

コンピュータとプログラミング に関する授業実践報告

埼玉県高等学校情報教育研究会

埼玉県立浦和第一女子高等学校 富田 平

目次

- **はじめに（研究会の活動紹介）**
- **今年度のテーマ決定の経緯**
- **授業案の紹介**

目次

- **はじめに（研究会の活動紹介）**
- **今年度のテーマ決定の経緯**
- **授業案の紹介**

研究会の活動紹介

- **研究委員** **情報科教員 10名程度**
- **活動内容** **毎年共通のテーマで、授業実践
等を考える**

これまでの歩み（16年間）

年	研究内容
2008 2009	情報モラルに関するアンケート調査
2010	情報機器を利用する上でのコミュニケーションの工夫
2011	コミュニケーション能力の向上を目的とした授業の工夫
2012	生徒の思考力・問題解決能力の育成を図る授業実践
2013	新教育課程に向けた年間指導計画の作成と分析
2014	LINE等ソーシャルメディアに関する指導について
2015	「社会と情報」の中で論理的思考力を養う授業案

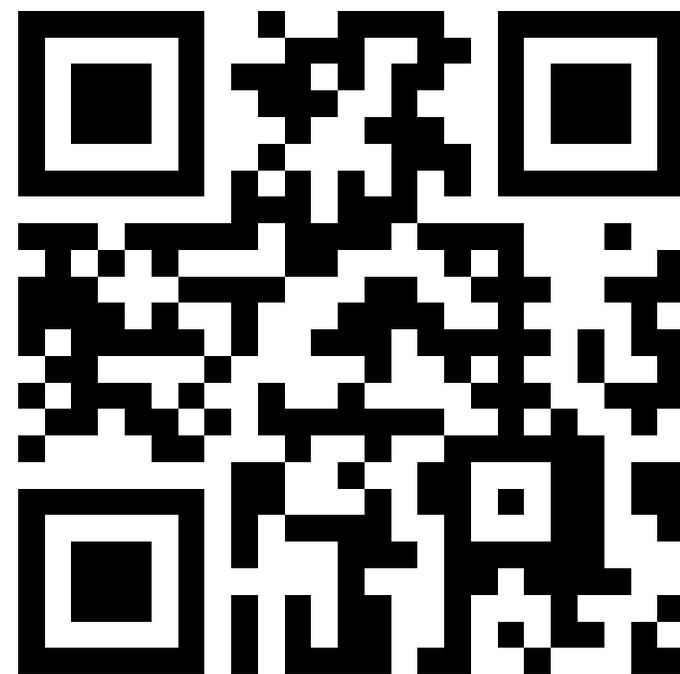
これまでの歩み（16年間）

年	研究内容
2016 2017	プログラミングに関するアンケート調査
2018	反転学習を意識した動画教材の作成
2019	プログラミング教育を見据えた動画作成の検討
2020	新学習指導要領「情報Ⅰ」（2）コミュニケーションと情報デザインに関する指導案の作成
2021	オンライン学習に向けた埼玉県立高校の取り組み状況
2022	共通テスト対策用の予想問題の作成
2023	コンピュータとプログラミングに関する授業実践報告

研究会の活動紹介

- 埼玉県高等学校情報教育研究会のWebサイトで過去の活動を公開しております。

(<https://www.saikojoken.net/>)

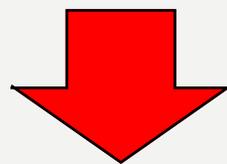


目次

- はじめに（研究会の活動紹介）
- **今年度のテーマ決定の経緯**
- 授業案の紹介

今年度のテーマ決定の経緯

2022年度 新学習指導要領
「情報Ⅰ」がスタート！



授業実践事例を共有したい

新学習指導要領「情報Ⅰ」

(1) 情報社会の問題解決

(2) コミュニケーションと情報デザイン

(3) コンピュータとプログラミング

(4) 情報通信ネットワークとデータの活用

新学習指導要領「情報Ⅰ」

プログラミングに関する授業実践事例
(もしくは授業案) を報告する

目次

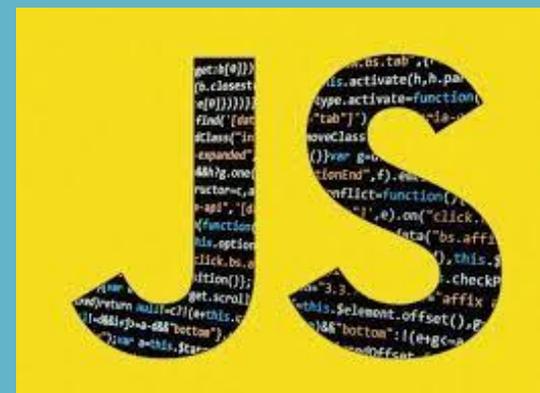
- はじめに（研究会の活動紹介）
- 今年度のテーマ決定の経緯
- 各校の授業案の紹介

7つの学校での言語の選択

- JavaScript 2件：事例①②
- Python3 4件：事例③④⑤⑥
- VBA 1件：事例⑦

実践事例①

JavaScriptを用いた
プログラミング体験

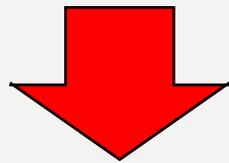


はじめに

- **実施年度** : **2021年度**
- **学年** : **3年**
- **科目** : **社会と情報（旧課程）で実施**
- **使用言語** : **JavaScript**

