GoogleColaboratoryにおけるPythonの導入実践 一斉授業型，動画視聴型の比較

山本　周[1,a)]　清水 克彦[2,b)]

概要：「情報Ⅰ」の導入により，プログラミングが必修となった情報の授業では，文部科学省が公開した教員研修用教材でPythonが使用されており，採択される可能性が高い．そこでPythonによるプログラミング指導の授業形態による効果の検討を目的とした．本実践は2019年度に一斉型，2020年度に動画配信型の形態において，“Google Colaboratory”を用いたPythonの導入授業を行った．プログラミング経験者が1割程度である高校3年生に対して，5時間の実施をした．“Google Colaboratory”を用いることで面倒な環境構築がなくなり，関連するライブラリのインストールも容易であった．また，動画視聴型の授業形態により，生徒のコーディングのエラー等の対応における教員負担が減少し，生徒においては個別最適化された学習が可能となった．“GoogleColaboratory”によるPythonの指導は，導入なども含めて有用であることが分かった．さらに，アンケートの結果からは生徒の興味を引くことができる等の結果も得た．

キーワード：情報I，Python，Google Colaboratory，プログラミング

1. はじめに

高等学校の共通教科情報科は，高度情報化社会に対応した人材を育成するために，情報の収集・分析から発信までを総合的に学ぶために，2003年に新設された教科である．当初は，情報教育の3本柱である「情報活用の実践力」を重視する科目「情報A」，「情報の科学的な理解」を重視する科目「情報B」，「情報社会に参画する態度」を重視する科目「情報C」の3科目の中から，1科目を選択必履修する形で始まった．その後，2009年の学習指導要領改訂 [1] では内容が整理され，選択必履修科目も「社会と情報」と「情報の科学」の2科目となった．そして，2018年に告示された新学習指導要領解説 [1] では，「問題の発見・解決に向けて，事象を情報とその結び付きの視点から捉え，情報技術を適切かつ効果的に活用する力を全ての生徒に育む」ことを目標とし，従来の科目選択制から共通必履修科目として「情報Ⅰ」が設定された．さらに発展的な内容の選択科目として「情報Ⅱ」を設けた．「情報Ⅰ」の学習内容の構成 [1，p.21] は以下の通りである．

内容（1）情報社会の問題解決
内容（2）コミュニケーションと情報デザイン
内容（3）コンピュータとプログラミング
内容（4）情報通信ネットワークとデータの利用

今回の改訂における中心的な変化としては，内

[1] 東京理科大学大学院理学研究科科学教育専攻
[2] 東京理科大学
[a)] 1719527@ed.tus.ac.jp
[b)] kats @ ma.kagu.sut.ac.jp

