サイエンスとプログラミングを合わせた小中学生向け教育の実践

サイエンス&プログラミング教室　ラッコラ

2019-03-23

株式会社スペースタイム

柳田拓人、中村景子

http://laccolla.com/

本資料について

- 本資料は、一般社団法人 情報処理学会 情報処理教育委員会 情報システム教育委員会主催による第11回情報システム教育コンテスト（ISECON2018）の本審査用資料を元に再編集されたものです。
- 本資料（柳田拓人，　中村景子，　「サイエンスとプログラミングを合わせた小中学生向け教育の実践」，ISECON2018, 2019-03-23）は、クリエイティブ・コモンズ 表示 4.0 国際 ライセンスの下に提供されています。

1. はじめに

近年注目を集める小中学生の習い事

- サイエンス教室
  - ある科学に関するテーマに沿って実験やアクティビティを行うことで、そのテーマやより科学一般について、知識や技能、考え方を学ぶ場を指す。

- プログラミング教室
  - 文字通りプログラミングの概念やプログラミング言語を学び、何らかのプログラムを成果物として制作する場を指す。

1. はじめに

ラッコラの特徴

- サイエンス教室&プログラミング教室
  - 各回の前半をサイエンス編、後半をプログラミング編と分け、同じテーマをそれぞれで関連付けて扱うカリキュラム。
  - 使用するプログラミング言語としてJavaScriptを採用した上で、初心者向けの開発環境（エディター）を独自に開発。

  - 2014年秋スタート
  - プログラミング未経験の
    小学3年生～中学生を対象

2. プログラミング教育の課題

どう動機付けるか？

「何のためにプログラミングを習得し、どんなプログラムを制作するのか」という動機付けが重要

- プログラミング教育の役割
  - 表現するための「メディア」
    - プログラミングは、自分自身が知識や概念をどのように理解しているかを「表現」するためのツールとして用いることができる。
  - 考える能力という「リテラシー」
    - プログラミングによって、プログラミングをする上で必要な「手順＝アルゴリズムを考える力」を向上させることもできる。

- 動機付けにおいても二つの役割を反映させるべき

2. プログラミング教育の課題

どの言語を使うか？

教育における学習の連続性、実際との連関を
意識させることを重視したい

- 学習に終わりはなく、学習者の意思で発展させられるべき。
- プログラミングは実学であり、学習の先には実用的なソフトウェア開発が見えているはず。それは学習者のモチベーションに繋がる。
- 単に職業としてのプログラマーを増やすことを意図しているのではない。

- 使用するプログラミング言語の要件
  - 実際のソフトウェア制作のためのプログラミングで用いられていること。
  - 現代のプログラミング言語が共通に持つ言語機能を備えていること。
- Scratchは
  - ソース・コードの記述を避けられるため、学習面でのメリットが大きい。
  - 教育に特化されており、現実のソフトウェア開発では利用されていない。

3. ラッコラの概要

動機付けへの工夫

サイエンスを学び、それを表現する手段として
プログラミングに取り組む。

- サイエンス編とプログラミング編とで同一の対象を扱い、その学びを交互に行い深化させ、プログラミング学習の二つの目的に同時に動機づける。
  - 応用対象への適用（メディア）としてのプログラミング
  - 手法（リテラシー）としてのプログラミング

- サイエンスを単に知識として身につけるのではなく、サイエンスの本質であり、かつ、深い理解に必要となる「考える力」の習得に繋げることができる。

3.ラッコラの概要

プログラミング言語の選択

- JavaScriptの採用
  - インターネット上を含めた様々な分野で使用されている、実用的なテキスト・ベースのプログラミング言語
  - このような実用言語では、あるプログラムを別のプログラムからライブラリ（プログラムの部品の集まり）として再利用することが可能
  - それにより、複数のプログラムを組み合わせて複雑な結果を得ることが可能
- ラッコラでは
  - 参加者は自らのプログラムを組みわせ、全体として複雑なプログラムを制作
  - カリキュラム内で使用する各種ライブラリも同じ言語で開発
    - 参加者は、自らのプログラムと同様に、中を「覗いて見る」ことが可能
  - 参加者の使用するアプリも、JavaScriptで開発
    - 参加者にとって、学んでいる言語が実用的なものである証拠