実践への思い

1時間の授業でプログラミングの面白さに触れさせたい！



既存のソースコードを改変しながらプログラミングを体験！

授業実践

```
*omikuji.html - メモ帳
ファイル(E) 編集(E) 書式(Q) 表示(V) ヘルプ(H)
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>おみくじ</title>
<script language="JavaScript">
<!-- Copyright (C) WEST MiRa http://www.west-mira.jp -->

function omikuji () {
    jikoku = new Date();
    var byou = jikoku.getSeconds();
    byou %= 10;
    var bun = new Array();
    bun[0] = "大吉！<br>絶好調です。何もかも貴方の思い通り！";
    bun[1] = "中吉！<br>何か良いことがありそう。";
    bun[2] = "吉。<br>穏やかな日々が過ごせそうです。";
    bun[3] = "小吉。<br>今日はゆっくりと過ごしましょう。";
    bun[4] = "凶…<br>忘れ物に注意";
    bun[5] = "大凶…<br>ミスを連発しそうです。";
    bun[6] = "中吉。<br>懐かしい人に出会いそう。";
    bun[7] = "小吉。<br>残り物には福がある。";
    bun[8] = "凶…<br>こんな日は大人しくしていよう。";
    bun[9] = "吉。<br>懐かしい人に連絡を取ってみよう。";

    document.open();
    document.write('<html><body><div align=center><font size="5">' + bun[byou]);
    document.write('<br><br>');
    document.write('<a href="omikuji.html">戻る</a></font></div>');
    document.write('</body></html>');
    document.close();
}
</script>
</head>

<form>
<input type="button" value="御神籤" onClick="omikuji();">
</form>
```

13行、2列 100% Windows (CRLF) UTF-8

授業実践

```
*omikuji.html - メモ帳
ファイル(E) 編集(E) 書式(O) 表示(V) ヘルプ(H)
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>おみくじ</title>
<script language="JavaScript">
<!-- Copyright (C) WEST MiRa http://www.west-mira.jp -->
```

エラーが起きにくい！

アルゴリズムの構造の
理解に集中できる！

与えられたコードを改変！
コピー & ペーストで編集！

```
document.write("</body></html>"),
document.close();
}
</script>
</head>
<form>
<input type="button" value="御神籤" onClick="omikuji();" />
</form>
```

13行、2列

100%

Windows (CRLF)

UTF-8

実践事例②

p5.jsを利用した
プログラミング入門

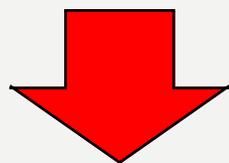
p5.js

はじめに

- **実施年度** : **2021年度**
- **学年** : **1年**
- **科目** : **社会と情報（旧課程）で実施**
- **使用言語** : JavaScript
- **実行環境** : p5.js (<https://p5js.org/>)

実践への思い

視覚的にプログラミングに
触れさせたい！

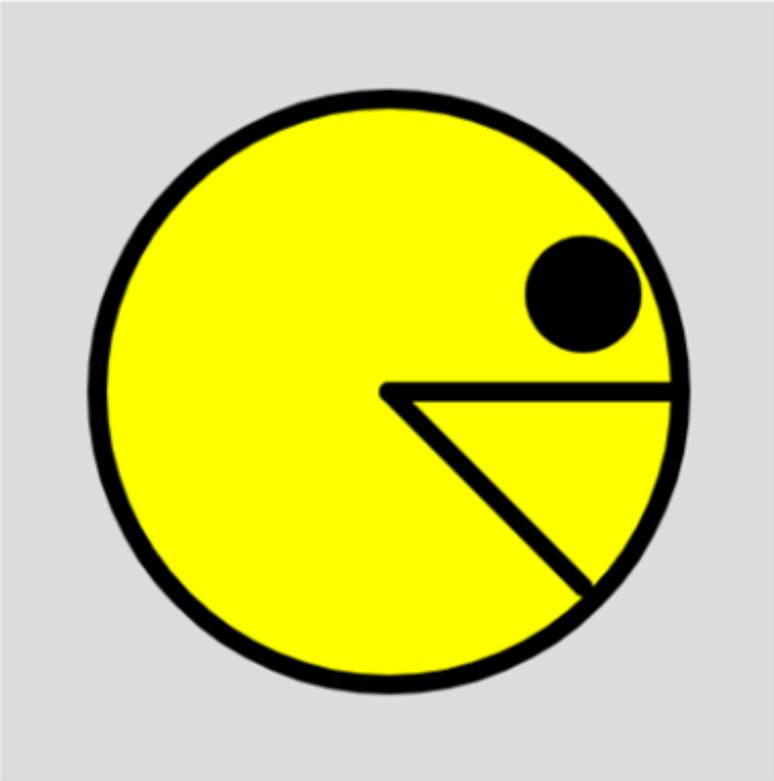


p5.jsを利用した
プログラミングで描画！

授業実践

指示したとおりに動いて
「楽しい」との声

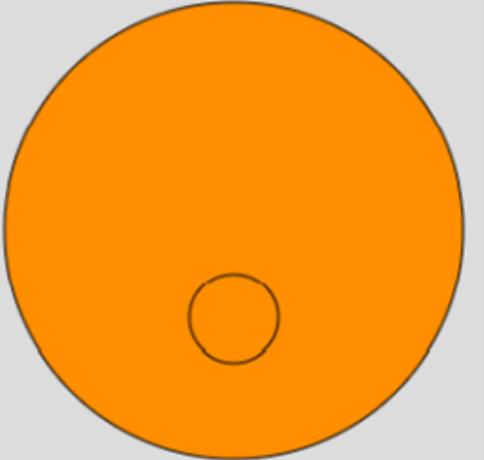
```
sketch.js Preview
1 function setup() {
2   createCanvas(400, 400);
3 }
4
5 function draw() {
6   background(220);
7
8   // rect(100,200,200,200);
9   // ellipse(60,90,50,100);
10  // line(30,30,100,30);
11  fill(255,255,0);
12  circle(200,200,300);
13
14  fill(0,0,0);
15  circle(300,150,50);
16  strokeWeight(10);
17  line(200,200,300,300);
18  line(200,200,350,200);
19 }
```



授業実践

- 拡大縮小
- 色を変えていく
- 位置を変えていく
- 分岐や反復による図形の描画を体験

```
> sketch.js Preview
1 let yanase;
2 yanase=100;
3
4 let niiza;
5 niiza=100;
6
7 function setup() {
8   createCanvas(400, 400);
9 }
10
11
12 function draw() {
13   background(220);
14
15   fill(yanase, 400-yanase, 0);
16   circle(yanase, 200, yanase);
17   circle(yanase, 250, 50);
18   circle(yanase, 300.50);
19
20   yanase=yanase+1;
21   niiza=niiza+2;
22
23 }
```



