

JavaScript のオンライン教材サイトの構 築について

0648016 大西未夕希

目 次

1 はじめに	1
1 本教材について	1
2 対象者について	1
3 JavaScript について	2
4 これまでの教材との比較	2
2 制作物について	3
1 Python との比較について	3
2 効果の見込み	4
3 構想と規模	4
4 背景となるコンピュータ技術	5
5 具体的な資料	5
6 作業の見通しと計画	7
3 本論	8
1 教材の詳細	8
2 教材の内容と構成	9
3 ファイルの構成	12
4 制作過程について	12
5 工夫した点	14
4 まとめ	17
1 ユーザー評価	17
2 自己評価	20
3 今後の課題	20
4 紹介していないものについて	21
5 最後に	21

1 はじめに

(1) 本教材について

本教材は、オンラインで JavaScript の学習ができるように制作したソフトである。記述方法やできることなどを、Python の場合と比較しながら、JavaScript の基本的な知識を学び、応用につなげられるまでの技術を習得することを目標として制作した。他のプログラミング言語も交えながらの解説となっているが、プログラミングの初心者にも、本教材を通して JavaScript に関心を持ち、今後の学習のきっかけとなるようなソフトにしたいと考えた。

(2) 対象者について

本教材は、人文情報学科のプログラミング演習を履修している人で、特にゼミで Web アプリケーション作成を目指している人に向けて制作している。

過去のゼミ生の卒業制作として、Python の教材が存在する。日本語に書き換えられたプログラミングを正しく表記し直すという出題形式のもので、私自身も上記の教材を使用し Python を学んだ。過去に目にしたことのある出題形式に似せた形にすることで、問題の解き方やプログラミングの記述方法の理解がしやすいと考えた。また、制作者本人もプログラミング演習を履修していることもあり、似た教材で学習してきた人に向けたものを制作しようと考えたことから、本教材はプログラミング演習を履修している人の方が理解もしやすいと思われる。

さらに、HTML についての基本的な知識がある事や、何らかのプログラミングに触れたことがあるという事なども対象条件としている。よってプログラムの概念についての説明は省いている。プログラミングの内

容については、Python との比較を例に挙げているので、特に、Python を学習したことのある人が望ましいといえる。

(3) JavaScript について

JavaScript は HTML ファイルに記入することでブラウザ上で実行することが可能である。サーバにアクセスすることなく編集でき、開発ソフトなども必要ないことから、比較的容易に作成することができる。HTML や CSS は、Web ページの文書構造や、表示方法について記述するものであり、ページに動きをつけることはできない。しかし、JavaScript は、に動きを加えたり、ユーザの操作によって反応を変えたりでき、ウェブページの表現の向上を可能にした言語でもある。現在、多くの Web ブラウザに搭載されている JavaScript は、我々にとって身近なプログラミング言語といえる。以上の点に加え、プログラムの作成にあたって、特別な環境を用意しなくても良いといった手軽さも、教材の題材として選んだ理由である。

(4) これまでの教材との比較

JavaScript の初心者用マニュアルサイトは、これまでのゼミ生の制作物の題材としても既に存在している。しかし、過去のソフトの多くは、他のプログラミング言語についての知識の有無は問わず、JavaScript の基礎技術を学ぶものとなっている。過去のマニュアルサイトの問題点としては、以下が挙げられる。

- ・ JavaScript はメモ帳などのテキストエディタがあれば作成できることから、学習者はファイルの作成手順の説明から学び、個々でファイルを用意しなければならない。

- ・ 個人でファイルを用意する必要はないが、JavaScript のプログラムを作り、動かすまでの環境設定についての解説はない。

また、ウェブ上に存在するマニュアルサイトでは、プログラミングを記述し、作成した実行結果が見られるものは存在したが、解答画面が用意されていたり、問題を解く形のものは見つからなかった。よって、以上の点を改善点とし、新たなサイトの構築を考えた。

まず、本教材は、プログラミングの知識がない人も対象ではあるが、何らかのプログラミングに触れたことのある人に向けても、これまで学習してきたプログラミングの概念と見比べられるように、Python との比較を例として挙げている。さらに、サイト内にプログラミングを書き込み実行できる空間を用意するなど、これまでのソフトとの差別化を図った。