3.ラッコラの概要（カリキュラムの構成とテーマ）

カリキュラム構成

- 1コース全5回
  - 各回150分
    - 前半60分サイエンス編
    - 後半90分プログラミング編
  - 毎回16ページ程度の
    ワークシートを配布
- テーマごとの5コース
  - 初級（3コース）
    - 初学者向け
  - 中級（2コース）
    - 初級を終えた参加者向け

初級 星びとコース（地学・天文学）

普段とは異なる視点で宇宙をイメージし、星の自転と公転が生み出す現象に迫る。プログラミングで天体の見え方や動きを表現する。

初級 葉っぱコース（植物学）

木の葉に注目し、形や色、葉の付き方の特徴を見つけ、紅葉の仕組みを実験で探る。プログラミングでその特徴や仕組みを表現する。

初級 雪そらコース（気象学）

雪の形から、雪ができる過程、雪ができる場所などを、実験などを通して探る。その自然現象をプログラミングによって表現する。

中級 生き物戦略コース

生き物の生存のために共通する方法「捕食、成長、繁殖、防衛」を探り、プログラミングで作った生き物でそれらを確かめ表現する。

中級 感じる生命コース

生き物が環境からの刺激に反応する仕組みを、様々なレベルを通して捉え、プログラミングで自分独自の微生物として表現する。

3.ラッコラの概要（コースの内容）

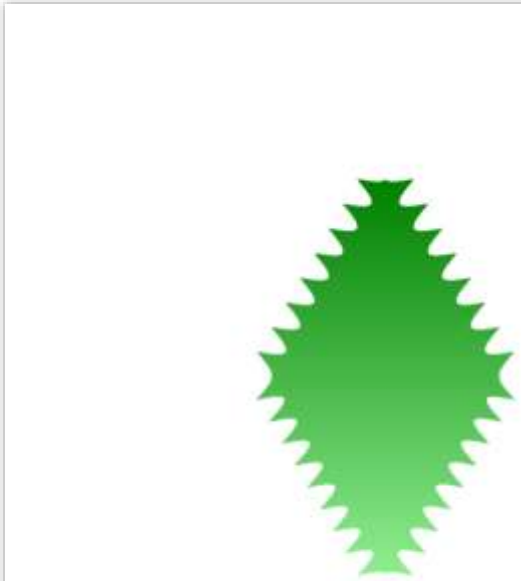
葉っぱコース

自分の分類 図鑑の分類表を使った分類 広葉と針葉	**第1回　葉っぱの分類学（仲間分け）** S: 分類学を体験しよう。 P: 順番になぞって葉っぱの形をかこう。	形を描く、色を塗る等の描画 タートル・グラフィック基礎 スタイル各種、逐次処理
葉身、葉脈などの外見的特徴 葉脈の役割 葉脈標本づくり、色水の実験	**第2回　葉っぱの形態学（形と役割）** S: 葉っぱの特徴と役割を覚えよう。 P: 線をくり返して葉っぱに葉脈をかこう。	線を描く、ライブラリの導入 タートル移動、エディタの基本 繰り返し処理
紅葉時の色素の変化 薄層クロマトグラフィー 紅葉と気温の関係	**第3回　葉っぱの生理学（色の仕組み）** S: 葉っぱが緑色の理由、紅葉する仕組みを知ろう。 P: 場合に合わせて葉っぱの色を変えよう。	色の変化で色が変わる 温度計ウィジェットの使用 条件分岐
光合成を効率良く行えるか 伸長成長、肥大成長 碁石によるアクティビティ	**第4回　木の発生学（形とルール）** S: 木の形が作られるルールを考えよう。 P: 自分で自分を呼び出して木をかこう。	関数呼び出し マウス・インタラクション 再帰処理
参加者自らが問題作成	**第5回　森の品評会（発表会）** S: サイエンス・クイズ大会をしよう。 P: 今までのプログラムをまとめて森を作ろう。	まとめプログラム（森） スプライト 各回のプログラムを発表

