

プログラミングの指導案

code.orgを用いた授業

埼玉県高等学校情報教育研究会

埼玉県高等学校情報教育研究会

- 教科「情報」で採用されたメンバーで構成されている
- 毎年共通のテーマを決めて指導案を研究している
- 一昨年は「論理的思考」を身につける授業案
- 昨年は「プログラミング」教えるにあたってのアンケートと授業案の検討

プログラミング教育が必須になる？

どこから？

どんなふうに？

どのレベルまで？

教えればいいのか？

29年度改定教科書では「社会と情報」でも
プログラミングを取り上げている教科書が増えた



「社会と情報」で いかにプログラミングを取り上げるか？

時間が少ない



効率よく学ばせたい

アルゴリズムで
難しいと感じる



入口は易しく

導入をやさしくできないだろうか？

- code.orgを使ってみる

<https://code.org/>



code.orgとは

- アメリカの非営利団体が立ち上げたサイト
- ビルゲイツやマークザッカーバーグらの寄付金を運営に充てている
- アメリカやイギリスの小学校で実際に利用されているらしい
- code.studioは、オンライン20時間の学習コース

code.studioとは

- オンライン20時間の学習コース
- 4歳から18歳までの初心者対象
- ブロックベースのプログラミング学習
- 順次処理 → 回数指定の繰り返し処理 (for文) →
終了条件指定の繰り返し処理 (while文) → 関数 (function) →
分岐条件 (if文) → 入れ子のループ → デバッグなど
が順を追って学習できる
- この学習方法はアメリカではバークレー大学やハーバード大学でも採用されている

学習方法

- ブロックでアルゴリズムを確認
- コード表示で実際に動いているプログラム (JavaScript) を確認



A screenshot of the Code.org 'Frozen' puzzle interface. The browser address bar shows 'https://studio.code.org/s/frozen/stage/1/puzzle/1'. The interface includes a character of Elsa on a blue background. A speech bubble says: 「実行」ボタンをおして、プログラムをうごかしてみてください。 Below this is an '実行' button and a slider. To the right, there are three movement blocks: '前に動く 100 ピクセル', '右に向く 90 角度', and '左に向く 90 角度'. Below the blocks is a character icon of Elsa and a message: こんにちは！アレンデルお国くのエルサです。1ぼんのせんをかくのをてつたってください。 At the bottom, there is a link for '助けが必要ですか？ビデオやヒントを見る' and a video thumbnail titled 'Hour of Codeの概要'. The bottom of the interface shows a language dropdown set to '日本語' and a copyright notice '著作権 | それ以上'.

code.studioを用いた 授業案の紹介

1. フローチャート→ブロック→プログラム確認
2. ブロック→フローチャート→プログラム確認
3. ブロック→プログラム確認
4. ブロック→プログラム→フローチャート→

仕事に応用する

授業提案1

フローチャートとビジュアルプログラミング

目標 フローチャートを用いて問題の整理



コンピュータに命令を出す

対象クラス 1年生

高校入学以前に学習につまづきを覚えた生徒多数

2学期に実施

実施の流れ(50分1コマ)