容 (3) に「コンピュータとプログラミング」が作られたため，すべての高校生がプログラミングを学ぶことである．内容としても，「アルゴリズムを表現する手段，プログラミングによってコンピュータや情報通信ネットワークを活用する方法」[1] に関する知識及び技能を身に付け，「目的に応じたアルゴリズムを考え適切な方法で表現し，プログラミングによりコンピュータや情報通信ネットワークを活用するとともに，その過程を評価し改善する」[1] など現行の学習指導要領には記載されていなかった内容も追加された．この改訂の背景には現行学習指導要領の成果と課題 [1] より，「情報の科学的な理解に関する指導が必ずしも十分ではない」[1] ことや「情報やコンピュータに興味・関心を有する生徒の学習意欲に必ずしも応えられていない」[1] といった課題があった．さらに,「生徒の卒業後の進路等を問わず, 情報の科学的な理解に裏打ちされた情報活用能力を育むことがいっそう重要となってきている」ことが改訂の理由としてあげられている．これらのことより,「情報 I」においては，すべての高校生に対して成果と課題で挙げたことを解決するようなプログラミングの授業が求められているとわかる．さらに，小学校では 2020 年度よりプログラミング教育が必修となることで，高等学校情報科に対して学習内容としてもより高度なものが求められることが予想される．また，「中学校技術・家庭科技術分野の内容「D 情報の技術」の学習を踏まえたプログラミングを扱う」とあり，小中高のプログラミング教育の接続も求めらている．そのため今後は今までよりプログラミング経験がある生徒が高等学校に入学してくる可能性が十分に考えられる．しかし一方で，現在のカリキュラムにおけるプログラミングにあたる部分は「情報の科学」であるが，情報教育に関する資料 [3] によると「社会と情報」が 80%,「情報の科学」が 20%程度であることから現状として高等学校におけるプログラミング教育は浸透していないことが分かる．また，高等学校共通教科情報の変遷と課題 [2] によると，教科「情報」担当教員の約 3 割が免許外，他教科との兼任は約 5 割となっていることからも，プログラミング経験が十分でない教員が担当していることが予想されるなど，多くの課題を抱えていることが分かる．そこで筆者は，プログラミング学習環境として位置付けられ，web ブラウザのみで動作可能な Google Colaboratory を使用し，初学者向けの授業に適したプログラミング環境を用いて，Python の授業実践を行った．今回はその授業実践と生徒の意識についての報告をする．Python は，近年プログラミングの入門言語として注目されている．その大きい要因としてまず，ライブラリが豊富であり，最近注目されているデータサイエンスなどで使用されていることが挙げられる．また，オフサイドルールによりコードが読みやすく書きやすいものになっている．情報 I の学習内容の構成 (3) において「プログラミング，モデル化とシミュレーション」があり，情報Ⅱの学習内容の構成 (3) においても「情報とデータサイエンス」があり，さらに「情報システム，ビッグデータ」などを扱うことが学習指導要領に明記されている．また，文部科学省が公開している高等学校情報科「情報 I」教員研修用教材 [4] に掲載されているサンプルプログラムには Python が使われているため，学校現場で指導されることが予想される．しかし，一般的な高校におけるコンピュータ教室において，Python の処理系や関連するライブラリを教師自身がインストールすることは困難であると思われるが，Google アカウントがあればインストールをせずに Python 環境を整えることができる Google Colaboratory(以下,「Colaboratory」と記す) を使用することで，環境構築の経験がない教員も授業で扱うことができる．本研究では授業前後に実施した生徒へのアンケート結果と共に授業実践に関する効果を報告する．

2. Colaboratory の概要

Colaboratory は，完全にクラウドで実行されるインタラクティブなノートブック環境であり，それは Jupyter ノートブックと呼ばれている [5].

2.1 Colaboratory のメリット

メリットとしては以下のものが挙げられる．

- 設定不要 (環境構築等) である
- Google アカウントがあれば， 無料で使用できる
- チーム内での共有が簡単である

これらのメリットを説明すると，Python の環境構築が不要であり，Numpy など機械学習で必要なほぼ全ての環境がすでに構築されており，さらに GPU(Tesla K80 GPU) も無料で使えること，次に必要なものは Google アカウントで，ブラウザのみですぐに機械学習を始めることが可能であるため情報Ⅱのデータサイエンスなどの題材においても活用できること，Colaboratory で書いたコードは，Google Drive で保存され，グループ内でのノートブックの共有などが非常に簡単で，かつ権限管理など Google Drive 上で行えることである．

3. 実践報告

3.1 対象学年

2019 年年度
高校 3 年生 (週 2 コマ (1 コマ : 50 分))，全 6 コマ
文系： 3 クラス (各 37，34，36 名)
理系： 2 クラス (各 25，27 名)

2020 年度
高校 3 年生 (週 2 コマ (1 コマ : 35 分))，全 6 コマ
文系： 3 クラス (各 29，30，25 名)
理系： 2 クラス (各 25，23 名)

文理のクラス数による変化はないが，今年度はコロナ対応として授業時間が 1 コマ 35 分となった．

3.2 生徒状況

2019，2020 年度におけるプログラミング経験は，ビジュアル，テキスト言語共に 1 割程度であった．(詳しくは 4.3.1 にて後述)

3.3 授業形態

2019 年度：一斉授業型
2020 年度：動画視聴型

昨年度の実践より，「生徒によって学習進度が大きく異なる」「プログラミングの指導においては教員の人員が必要である」がわかった．よって今年度はあらかじめ授業内容を動画にし，生徒が授業内で各々のペースで進める動画配信型授業の形態とした．