3.ラッコラの概要（コースの内容）

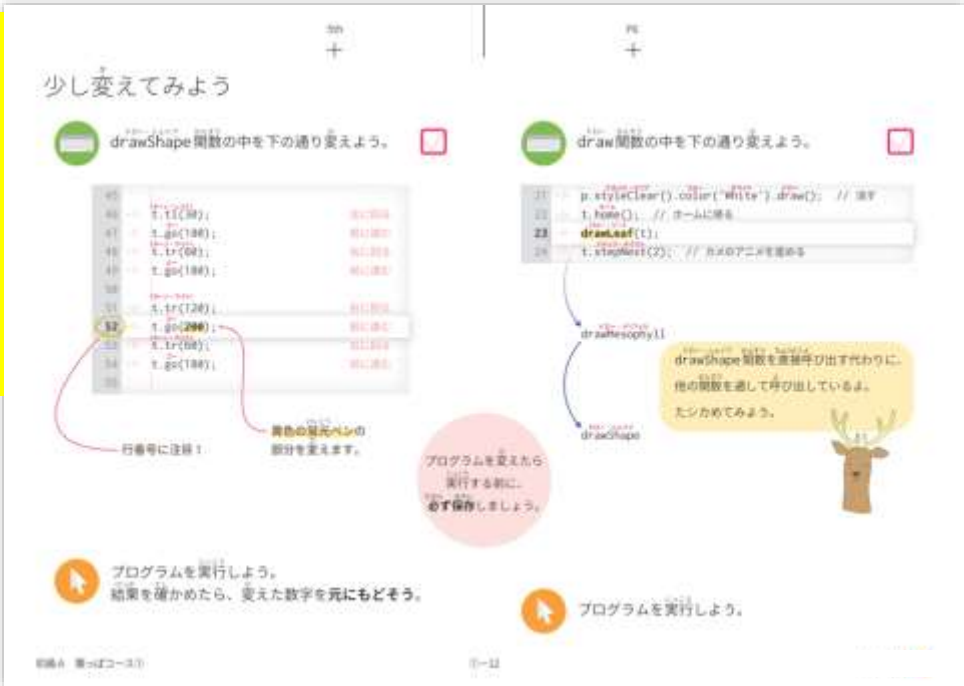
葉っぱコース①

サイエンス編の
ワークシート



プログラムの実行結果

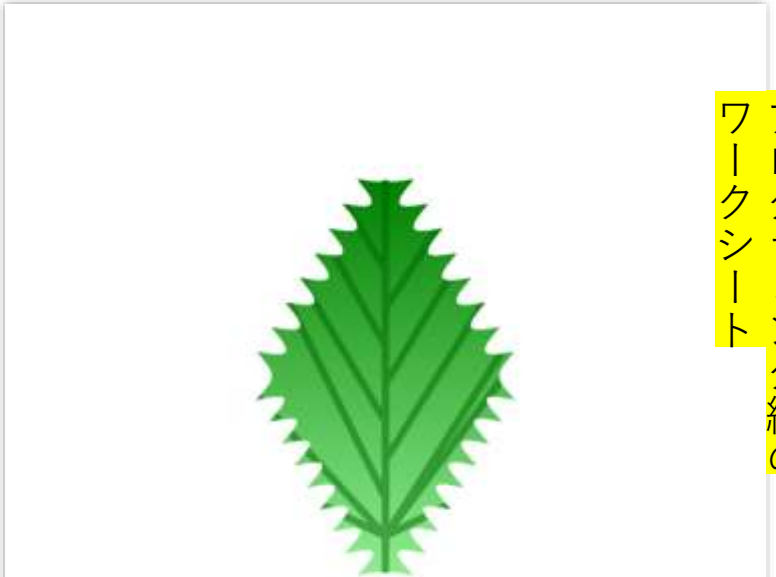
プログラミング編の
ワークシート



3.ラッコラの概要（コースの内容）

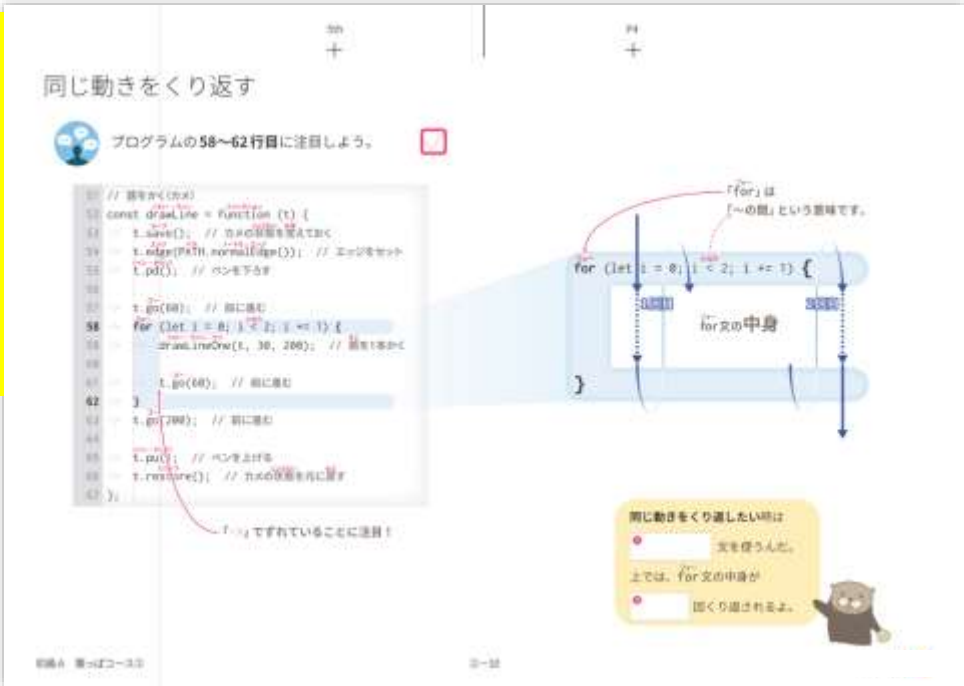
葉っぱコース②

サイエンス編のワークシート



