授業改善セミナー　教科「情報」　実践発表

北海道札幌北高等学校　前田　健太朗

1 今までのプログラミングの指導経験から

　プログラミングの授業をしたあとの感想や学習の振り返りのコメントから、何かをつくると達成感を味わうことができ、楽しかったと生徒は感じるようです。また、プログラムの例題を打ち込むよりも、自分である程度プログラムを考えるような問題が楽しいようです。一方で、ちょっとしたプログラムなのにこんなに入力するのかという驚きや、カッコやダブルクォーテーションなどプログラムの記法が面倒だと感じる生徒もかなりいます。
　そこで、簡単に取り組めて、生徒が主体的に取り組む場面があり、エディターは入力の補助やエラーメッセージを表示するようなものを利用するとよいと考えています。

2 すぐにプログラミングの授業ができる環境

　学校のリースPCにソフトをインストールすることをためらう方が多いのではないでしょうか。また、今年度の教育課程編成・実施の手引きでは「学びの重点化」がキーワードになっています。学校だけでなく家庭等でも学べる環境が重要になります。学校のPCだけでなく自宅のPC等からも学べるプログラミングの環境がますます重要になると思います。

3 初心者向けプログラミングの授業の実践例

　初めてプログラミングを学ぶ生徒向けで、学校だけでなく家庭等でも取り組めるプログラミングの実行環境がある教材を紹介します。

3-1 アルゴロジック

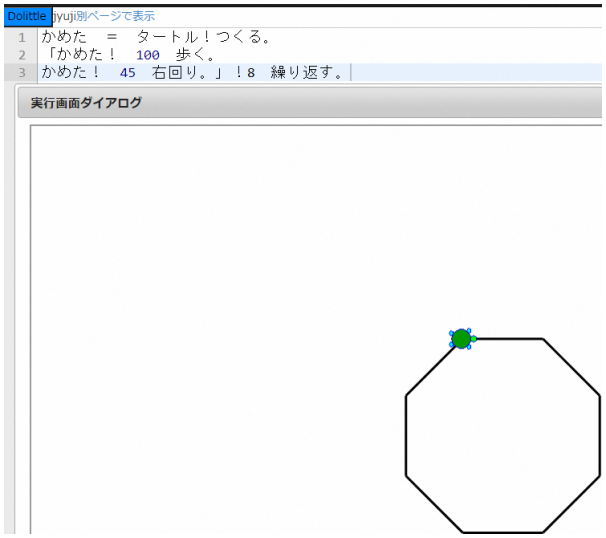
　アルゴロジックは、一般社団法人　電子情報技術産業協会（JEITA)が開発したソフトです。ネット環境とPCにFlash Playerがインストールされていれば無料で使用できます。しかし、Flash Playerは2020年末にアドビ社のサポートが終了する予定です。そこでJEITAは今年の7月末にHTML5で動くアルゴロジック（https://home.jeita.or.jp/is/algo/）を公開しました。
　アルゴロジック1のジュニア問題には23問、チャレンジ問題には49問が用意されており、小学生から大人まで楽しめる問題が用意されています。アルゴロジックは、小学生はもちろん札幌北高校の生徒も楽しく取り組める教材です。タイトルからわかりますがアルゴリズムを考えるための教材ですが、テーマがロボット操作のため、プログラミングで何をするかという目的が限定的なので、初めてプログラミングを学ぶ人向けで、プログラミングの導入のための教材と言えるでしょう。

3-2 ドリトル

　ドリトルとは大阪電気通信大学の兼宗進教授が開発した教育用に設計されたプログラミング言語です。その最大の特徴は、日本語でコードを記述できることです。英語がわからない児童、英語に苦手意識のある生徒でも取り組みやすいプログラミング言語です。
　ドリトルはBitArrow（https://bitarrow.eplang.jp）を利用すればネット上でプログラミングができるので、PCへのソフトのインストールは不要です。また、ゲストアカウントでも利用できるので、事前に登録作業等も不要です。
　かめを表示し、それを動かして線を引かせれば、アルゴロジックで取り組んだアルゴリズムをプロ

