初等・中等教育における
プログラミング言語の傾向と
イベント処理プログラミングについて

帝塚山学院大学
喜家村奨、稲川孝司

太成学院大学
西野和典

NPO法人 学習開発研究所
高橋参吉

初等・中等教育の接続性を考慮した
プログラミング言語の選択について

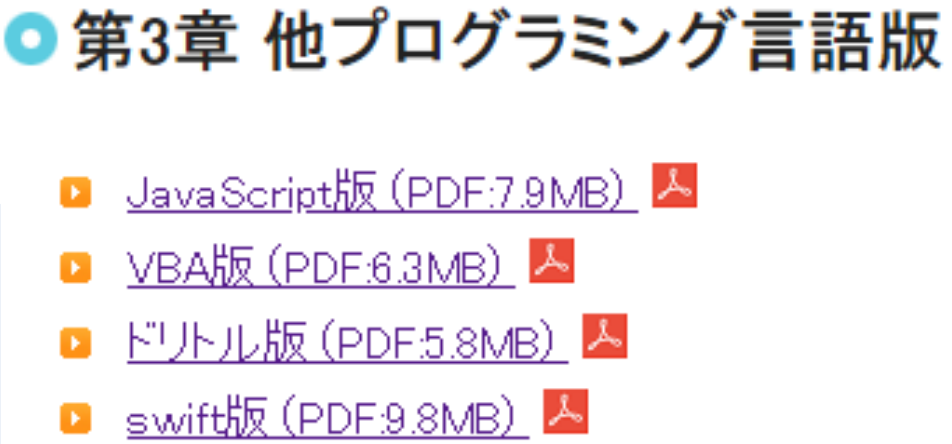
中学技術・家庭科の教科書のプログラミング言語の傾向

- 多くの教科書で、プログラミング言語を複数紹介している。
- 紹介されているプログラミング言語は、日本語入力型言語（なでしこ、ドリトル）、スクラッチ、JavaScript
- どの言語を使って、授業を実施するかは、教員よってまちまちであり、生徒がどのプログラミング言語を経験するかもまちまちであることが予想される。