授業実践

- 拡大縮小
- 色を変えていく
- 位置を変えていく
- 分岐や反復による図形の描画を体験

結果を視覚的に反映
→理解度の向上

```
21   niiza=niiza+2;  
22  
23 }
```

授業の流れ(6時間)

1時間目	プログラミングとは？ どこで使われている？開発環境の確認
2～3時間目	p5.jsで命令がコンピュータに 伝わっていることを体験
4時間目	「順次」「分岐」「反復」の説明。 「変数」の概念を理解
5～6時間目	p5.js において変数を用いた図形描画を体験させる。 分岐や反復による図形の描画の仕方を体験させる。

授業で使用した教材(例)

プログラミングで図形を描いてみる

今回使うのは「p5.js」

インターネット上で
プログラムを書ける



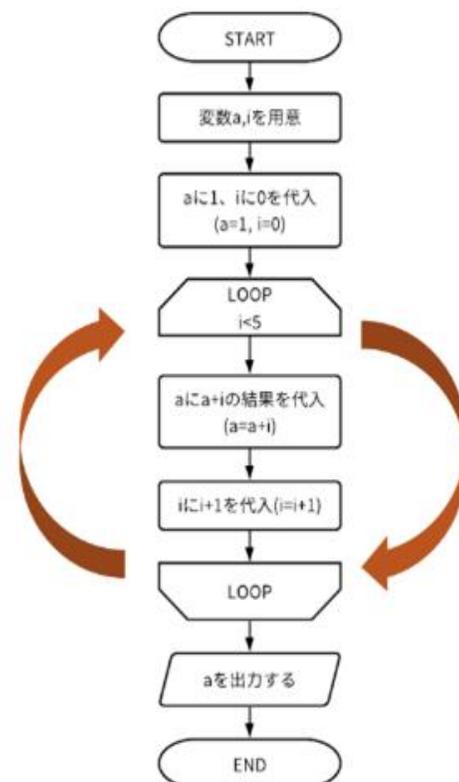
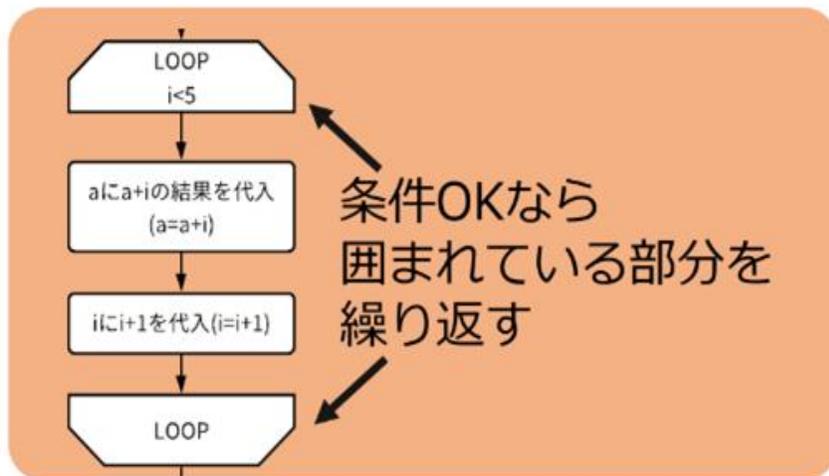
p5.jsの説明

授業で使用した教材(例)

フローチャートで解説

プログラムの3大要素 反復

条件が成り立つ間、処理を繰り返す



実践事例③

paizaラーニングを使用した授業実践

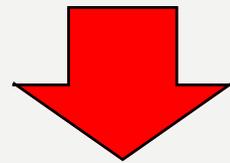


はじめに

- **実施年度** : **2022年度**
- **学年** : **1年**
- **科目** : **情報Ⅰで実施**
- **使用言語** : Python3
- **実行環境** : paizaラーニング (<https://paiza.jp/works>)

実践への思い

生徒の個別対応や評価方法、
入試対策など多くの課題



外部の学習サービスを活用
することで解決！

The screenshot shows the paiza.io Python editor interface. The browser address bar displays the URL `paiza.jp/works/python3/primer/beginner-python2/4010`. The page title is "Python入門編2: 条件分岐、比...". The user's progress is shown as "Python3入門編 (全11レッスン)の進捗度 100%" and "学習レベル70".

Four callout boxes highlight specific features:

- 動画で学習** (Learn with video): Points to the video player area showing a code editor with Python code and a terminal output.
- プログラム打ち込み** (Program input): Points to the code editor area where the user can enter or paste code.
- ヒント等** (Hints, etc.): Points to the "Tips" section at the bottom of the page, which includes a section titled "# 01: 数値が一致した場合、メッセージを表示".
- 実行結果** (Execution results): Points to the terminal output area showing the execution of the code and the resulting error message: "Runtime error(Exit status: 1)".

```
1 # coding: utf-8
2 # if文による条件分岐
3 number = 2
4 if number == 1:
5     print("スキップ")
6 else:
7     print("クリア")
```

出力 実行時エラー 入力 コメント 0 (0.03 sec)

```
File "Main.py", line 4
if number == 1
^
Runtime error(Exit status: 1)
```

Tips

次のチャプターへ 演習課題へ進む

有料会員になるとエンジニアに質問ができます 入力

実行結果

コード判定詳細

テスト番号	ジャッジ結果	実行時間
1	Success	0.03 秒
2	Success	0.03 秒
3	Success	0.03 秒
4	Success	0.03 秒
5	Success	0.03 秒

テストケース通過率: 100 %



奇跡の騎士
演習課題をクリア!