3.4 実践環境

2019，2020 年度共に以下の通りである．

- デスクトップ型のパソコンを 1 人 1 台使用できる．
- 入学時に学校から個人の Google アカウントが与えられている．

G Suite for Education が導入されており，情報の授業では Google classroom にて授業のプリントや課題の配布等の管理を行っている．他の科目においても活用されている．

3.5 授業内容

表 1 実践授業内容

授業数	内容
1	Python とは，四則演算，比較演算子の学習
2	変数，リストの学習
3	分岐 (if,elif,else) の学習
4	反復 (for,range) の学習
5	最終課題
6	予備

主な授業の流れは，2019，2020 年度共に上の表 1 の通りである．最終課題を数学の整数問題に設定し，その問題を解決するためにプログラミングの基本的な考え方である逐次・分岐・反復処理のツールを 4 回の授業に分けて学んでもらい， 5 回目の授業で最終課題に取り組むという形で行った．授業内で扱う問題は両年度同様のものを使用した．動画の長さは授業時間が 35 分であることを踏まえ，1 つ 10 分以内の 5 本とした．

3.5.1 1 回目 (目安)

実社会における Python の活用例の紹介，プログラミング言語における Python の特徴等の説明を一斉授業形式で行った．その後は，生徒各自で動画を視聴しながら課題に取り組む．

以下，四則演算の練習問題

練習問題

問題 1：10−2

問題 2：10 × 2

問題 3：10 ÷ 2

問題 4：11 ÷ 2 の余りの表示

問題 5：2^{10}

授業中や授業後に提出された生徒のコードを確認すると，問題 3 と 4 において使用する演算子の違いがわからない生徒が見られた．また，問題 5 に関しては「*」を 2 つではなく，2 の後ろに「*」を 10 個つけている生徒もおり，さらに，キーボードで「*」,「%」の位置がわからない生徒も一部いた．

以下，比較演算子の練習問題

練習問題

問題 1：2==2

問題 2：2!=2

問題 3：2>2

問題 4：2>=2

問題 5：i =100

i%2==0

問題 6：i =101

i%2==1

比較演算子に関してはコードを写す形にし，返ってきた答えに対してどうしてそのような結果が返ってくるのかを考えさせる記述問題も追加した．問題 5，6 において，はじめはなぜ“true”や“false”が返ってくるのかわからない生徒もいたが机間指導の中で理解していた．

四則演算と比較演算子を実行するにあたり共通することとしては“SyntaxError: invalid character in identifier”である，数字や「+」,「=」などの記号が半角ではなく，全角入力されていることで起こるエラーが多数見られた．初めに「1+1」を実行するところとそれを全角で入力した際には同様の“SyntaxError”エラーが起こるということを説明していたが，助けを求める生徒もいた．

3.5.2 2 回目 (目安)

以下，変数の練習問題

練習問題

問題 1：name1=”自分の姓”

問題 2：name2=”自分の名”

問題 3：type(name1)

問題 4：type(total)

問題 4 に関しては，変数の導入の際に total には，数字を入力させる練習を行った．変数において type まで意識させた点は，次にリストの学習をさせる際に「string 型」と「integer 型」は，「integer 型」を「string 型」に直して実行しないとエラーが起こってしまうためであるのでここで学習させることとした．ここでのエラーとしては，一度「type=name1」を実行してから間違いに気づき，「type(name1)」を実行し，“’str’ object is not callable”とエラーがでてしまい，どうしてエラーが起こったのかわからない生徒もいた．

以下，リストの練習問題

練習問題

- 名前：○○○○
- 生年月日：○月○日
- 性別：男性
- 趣味：○○○○○

解答コード例

空のリスト，リストの先頭は 0 から始まること，リストの中から自分が取り出したいものを表示させる方法，自分の好きな情報 (部活や趣味) をリストに追加し表示させるという練習問題を各自で行った．ここで生年月日を表示させる際に変数で行った“string”と“integer”の内容が使われるようになっており，エラーになった際には動画を見直し，解決する生徒もいた．さらに初めは気が付かずに実行したが，エラーを見て解決した生徒も多くみられた．理由としては，エラーが“must be str, not int”と比較的わかりやすいエラーメッセージであったことが挙げられる．