このような状況の中、高校のプログラミング教育における言語選択を、どう考えればよいか？

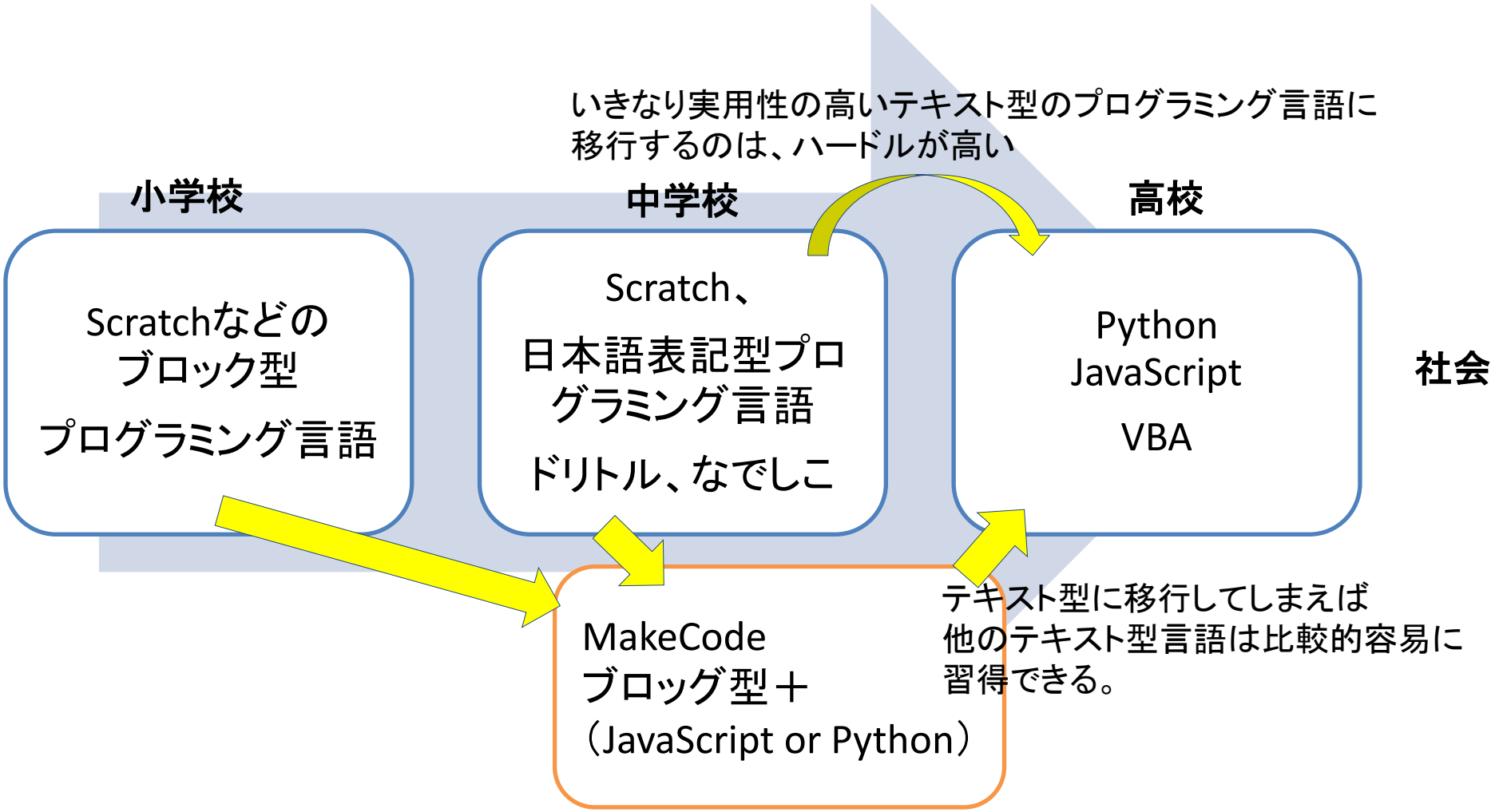
高等学校情報科「情報Ⅰ」教員研修用教材

- 本編には、Pythonというプログラミング言語が使われている。制御のところではmicro:bitが紹介されている。
- 以下の4種類の言語も例示されている。



ドリトル以外は
実際のアプリケーション開発に
使われているテキスト型の
プログラミング言語

教科書で使用されている言語の傾向



ブロック型、日本語入力型からいかにスムーズにテキスト型に移行できるかが課題

懸念点

- 高校では、（実用性が高い）テキスト型のプログラミング言語が多く使われているが、中学校までは、テキスト型のプログラミング言語の利用は少ない。
- クラス内で生徒によって、それまでに習ってきた言語がまちまちである可能性もあり、一つの言語で授業をすすめることが難しい。

MakeCodeエディター

- ブロック型、テキスト型（JavaScript、Python）を切り替えて、プログラミングが可能

https://makecode.microbit.org/#editor

ScratchからMicro:bitへの接続性を考慮したイベント処理ブロックの開発

イベント処理の方法

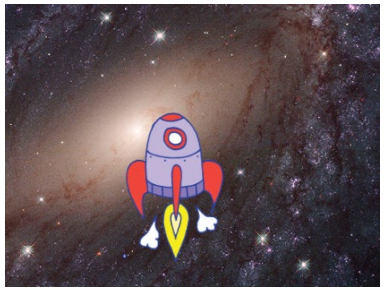
- イベントを処理する方法としては、大きく分けて以下の2つの方法がある
  - イベント駆動型：システムがイベントを検知して知らせてくれる。ユーザは発生したイベントに対応した処理をイベントハンドラに書く
  - イベント検知型：自分のプログラム内でイベントの発生を検知する処理を書く

※イベント検知型とは、あまり言いませんが、
ここでは、区別のために、そう呼んでいます。

イベント処理の例

- 以下のURLをクリックしてプログラムを実行してみてください。どちらのロケットも矢印キーで操作できますが、処理の仕方が異なります。
- https://scratch.mit.edu/projects/520990142
- 左のロケット：イベント駆動型プログラム
  - Scratchのシステムがイベント検知して知らせてくれる
- 右のロケット：イベント検知型プログラム
  - 自分のプログラム内でイベントの発生を検知する処理を書く

