

# 学び方を学び問題解決能力を育成するプログラミング学習の実践

水野 修治

愛知県立岡崎商業高等学校

jy222@aichi-c.ed.jp

初等中等教育においてプログラミング教育の重要性が高まる中、指導内容や使用するプログラム言語、教材の確保や指導者の育成など、さまざまな課題が指摘されている。そのような中、専門高校の課題研究で取り組んだ、HTML の基礎からスマホアプリの制作までの体系的なプログラミングの学習をとおして、問題解決や探究活動に主体的、創造的、協働的に取り組ませる授業実践を報告する。

## 1. はじめに

工業や商業高校などの専門高校では従来からプログラミング教育が実施されてきた。しかし、学校の特質上、国家試験への対応などに偏重する取組も見られ、残念ながら問題を解くことに目的が置かれるケースも見られる。

このような中、次期学習指導要領では小中高におけるプログラミング教育の在りようが大きな課題となっている。このプログラミング教育が小中高において継続的に取り扱われれば、論理的な思考力や、順序だてて問題解決する能力、さらにはモデル化や抽象化する能力を育成するばかりか、自分が考えたアイデアを具現化するという創造性も培うことができると考える。

## 2. プログラミング学習の概要

### 2.1 プログラミング学習のポイント

自ら課題を見出し追究する問題解決能力は、これからの現代社会を生き抜く児童生徒において必要な資質能力である。このような力を育むために、次のようなポイントに重点を置いたプログラミングの学習に取り組んだ。

- ・学び方を教えることを徹底する。
- ・まず自分で考え、解決しなければ自分で調べさせる。どうしても自己解決できなければ他の生徒に尋ねさせる。
- ・自分で工夫させ、アイデアを具現化させることによって内発的動機づけを引き起こす。
- ・効率及び優先順位の意識を高めるために時間の制限を設ける。
- ・学んだ内容を次に生かすために、自分で利用しやすいように整理させる。
- ・生徒のコードを可能な限り尊重する。

### 2.2 体系的なプログラミング学習

この授業の取組では、HTML の基礎からスマホアプリの開発まで、学んだ内容が次に生かせるような図 1 のような体系的なプログラミングの学習を行った。また、各フェーズにおいて問題解決が必要とされるようトラップを仕掛け、自己解決しな

がら段階的に学んでいけるように工夫を凝らした。

課題アプリは、仕様の説明が不要であるとともに、機能追加においてアイデアが出しやすい身近な「電卓」とした。

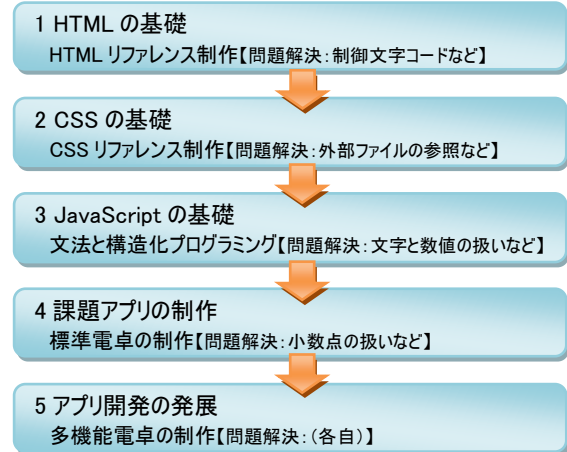


図 1 プログラミング学習の流れ

### 2.3 開発環境と教材

このプログラミング学習では、最終的な制作物を生徒の興味・関心を高めるためにスマホアプリとした。プログラム言語は、iPhone や Android など各々が持つスマホでも稼働できるハイブリッドアプリとするため HTML5 を利用することにした。また、開発環境には、インストール等の負担や生徒が自宅等でも継続的に学ぶ環境が構築できることを重視しクラウド型アプリ開発環境の Monaca を利用した。

教材は、授業展開や様々な問題解決に対応するとともに、各自の興味・関心や技能に合う教材を見出す能力を育むために、Web 上に存在する膨大に技術情報を利用した。

## 3. 授業の展開と問題解決(トラップ)

### 3.1 HTML の基礎(HTML リファレンス制作)

HTML の基本的な構造やタグの記述等を学んだ後、各自が利用しやすい HTML リファレンスを HTML で作成させる。(メモ帳を利用)