3.5.3 3 回目 (目安)

以下，if の練習問題

練習問題

80 点以上・・・「優」

60 点以上〜80 点未満・・・「良」

60 点未満・・・「不可」

を表示するようなプログラムの作成

解答コード例

教員研修用教材 [4] の分岐のコードを参考にし，問題を作成した．例えば，練習問題としては“elif”を使い，3 つ以上の分岐するような練習問題とした．さらに“input”も教えることでコードの実行後，いろいろな値に変えた値を入力し実行結果より自分のコードが正しく書けているかを確認させた．ここでのエラーとしては，インデントによるエラーが多かった．Python には可読性を高めるためにインデントがあるが，今回はそれによりコードはあっているにもかかわらずエラーが返ってきてしまい，困惑してしまう生徒がいた．

3.5.4　4 回目 (目安)

以下，for 文の練習問題

練習問題

問題 1：0,1,2,3,4,5,6,7,8,9,10 を表示
問題 2：5,6,7,8,9,10 を表示
問題 3：1,3,5,7,9 を表示
問題 4：問題 3 と別の書き方
問題 5：1 から 20 までの中で，3 の倍数を表示
問題 6：問題 5 に加えて，3 の倍数の中でさらに 2 の倍数の時は，数字を出力させた後，”6 の倍数！”と表示

穴埋めコード

```
for i in range(?):
  print(i)
```

図 1　for 文　穴埋めコード

for 文の基本的な問題から，最終問題を解く際に使用する“range”の練習問題とした．「range」を使い最後の数字を表示する際には一つ先の数字をコードに書く必要があるが，リストの理解が不十分な生徒はなぜ一つ先の数字を入れる必要があるのかわからない生徒が多数見られた．しかし，穴埋め式のコードを用いて，トライアンドエラーを繰り返しながら行うことで多くの生徒が全ての課題に取り組むことができていた．

3.5.5　5 回目

以下，最終問題

最終問題

3 以上 9999 以下の奇数 i で，$i^2 - i$ が 10000 で割り切れるものを全て求めよ．

解答コード例

```
[1]  for i in range(3, 10000):
       if i % 2 == 1:
         if (i**2-i) % 10000 == 0:
           print(i)

     625
```

```
[2]  for i in range(3, 10000, 2):
         if (i**2-i) % 10000 == 0:
           print(i)

     625
```

図 2　最終課題　解答

最終課題は東京大学の入試問題 [6] を取り扱ったが，初めは入試問題であることは伏せて行った．授業の最後に入試問題であることを伝えると特に文系クラスにおいて反応がよかった．コードの解答例としては，if 文を 2 回使うものと「range」を用いる 2 パターンであった．多くの生徒が前者である if 文を 2 回使うコードであった．理由としては，前の授業で行なった「問題 6」が最後にあったためであると考えられる．さらに，机間指導していく中で最終課題は幾つの条件から出来ているかと声かけをしたことも理由として挙げられる．また，「(i**2-i)」の「()」がない生徒も多く見られた．さらに，「()」がない場合においてもエラーが返ってこないため，間違いに気がつく生徒は少数であった．また，この問題は数学の問題として解くと難解で解答が長いが，Python で行うと短いコードであれば 4 行で書くことができることから，

プログラミングの有用性や面白さを実感している生徒が多く見られた．

昨年度と比較し，全体的に教員に対する質問が少なかった．それは生徒が繰り返し動画を見直しながら課題に取り組んでいたためである．

4. 授業アンケートから見る生徒の実態

授業アンケートはプログラミングの授業の開始する 1 番初めと前半授業 5 回終了後に行った．

4.1 事前アンケートの項目

Q1:プログラミングに興味があるか

Q2:ビジュアルプログラミングをしたことがあるか (スクラッチ, ビスケット, グーグルブロッキーなど)

Q2’:Q2 にてビジュアルプログラミング経験したことがあると回答した人に対して，順次構造・分岐構造・反復構造の経験があるか

Q3:テキストプログラミングをしたことがあるか (C++、Java、python など)