①プログラミングとは？

②フローチャートを考える

③ブロックを操作して

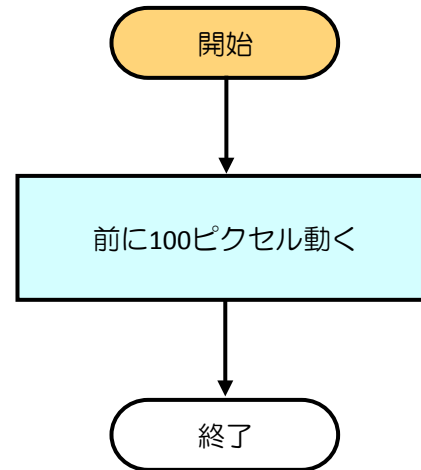
ビジュアルプログラミングを動かす

④プログラミングの記述を確認する

フローチャートを書いてみる



こんにちは！アレンデルお国このエルサです。
1ぽんのせんをかくのをてつだってください。



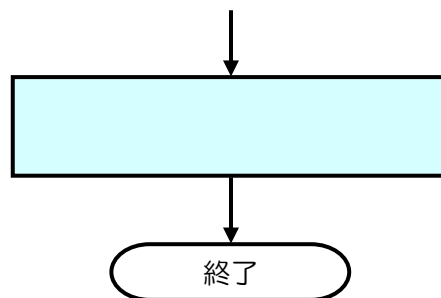
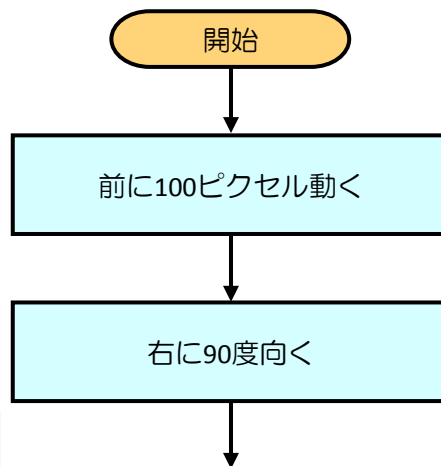
```
moveForward(100); //前に100ピクセル動く
```

四角を描いてみる

～順次処理～



しかくをかくのにはんぶんのところまでできたみたいだわ。
さあ、しかくをかくために、4ほんのせんをひきましょう！



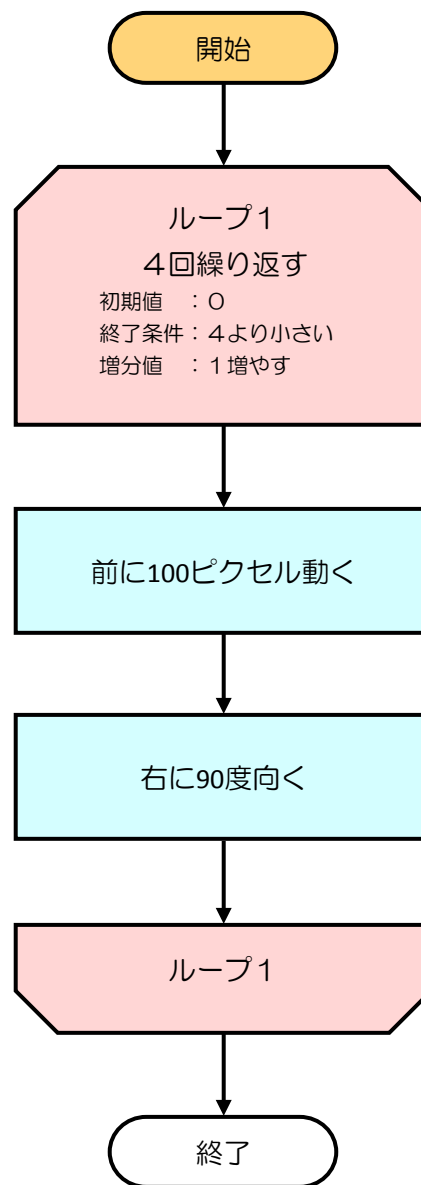
```
moveForward(100);
turnRight(90);
moveForward(100);
turnRight(90);
moveForward(100);
turnRight(90);
moveForward(100);
```

四角を描いてみる

～繰り返し処理～



こんにちは、わたしはアレンデルお国のアナよ！
「くりかえし」ブロックをつかって、
さっきよりすくなくブロックでしかくをかいてみましょう。
なんか「くりかえし」ブロックのなかのブロックをくりかえせば、
しかくがかけるかしら？