イベント駆動型のプログラム



- 右のプログラムは、イベントブロック「〇〇が押されたとき」を使い、キーが押されたらロケットの座標を動かすイベント駆動型のプログラム
- イベント駆動型のプログラムのメリットは
  - イベントの検知処理を自分で書く必要がない。
  - そのイベントに関する処理だけをひと固まりにかける
  - コードが比較的短くなる
- イベント駆動型のプログラムのデメリットは
  - 複数のイベントを処理する場合、それぞれの処理が並列で動くので、並列処理を意識したプログラミングをする必要がある
  - イベントの発生を無視できない

イベント検知型のプログラム

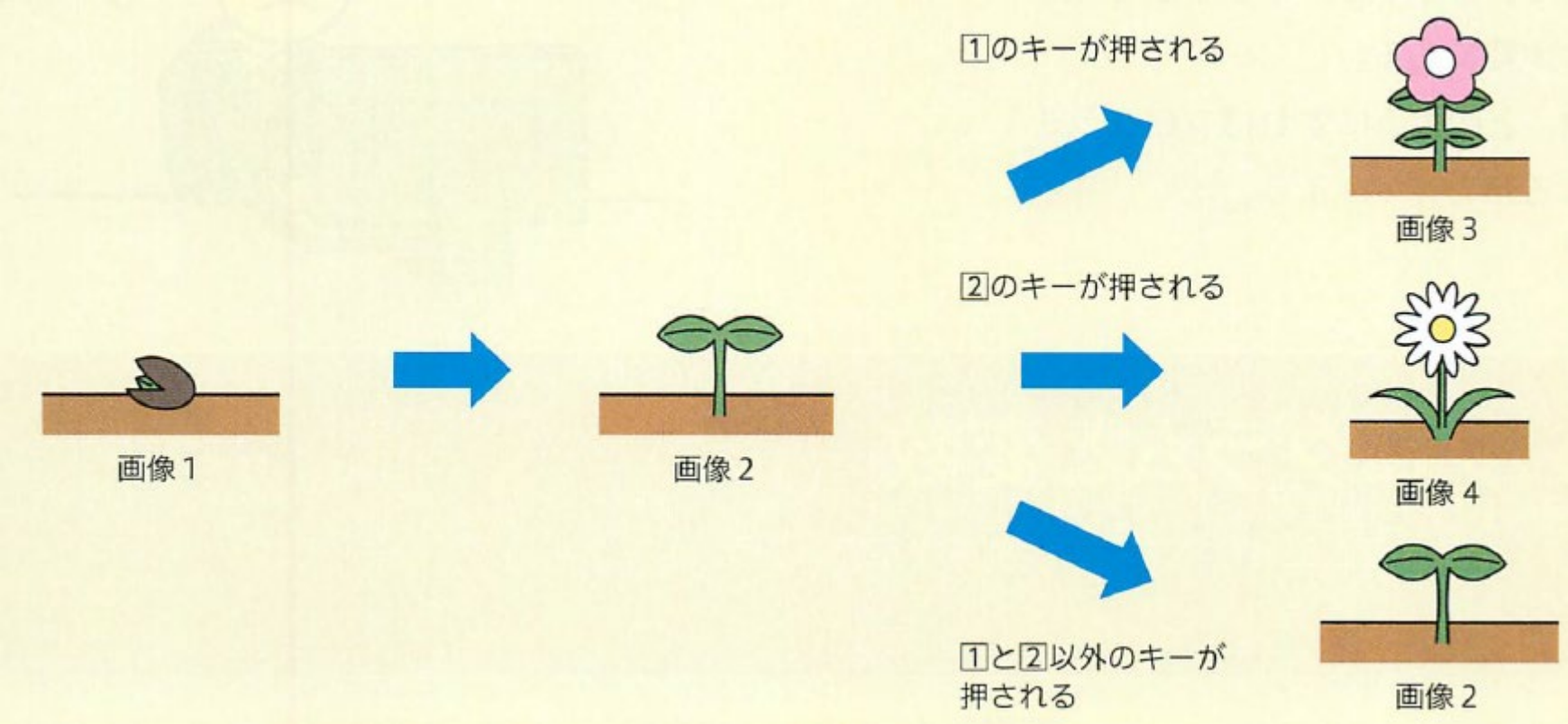
- 「キーが押された」を使い、自分のプログラム内で、キーが押されたかを繰り返し確認し、もし、キーが押されたら、ロケットの座標を動かす。
- イベント検知型のプログラムのメリットは
  - 複数のイベント処理を逐次的に記述できるので、プログラム全体の見通しがいい。
- イベント検知型のプログラムのデメリットは
  - 自分の処理の中でずっとイベントの発生を検知しないといけないので、CPU時間を浪費する。
  - コードが長くなりがち