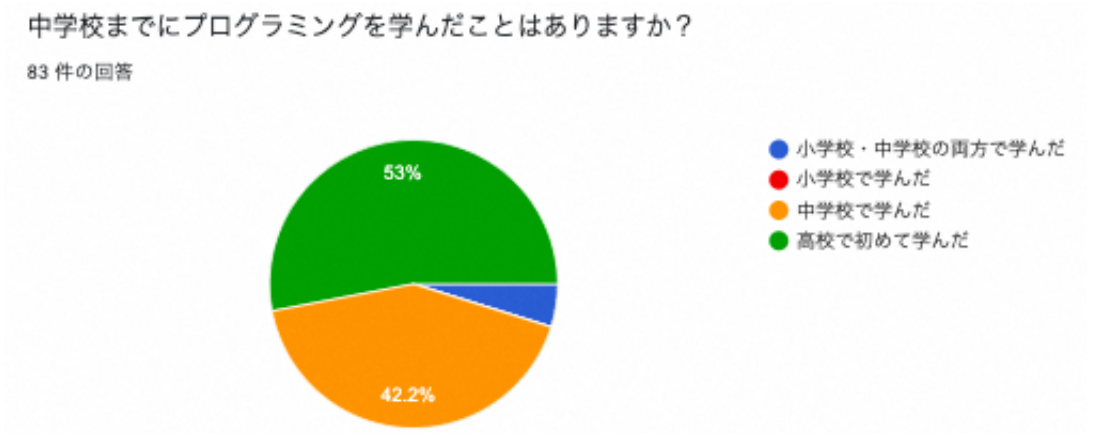
グラミングで再現できます。アルゴロジックで解いた問題を記録シートにメモさせ、それをもとにドリトルでプログラムをつくれば、生徒は自分の進度に合わせて問題に取り組むことができます。プログラミングは生徒の進度に差が出やすく、教員が一人だと対応しづらいことが多いですが、アルゴロジックとドリトルを利用すればこの問題を解決できます。

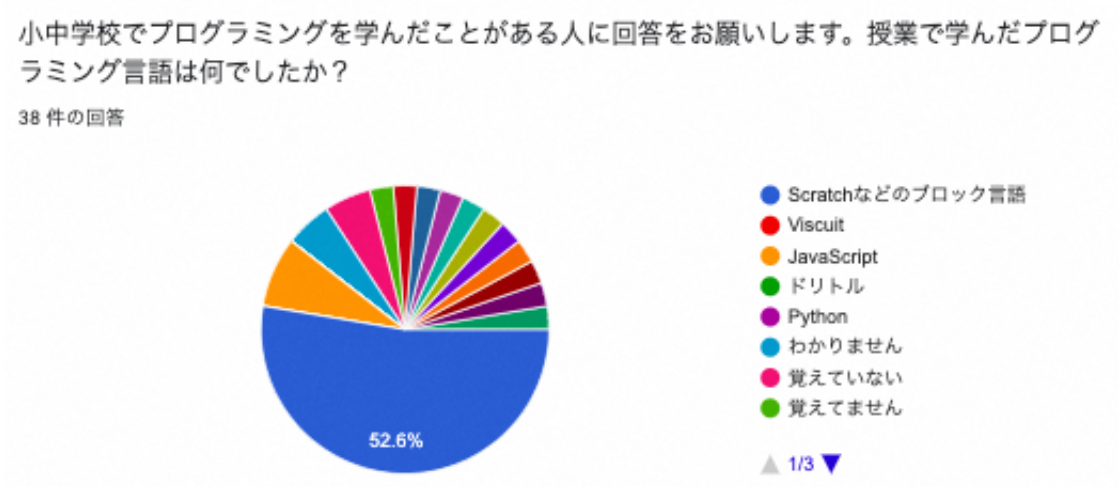
図1　ドリトルでアルゴロジックチャレンジ問題入門7問目を再現したプログラムと実行画面