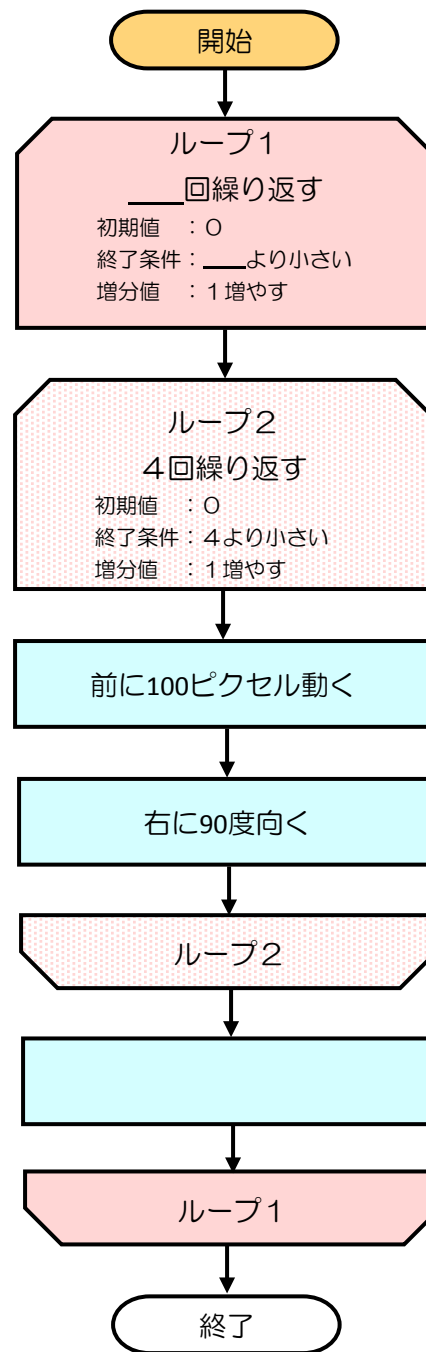
```
for (var count = 0; count < 4; count++) {  
  moveForward(100);  
  turnRight(90);  
}
```

//countの値は0からスタート
//Countの値が4より小さい間、 { } 内の処理を繰り返す
//繰り返すたびにcountの値を1増やしていく

「四角を3つ描く」



しかくを3つかいてみましょう。
つぎのしかくをかくまえに、かならず120どむきかえるのよ。



```
for (var count2 = 0; count2 < 3; count2++)  
{  
  for (var count = 0; count < 4; count++) {  
    moveForward(100);  
    turnRight(90);  
  }  
  turnRight(120);  
}
```


留意点

- JavaScriptは記述させない
- プログラムコードは見せるだけ（プログラムを意識させる）
- 生徒の能力に応じたペースで作業をさせる
- 生徒同士の話し合い(教え合い)歓迎

振り返り(教師)

- なんでプログラムを学ぶかの説明不十分
- 「わかると楽しい」という経験が学習意欲につながる
- 指示通りの結果がでて、「もっと少ないブロック数でできる」というメッセージが出ると悔しがって改良の努力をしていた
- 5まではワークシートを作ったが6以降(5の応用)は自分たちでやらせたら難しくてできなかった
- ページの読み込みに時間がかかった
- この課題を発展させて、どのように論理的思考力を養うかが課題

生徒の感想

やってみたい

- 今回は簡単なものばかりだったが、今度はもっと難しいものに挑戦したい。
- アナとエルサのゲームでやると**楽しい**。
- わかると意外と**楽しい**。

- 楽しい**けど難しかった
- 前よりもプログラミングが**楽しそう**に見えてきた
- プログラミングは**大変** 自分に向かない

- 意外と**楽しかった**。
- 楽しい**けど難しい。
- 頭を使うので**大変**だったけど**すごく楽しくできて**良かった。
- エルサのゲームでやるとちょっとだけ理解できたけど、だんだん難しくなり、一人では**無理**かも。
- かなり**面倒**。もうやりたくない。
- むずかしすぎる。パソコン本当に**無理**。

- 思っていたより**楽しかった**。家でもやりたい。
- 中学でやったのは難しくて嫌だったが今日のは簡単だったので**楽しかった**。
- 数学苦手でよくわかんないけど**楽しかった**。
- 難しかったけど**楽しかった**。
- 思ったより**楽しかった**けど、**やろうとは思わない**。
- 難しかった**。

