

## 普通科高校での Web プログラミング

愛知県立安城南高等学校，授業者：田中 健（現：愛知県立瀬戸北総合高等学校）

校種：高等学校（普通科 情報活用コース）， 実施教科：情報

実施単元：コンピュータによる処理手順の工夫（情報 B）， 課題研究， 実施学年：2， 3 年生

### 使用したプログラミング言語や実行環境

- HTML/CSS, JavaScript.
- テキストエディタ（生徒はメモ帳を使用。また，有用なプログラミング用エディタとして，EmEditor を教員機にインストールし授業内で紹介）。
- Web ブラウザ（Internet Explorer 9（2012～13 年））。

### 教室の設備・環境

- **実施場所**：第 2 コンピュータ室
  - ごく一般的なコンピュータ室が備える環境と大差なく，Windows 搭載教員機 1 台，生徒機 40 台，Windows Server 2008 R2 搭載サーバ，スクリーン，プロジェクタを備えた環境。
  - 各生徒には個別ログインアカウントと 1GB 程度のファイルサーバ領域を割り振り。
  - 3 年次の Web プログラミングに関しては他の生徒の制作状況を確認可能。
  - 教室管理の問題上，サーバサイドプログラムの使用は不許可。

2012 年から 2013 年にかけてこの授業を実施した高等学校は，授業者の前任校である愛知県立安城南高等学校であり，当校には普通科でありながら 3 年間で 14 単位（普通教科 2 単位，専門教科 12 単位）情報の科目を履修する情報活用コースが設定されている。文系科目が多く取り入れられたカリキュラムを展開する情報活用コース所属の生徒の進路希望先は，文系国公立大学進学から就職まで多岐にわたっており，各生徒が獲得することを望む情報活用能力にも差異がある。

そこで，日常的に目にする Web サイトがどのような仕組みで作られているかということの知識と理解を，情報活用コースでの学習を修了した生徒が最低限持ち合わせておくべきプログラミングの知識と捉え，2 年次の普通教科科目「情報 B」で HTML でのマークアップの基礎と JavaScript による簡単な動的処理を学び，3 年次の専門教科科目「課題研究」で HTML/CSS と JavaScript を工夫して組み合わせ，各自がクラスメートに紹介したいものについての Web サイトを作成し，最後に相互評価させた。

本稿では，特に 2 年次の「情報 B」での取り組みについて触れる。

### 授業の様子

当该校では情報の授業をすべて Team Teaching で行っており，教壇で解説をする教員と机間指導を行う教員に役割を分けることで，生徒の初期段階でのつまづきを防ぐ効果があり，エラー対応における授業の中断は比較的少ない。得手とする生徒が他の生徒を教えるという姿も見られる。

なお，必要なものは教科書・教員作成のハンドアウト・Windows 付属のソフトウェアであり，特殊な機材は一切使用しないため，自宅に持ち帰っての復習も可能である。



実行結果を周囲の生徒と確認する様子



全体講義の様子



指示に従って配列をつくる様子

全体指導計画（情報 B：全体 6 時間計画，課題研究：全体 30 時間計画）

●2 年次「情報 B」

1. マークアップの基礎 …… (1 時間) 2. インタプリタ処理の理解と繰り返し処理… (2 時間)  
3. 配列の処理【本時 2/2】 … (2 時間) 4. まとめ… (1 時間)

●3 年次「課題研究」

1. 年間計画の提示と 2 年次の復習 (2 時間) 2. HTML/CSS の演習 …… (10 時間)  
3. JavaScript の演習 … (8 時間) 4. 課題制作… (8 時間) 5. 作品の相互評価 … (2 時間)

授業の概要

**授業の目標：**繰り返し処理と配列を利用して，買い物の税込合計金額を返すプログラムを作ることができる。

評価規準

- A・・・JavaScript を利用した繰り返し処理と配列の処理方法を理解し，自身で創造的に目的のプログラムを作成することができる。  
B・・・JavaScript を利用した繰り返し処理と配列の処理方法を理解し，他の生徒との相談や助言者からのアドバイスを通じて目的のプログラムを作成することができる。  
C と判断した生徒への手立て・・・前回までの授業内容におけるつまづきの部分を見つけさせる。また，完成時のプログラムと当該生徒の作成したプログラムを対比させ，どのように処理が違ってくるのかを理解させる。