プログラムの実行結果

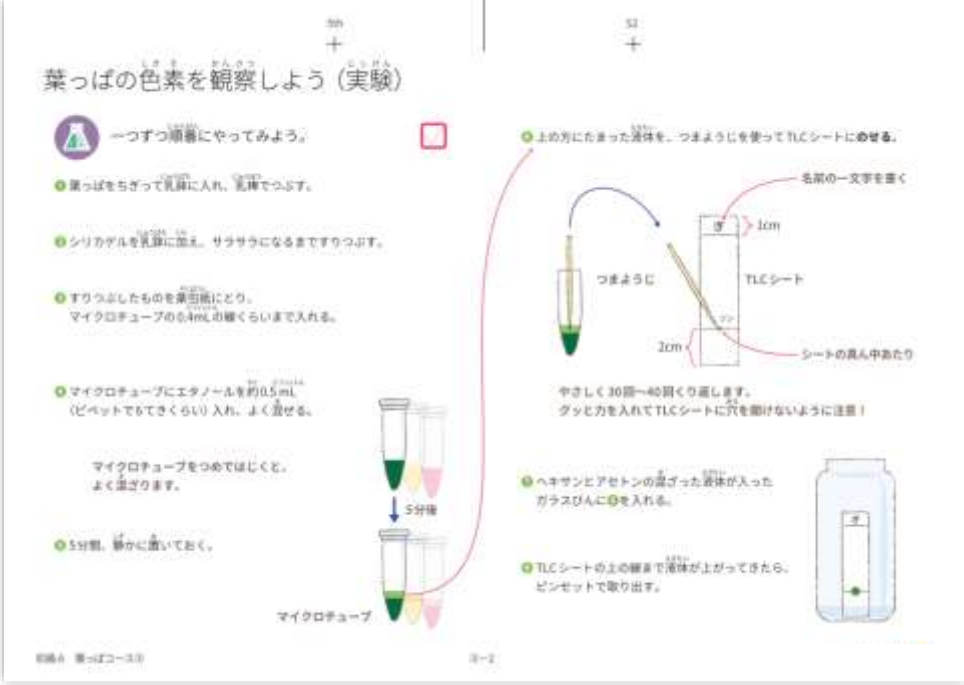
プログラミング編のワークシート



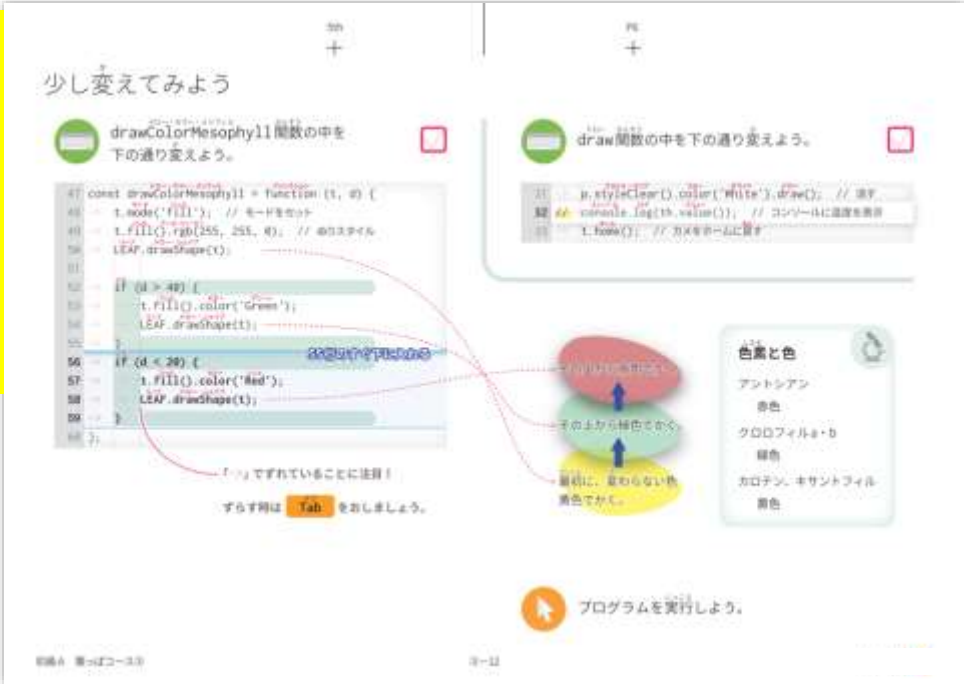
3.ラッコラの概要（コースの内容）

葉っぱコース③

サイエンス編のワークシート



プログラミング編のワークシート



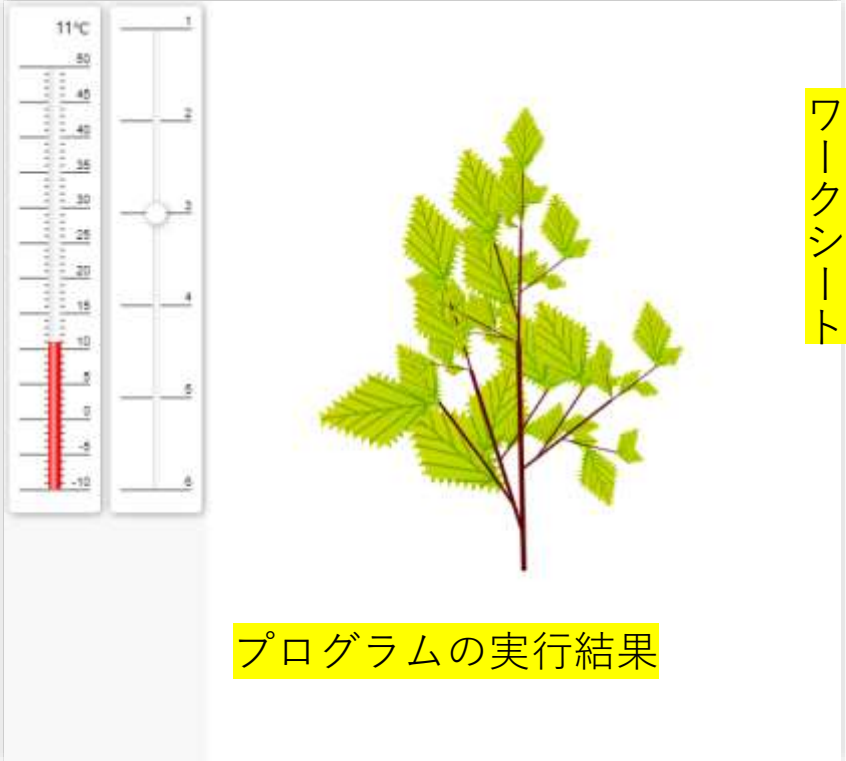
プログラムの実行結果

3.ラッコラの概要（コースの内容）