　文部科学省の「情報I教員研修用資料」で扱われたプログラミング言語は、Pythonだけでなく JavaScript、ドリトル、Swiftもありました。高校においても生徒の実態等に合わせてドリトルを選択するということが考えられると思います。ちなみに、小中学校ではブロック型言語を体験しているようです。高校で初めてプログラミングを学んだと回答する生徒が多いことは気になります。（学んだけれどもそれがプログラミングだったとわかっていない生徒もいるように思えます。）

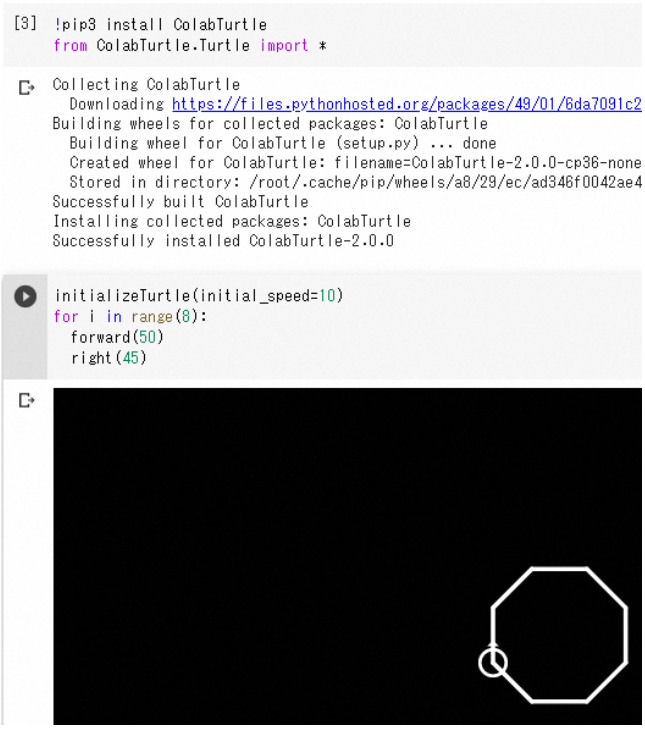
図2,3 札幌北高生へのプログラミングの学習の経験に関するアンケート結果から





3-3 Colaboratory & Python（ColaboTurtle）

Colaboratoryとは、Googleが機械学習の教育、研究を目的として開発したツールです。これを利用すれば無料でJupyter Notebook環境を利用することができます。ネット環境さえあれば、ソフトをインストールすることなく利用できます。ただし、Googleのアカウントが必要になります。Pythonはもちろん、このColaboratoryのPythonもTurtleを利用することができます。ドリトルではなくPythonをアルゴロジックと組み合わせてプログラミングの授業を展開することもできます。
図4　Colaboratoryでアルゴロジックチャレンジ問題入門7問目の問題のプログラムと実行例



3-4 micro:bit & JavaScript

アルゴロジックとタートルグラフィックスを利用したプログラミングは、ロボットやかめを動かす命令が中心となるので、特殊な命令が多いと思います。そこで、マイコンボードを動かして実行結果を表示させるmicro:bitを利用したプログラミングもしました。micro:bitは実物がなくてもWeb上で動かすことができます。LEDを光らせる、ボタンによって動作する、変数を利用して条件分岐させるといったプログラムが数行でつくれます。

micro:bitでは、最初に数字やアイコン、文字をLEDで光らせて表示するプログラムをつくりました。

```
basic.showNumber(2020,100)
```

```
basic.showIcon(IconNames.Heart)
```

```
basic.showString("Hello!",100)
```

次に、変数を利用して値を計算させるプログラムをつくりました。

```
let atai = 0
basic.forever(function () {
    atai = atai + 1
    basic.showNumber(atai)
})
```

さらに、micro:bitに用意されているボタンを押したときにカウントするプログラムをつくりました。

```
let atai = 0
input.onButtonPressed(Button.A, function () {
  atai = atai + 1
    basic.showNumber(atai)
})
input.onButtonPressed(Button.B, function () {
  atai = atai - 1
    basic.showNumber(atai)
})
```