### 【問題解決(トラップ)の一例】

HTML リファレンスでタグの説明を記述する場合、「 $\lt$ 」や「 $\gt$ 」の山括弧をどうやって表すか。  
⇒特殊文字コードを利用 (自己解決率 100%)

### 3.2 CSS の基礎(CSS リファレンス制作)

style 属性や $\lt$ style $\gt$ 、id と class によるスタイル指定など基本的な記述の方法を学んだ後、各自がアプリ開発時に利用する CSS リファレンスを HTML+CSS で制作させる。(メモ帳を利用)。

### 【問題解決(トラップ)の一例】

統一感あるデザインの Web ページを複数作る場合、効率良く CSS の記述をする方法はないか。  
⇒外部ファイルの参照 (自己解決率 75%)

### 3.3 JavaScript の基礎

JavaScript の基本的な文法を説明した後、form のテキストボックスに入力した数字の和を求めるプログラムを制作させる。

### 【問題解決(トラップ)の一例】

図 2 のような結果になってしまうのはなぜか。



図 2

⇒eval 関数で文字を数字に変換 (自己解決率 75%)

### 3.4 課題アプリ(標準電卓の制作 1)

図 3 のような 0 から 9 までのボタンを使い 1 桁どうしの和を求めるプログラムを作成し、ユーザー定義関数の使い方を学ばせる。

### 【問題解決(トラップ)の一例】

1024+64 のように複数桁どうしの計算ができるようにするにはどうすればよいか。

⇒数字のボタンが連続して押されたら変数に格納されている数値を 10 倍して足す (自己解決率 63%)

### 3.5 課題アプリ(標準電卓の制作 2)

図 4 のように複数桁どうしの四則演算ができるようにする。続いて小数点の数値計算に対応するプログラムに取り組ませる。

### 【問題解決(トラップ)の一例】

「0.3」が「0.300000000000000004」と表示されてしまうがどうすれば良いか。

⇒IEEE754 浮動小数点数への対応 (自己解決率 25%)

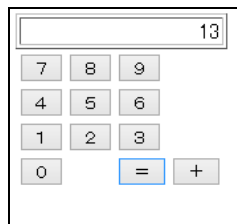


図 3

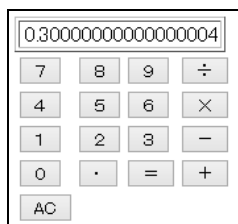


図 4

### 3.6 アプリ開発の発展(多機能電卓の制作)

標準電卓を制作した後は、生徒の興味・関心に応じた多機能電卓に取り組ませる。実際の電卓をイメージしながら機能を追加したり、アプリならではのメモリ機能や履歴機能を実装する作品が見られた。

(図 5)

この取組を通して自ら考えた追加機能の実装において、さまざまな問題に直面するが、自己解決に向けて精力的に取り組む姿勢が見られた。



図 5 生徒が制作した多機能電卓の例

### 4. 今後の課題

HTML5 ハイブリッドアプリを最終制作物とするこの系統的なプログラミング学習の流れは、生徒の興味・関心が高いばかりか、派生的にさまざまな Web 技術を学ぶことができる。しかし、今回題材とした「電卓」アプリの制作においては、構造化プログラミングの中の「反復処理」を利用する機会がなく、プログラミングの真骨頂をアプリ開発において体験できていない。また、自己解決率が低い問題解決において、内発的動機づけを維持しながら取り組ませる適切なコードの提供とアドバイスの仕方などについても工夫が必要であると考えられる。さらには、生徒のコードを尊重する程度についても検討が必要である。

### 5. おわりに

これまでプログラミングを行うには高度な知識と技術が必要であったが、開発ツールの発展などによりその敷居はかなり低くなっている。プログラミングを学ぶことによって、自らが必要とするプログラムを DIY のような感覚で工夫し作り上げることも可能となってくる。プログラミングの知識・技術の有無によって、その人の可能性が大きく左右される、そんな時代がもう現実となってきたと考えている。

#### 参考文献

- (1) 文部科学省：高等学校学習指導要領解説情報編，2010.
- (2) Monaca：https://ja.monaca.io/.