どちらのイベント処理方法を使うか

- 一概にどちらがいいとは言いにくのですが、並列処理を意識したイベント処理をちゃんと書くのは難しい（排他制御のためにセマフォを使うとか、イベントキューを使うとか）。
- 次に示す例のように、ある状態から、発生するイベントによって処理をかえるようなとき（自動販売機などの順序機械のプログラミング）は、イベント検知型のほうが理解しやすいように思う。例で説明します。

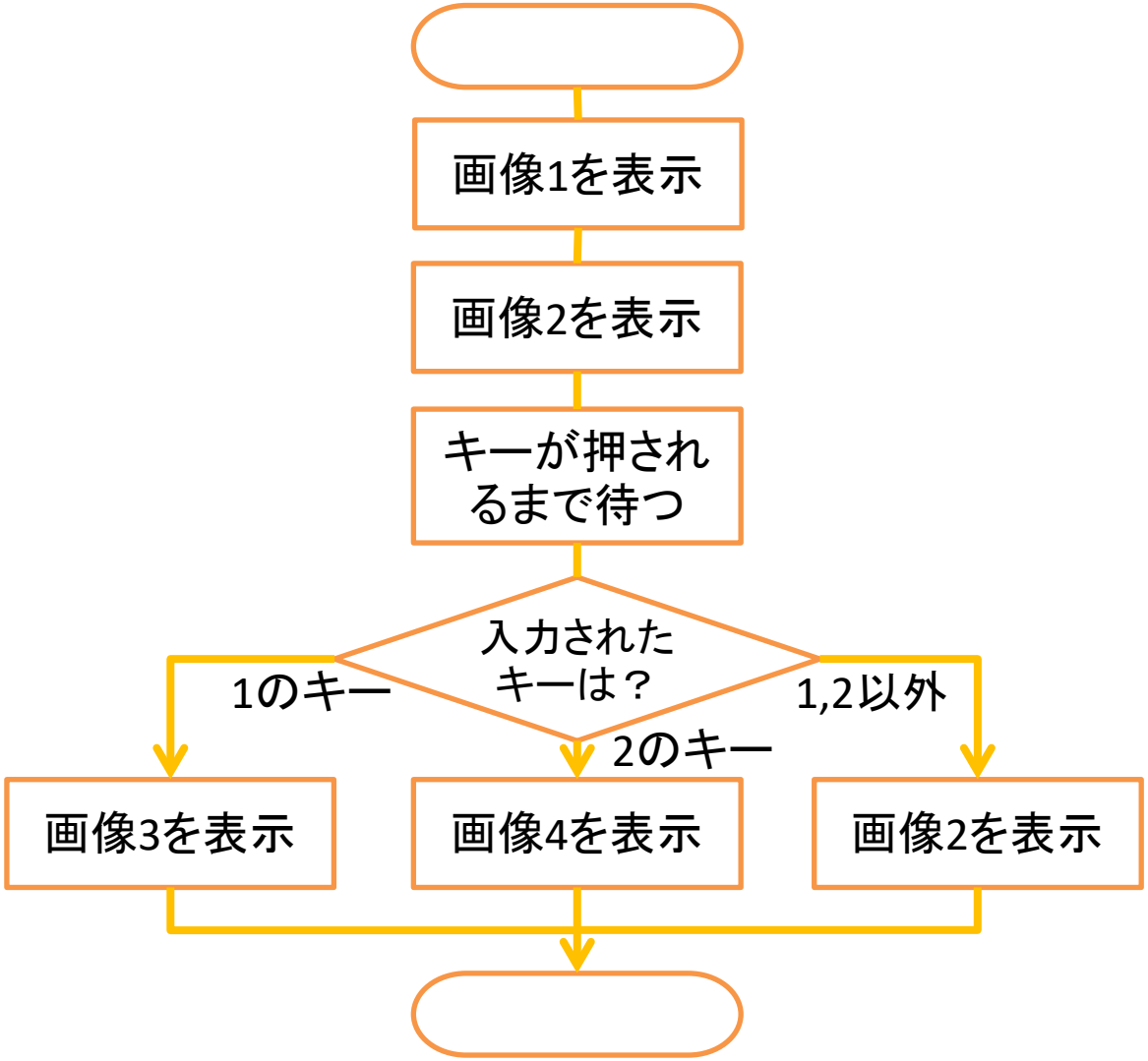
スクラッチのイベント処理プログラムの例