他にも、乱数を利用するプログラムをつくりました。

```
let atai
atai = randint(1,6)
basic.showNumber(atai)
```

それから、条件分岐するプログラムもつくり、最終的にはおみくじのプログラムをつくりました。

```
let atai
atai = input.temperature()
if(atai > 25){
    basic.showString("hot")
}else{
    basic.showString("just right")
}
```

```
let atai = randint(0,2)
if(atai == 0){
    basic.showString("Lucky")
}else if(atai == 1){
    basic.showString("Usual")
}else{
    basic.showString("Bad")
}
```

micro:bitでできることは限られていますが、それが初心者でも取り組みやすいようです。また、実行結果がLEDの点灯でわかることから、達成感を感じられるようです。一方で、生徒が自由にプログラムをつくることには向いておらず、できることが制限されていることに不満を感じる生徒もいるようでした。

3-5 Python

プログラミングをどのくらいまで指導するかというのはなかなか難しい問題です。本校でのプログラミングの単元の配当時間は5時間程度なので、指導できる内容に限りがあります。文部科学省が発行した高等学校情報科「情報I」教員研修用教材では、分岐や反復、配列、乱数、WebAPIの利用、探索、ソートのプログラム例が記載されており、ある程度の配当時間が必要だと思います。そこで、探索やソートは分岐や反復、配列を利用したプログラムといえるので、探索やソートを教材とするよりも分岐、反復、配列を扱う教材を考えるのがよいと個人的には考えています。

昨年度まで、基本的なプログラムとして分岐、反復、乱数を利用して数当てゲームをつくっています。はじめに変数や代入、表示、次に乱数の生成を説明します。さらに、キーボードからの入力と条件分岐について説明し、下のプログラムを作成します。

```
from random import randint
pc = randint(0,10)
hum = input('数値を当ててね')
hum = int(hum)
if pc == hum:
    print('当たり',pc)
else:
    print('はずれ',pc)
```

このプログラムを応用して、じゃんけんゲームをつくります。

```
from random import randint
pc = randint(0,2)
hum = input('グーなら０・チョキなら１・パーなら２、どれにする？')
hum = int(hum)
print('あなた',hum)
print('comは',pc)
result = hum - pc
if result == 0:
    print('ひきわけ')
elif result == 1:
    print('負け')
elif result == 2:
    print('勝ち')
elif result == -1:
    print('勝ち')
elif result == -2:
    print('負け')
```

さらに、勝ち負けの判定に利用している変数resultの計算を工夫すれば、勝ち負けの判定がスッキリします。

```
from random import randint
pc = randint(0,2)
hum = input('グーなら０・チョキなら１・パーなら２、どれにする？')
hum = int(hum)
print('あなた',hum)
print('comは',pc)
result = (hum - pc + 3) % 3
if result == 0:
    print('ひきわけ')
elif result == 1:
    print('負け')
elif result == 2:
    print('勝ち')
```

　このプログラムを生徒に考えさせると、生徒によって進捗状況に差が出てきます。そこで、事前に反復を説明しておき、ジャンケンゲームを連戦にしたり、勝率を求めたりするよう指示しておくと、進んでいる生徒が手持ち無沙汰にならないでしょう。
　次に配列を利用したプログラムをつくります。配列を用意し、ランダムでその中のどこかに「あたり」を入れ、それを探し当てるというゲームです。

```
from random import randint
masu1 = [0,0,0,0,0]
x = randint(0,4)
masu1[x] = 1
print(masu1[0],masu1[1],masu1[2],masu1[3],masu1[4])
yosox = int(input("どこかな？"))
if masu1[yosox] == 1:
    print('あたり')
else:
    print('はずれ')
```

　このままだと、プログラムの実行時に「あたり」が入っている場所が分かってしまうので、表示用として配列を追加したものが下のプログラムです。

```
from random import randint
masu1 = [0,0,0,0,0]
masu2 = ["*","*","*","*","*"]
x = randint(0,4)
masu1[x] = 1
print(masu2[0],masu2[1],masu2[2],masu2[3],masu2[4])
yosox = int(input("xどこかな？"))
if masu1[yosox] == 1:
    print('あたり')
else:
    print('はずれ')
    masu2[yosox] = 0
```