Q3’:テキストプログラミング経験したことがあると回答した人に対して，順次構造・分岐構造・反復構造の経験があるか

4.2 事後アンケートの項目

Q4:授業を受けてプログラミングに興味を持ったか

Q5:プログラミングは楽しかったか

Q6:四則演算・リスト・分岐構造・反復構造・分岐と反復の組み合わせ問題それぞれに対する難易度

Q7:プログラミングを行い、達成感を得られたか

Q7’:Q7 において達成感が得られたと回答した人に対して，具体的には何があるか

Q8:プログラミングの授業が終わってもプログラミングをしてみたいか

Q9:プログラミングをしていて大変だな、または難しいなと思うところはあったか (2 択)，大変, 難しいと感じた人は具体的には何があるか

Q9’:Q9 において大変, 難しいと回答した人に対して，具体的には何があるか

Q10:情報の授業以外で自分でプログラミングの勉強をしたか (情報の授業を行ってから)

Q11:今後プログラミングでしたいことはあるか (あれば具体的に)

4.3 集計結果

全てのアンケートは欠損値を除いた．2019 年度 : n=145[11]，2020 年度 : n=105 で分析を行なった．

4.3.1 対象生徒のプログラミング学習経験 (Q2,2’,3,3’)

	ビジュアルプログラミングをしたことがあるか(スクラッチ,ビスケット,グーグルブロッキーなど)		テキストプログラミングをしたことがあるか(C++、Java、pythonなど)	
	ある	ない	ある	ない
2019年度	13.1%(19)	86.9%(126)	10.3%(15)	89.7%(130)
2020年度	12.4%(13)	87.6%(92)	17.1%(18)	82.9%(87)

それぞれの質問において「ある」と答えた生徒に対して，「順次処理」，「分岐構造」，「反復構造」の経験があるかと聞いたところ 2019 年度はほとんどの生徒がわからないと解答し，2020 年度は経験があると多くの回答が得られた．その理由としては，プログラミング部に所属している生徒が部活内で経験したことが挙げられる．

4.3.2 授業前後におけるプログラミングへの興味 (Q1,4)

Q1 \ Q4	とてもある	ある	どちらかと言えばある	どちらかと言えばない	ない	全くない
とてもある	15.2%(16)	7.6%(8)	0.0%	1.0%(1)	0.0%	0.0%
ある	3.8%(4)	20%(21)	7.6%(8)	1.0%(1)	0.0%	0.0%
どちらかと言えばある	0.0%	8.6%(9)	8.6%(9)	1.9%(2)	1.0%(1)	0.0%
どちらかと言えばない	0.0%	1.0%(1)	2.9%(3)	4.8%(5)	1.9%(2)	0.0%
ない	0.0%	0.0%	0.0%	0.0%	1.9%(2)	0.0%
全くない	0.0%	0.0%	2.9%(3)	1.9%(2)	1.9%(2)	4.8%(5)

2019 年度は授業の前後におけるプログラミングに対する興味の変化を見ると，「興味がある」は 31 ポイント上昇し，「興味がない」は 9 ポイント減少した．さらに，「興味がある」と「どちらかと言えば興味がある」を合わせると，50.3%から 80.7%と 33.4 ポイント上昇した．一方，2020 年度は表のように授業前に「全くない」の層がプラスの変化をした部分もあるが大きな変化はなかった．この結果としては 2019 年度においては授業前におけるプログラミングの興味が 5 割であったのに対し，

2020 年度においては 7 割であったことが挙げられる．この結果から年々高校生におけるプログラミングへの興味が高まっていることがわかる．

4.3.3 プログラミングをしていて大変だな、または難しいなと思うところはあったか (Q9,9')

	あった	特になかった	合計
2019年度	81.4%(118)	18.6%(27)	100.0%(145)
2020年度	73.3%(77)	26.7%(28)	100.0%(105)

具体的なこととしては，数式などの記号を覚えること，エラーの修正 (半角)，エラーの内容がそもそもわからない，タイピング，インデント，コードの理解，問題の解答としてはおかしいがコードによるエラーはなかったためエラーが返ってこなかった ((a**2-a)%10000 の () が必要であったこと) など様々挙げられた．さらに，今年度は動画配信型であったため，動画を見ながらでは画面が小さい，動画の中で開設されていない問題は全く手がつかないなどがあった．

