。

プログラムの内容では変数の部分が難しかったという感想が多かった。2回目から5回目にかけて難しいプログラムを作成していくと、変数が徐々に増えていく。そのため、何をどの変数にしたか確認しながらプログラムを作成するため難しく感じたのだと考える。

第6章 教えるための要点

この章では主に実際に小学生にプログラミングを教えるための注意点をまとめている。

6.1 教える上で注意すること

授業担当者と授業アシスタントの両方の視点から述べていく。

6.1.1 授業担当者

授業担当者は、小学生に教えるということで内容を噛み砕いて教えることが重要になってくる。授業は基本的に操作など全員一緒に進んでいくため常に小学生の顔を見るようにする。すると操作が終わっているのか、まだ作業の途中なのかなど確認することができる。進み具合をみて、かなり遅れている場合はアシスタントに頼んでサポートしてもらおう。授業中は、ただ話すのではなく、常に問いかけたり、手を上げてもらったりと反応を見るようにする。話を聞く、ノートを取る、ホワイトボードに書き込む、プログラムを作成するなど指示をしっかりと出して、何をするのか明確にすることが大切である。

6.1.2 授業アシスタント

授業アシスタント者は操作、授業で理解できなかった部分の説明など個人に合わせてサポートする。教える小学生に対して目線を合わせて教えることが大切である。難しい内容でも分かりやすい言葉を使う。プログラム作成の時は、全てサポートするのではなく考えてもらう時間を大切にする。操作が遅くて焦ってしまう小学生がいた場合は、声をかけて、ゆっくりでも問題ないことを伝えて、確実に操作してもらおうようにする。

6.1.3 小学生の理解度に合わせる

一人一人授業の内容の理解度、操作速度などは違う。そのため一人ひとりに合わせた教え方をしなければならない。

- 理解が早い小学生

理解が早い小学生には、サンプルのプログラムを改造してもらおうということをして、何もしていない時間をなくすようにした。最初にどの部分を変えると良いのかヒントを出しながら少しずつ教えていった。

- 操作が遅い小学生

操作が遅いと追いついていかなければと焦ってしまう小学生がいる。そのときは、どこまでできているのか確認し、話しかけながら操作のサポートをする必要がある。

6.2 その他

その他として授業が計画通りにいかなかった場合、参加者が来れなくなった場合、PCが調子悪くなった場合の対応を述べる。

6.2.1 授業の内容が予定より進まなかった場合

どこを中心に教えるのかということを考える。説明に時間がかかりすぎてしまったときは、次の授業でどのぐらいの時間で説明するかのと打ち合わせをして教える範囲を全て5回で終わらせるようにする。

6.2.2 参加者が来れなくなった場合

全5回の授業で1つどこかの回の授業に参加できなかった場合は次回の授業の**30分前**に来てもらい簡単な授業を受けてもらい、その後次の授業に参加してもらった。すると、次の授業においていかれることなく参加してもらうことができた。

6.2.3 利用しているPCの調子が悪くなった時

インターネットを使いWebページを見る時に無線LANではうまく繋がらない場合があった。そのときは、有線で接続をした。次の授業からは予備のパソコンを使い対応をした。

第7章 実際に行うための一例

Ruby teraこったの活動を通して良かった部分、悪かった部分があった。そのことを踏まえて企画、運営を円滑に行うための提案をする。プログラミング教室を企画から開催するまでの大まかな流れ、内容をまとめる。詳しい部分は前の章を参考にすると良い。