これで配列を扱うことができますが、最近の大学入試センター試験「情報関係基礎」の問題を見ると、二次元配列が出題されています。そこで、このゲームも二次元配列を利用した改造版をつくります。

```
from random import randint
masu1 = [[0,0,0,0,0],[0,0,0,0,0],[0,0,0,0,0],[0,0,0,0,0],[0,0,0,0,0]]
masu2 = [["*","*","*","*","*"],["*","*","*","*","*"],["*","*","*","*","*"],
["*","*","*","*","*"],["*","*","*","*","*"]]
x = randint(0,4)
y = randint(0,4)
masu1[y][x] = 1
for i in range(0,5):
    print(masu2[i][0],masu2[i][1],masu2[i][2],masu2[i][3],masu2[i][4])
flag = 0
while flag < 1:
    yosox = int(input("xどこかな？"))
    yosoy = int(input('yどこかな？'))
    if masu1[yosoy][yosox] == 1:
        print('atari')
        masu2[yosoy][yosox] = 1
        for i in range(0,5):
            print(masu2[i][0],masu2[i][1],masu2[i][2],masu2[i][3],masu2[i][4])
        flag = 1
    else:
        print('hazure')
        masu2[yosoy][yosox] = 0
        for i in range(0,5):
            print(masu2[i][0],masu2[i][1],masu2[i][2],masu2[i][3],masu2[i][4])
```

このプログラムを実行すると、25の配列の中から1つを選択することになるので、当たる確率がかなり低くなります。そこで、当たるまで繰り返してゲームができるようにしてあります。それでもなかなか当たらないので、どこに「あたり」があるかヒントを出すようにします。

```
from random import randint
masu1 = [[0,0,0,0,0],[0,0,0,0,0],[0,0,0,0,0],[0,0,0,0,0],[0,0,0,0,0]]
masu2 = [["*","*","*","*","*"],["*","*","*","*","*"],["*","*","*","*","*"],
["*","*","*","*","*"],["*","*","*","*","*"]]
x = randint(0,4)
y = randint(0,4)
masu1[y][x] = 1
for i in range(0,5):
    print(masu2[i][0],masu2[i][1],masu2[i][2],masu2[i][3],masu2[i][4])
flag = 0
while flag < 1:
    yosox = int(input("xどこかな？"))
    yosoy = int(input('yどこかな？'))
    if masu1[yosoy][yosox] == 1:
        print('atari')
```

```
        masu2[yosoy][yosox] = 1
        for i in range(0,5):
            print(masu2[i][0],masu2[i][1],masu2[i][2],masu2[i][3],masu2[i][4])
        flag = 1
    else:
        print('hazure')
        masu2[yosoy][yosox] = 0
        for i in range(0,5):
            print(masu2[i][0],masu2[i][1],masu2[i][2],masu2[i][3],masu2[i][4])
        if yosox < x :
            print('もっと右だよ')
        elif yosox > x:
            print('もっと左だよ')
        else:
            print('その行だよ')
        if yosoy < y :
            print('もっと下だよ')
        elif yosoy > y:
            print("もっと上だよ")
        else :
            print('その列だよ')
```

4 おわりに

　プログラミングは授業の展開や教材を一工夫すると楽しい授業となるでしょう。（次の図は本校生徒にJavaScriptとmicro:bitの授業についてアンケートしたものと、Webページ作成の後にJavaScriptの学習をしたときの振り返りをテキストマイニング（共起ネットワーク図）したものです。これらを見ると、生徒の多くがプログラミングの授業を楽しんだと思えます。）アルゴリズムを考えてプログラムを入力するだけでも生徒の主体性を引き出すことができると思います。さらに生徒自身がプログラムを改造してもっとこんなことはできないかという疑問や学ぶ意欲、深い学びへとつなげていくことができるでしょう。そこに、学校だけでなく自宅等でも学習を続けられる環境があれば、意欲の高い生徒は実際に取り組んでくれると思います。ネットを利用したプログラミングの実行環境を利用するメリットは大きいと思います。