👉 次の演習課題へ

🐦 ツイートする

採点は自動なので、業務
軽減が期待できる！

学習ステータス



 **Lv.70** 奇跡の騎士

 学習チケット所持数 6 / 6枚

 レッスン完了数 11 / 11レッスン

 演習課題完了数 144 / 144問

キャラが育っていくので
モチベーションアップ！

授業の基本的な流れ



時間	内容	留意点
5～10分	動画の内容を解説しながら入力	生徒がより興味関心を示すように、プログラムの内容を一部変更しながら説明する。
5～15分	演習課題	課題の進行に差が生じるため、早く終わった生徒は、サポートに回ったり授業未実施箇所を進めたりする。

大学入学共通テストでの出題が考えられる内容 ※

- 変数
- 順次構造
- 反復構造
- 選択構造
- 入れ子構造 (ネスト)
- 配列
- 組み込み関数 (print関数など)
- ユーザー定義関数

抜粋して
実施



※ 令和7年度試験の問題作成の方向性, 試作問題等より

授業の流れ（12時間）

授業	内容	講座	チャプタ
1時間目	アカウント登録・アルゴリズムとプログラミング・フローチャート・基本構造		教科書を使用
2時間目	順次構造・表示(print)・エラー・コメントアウト・数値	体	02初めてのプログラミング 03間違いやすいポイント 04コメントを書く 05数値を扱う
3時間目	計算・演算子・変数・標準入力	体	06プログラムで計算する 07変数にデータを入れる 08データを受け取る
4時間目	選択構造(if,else)	体	10条件に一致したら処理を実行する 11条件に合わせて処理を変える 12数値を分類する
5時間目	モジュール・random・選択構造(if,elif,else)	入1	06サイコロを作ろう
		入2	04おみくじを作ってみよう
6時間目	反復構造(for,while)・forとifの組み合わせ	入3	01数値を繰り返し表示してみよう 02条件に合わせてくり返してみよう1
		体	15複数データを分類する

7時間目	リスト	入4	01リストとは何かを学ぼう 02リストを作ろう 03リストの要素を取り出してみよう 04リストを操作しよう
8時間目	リストと反復構造・split・2次元リスト	入4	05ループでリストを処理しよう 06カンマ区切りデータを、splitで分割しよう
		入6	01 2次元リストとは何か学ぼう 02 2次元リストを作成する
9時間目	2次元リストと反復構造	入6	03 2次元リストを操作する1 04 2次元リストを操作する2 教科書：2次元配列と反復
10時間目	関数	入7	01関数について学習しよう 02関数を作ろう 03引数と戻り値を追加しよう 04スコープを理解しよう
11時間目	探索のプログラム		教科書を使用
12時間目	整列のプログラム		教科書を使用

- ※ paiza ラーニングの講座名は次の通りに省略する。
python 体験編：体 python3 入門編：入
- ※ python3 入門編の講座には、レッスンの階層が存在するため、次の通りに表す。
Python3 入門編の講座, レッスン3 の場合の例：入3



無料で学習効果の高い
サービスはたくさんある

プログラミング学習サービスを 活用したメリット

環境設定
なし

自動採点

共通テスト
対策

能力差
あっても◎

動画教材
自宅学習も◎

実践事例④

Google Colab を使用した
プログラミング演習

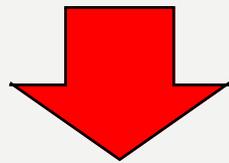


はじめに

- **実施年度** : **2022年度**
- **学年** : **1年**
- **科目** : **情報Ⅰで実施**
- **使用言語** : Python3
- **実行環境** : Google Colaboratory

実践への思い

理解度に応じて授業展開したい



自作教材の解説と演習問題を
生徒主導で取り組ませる！

授業実践

The image shows a screenshot of a Python programming lesson slide. The slide content includes a table of contents, an introduction to conditional branching, and a code example. Two large orange arrows are overlaid on the right side: one pointing up labeled '解説' (Explanation) and one pointing down labeled 'サンプルプログラム' (Sample Program). A speech bubble on the left contains the text 'プログラムを動かしながら理解!' (Understand while running the program!).

+ コード + テキスト

- 第3回 Pythonプログラミング講座 【条件分岐】
- 分岐構造 (簡単な分岐)
- if文を理解しよう

コンピュータの世界では、特定の条件を満たすときだけ処理を実行することがあります。これを条件分岐といいます。Pythonでは、条件分岐をif文で以下のように記述します。

```
if 条件:  
    条件が成り立ったときの処理
```

例) 点数xが60点以上ならば、「合格」を表示するプログラム

```
[1] # 実行してみよう  
  
x = 80          # 変数xに数値「80」を代入する  
  
if x >= 60:    # もし、条件「xが60以上」を満たしていれば以下のブロックを実行する  
    print("合格") # 文字列「合格」を表示する
```

合格

解説

サンプルプログラム

プログラムを動かしながら理解!

授業実践

コード + テキスト

行末の「: (コロン)」を入力してから改行すると、次の行に自動的にスペースができます。このスペースをインデントといいます。このインデントした範囲は条件が成り立った時の処理を行います。この範囲をブロックといいます。

また、行頭のスペースを削除してインデントのない状態にすると、ブロックから出たことになります。その部分はif文とは関係なくなるので、常に実行されることになります。

【プログラム】

```
x = 80
if x >= 60:
    print("合格")
    print("すごい")
print("お疲れさま")
```

■ …インデント

コロンを忘れずに!

■ print("合格") ブロック内

■ print("すごい") ブロック外

print("お疲れさま")

【フローチャート】

```
graph TD
    A[x = 80] --> B{x >= 60}
    B -- True --> C[print("合格")]
    C --> D[print("すごい")]
    B -- False --> E[print("お疲れさま")]
    D --> F[ ]
    E --> F
    F --> G[ ]
```

分岐が多くなると処理の流れが把握しづらくなるため、上記のように「フローチャート (流れ図)」で視覚的に表現する方法があります。また、条件の中では2つの値を比較するための記号「比較演算子」が使われます。

比較演算	意味
==	左辺と右辺は等しい
!=	左辺と右辺は等しくない
<	左辺は右辺より小さい
<=	左辺は右辺以下である
>	左辺は右辺より大きい
>=	左辺は右辺以上である

▼ ◆ [演習] 次の手順に沿ってプログラムを作成してみよう

～手順～

- ① 変数ageに年齢を代入する。(年齢は自由に設定する)
- ② もし、条件「ageが18より小さい」を満たしていれば、文字列「未成年です」を表示する。

```
[ ] # 解答用セル (この下に解答してください)
```