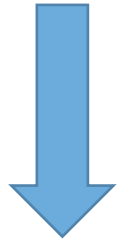
経験あり

授業提案2

code.orgを活用してフローチャートを作成する

目標

- ① **ブロック操作** ビジュアルプログラムを動かす
- ② Wordで**フローチャート**を表現する
順次・反復・選択処理を理解する



コンピューターリテラシーを高める

授業の流れ(50分4コマ)

- 1限目 code.orgの紹介
実行・「順次構造」の理解
基本的な言葉の理解
- 2限目 「反復構造」の理解
- 3限目 「選択構造」の理解
- 4限目 コード表示 → プログラムで動いていることを確認

ドリル形式の教材 順次構造

STUDIO



実行

ブタを捕まえるのを手伝ってくれない？「前に動く」ブロックをつみあげて、「実行」ボタンをおしてね

実行した時
まえにすすむ
まえにすすむ

フローチャートを書いてみよう

年 組 番 氏名

フローチャートとは？
プログラミング言語を使わずにアルゴリズムを記述する方法。

利点
考えているアルゴリズムが正しいかどうか処理や手順に間違いがないかを目で確認する事ができ、わかりやすく伝える事ができる。

ミッション
code.org の迷路 (Stage1~20) を解いてそれをフローチャートで表してみよう！

必要な記号

- 開始 (オレンジの楕円)
- 終了 (オレンジの楕円)
- 処理記号 (緑の長方形)
- 前に進む (緑の長方形)

端子記号
アルゴリズムの開始と終了を表す記号

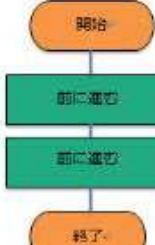
処理記号
処理を表す。記号の中に具体的な処理内容を書く

流れ線
記号同士を結び処理の流れを示す。
水平や上から下へ向かうときは矢印を使う

★必要な記号をコピー＆ペースト、編集してフローチャートを書いてみよう

Stage1 (例)

ポイント：フローチャートの開始と終了を記入する！



```
graph TD; Start([開始]) --> Step1[前に進む]; Step1 --> Step2[前に進む]; Step2 --> End([終了]);
```

ドリル形式の教材 反復構造



②フローチャートを書いてみよう+

Stage6~Stage13 解答編

年組番氏名

ミッション code.org の迷路を解いてそれをフローチャートで表してみよう！+

必要な記号

- 開始 (Start) 終了 (End) 端子記号 (Terminal symbol) アルゴリズムの開始と終了を表す記号
- 前に進む (Move forward) 処理記号 (Process symbol) 処理を表す。記号の中に具体的な処理内容を書く

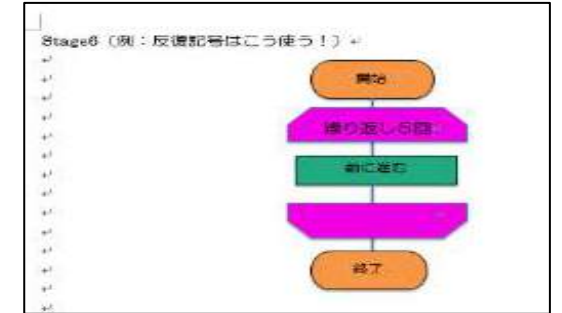
流れ線 (Flow line) 記号同士を結び処理の流れを示す。水平や上から下へ向かうときは矢印を使う

新しく必要な記号

- 繰り返しの記号 (Repeat symbol) ループ記号 (Loop symbol) 反復構造の開始、終了を表す記号。中にループの条件を記入する

Point!