- 以下のような、キー入力を待ち、入力されたキーの応じて、処理をするプログラムを考える。

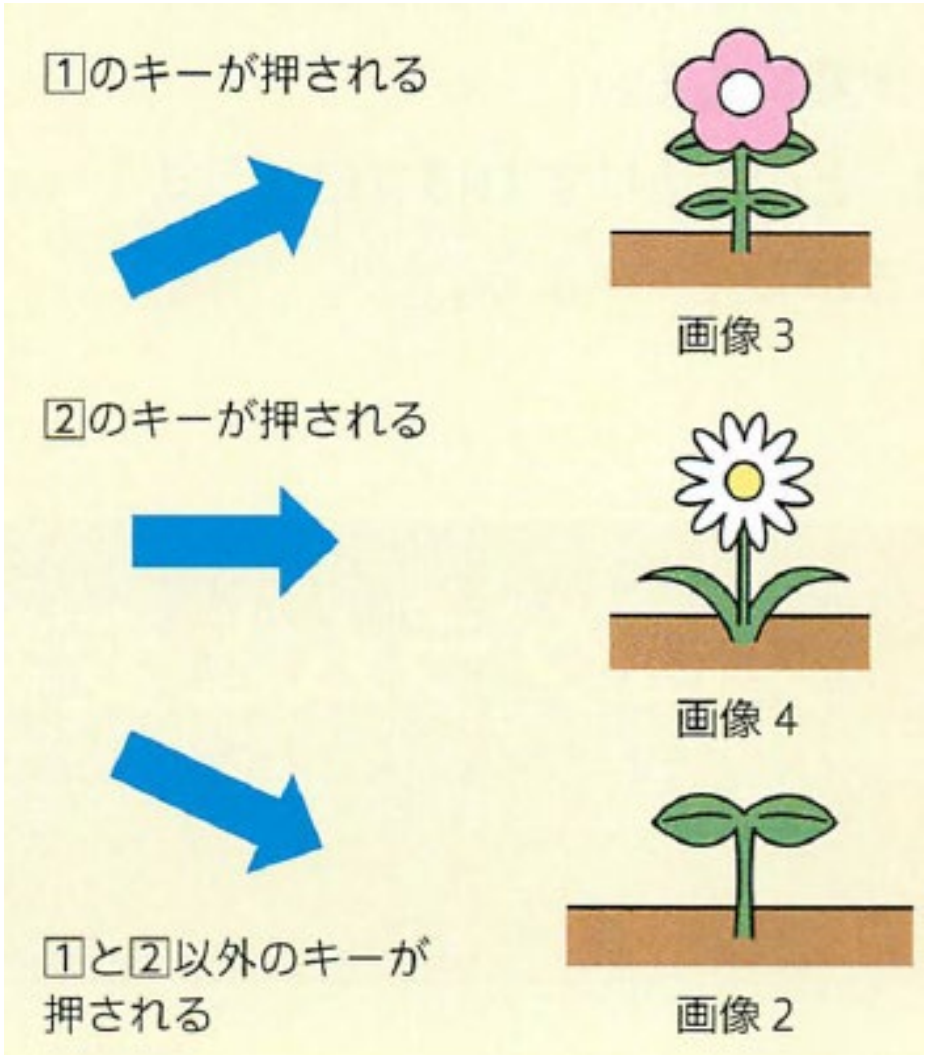


図説 情報 I　実教出版社より

花が咲く処理のフローチャート



スクラッチのプログラム



フローチャートで表現された順序機械の実装

- フローチャートは設計図であり、その手順通り、プログラムが書けることが望ましい。
- 例で示したようなプログラムで、イベント駆動型にすると（イベントハンドラを使うと）、同期処理のために共有変数を使ったり、イベントの発生後の処理を条件で制限したりする必要がある。