図5,6 micro:bitを用いた授業は楽しいかという質問への回答結果

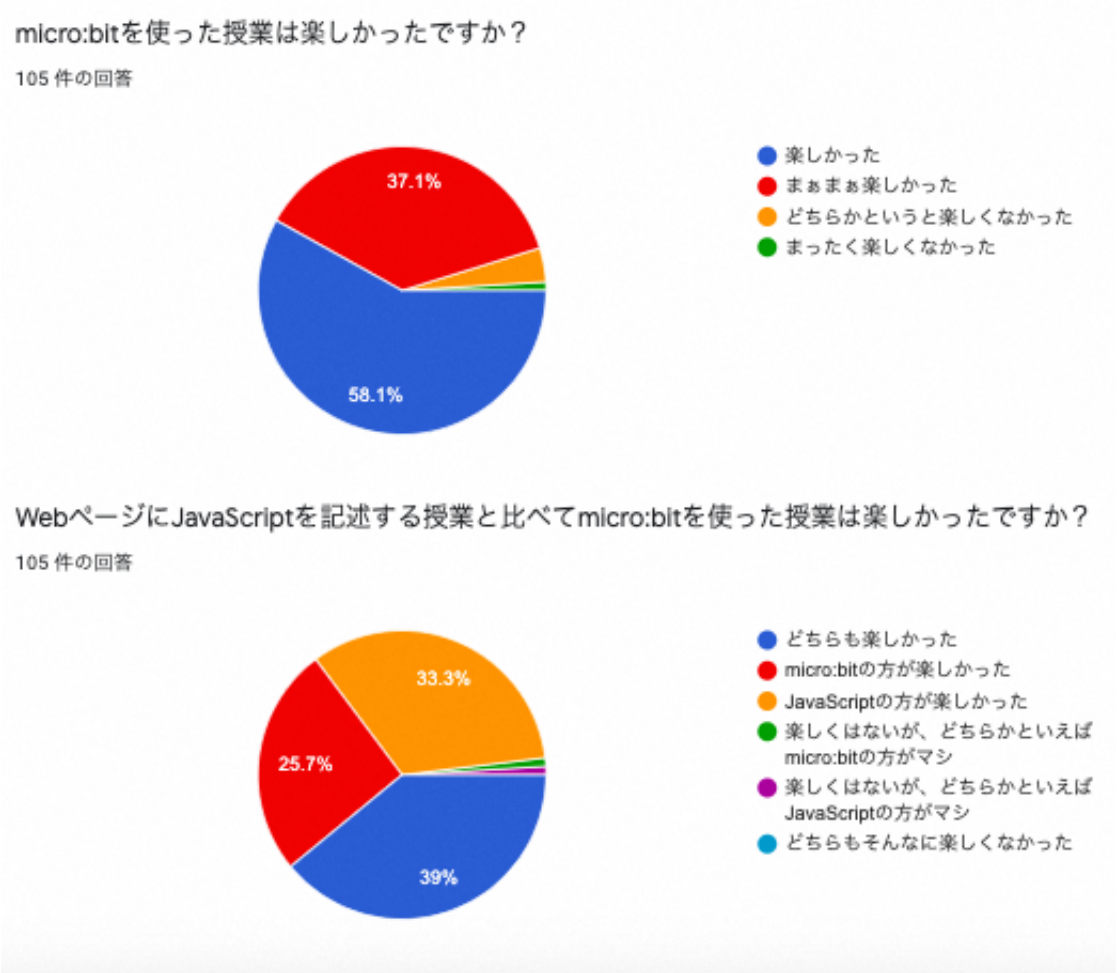
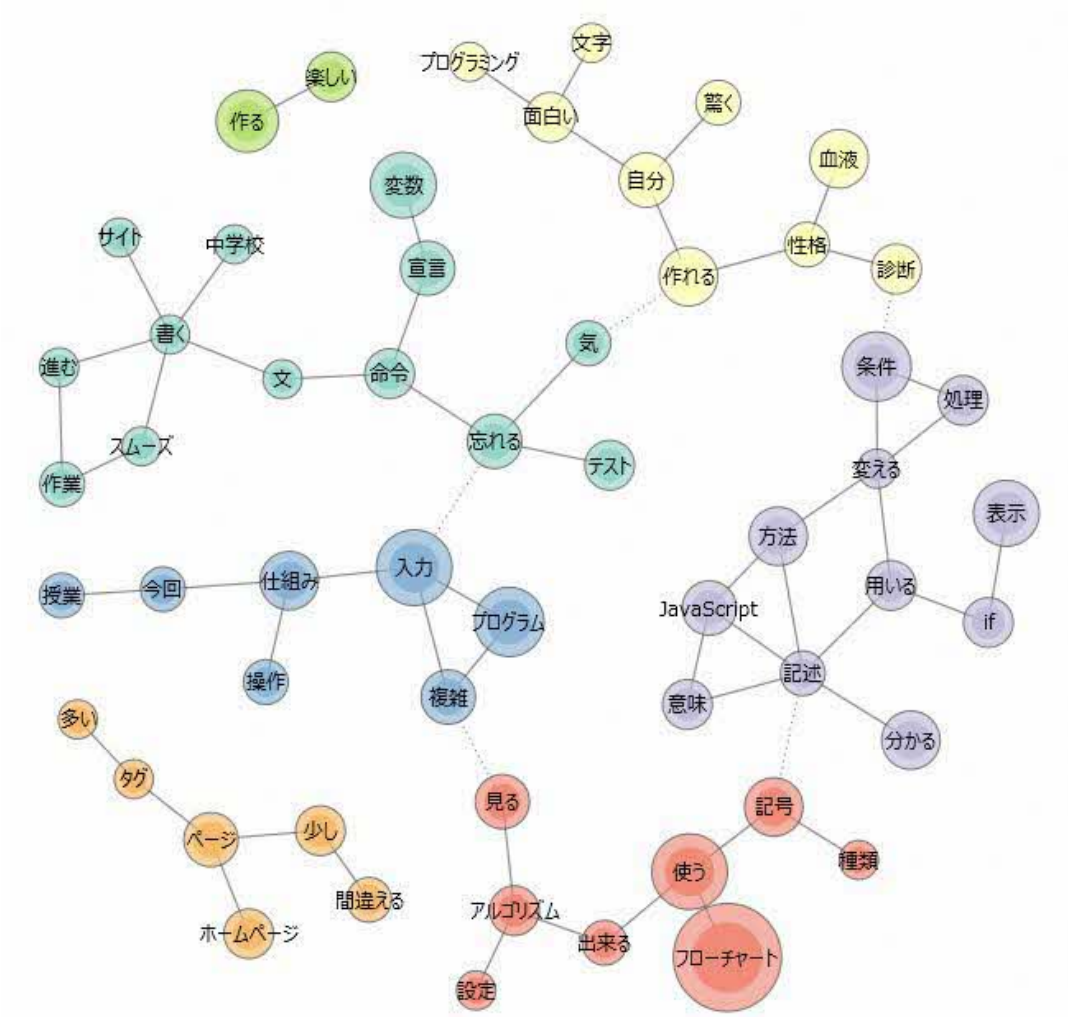


図7 JavaScriptの授業の振り返りのコメントをテキストマイニングした結果（共起ネットワーク図）


楽しい
作る
プログラミング
文字
面白い
驚く
自分
血液
変数
性格
サイト
中学校
宣言
作れる
診断
書く
気
条件
進む
文
命令
処理
スムーズ
忘れる
テスト
変える
作業
表示
方法
入力
授業
今回
仕組み
用いる
JavaScript
if
プログラム
操作
記述
複雑
意味
多い
分かる
見る
記号
タグ
使う
ページ
少し
種類
間違える
アルゴリズム
出来る
フローチャート
ホームページ
設定