簡単な演習問題に
挑戦する

解説

演習

授業の基本的な流れ

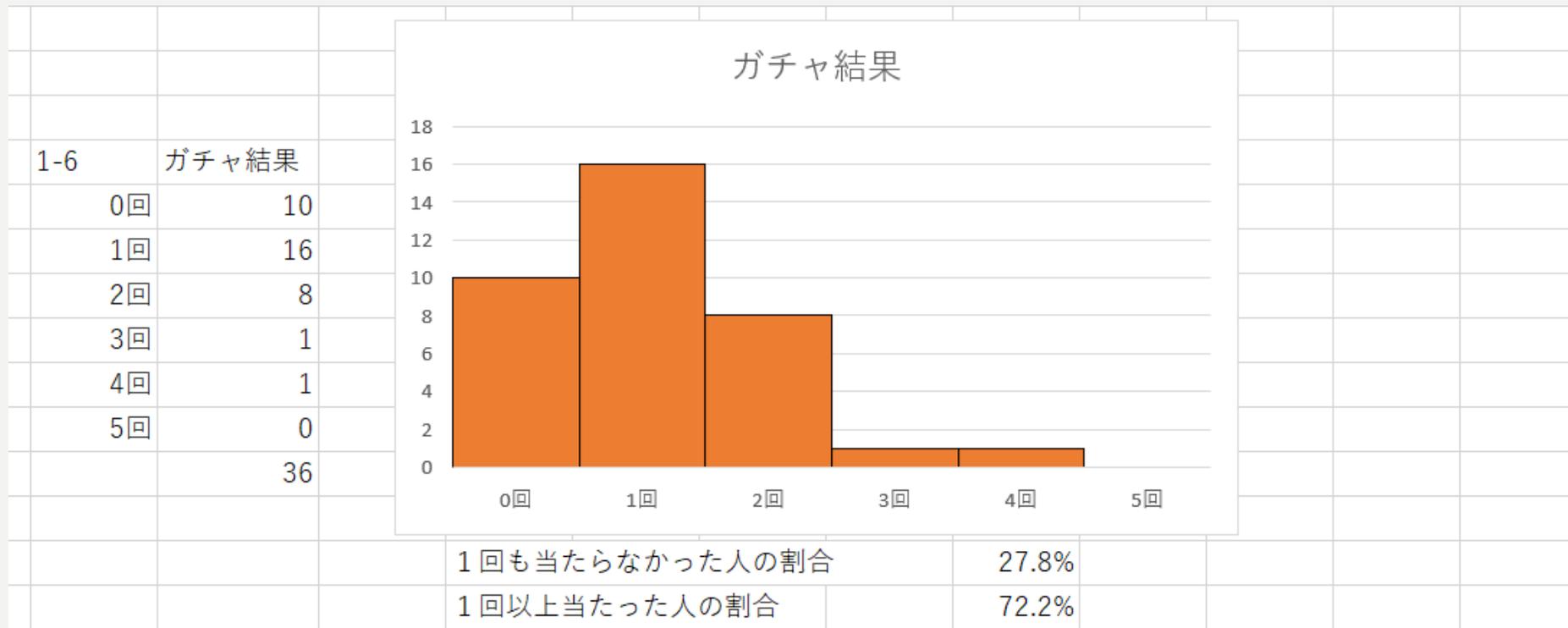
10～15分	理解のポイントを伝える 要点スライド1枚提示
35～40分	演習課題

教員、早く終わった生徒は
終わっていない生徒のサポートにまわれる！

授業の流れ（8時間）

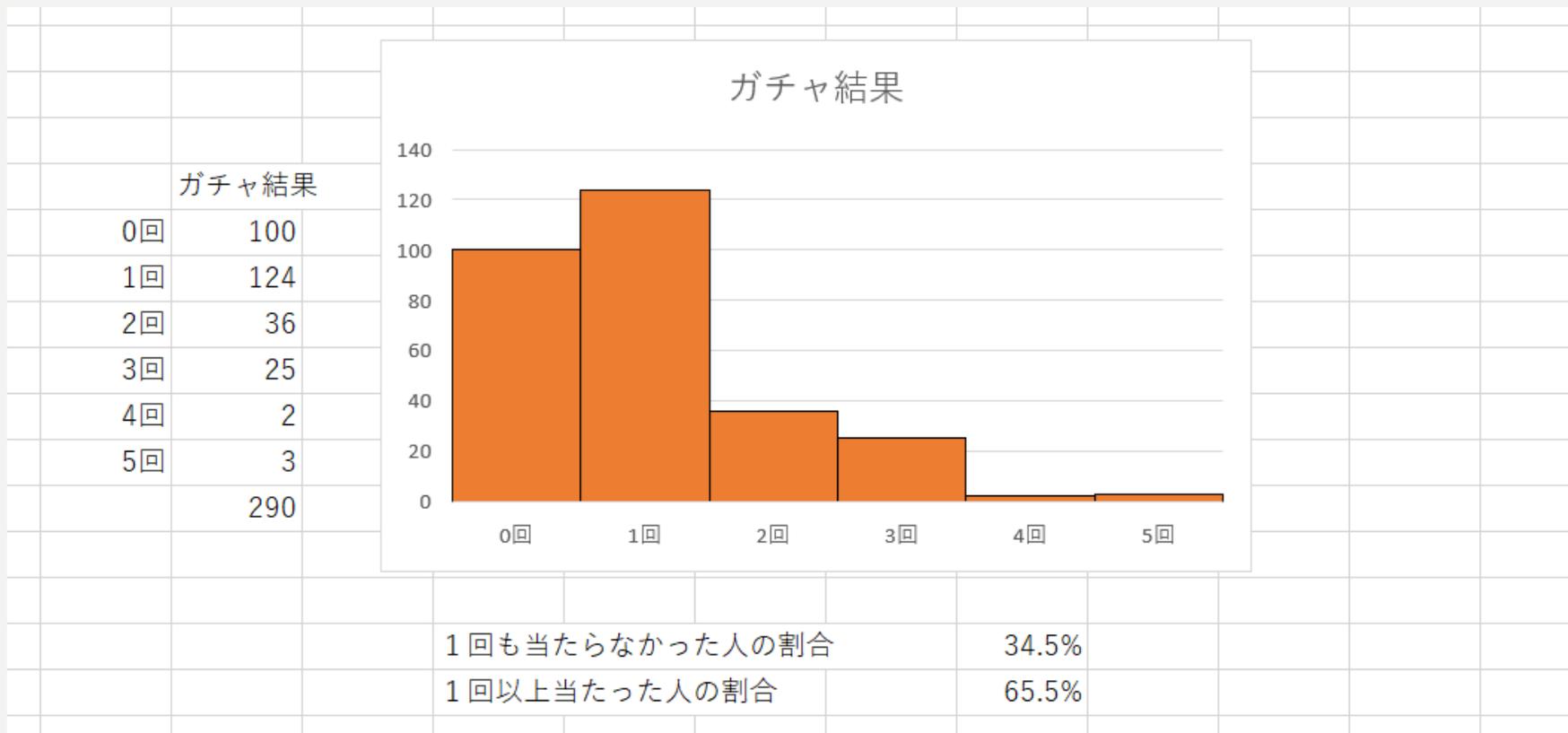
時数	内容	キーワード
1	コンピュータのしくみ	ハードウェア、ソフトウェア、浮動小数点数
2	文字列・数値の表示	バグ、デバッグ、算術演算子
3	変数と代入	変数、代入、代入演算子
4	条件分岐	フローチャート、条件分岐、比較演算子
5	繰り返し	繰り返し、ループカウンタ
6	配列(リスト)	リスト、要素、インデックス(添字)
7	10連ガチャの実装	モデル化、シミュレーション、乱数、モンテカルロ法
8	演習問題	整列や探索など