micro:bitでの実装

- 先ほどのプログラムをmicro:bitで実装することを考える。1または2のキーを押す代わりにmicro:bitのAボタンとBボタンを使ったとしても、MakeCodeのブロックでは、イベントの入力を待つブロックはイベントハンドラのみであり、処理が並列になる。
- ボタンを押されたかを検知するプログラムを書くこともできるが、スクラッチのプログラムより煩雑になる

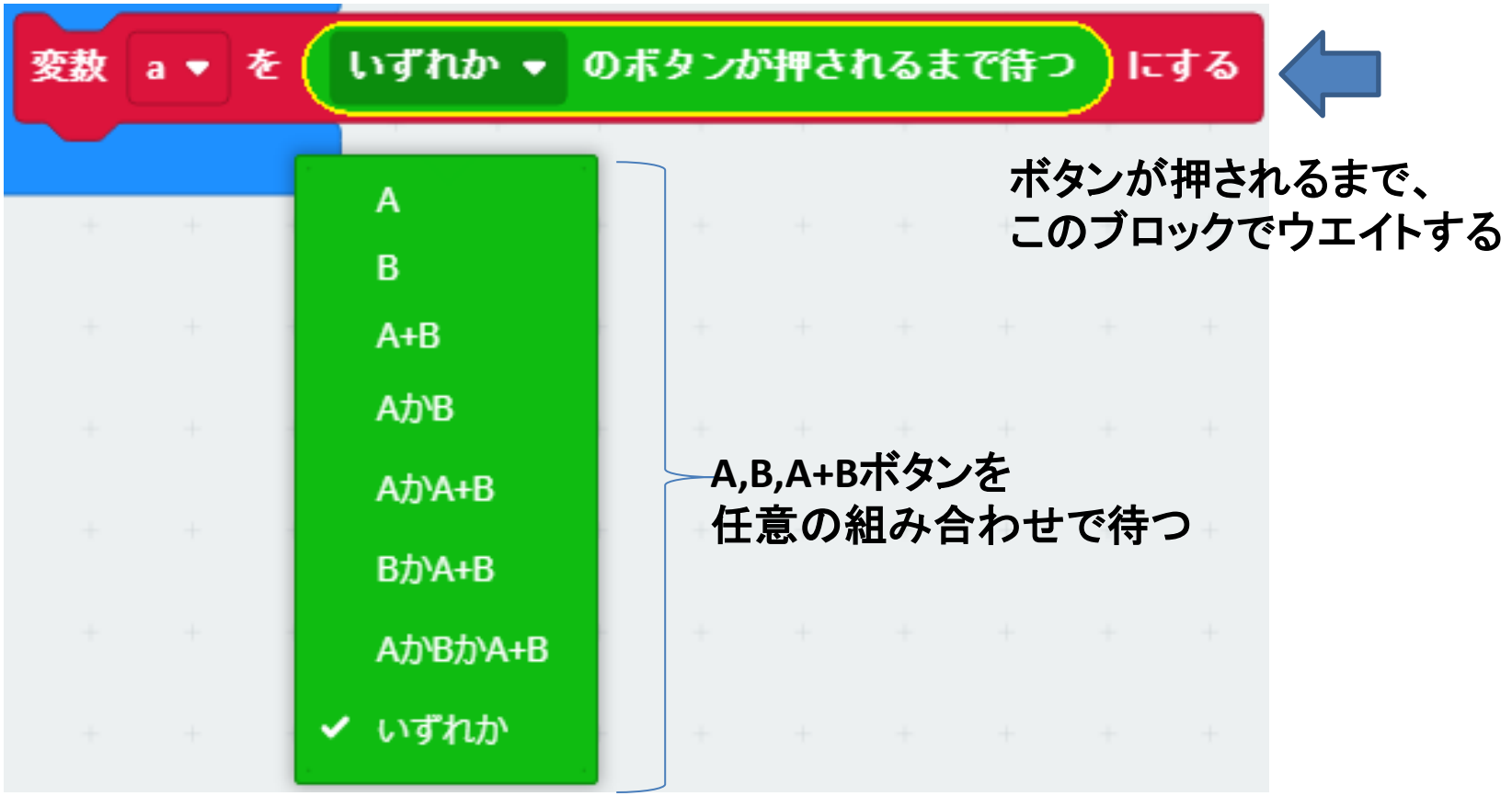
⇒

ボタンの入力待ちのための拡張ブロックを作成

MakeCodeのボタンの入力を待つブロック

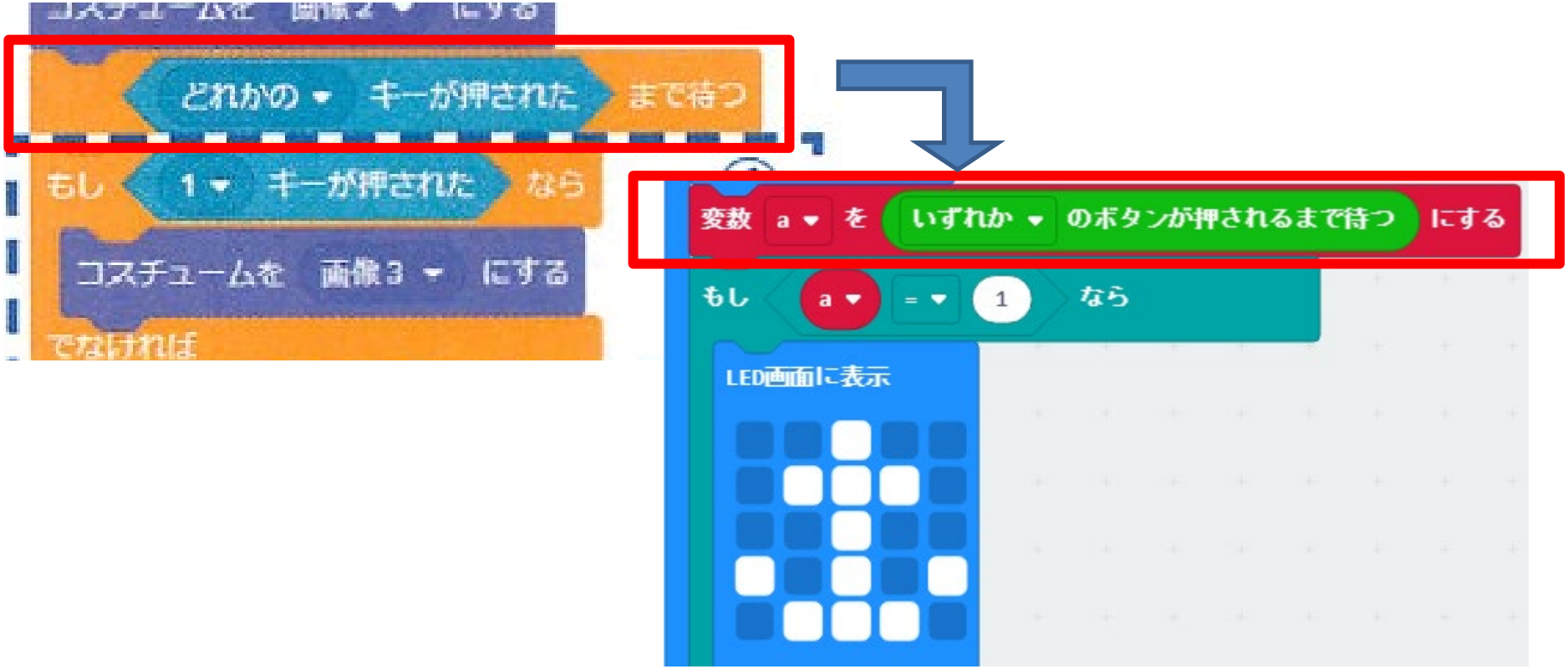