コンピュータは繰り返すのが得意！
間違えずに何度でも素早く繰り返す事ができる。



ドリル形式の教材 選択構造



実行

新しいもし*ブロックを使い、いつ曲がるかを決めてください。ヒント: あともう1つのブロックが必要です。どう設定しているかを調べば、次はひとりでできるようになります。

実行した時

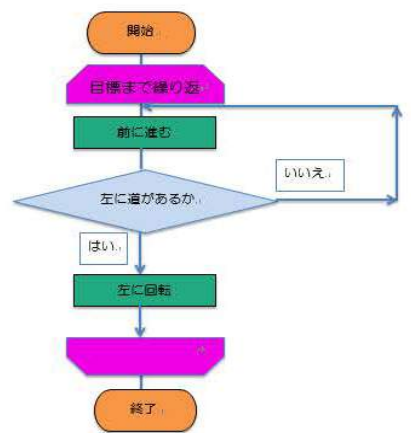
とどくまでくりかえす

やること

- まえにすすむ
- ひだりにみちがあるなら
- やること
- ひだりにまがる

Stage14 (例: 判断記号はこう使う!)

判断記号には必ず「はい」「いいえ」を書く!



```
graph TD; Start([開始]); Repeat[/目標まで繰り返す/]; Move[前に進む]; Decision{左に道があるか?}; Turn[左に回転]; End([終了]); Repeat --> Move; Move --> Decision; Decision -- はい --> Turn; Turn --> Repeat; Decision -- いいえ --> Decision;
```



```
while (notFinished()) {  
  moveForward();  
  if (isPathLeft()) {  
    turnLeft();  
  }  
}
```


留意点

- フローチャートやWordに不慣れな生徒のために実際に作業しながら説明することが必要
- 教材を回収してフィードバックすることが必要
- 作業の進行に差が出る→
早く終了した生徒は他の生徒のサポート役にする

評価対象

- Code.orgのステージをクリアできたか
- Wordでフローチャートを描けたか

授業提案3

code.orgを利用したプログラミング

目的 プログラミング教育



コーディング → 実行 → 改善

思考力 問題解決能力(分析・判断力)を養う

授業の流れ (50分5コマ)

ブロック→プログラミング

- 1限目 順次構造 (アナとエルサ 1~3)
反復構造 (アナとエルサ 4~7)
- 2限目 順次構造 (古典的な迷路 1~5)
分岐構造 (古典的な迷路 6~15)
- 3限目 プログラミング演習1
プログラムの紹介 (1~nまでの素数判定, 月齢計算)
プログラムの作成 (1~nまでの合計計算)
- 4限目 プログラミング演習2 復習 台形の面積を求める
アルゴリズムの解説 分岐構造
- 5限目 プログラミング演習3 数値nの奇数偶数判定
1~100までの整数を表示する
アルゴリズムの解説

実施の流れ

- code.orgの実行
プリントでの復習（1・2限目）
- VBScriptを利用した
プログラミング学習（3～5限目）

留意点

エラー表示(行・内容)の確認



問題を認識して改善できる

プログラミング特有の表現については

×詳細な説明(時間がかかる)

○例題の表示

授業提案4

アルゴリズム学習から問題解決学習へ

題材の目的

文化祭の企画内容の振り返り



当日までの作業をリスト化する



フローチャートの作成(流れを理解する)