指導過程：（3. 配列の処理【本時 2/2】）

	教員の働きかけ	予想される生徒の反応	指導上の留意点
導入 (5 分)	1.日常生活で繰り返し処理と配列処理が使われる場面にはどのようなものがあるか考えさせる。  【導入場面のここがポイント】 プログラムが日常生活にいかに関わっているかを，毎時間説くことで，生徒が持つプログラミングへの精神的なハードルが下がる。難しそうという意識を取り除くことが重要である。	1.日常生活の中から考える。 ○：図書館で本を借りる時の貸出手続 ×：縄跳び（「繰り返し」に着目？） →配列の概念を理解するのが難しいようである。	1.コンビニやスーパーでのレジ打ちの場面を想起させる。 ・買い物カゴに商品が入っている間は 1 つずつカゴから取り出し，バーコードを読み取って袋へ入れていく ・全て読み終えたら合計金額を表示させ，客に知らせる という流れを確認させる。
展開 (40 分)	2.本時の課題を提示する。  <b>配列を使って計算を行い，結果を表示させてみよう</b>  3.これまでに作成したプログラム（基本構造）をスクリーンに表示しておき，机間指導を行って生徒作成プログラム中の細かなエラーを発見，排除させる。  4.完成したプログラムを実行させ，完成作品を自身のディスクスペースに提出させる。提出後は未完成者の助言に当たらせる。  【展開場面のここがポイント】 課題を見た途端にやる気をなくす生徒が多く出ると，その雰囲気打破するのがとても難しいので，課題設定と出し方には細心の注意を払う必要がある。時間ごとに課題作成に係るヒントを小出しにしていくことで，スムーズにプログラム修正が進む。	2.本時の課題を提示する。  3.教室前方スクリーンに表示したプログラムの基本構造やハンドアウトを適宜使用し，プログラミングを行う。  4.プログラムを実行し，期待される結果が返ってきたことを確認の後，提出する。課題完成後は机間指導の補助を行う。	2.プログラム本体ではなく実行結果を見せ，最終的に合計金額が返るようになることが本時の目標であることを示す。  3.机間指導を行う場合にはすぐに解答を示すのではなく，「箇所を指摘するにとどめる」「周りの生徒のプログラムと見比べさせる」など，間違いの深さによって指導を変えると良い。
まとめ (5 分)	5.プログラム作成で困った部分，今後活かそうな気づいた部分について，テキストエディタに記入し自身のディスクスペースに提出させる。  【まとめ場面のここがポイント】 今や SNS やテキストメッセージアプリ全盛であるが，構築されたウェブサイトの根幹を成すのが Web プログラミングである。本授業で学んだことを基礎部分として多くのウェブサイトが作られ各自が利用していること，学習し続けることで自身にもインターネットで公開できるものが作れるかもしれないという期待感を持たせること，また，次の時間のまとめにおいて，サーバサイドプログラムについても触れることを予告することで，興味を喚起できる。	5. 本時の授業で得たことや感想を簡潔に記入する。	

## 生徒の作品例

## どのようなプログラムか

HTML/CSS と JavaScript を利用し、他者に紹介したいものや事柄についての Web サイトを作成することを課題とした。下図が生徒の作品例である。この生徒はある国立大学の情報学群に進学した生徒であり、トップページから読み込む外部スクリプトを用意し画面上に雪を降らせたり、ウインドウ制御を駆使したフォトアルバムを設置したりするなど、好評を博する作品を制作した。

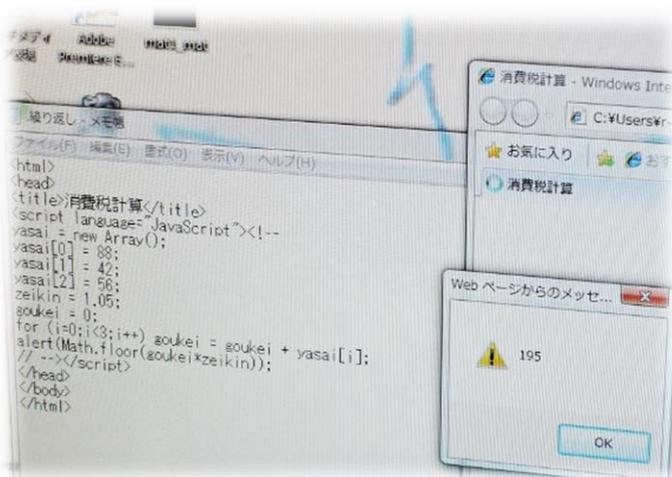
他方、当該課題は自由裁量の制作であるため、プログラミングの得手不得手が如実に現れた。JavaScript を使いこなせず、HTML/CSS のみのウェブサイトで作成が止まってしまう生徒も少なからずおり、「prompt()」で呼び出されるテキスト入力用ダイアログボックスを用いて名前を入力し、「ようこそ、〇〇さん」と表示させるプログラムを組み込むことを最低限の条件とした。



## どのように動くか

JavaScript を学んで 5 時間目となる本時の課題の模範と実行結果は下図・写真のとおりである。

- 商品（野菜）を 3 点購入する時の税抜金額をそれぞれ事前に配列 yasai[0], yasai[1], yasai[2] に格納する。
- 消費税率を定め（当時は 5%）、税抜金額を合算した後に税金額を計算させるために、zeikin に格納するデータを 1.05 とする。
- 合計金額を示す goukei の初期値を 0 とする。
- yasai[0] のデータを goukei に加え、yasai[1] とする。同じく、yasai[1], yasai[2] のデータを goukei に加え、繰り返し処理を終える。
- goukei に入っているデータに 1.05 を掛ける。
- alert で呼び出される警告ダイアログ上に実行結果である「195」を表示する。