7.1 企画

企画ではこのような流れで役割、教える範囲などを決めた。

1. 活動するメンバーを集める

活動をするにあたってどのようなメンバーが必要か考え集める。

2. 責任者の選出

リーダー、副リーダー、企画班リーダー、授業班リーダーを決める。

3. プログラム教室の目的を明確にする

プログラミング教室を通して小学生に何を学んで欲しいのか、身につけた力などを考える。

4. 教える範囲を決める

教える範囲は小学生でも理解しやすい内容、最終的に小学生にどのようなプログラムを自分で考え作成できるようになってもらうのかを決める。

5. 役割を決める

企画班 (書類作成係、連絡事項係、備品作成係) 授業班 (授業担当係、教材作成・管理係) を決める。

6. 授業について決める

教える範囲をどの順番で教えていくか考える。それぞれの授業で教えるイメージ図についてなどを決める。

7. 日程について決める

日程は全5回の授業であるため小学生が参加しやすい日程を考えて組み立てる。

8. 資料の配布について決める資料を作成したら配布する範囲、方法など決める。

7.2 運営

運営の流れは以下の通りである。

1. 役割決め

運営が始まると外部の人と連絡が取ることが多くなるため連絡をして日程調整などをする役割を決める。

2. 必要な道具の決定

授業に必要な道具 (パソコン、USB メモリ、マグネット、ホワイトボード、教科書など) を決める。

3. 教える体制を決める

どのような体制でアシスタントを何人にするか決める。

4. 内容の吟味

授業担当者が考えてきた授業の構成、教える内容、タイムテーブルなどを話し合い決める。

5. サンプルプログラムを決める

授業のポイントが入ったサンプルプログラムを決める。

6. 模擬授業

実際の授業を想定して、作成したタイムテーブルをもとに授業をする。スムーズにできるまで繰り返す。

7.3 広報関係提案

広報関係では本学から教育員会に資料を持っていた。その後教育委員会から各小学校へポスターを配布した。ポスターには以下のような内容をまとめる。

- 内容
- プログラミングとは何か
- 日程
- 参加条件
- 時間
- 場所
- 問い合わせ先
- 申し込み方法
- 申し込み締め切り

小学5,6年生向けプログラミング教室

Ruby てらこた **無料**

Q. プログラミングってなに?
A. 今みんなが当たり前に使っているYoutubeやゲーム、スマホアプリはプログラマーと呼ばれる人達によって作られているんだよ。そして、そのプログラムを作る作業のことをプログラミングというよ。

Q. プログラミングができて...?
A. ゲームを作ったりクラスのみんなができないようなことができるよ。それに、算数や英語を使うから普段の勉強にも活かせるよ!

真のプログラマーへの第一歩!!

ACCESS MAP

A日程: 8月1日, 2日, 3日, 4日, 5日
B日程: 8月4日, 5日, 6日, 7日, 8日
定員: 20名(各日程10名)
参加条件: 無料
参加条件: 余5回出席できること
小学生5,6年生であること
時間: 10:00-12:00(休館日)
日日程の4日, 5日のみ14:00-16:00(休館日)
内容: Rubyを用いたプログラミング
場所: 東北公益文科大学(酒田市飯森山3丁目5の1)
問い合わせ先: terakotta@e.koeki-u.ac.jp
Web: <https://www.koeki-pj.org/ruby/>
締切: 7月27日(金)
申し込み: 上記URLからメールまたはWebフォームにて
※お問い合わせはお気軽どうぞ



7.4 ミーティング

週に1回メンバーと約1時間ほどのミーティングを行う。内容としては以下のような項目である。ミーティングした内容は、メンバー全体に共有できるように必ず記録を取った。

- 成果物の発表
- 成果物に対して意見をもらう
- 授業に向けて
- 次回まで準備するもの

4/17 議事録

○企画班

- ・名刺は人数分完成して発注済み→来週配布
 - ・ゴールデンウィーク前までにポスターの原案を作っておくこと。
- 5月中に小学校にアポを取り、配りにいく予定。

○教材班

- ・各担当ごとに進行状況を確認
- 第1回:担当者不在により不明。
 第2回:発表用スライドは完成。while文はS4にあるプログラムを使用予定。
 第3回:プログラムは以前S4に載せたものを使用予定。スライドはまだ完成していない。
 第4回:スライドを使わないで見本を見せながら進めることをイメージしている。
 第5回:ゲーム様式のプログラムを使う予定。プログラム自体はできている。スライドはまだ完成していない。

※課題

- ・企画班は1週間後までポスターを制作し、先生と確認する。
- ・教材班は来週の火曜日までに、プログラムとスライドをS4にあげること。
- ・Webは今日中にあげること。
- ・自分の成果物をメモし、把握しておくこと。