無駄な部分・失敗点の改善案
パターン化・共有 → 新しい工夫

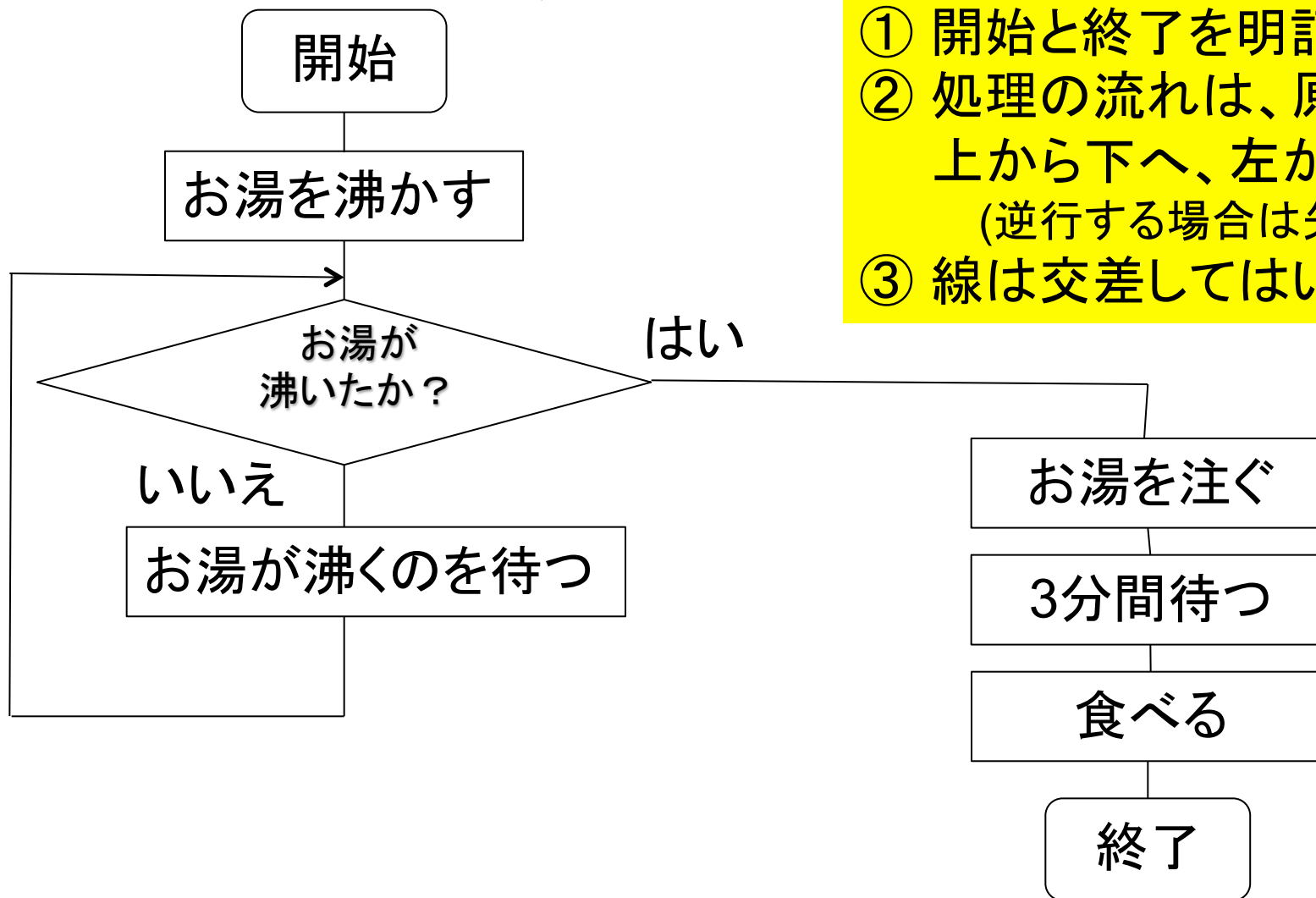
題材設定の理由

- フローチャートからcode.orgへ
プログラミング実習へスムーズに移行できた
- ゲーム感覚でできる(解説が容易)
ブロック操作→動画で確認→完成したコードを見る
フローチャートより達成感がある
- 実際にプログラミング学習をするより時間が短縮できる

実施の流れ








- 1～2限 アルゴリズムとプログラミング
code.org(アナとエルサ・古典的な迷路)を使用して学習する
- 3限 フローチャートの描き方説明・班分け・企画の振り返り
- 4～8限 作業のリスト化・手順の並べ替え
ワークシートの作成・グループごとの発表

フローチャート(流れ図)



- ① 開始と終了を明記する
- ② 処理の流れは、原則として、上から下へ、左から右へ
(逆行する場合は矢印をつける)
- ③ 線は交差してはいけない

フローチャートの記号

	端子 フローチャートの始まり及び終わりを表す		ループの開始
	処理 計算、代入などの処理を表す		ループの終了
	サブルーチン 定義済みの処理を表す		入出力 ファイルへの入出力を表す
	判断 条件による分岐を表す		ページ内結合子 フローチャートが長くなり、ページ内で二列にするときなどに使う。
	表示 コンソール上への結果の表示を表す		ページ外結合子 フローチャートを次のページに続けるときに使う。