モデル化とシミュレーションに繋げる



「100連すれば絶対1回は出るはず！」と震えながらガチャを回そうとしますが、100回ガチャを引いて排出率1%のキャラが1体以上手に入る確率はたったの63%。
→計算式は「100回連続で引けない確率 (0.99の100乗) を1から引く」

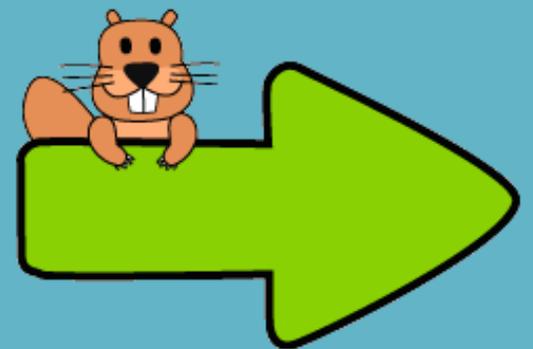
モデル化とシミュレーションに繋げる



「100連すれば絶対1回は出るはず！」と震えながらガチャを回そうとしますが、100回ガチャを引いて排出率1%のキャラが1体以上手に入る確率はたったの63%。
→計算式は「100回連続で引けない確率 (0.99の100乗) を1から引く」

実践事例⑤⑥

アルゴリズムと
Bit Arrowを用いた授業実践

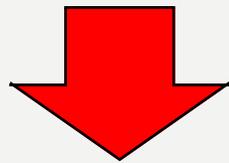


はじめに

- **実施年度** : **2022年度**
- **学年** : **1年**
- **科目** : **情報Ⅰで実施**
- **使用言語** : Python3
- **実行環境** : Bit Arrow (<https://bitarrow.eplang.jp/>)

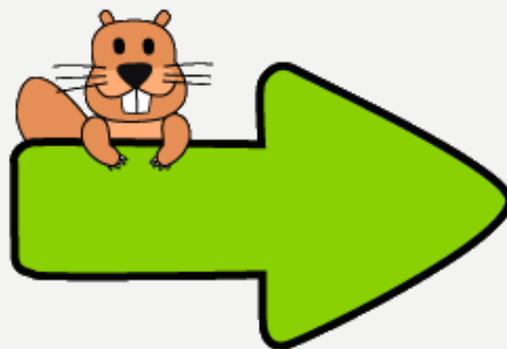
実践への思い

プログラミングは「難しい」というイメージが先行している・・・



アルゴリズムを理解してプログラミングを楽しんでもらいたい！

実践事例



BIT ARROW

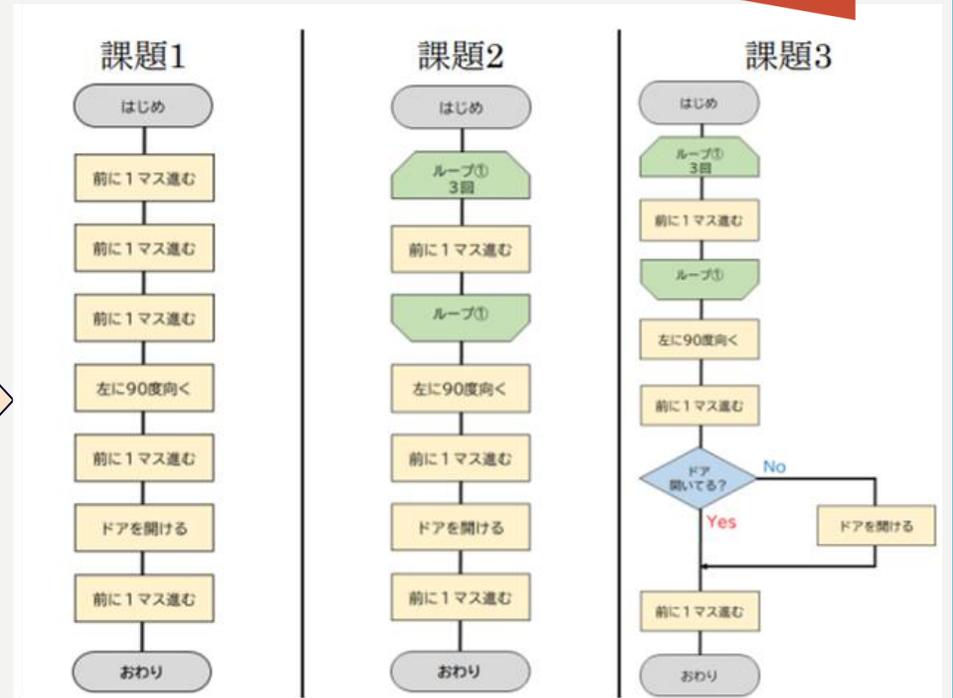
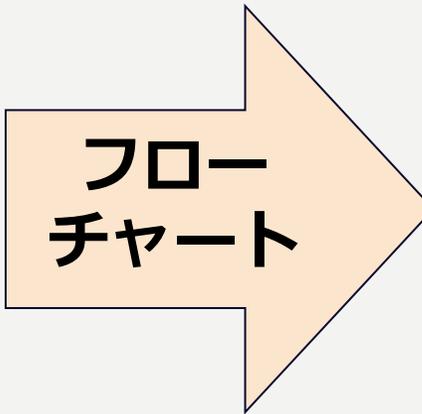
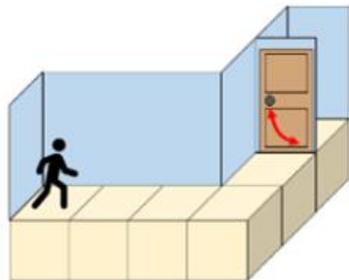
環境設定不要！