4.3.4 プログラミングは楽しかったか (Q5)

	とても楽しかった	楽しかった	どちらかと言えば楽しかった	どちらかと言えば楽しくなかった	楽しくなかった	全く楽しくなかった	合計
2019年度	24.8%(36)	29.0%(42)	26.2%(38)	4.8%(7)	5.5%(8)	9.7%(14)	100.0%(145)
2020年度	21.9%(23)	36.2%(38)	29.5%(31)	7.6%(8)	0.0%(0)	4.8%(5)	100.0%(105)

今年度の方が全体的に楽しかった割合が大きいが，特に「楽しくなかった」「全く楽しくなかった」の層が大きく減少した．これは今年度，動画型の個人学習にしたことにより昨年度では一斉型の授業でついてこれなかった層が自分のペースで学習できたことが要因であると考えられる．

4.3.5 プログラミングの授業が終わってもプログラミングをしてみたいか (Q8)

	とてもしたい	したい	どちらかと言えばしたい	どちらかと言えばしたくない	したくない	全くしたくない	合計
2020年度	17.1%(18)	31.4%(33)	26.7%(28)	13.3%(14)	5.7%(6)	5.7%(6)	100.0%(105)

2019 年度は 5 件法で取ったため完全に比較することができないが，全体として「したい」のグループに 5 ポイント上昇した．さらに，「とてもしたい」,「したい」が 48.5%，「どちらかと言えばしたい」を含めると 7 割以上から肯定的な回答が得られたことから初学者におけるプログラミング例としてはよかったと考える．

4.3.6 プログラミングを行い、達成感を得られたか (Q7,7')

	得られた	得られなかった	合計
2019年度	75.9%(110)	24.1%(35)	100.0%(145)
2020年度	82.9%(87)	17.1%(18)	100.0%(105)

「練習問題，最終課題が解けた時」,「コードエラーを直せた時」,「トライアンドエラーを繰り返している時」,「エラーなく正しいコードを打てた時」,「コードの意味がわかった時」などが多数挙げられた．問題が解けた際に「できた！」など喜んでいる姿を多数見ることができたことからも多くの生徒が達成感を得ることができたと考える．

4.3.7 各項目の難易度に関する反応 (Q6)

下記の内容の難易度	[四則演算](n=145)	[リスト](n=145)	[if文](n=145)	[for文](n=145)	[if文とfor文の組み合わせ](n=145)
簡単	33.1%(48)	24.8%(36)	17.2%(25)	13.8%(20)	7.6%(11)
やや簡単	17.9%(26)	17.9%(26)	12.4%(18)	11.0%(16)	6.2%(9)
普通	22.8%(33)	28.3%(41)	29.0%(42)	25.5%(37)	15.2%(22)
やや難しい	5.5%(8)	7.6%(11)	15.9%(23)	20.7%(30)	22.8%(33)
難しい	20.7%(30)	21.4%(31)	25.5%(37)	29.0%(42)	48.3%(70)

図 3 各項目の難易度 (2019 年度)

	四則演算 (n=105)	リスト(n=105)	if文(n=105)	for文(n=105)	if文とfor文の組み合わせ (n=105)
簡単	25.7%(27)	13.3%(14)	10.5%(11)	6.7%(7)	5.7%(6)
やや簡単	18.1%(19)	27.6%(29)	10.5%(11)	9.5%(10)	6.7%(7)
普通	32.4%(34)	33.3%(35)	32.4%(34)	32.4%(34)	21.9%(23)
やや難しい	15.2%(16)	15.2%(16)	33.3%(35)	38.1%(40)	29.5%(31)
難しい	8.6%(9)	10.5%(11)	10.5%(11)	13.3%(14)	36.2%(38)

図 4 各項目の難易度 (2020 年度)

両年度において問題の難易度は授業回数の増加に応じて，難しいと思う割合も増加し，一方で簡単だと感じる割合が減少している．さらに，どの「難しい」項目においても 2019 年度に比べ 2020 年度が 10 ポイント以上低くなっている．このことからも動画による学習方式による効果であると言え