葉っぱコース④

サイエンス編の
ワークシート

プログラミング編の
ワークシート



プログラムの実行結果

既存の開発環境の問題点

- プログラミングに必要なアプリ
  - プログラムを入力・編集するためのアプリケーション、開発環境（エディター）が必要となる。
  - JavaScript によって書かれたプログラムは、通常ウェブ・ブラウザ上にて実行され、その際には、HTML ファイルや CSS ファイルが合わせて必要となる。

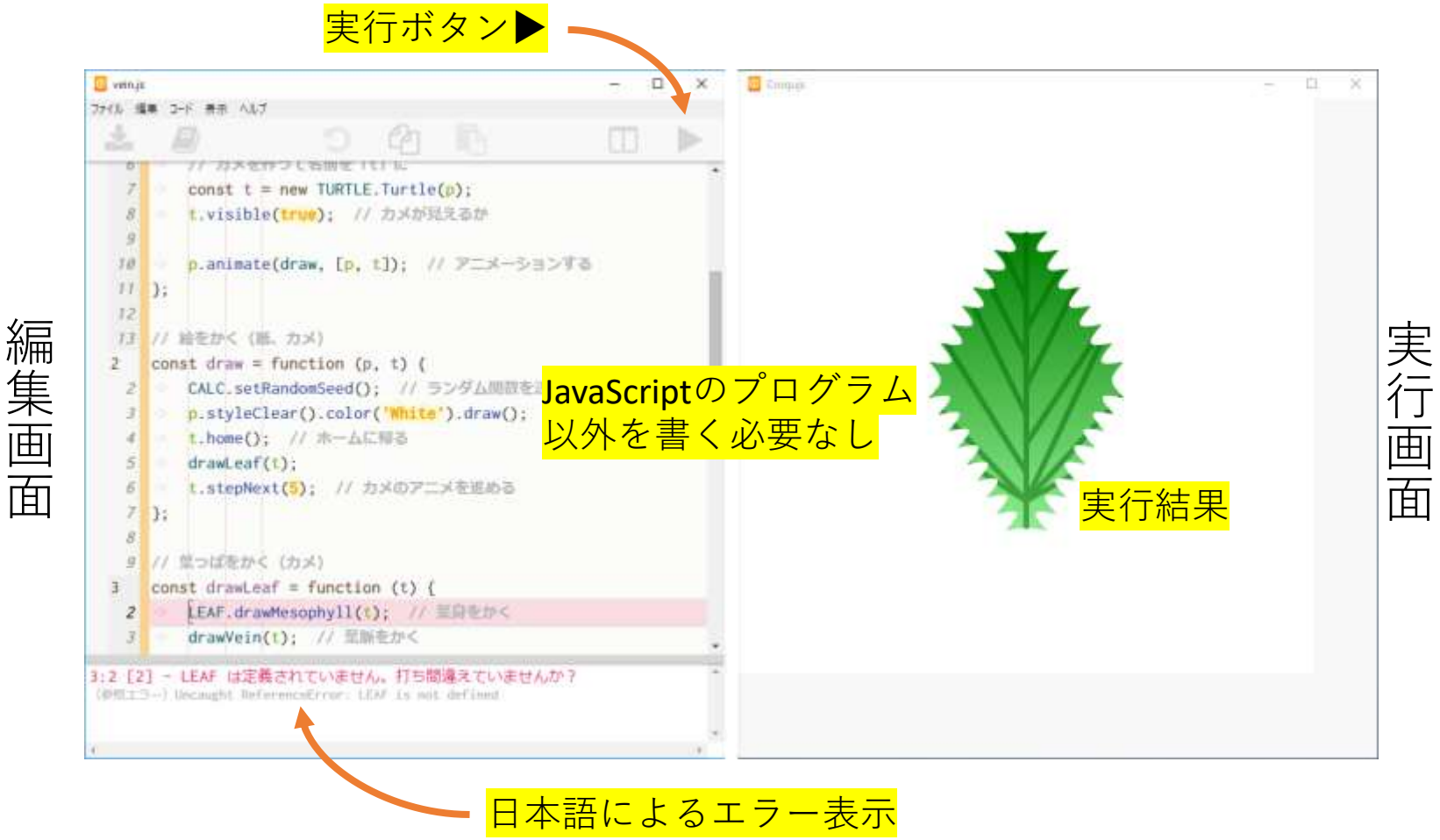
- 既存の開発環境の問題点
  - HTML ファイルなど別に用意する必要があり、プログラムを入力するだけですぐに実行してみるということが難しかった。
  - エディタ画面の表示を変更して、参加者が使用するワークシートの紙面と、見た目を一致させることが難しかった。
  - 学習時に不必要な機能を含むため、利用者が戸惑う恐れがあった。
  - 編集時のエラー・チェック表示やプログラム実行時に発生するエラー・メッセージが英語で表示されることがあった。