フローチャートを書いてみよう

例) 遠足のカレー

作業内容(順番を考えよう)

- 野菜を切る
- 肉を切る
- 材料を炒める
- 鍋に水を入れる
- ルーを入れて煮込む
- 米を炊く
- 火をおこす
- 使った皿や鍋を洗う
- 調理器具を用意する
- 使った調理器具を洗う
- 全員で「いただきます」
- ごみを捨て、流しを掃除する

- テーブルを拭く
- お皿に米とカレーをよそう
- 皿をテーブルまで運ぶ
- 味見する
- 米を研ぐ
- 煮込む
- あくをとる



確認事項

- 材料に火は通ったか
- 全員分のカレーはそろったか
- 片づけは終わったか...

フローチャートの特徴とメリット

- 作業を一つ一つバラバラに分ける
- 作業の流れをあらわす



- 複数人で作業する場合は分担ができる
- 時間の流れがわかるため、いつまでに何をすれば良いか、スケジュール管理ができる



- より効率的に作業を行うことができる

課題：文化祭のクラス企画を振り返る

- 文化祭準備～当日までの期間に、何をしたか？
- いつ、どんな作業を、誰が担当していたか？
- 自分がした仕事、係りで担当した仕事、クラス全体でした仕事、誰もやらなかった仕事・・・
- 無駄な部分はどこか、失敗した点はあるか、それに対する改善案はあるか・・・

できるだけ具体的に書き出してみよう!!

全体の動き

- 6月
 - 企画決め
 - 書類の提出
- 7月
 - 情報の共有
- 8月
 - 仕事分担
- 9月
 - 進捗状況の把握
- ...

どの時期に、誰が、
何をすれば良い？

係りの動き

- 装飾
- 衣装
- 買い出し
- Tシャツ
- 看板
- ポスター
- ワンショット
- 当日の係り...