作成したボタン入力待ち拡張ブロック

- micro:bit A、Bボタンが任意のパターンで押されるまで待つブロック。

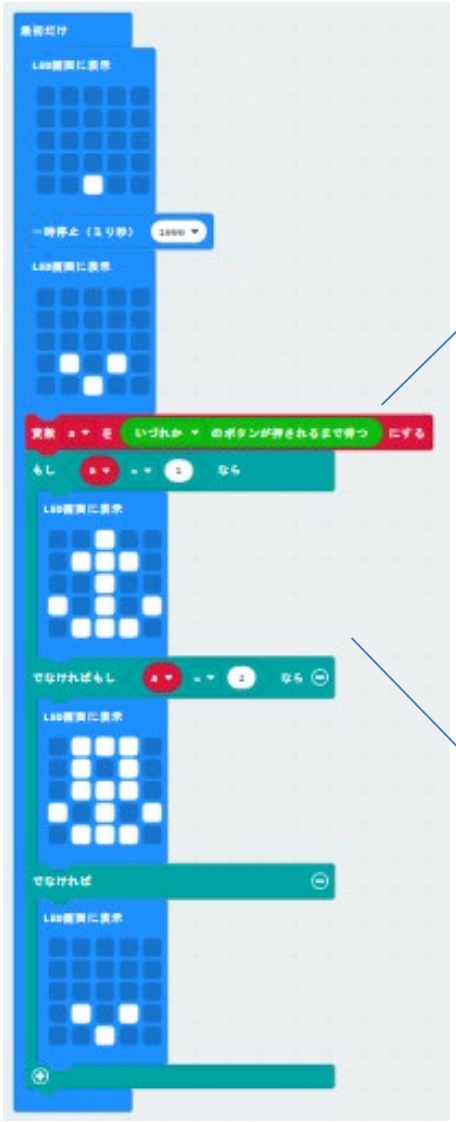


ボタン入力待ち拡張ブロックの利用

- 拡張ブロックを利用すると、先ほどのスクラッチのプログラムと同様に、逐次処理プログラムとしてmicro:bitでも実装できる



拡張ブロックを用いた逐次処理プログラム



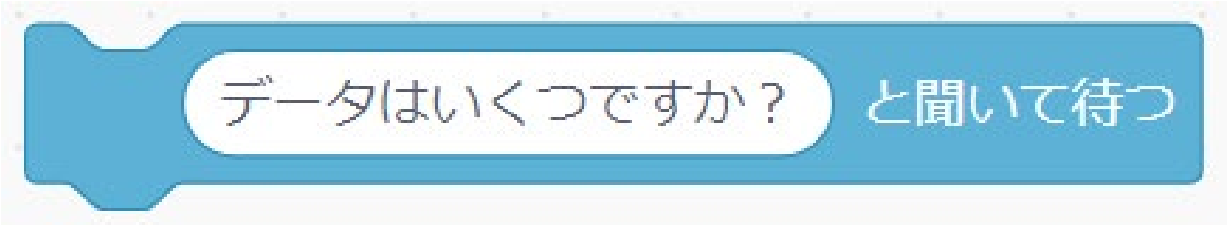
その他の作成した拡張ブロック

- スクラッチの〇〇と聞いて待つブロックの置き換え用にmicro:bitのLEDにメッセージを表示して、その後、数字を選択するブロックも作成した

仕様

- メッセージを表示し、その後、0～9までの数値が入力（選択）されるまで、このブロックで処理を継続
- 数値の入力はAボタンを押すと0、1、2・・・9と順番に選択でき、Bボタンで確定

スクラッチの〇〇と聞いて待つブロック



作成した〇〇と聞いて数字入力を待つブロック

まとめ

- 初等・中等教育の接続性を考慮したプログラミン言語の選択について、micro:bitの開発環境makeCodeエディタの有効性を紹介した
- Scratchからmicro:bitへの接続性を考慮したイベント処理ブロックの開発について紹介した

おわり

本研究はJSPS科研費JP20K02528の助成を受けています

.