模範プログラムと実行結果

```
<html>
<head>
<title>消費税計算</title>
<script language="JavaScript"><!--
yasai = new Array();
yasai[0] = 88;
yasai[1] = 42;
yasai[2] = 56;
zeikin = 1.05;
goukei = 0;
for (i=0;i<3;i++) goukei = goukei + yasai[i];
alert(Math.floor(goukei*zeikin));
// --></script>
</head>
</html>
```

## 生徒の変容や生徒の感想より

### 生徒の変容

当初はよくわからない文字列の羅列を前にプログラミングに対して構えがちではあったが、プログラムを作成し、更新・実行する作業を重ねることで「よくわからない文字列」が「自分の思ったとおりに動かせるもの」へと意識が変わると、理解が格段に速くなり、深まっていくことが見受けられた。初期段階では何よりもプログラミングへの垣根を取り払う工夫が肝要である。

座学での学習が苦手な生徒がプログラミングには才能を発揮する例も見られた。この場合、普段は教えられる立場の生徒が他の生徒より先んじてプログラムを完成させ、他の生徒を教える側に回るといった逆転現象が発生することになり、当該生徒が学習活動に自信を持つきっかけにもなり得る。また、生徒同士のコミュニケーションを活発化させる効果も期待できると推察される。

### 生徒の感想

プログラミングに対して前向きな意見から、どうしてもやる気になれないという意見まで多様な感想が集まったが、経験できたことは良かったとする意見が大半を占めた。また、普段から進路意識を高めることを筆者から説かれているクラスの生徒のため、プログラミングを行う職業についての感想も多く見られた。

- プログラミングをすると聞いて、最初は難しそうだったと思うが、案外簡単にできておもしろかった。
- ソフトを買う必要がないなら、自宅でもやってみたい。
- 自分でアプリを作ってみたい。
- 友人の力を借りて完成はしたが、やはり自力ではできそうにない。
- どうしてもエラーが出てくるので途中でイヤになった。
- プログラマという職業に興味を湧いた。しかし1日中プログラミングをするのは大変そう。

## この授業のお勧めポイントと実施上の留意点

### 題材のおすすめポイントと留意点

本稿では、教科書に掲載されているプログラム例を元に、誰もが利用できる環境を用いて、課題プログラムを制作した。授業者が知るところによれば、総じて、教える側にも教えられる側にもプログラミングに対する畏怖や、定期考査で扱いづらいという忌避が多く、多くの学校で見受けられ、教科書に例が示されているにも拘らず、結果的に授業内で取り扱わないことが少なくないようである。

しかし、プログラミングには特殊な機材も技能も必要ない。授業者の例のように、メモ帳やWebブラウザなど、各学校に配備されている一般的なコンピュータ室の設備で実現できるものである。生徒にとっては座学だけでは得られない経験ができ、グループ活動として授業展開を行えば、学習指導要領に示された言語活動の充実につなげることも容易である。また、何よりも初めてプログラミング授業を行う教員にとっても新しい発見があることであろう。

### 指導方法のおすすめポイントと留意点

課題を設定するにあたって課題のレベルをどうするかという点には、年度をまたいで苦心を重ねた。プログラミング実習を行うにあたり1番の障壁となるのが、生徒が間違ったプログラムを作成し、実行エラーとなった場合のデバッグ対応である。処理が中断され（もしくは無事に終わり）、生徒のプログラムを作り直すことで直るものであれば良いが、繰り返し処理のプログラムを扱う場合、不等号の向きが1箇所違うだけで無限ループが始まり、コンピュータ自体が動作不能になることがある。また、ダイアログボックスを頻繁に使用して起きたエラーを修正する際には、確認作業が煩雑になって手間取り、プログラミングに対する気概を喪失してしまう生徒も見られた。何かの拍子で全角スペースが打たれている、文字が1文字違うというだけでもエラーが発生するのがプログラミングのハードルが高くなる要因であり、エラーのたびに感情的になったり、否定的な発言・態度をとったりする生徒も出ないわけではない。その雰囲気が周囲に伝播しないようにフォローする役割の補助員を置くことができると授業が円滑に進むようである。

机間指導を行う補助員の人数にもよるが、課題プログラムは易しすぎず、難しすぎず、修正する際に深刻なエラーの発生し難いものにする必要がある。エラーを起こした1人に時間が取られ、結果として教室全体に指導が行き渡らないということも見受けられる。生徒の習熟度、プログラミングへの姿勢などを考慮し、慎重に授業計画を練ることが求められる。