独自のプログラミング環境

- 開発環境 Croqujs（クロッキー）
  - プログラムを入力・編集し、ボタンを押すだけで、内部的に生成されたHTML を通じて、別のウィンドウに組み込まれたウェブ・ブラウザでプログラムを実行できる。
  - プログラムを「ライブラリ」として保存する機能と編集中のプログラムにライブラリを導入する機能を提供する。
  - 著者らは、Croqujs自体をJavaScript（および HTML, CSS）によって実装した。

3.ラッコラの概要（プログラミング環境の開発）

プログラミング環境の画面



4. 考察

サイエンスとプログラミングの組み合わせ

効果を定量的に求めるのは困難だが、
参加者の興味の対象が分かれる場合もモチベーションが持続

- サイエンスなしで行った試験的実施では……

  ラッコラの前に、試験的にサイエンスの要素を取り入れず、代わりに「絵を描く」こと自体をテーマとしたコースを実施したが、参加者の興味や関心は、プログラミングという行為自体にとどまっていたように思われた。

- ラッコラでは……

  サイエンスの方が好き、あるいはプログラミングの方が好きと、参加者の興味の対象が分かれる場合でも、カリキュラム上、相互に関連があることによって、モチベーションが持続する様子が見られた。

4. 考察

テキスト・ベース言語の是非

- JavaScriptのメリット

  参加者が直接編集するプログラム以外も同じ言語で制作したことの結果として、中級コースにおいて、参加者が著者らによって制作されたライブラリを見たり、あるいは編集してみたりする場面が見られた。

- JavaScriptのデメリット

  プログラムの打ち間違いによるエラーの発生とその修正の手間がかかる点である。エラー・メッセージを日本語で表示するなどの対応をしているが、参加者が自力でエラーを修正できない場面が多々見られるため、不十分であると考えている。

4. 考察

キーボード入力

- キーボード入力は……
  - ラッコラは対象者を小学 3 年生以上としており、小学校で英語教育がなされるようになった現在においても対象者の多くはアルファベットの読み書きが容易ではない。
  - そもそも入力デバイスとしてのキーボードに不慣れである。

プログラミング教育において、キーボードは、
大きな障壁ではない。

- 多くの参加者は、1コースを修了するまでの間に、アルファベットのキーボード入力をある程度習得する。

5. おわりに

- サイエンス&プログラミング教室
  - 小中学生を対象
  - サイエンス教室とプログラミング教室を組み合わせたカリキュラム
  - 独自の開発環境でJavaScriptを採用

- 今後の課題
  - エラー情報の収集・解析によるエラーの自動的修正
  - より初心者にとって分かりやすいエラー表示等
  - 新たなカリキュラムの開発