**ユーザ登録により生徒の進捗確認可能
PC、タブレット、スマホでも実行可能**

実践事例



	条件
共通	できる行動(処理)は、「前に1マス進む」「左に90度向く」「ドアを開ける」の3つ
課題1	<ul style="list-style-type: none">・ドアは常に閉まっている・順次構造のみを利用
課題2	<ul style="list-style-type: none">・ドアは常に閉まっている・順次と反復構造を利用
課題3	<ul style="list-style-type: none">・ドアは開いているかもしれないし、閉まっているかもしれない・順次、反復、分岐構造の全てを利用

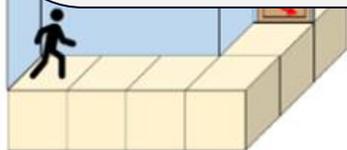


実践事例

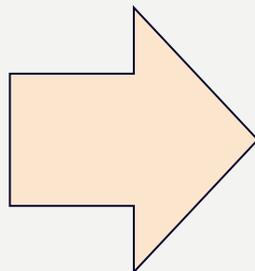
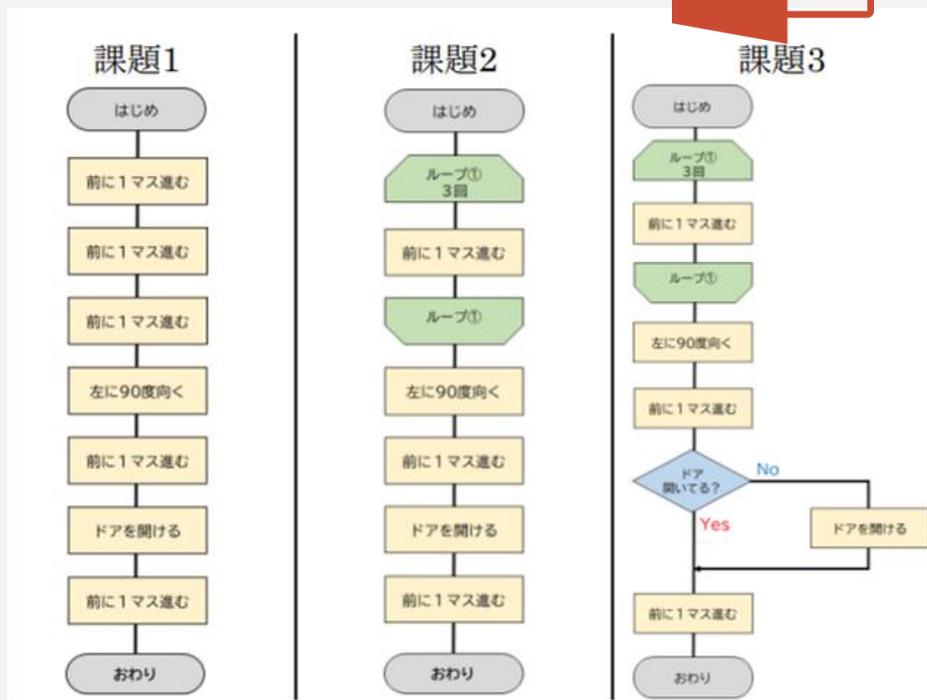
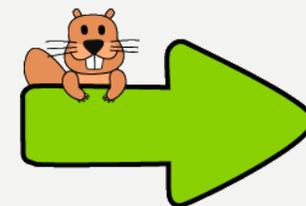


	条件
共通	できる行動(処理)は、「前に1マス進む」「左に90度向く」「ドアを開ける」の3つ
課題1	・ドアは常に閉まっている ・順
課題2	・ド ・順
課題3	・ド 閉 ・順