5/8 議事録

○グッズ

アンケート作成

- 内容・公益大を知っていますか?
 ・プログラムしたことありますか?等

○教材班

- ・毎週授業始め5分でタイピング練習をする
 - ・マニュアル(教科書)を作る
- 小学5・6年の社理数を盛り込む

・リハーサル1人20分の予定
 スライドを使って一連の流れを通す、ギャグも入れ込む
 ○てらこったスケジュール
 第1回6月10・17・24 7月1・8
 第2回8月1・2・3・4・5・6・7・8・9

5/15 議事録

○企画班

- ・アンケートの修整
- ・授業で使うマグネットの作成

○授業班

- ・タイピングのシステムを統一
- ・前回のまとめ課題を授業の最初に解説
- ・スライドは図や絵を多くしたほうが好ましい
- ・キャラクターを使う
- ・文字のフォントはかわいいものが好ましい

○Web班

- ・「無料」を大々的にアピール
- ・5回全て出席することを説明
- ・全ての項目において詳細を載せる

○その他

- ・教室は前2列を子供、後側を保護者用で分ける
- ・アシスタントは子供達のとなりに待機
- ・お菓子テーブルを設置
- ・保険は子供達の名前と住所が確定したら加入する
- ・アシスタントの最小人数は5人

○準備するもの

- ・名札
- ・ノート
- ・筆記用具

7.5 申し込み方法の提案

申し込み方法で情報、確認が確実なのは、申し込みを Web で行い、参加者が決定後に電話でやり取りするのが良い方法だと考える。その理由としては、Web でおこなうことで必要な情報を残すことができるからである。Web フォームを用意し、名前、学校名、住所、メールアドレス、電

話番号などを入力してもらった(図 7.1)。第1回の運用実績から、申し込み時期の確認は電子メールが、開催中の予定変更連絡などは電話が確実であると判断したため両方を入力してもらうことにした。

上記の記載事項に同意いただける場合は、下記のお申し込みフォームをお書きください。

参加者氏名	<input type="text" value="例) 寺子 太郎"/>
参加者氏名 (ふりがな)	<input type="text" value="例) てらこ たらう"/>
性別	<input type="text" value="性別を選択 ▼"/>
メールアドレス	<input type="text" value="例) terakotta@e.koeki-u.ac.jp"/>
電話番号	<input type="text" value="例)01234567890"/> ※半角数字で入力 ※電話番号は、市外局番からハイフン (-) なしで記入してください
住所	<input type="text" value="例) 〒998-8500 山形県酒田市飯森山3丁目5番1号"/> ※住所は、郵便番号から記入してください
学校名	<input type="text" value="例) てらこった小学校"/>
学年	<input type="text" value="学年を選択 ▼"/>
クラス	<input type="text" value="例) 1"/> ※1から9までの半角数字で入力 (1クラスしかない場合 → 1)
出席番号	<input type="text" value="例) 01"/> ※2けたの半角数字で入力 (出席番号1番 → 01)
参加日程	<input type="text" value="参加日程を選択 ▼"/>
備考	<input type="text" value="電話対応可能な時間や、その他連絡事項などあればお書きください。"/>

図 7.1: Web 申し込みフォーム

7.6 授業

授業では、分かりやすく丁寧ということを意識する。小学生の反応、操作の進み具合などを確認して教える速度を調整する。アシスタントは、小学生が途中で困らないように声をかけながらサポートする。

7.7 授業後のミーティング

授業後のミーティングは、授業担当者、アシスタント、タイムキーパーなど全員が授業を振り返る時間にする。悪かった部分に対しては、改善策を考えるようにする。

第1回の反省点

小文字がわからない

おやつを小分けにする

アシスタントが介入しやすいようにする

90分のつもりで進行

アシスタント 2,3人じゃ足りない TA1人に対して児童2人くらい欲しい

ネットが繋がらなかった
 教科書等を持ち帰らせる
 アシスタントはもっと盛り上げる
 C-x C-fは2回まで
 C-x C-sは大丈夫
 英語入力と日本語入力の切り替えが難しい 日本語使うときには前もって言うしておく もしくは切り替えを減らす
 重要なところはノートに書いてねと言う 小学生はどこが重要かわからない
 講師は1回の指示に一つまでにしておく
 指示は3回言う
 アシスタントの守備範囲を決める（親も守備・対応範囲）
 タイムキーパーが欲しい
 飽きないような工夫 入力させるものを増やす
 自分なりの作業
 PCは同じ場所に座らせる