る．一方，if 文と for 文の組み合わせにおいて難しいと感じている生徒の割合が多く，授業中においても悩んでいる生徒が多くいた．答えが複数あることが逆に生徒は悩んでしまう要因であると考えることができる．

4.3.8 情報の授業以外で自分でプログラミングの勉強をしたか (情報の授業を行ってから)(Q10)

授業以外でプログラミングをした一部であり，「GAS を使ってみた」,「配布した資料を見た」,「インターネットにてプログラミングに関することを調べた」など具体的な回答の生徒は事前アンケートにて過去にプログラミング経験があると答えていた．高 3 ということもあり，授業で疑問に残ったことや気になったことを調べたり，配布資料をもとに復習することはほとんどなかったと考える．

4.3.9 今後プログラミングでしたいことはあるか (Q11)

web ページの作成，ゲームの作成，アプリケーションの作成，データの集計，ハッキング，レゴを使ったプログラミングなどが挙げられた．また，何かしたいが何ができるかわからないという回答も多数見られた．

4.4 まとめ

web ブラウザのみで動作可能な Google Colaboratory を使用し，初学者向けの導入授業を行なった．教師の立場から考えると，Python を行うための環境設定は一切なかったことから負担が少なかった．また，G Suite が導入されており，Google classroom にて課題や見本コードの配布，生徒の作成物の管理が大変楽であった．生徒においては，「難しかったが，楽しかった」や「達成感があった」など肯定的な声が多数あり，導入の授業としては適していることがわかった．さらに，一斉型と動画視聴型の授業形式を 2 カ年に渡って行った．生徒アンケートからも後者の方が生徒個人が各自のペースで学習を進めることができ，プログラミングの授業においては大変効果的であったことがわかった．

参考文献

[1] 文部科学省．(2018). 新学習指導要領解説 ．http://www.mext.go.jp/component/a_menu/education/micro_detail/__icsFiles/afieldfile/2019/03/28/1407073_11_1_1.pdf(2019 年 11 月 1 日確認)

[2] 中野由章．高等学校共通教科情報科の変遷と課題. 2018．https://www.ipsj.or.jp/magazine/9faeag0000005al5-att/5910peta.pdf(2019 年 11 月 1 日確認)

[3] 文部科学省．情報教育に関する資料．2015. http://www.mext.go.jp/b_menu/shingi/chukyo/chukyo3/059/siryo/__icsFiles/afieldfile/2015/11/11/1363276_08_1.pdf(2019 年 11 月 1 日確認)

[4] 高等学校情報科「情報 I」教員研修用教材. 2013.http://www.mext.go.jp/component/a_menu/education/micro_detail/__icsFiles/afieldfile/2019/10/09/1416758_005.pdf

[5] **Colaboratory** へようこそ．2013.https://colab.research.google.com/notebooks/welcome.ipynb?hl=ja(2019 年 11 月 1 日確認)

[6] 東京大学 (理系) 前期入試問題.2005.https://www.densu.jp/tokyo/05tokyospass.pdf(2019 年 11 月 1 日確認)

[7] 本多佑希．オンラインプログラミング環境「**Bit Arrow**」の **Python** 対応. キミのミライ発見．https://www.wakuwaku-catch.net/jirei19133/(2019 年 11 月 27 日確認)

[8] 阿部百合．問題解決への利用を目的とした統計の学習 (**Python** を利用して)．2018. http://www.johobukai.net/20181227/181227_03_01.pdf(2019 年 11 月 27 日確認)

[9] 遠藤優一．**2020** 年度必修化に向けた高等学校プログラミング授業の実践報告．デジタル教科書学会．pp. 43-44. 2019https://www.wakuwaku-catch.net/%E6%8E%88%E6%A5%AD%E4%BA%8B%E4%BE%8B-%E3%83%90%E3%83%83%E3%82%AF%E3%83%8A%E3%83%B3%E3%83%90%E3%83%BC/(2019 年 11 月 27 日確認)

[10] オンラインプログラミング学習環境 **Bit Arrow**．https://bitarrow.eplang.jp/(2019 年 11 月 27 日確認)

[11] 山本周．オンラインプログラミング環境 **Google Colaboratory** における **Python** の導入実践と生徒の実態．2019.