まずは、しっかりと
アルゴリズムを理解する！

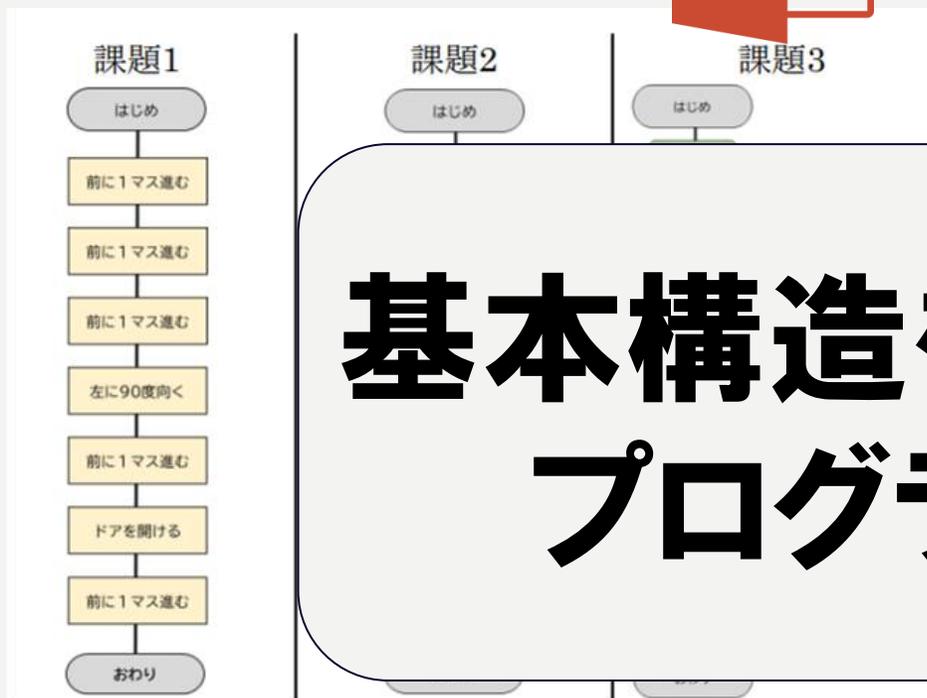
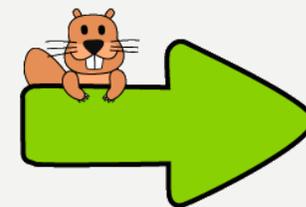


実践事例



```
Bit Arrow  ファイル  実行  保存  設定  ツール  教員  使用方法
Test1/
ex1
ex2
ex3
ex4
ex5
ex6
ex7
ex8
kadai1
kadai2
kadai3
kadai_plus1
Python ex1 別ページで表示
1 print(2022)
2 print("hello")
```

実践事例



基本構造を理解したうえで、
プログラミング実践！

教員 使用方法

実践事例

オリジナルの例題に
挑戦する

例題1	「2022」、「hello」と表示させる
例題2	変数nameに名前を代入し表示させる
例題3	「名前は?」の表示のあとに入力すると、「〇〇さん、こんにちは」と表示させる
例題4	「+」「-」「*」「/」「**」を使った計算
例題5	三角形の面積の計算 (底辺・高さはあらかじめ代入)
例題6	三角形の面積の計算 (底辺・高さはinput関数で入力)
例題7	変数a,bの大小を比較して、それぞれに応じた計算をする

例題8	1~5までの数字を順に表示させる
課題1	円の面積の計算 (円周率は3.14、半径はinput関数で入力)
課題2	18歳以上は「adult」、18歳未満は「child」と表示させる(年齢はinput関数で入力)
課題3	50より小さい偶数を順に表示させる (2,4,6,...,46,48)
応用1	西暦を和暦(令和)に変換し表示する (西暦はinput関数で入力)
応用2	パスワードが正しければ「ログイン完了」、正しくなければ「違う」と表示する (パスワードはinput関数で入力)
応用3	九九の5の段を順に表示する ($5 \times 0 = 0$, $5 \times 1 = 1$, ..., $5 \times 9 = 45$)

授業の流れ①(10時間)

1～2時間目	フローチャート、アルゴリズムの基本構造
3時間目	プログラミングとは何か
4～7時間目	プログラミング（例題） 例題のプログラムを入力・実行させる
8～10時間目	プログラミング（課題） 早く終わった人は応用課題に取り組む

授業の流れ②(6時間)

1 時間目	プログラムの基本構造1 演算、値の使い方、制御構造(順次・分岐・反復)
2 時間目	プログラムの基本構造2 分岐構造、反復構造のプログラミング実習
3～4 時間目	発展的なプログラム1 制御構造の組み合わせ、リストと配列
5～6 時間目	発展的なプログラム2 乱数、関数

実践事例⑦

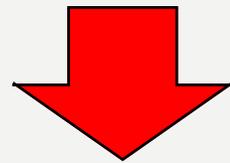
表計算ソフトウェアを活用した
プログラミング 学習

はじめに

- **実施年度** : **2022年度**
- **学年** : **1年**
- **科目** : **情報Ⅰで実施**
- **使用言語** : **VBA (MS-EXCEL2016)**

実践への思い

プログラミングの授業を通して、
問題解決を体験させたい！



自分の力でプログラミングして
実行するまでを体験させる！

授業の流れ(5時間)

生徒同士の
教え合い
推奨!

1 時間目	フローチャートの作成 変数の考え方、順次、分岐、反復の説明
2～3 時間目	プログラミング環境の設定 簡単な流れ図を基に実際にプログラミング
4～5 時間目	エクセルシート操作を伴うプログラミング

操作をしながら
アルゴリズムの検討やプログラミング!

振り返り①

限られた授業時間数でどのように教えるかという点においてアプローチに違いがみられたが、本質的に習得させたい能力は共通したものであることがわかった。また、単に演習問題をやらせるだけではなく、**自分で何かを生み出す経験があると夢中に取り組む傾向があった。**

**使用言語よりもプログラミングによって
様々な問題解決を体験させることが大事**

振り返り②

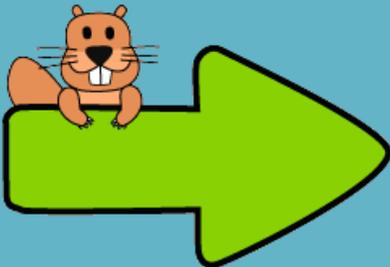
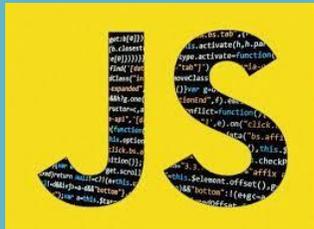
生徒間の能力差を配慮することは重要

- **プログラミングが得意な生徒**
授業内で扱わない演習問題を用意しておいたり、仲間に教えるように指示したりする工夫があるとよい。
- **プログラミングが苦手な生徒**
教員や他の生徒がサポートする環境を作れるとよい。また、グループ活動などで自分たちの力で何かが作れるという体験ができるとうよい。

おわりに

7つの授業実践をご紹介します

先生方の日頃の授業のヒントに
してただけたらうれしいです



ご清聴ありがとうございました



埼玉県高等学校情報教育研究会Webページ
URL : <https://www.saikojoken.net/>