7.8 6月にプログラミング教室を開催の実例

2018年度の活動日程をまとめたものである。

表 7.1: 開催までの流れ

日付	企画	運営
2月23日	メンバー募集	
2月28日	ミーティング(企画について)	
3月8日	責任者選出	運営計画
3月8日		Web作成開始
3月26日	授業内容の決定	必要な道具を決める
4月2日	授業内容の吟味	
4月10日	授業担当者の決定 授業体制の決定 教材作成者の決定	使用するOSの決定 必要な道具発注
4月17日	資料配布小学校決定 授業内容の吟味	ポスター作成開始
4月19日	配布ための連絡調整	
4月24日	配布資料完成 授業の大まかな流れ確認	
5月1日	成果物の確認	連絡状況の確認
5月8日	模擬授業について 教科書作成開始	

表 7.1: 開催までの流れ

日付	企画	運営
5月10日		各小学校へ資料配布
5月15日	マグネットシートの作成	授業スライドの確認
5月15日		使用する教室の予約
5月22日	各授業の確認	当日の日程について
5月28日	当日に必要なモノの確認	
5月29日	申し込み締め切り	
6月5日		参加者への連絡
6月7日		Web公開
6月8日		1回目模擬授業
6月10日		1回目授業 1回目授業についてミーティング
6月12日	2回目の授業に向けて	2回目模擬授業
6月17日		2回目授業
		2回目授業についてミーティング
6月19日	5回目授業の必要なモノの確認 3回目の授業に向けて	
6月21日		3回目模擬授業
6月24日		3回目授業
		3回目授業についてミーティング
6月26日	4回目の授業に向けて	4回目模擬授業
7月1日		4回目授業 4回目授業についてミーティング
7月3日	5回目の授業に向けて	
7月8日		5回目授業 5回目授業についてミーティング

第8章 課題の考察

プログラミング教室を企画運営をし、1.3で述べた課題について考察をする。

プログラミングツールに関する課題

Ruby であらったでは Ruby をツールとして使うのを決めた理由は3つある。1つ目の理由は、本学では2.1.1で述べたように Ruby を用いたプログラミングを学んでいる。学生が教えるということを前提としたので Ruby とした。2つ目は、他ではあまり Ruby を教えてないということである。1.2のようにScrachを教えているプログラミング教室が多いことから他のプログラミング言語にした。3つ目の理由は、小学生のうちに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考えたからである。

プログラミング環境に関する課題

一人一台のPCを用意した。私立大学ブランディング授業で補助金でPCを揃えた。プログラミング教室を行う程度であれば、中古の2万円前後のPCで十分である。そのため、中古のPCを参加想定人数分を準備した。

プログラミング教育を行う教員の課題

小学生に指導する大学生は、分かりやすく教えるために3.3で述べたように授業準備を行った。準備をするときは、本番の授業をシミュレーション(タイムテーブルなど)して行った。

第9章 付録

使用した補助教材は、授業で使用した、コマンド集とアルファベット対応表を添付する。

9.1 コマンド集

コマンド集(簡単 Ver.)

イーマックス
Emacsへ移動
C-1

ケーターム
ktermへ移動
C-2

コンソール
consoleへ移動
C-3

Emacs 編

C-x C-f	ファイルの作成 / ファイルを開く
C-x C-s	ファイルの保存

kterm 編

cd ディレクトリ	[ディレクトリ]に移動
cd	ホームディレクトリに移動
mkdir ディレクトリ	[ディレクトリ]を作成
ls	今いるディレクトリの中にある ファイルを表示
ruby ファイル名	[ファイル名]を実行

console 編

exit	ログアウト
shutdown -p now	電源をおとす

9.2 アルファベット対応表

－ アルファベット表 －

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		

参考文献

- [1] 上松理恵子 久野靖 萩谷昌己 . 「小学校にプログラミング教室がやってきた!」 株式会社三省堂,2016,p144
- [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] 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.
- [4] 広瀬雄二. 「Ruby プログラミング基礎講座」, 技術評論社,2006,p312
- [5] 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).
- [6] 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 I T 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.