班ごとの係り分担

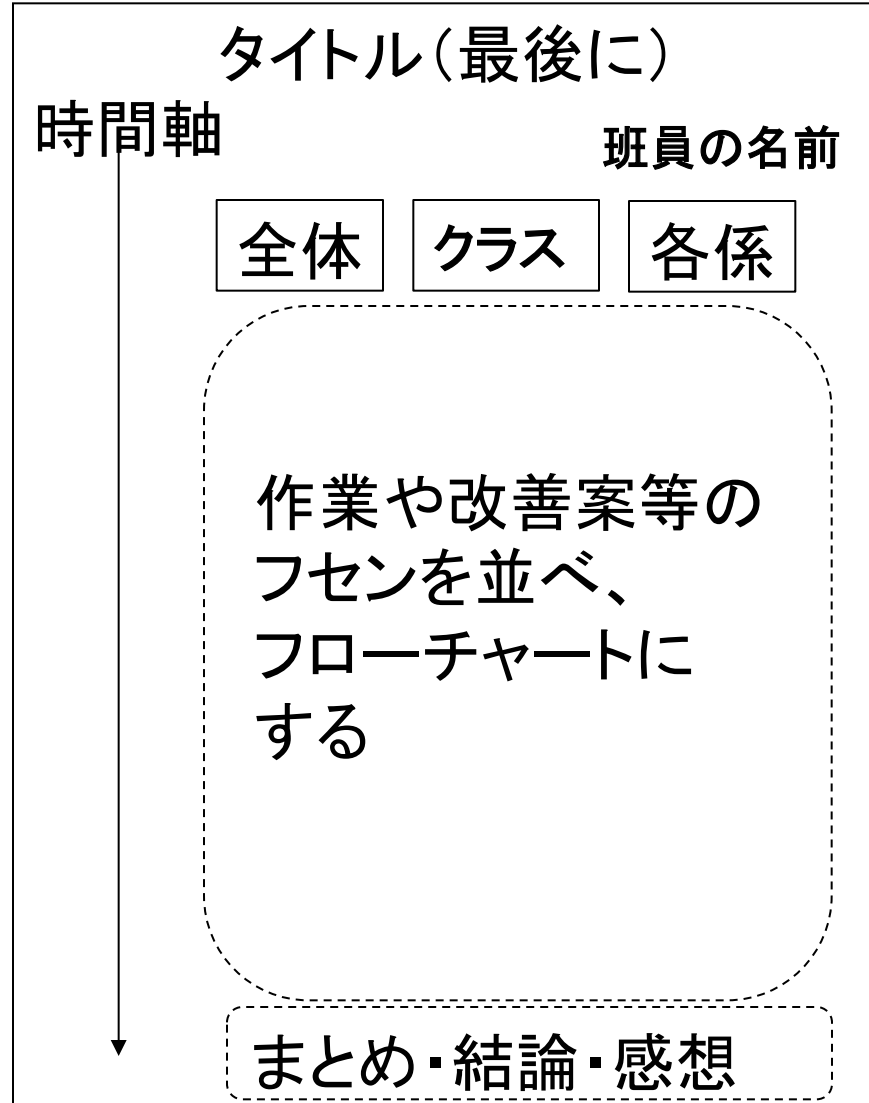
1	司会	話し合いをスムーズに進めよう
2	記録 (ワークシート)	話し合いで出た意見などをワークシートに まとめよう
3	記録① (模造紙)	発表に使用する模造紙を完成させよう
4	記録② (模造紙)	発表に使用する模造紙を完成させよう
5	発表①	ワークシートを使って発表しよう
6	発表②	模造紙の発表されている部分を指し示す

※5人グループの場合は発表②の担当を誰かが兼任

グループでフローチャートを作ろう

- 各自が作業を1個ずつ付箋に書く(全員)
- 分岐や繰り返しのある部分は条件を書く
- 模造紙には6月から9月までの時間軸を書く
- いつやるべき作業か考え、付箋を取り組む順番に並べていく(模造紙に貼る)
- 誰が担当する作業なのか分類する
- うまくいった点、良かった点は目立たせる
- 無駄な部分や失敗した点については、改善案を考え、書き足していく

模造紙の書き方



配布物

ワークシート
模造紙
マジック
付箋
新聞紙

フローチャートで作業をリスト化する目的は

× 作業を機械的に処理できるようにすること

○ パターン化できる部分をパターン化し、
共有できるようにすることで、
話し合いを可能にし、
今までできなかった部分に意識を向け、
新しい工夫を生み出すことである。

組織で働くということ

- 共通のゴールのイメージを持つ
- 建設的な意見やアイデアを出し合う
- やるべき仕事やスケジュールを全員が把握する
- 作業は分担して効率的に行う(一人で抱えない)
- 途中経過を報告・相談し、相互に作業内容をチェックする (任せっぱなしにしない)
- 結果を評価する
(評価基準は事前に明確にしておく)

発表しよう

- 各班5分ずつ1～6班が発表。
- 発表者以外のメンバーが模造紙を持つ。
- 聞き手は模造紙前に集合。静かに聞くこと。
- 全員感想を書いて提出

留意点

- code.orgを使用する際は、ゲーム感覚になりすぎないように、適宜作業を止めて解説を行う
- グループ作業では全員に何かしら取り組める役割を与え、協力して作業を進められるように配慮する
- クラス代表・会計担当・文化委員・生徒会役員など役割ごとに把握する情報を分け、それぞれが担当する情報を班員に説明させる

評価

- ワークシートと授業の取り組み状況
- まとめの発表内容

おわりに

目的 「論理的思考」「問題解決能力」を身につける

手段 「プログラミング学習」

生徒にも教員にも難解！！

目標 効率よく

興味関心を高める

理解を深める

授業案づくり

今年度へ続く