2 制作物について

(1) Python との比較について

Python は JavaScript のように、Web ページに動きを加えることに関して主に使われている言語ではないが、類似している点がいくつか見られる。まず両プログラミング共にコンパイルの必要がなく、ファイルが作成しやすいプログラミングということが共通している。また、Python には変数の宣言が必要なく、JavaScript も、必ずしも変数の宣言をする必要がない。また、演算子や制御構文など、基本的な部分で比較できる点も多く、さらに、私自身が Python を学習したことがあり、参考にしやすかったことから、例として挙げる言語として Python を選ぶことにした。

(2) 効果の見込み

解説を見ながら解ける問題や、プログラムの動きに触れることで、JavaScript の基本的な構造や知識が体感できると考える。さらに、JavaScript に興味や関心を持ち、応用技術の意欲につながることで、個人の表現の幅の広がりが見込めると考える。

(3) 構想と規模

本教材では、JavaScript の基本技術を身につけられることを目標とした。また、教材を通して、これまでにプログラミングに触れたことのない人は、HTML や CSS とは違った動きが体感でき、プログラミングに触れたことはあるが JavaScript は初めて学習するという人には、他のプログラミングと見比べながら学べるという教材を目指して制作した。

JavaScript は、現在も多くのウェブページで使われているプログラミング言語であり、2005年には、「Ajax (エイジャックス)」という技術の登場で再び注目を集めた言語でもある。Ajax の例としては、Google マップや Google サジェストが挙げられる。Google マップでは、地図をドラッグしたり、拡大や縮小をしたりなどといったことがページ遷移なしでリアルタイムに行うことができ、また、Google サジェストでは、検索語の候補をリアルタイムで表示してくれる。Ajax は、JavaScript を中核とする技術を組み合わせた Web ページの新しいアプローチだといえる。しかし、新しい技術ということで、JavaScript に比べて資料が少なく、さらに、Ajax の実行には Web サーバが必要である。以上の点に加え、本教材は、プログラミングの初心者も対象にしたことから、Ajax の紹介は省き、JavaScript のみの学習を目的と

して制作しようと考えた。

(4) 背景となるコンピュータ技術

本マニュアルは、オンライン上で利用できる教材として制作し、ウェブページの形式で作成した。サイトのデザインや外観の制作にあたっては、主に HTML と CSS が使用されている。マニュアルの制作にあたっては Python と JavaScript が使われており、マニュアルの題材である JavaScript にいたっては、デザインやコンテンツに動きを加えるため、サイトの外観部分とマニュアル部分の両方に使用されている。また、マニュアルには、HTML や Python の内容も含まれているので、利用者には、HTML や、プログラミングの知識があるということを前提としている。

(5) 具体的な資料

主な資料としては、『JavaScript プログラミング入門』（大津 真著、オーム社出版、2006）を利用した。本書は歴史的な背景や概要など、JavaScript についての詳細と、プログラミングの基礎から実践までが丁寧に説明された初心者向けの解説書である。また、JavaScript の特徴や使い道に加え、ブラウザに合わせたそれぞれの設定の仕方も詳しく説明されている。教材を制作するにあたり、自身も基礎から JavaScript を学ぶ必要があったことや、利用者に向けての解説の資料としたかったことなどから、本書を参考にした。

学習の参考だけでなく、Python との比較部分である演算子や用語の確認、出題部分には Web ページも資料とした。参考とした Web ページの詳細は以下の通りである。

(i) JavaScript 例文辞典(<http://www.openspc2.org/reibun/javascript/>)

本サイトは、JavaScript のサンプルが項目別に紹介されている Web ページである。外部ファイルの呼び出しやそれぞれのブラウザに対応させる記述の仕方などといった基本の部分から、フォーム編、リンク編といったように項目が細かく分けられて用意されている。プログラムの紹介ページには、サンプルだけでなく、ポイントとなる部分や、詳しい説明も一緒に掲載されている。また、サンプルのプログラムは実行結果も見られるので、動かした場合にどう表示されるのかも参考にすることができる。

(ii) Python 例文辞典 (<http://www.openspc2.org/reibun/Python/>)

本サイトも JavaScript 例文辞典と同様に、プログラムの紹介は「計算」や「関数」といったように項目別にまとめられており、説明やサンプルに加え、書式の紹介もされている。「JavaScript 例文辞典」と合わせ、Python の文法の確認や、出題部分の参考に本サイトを利用した。

(iii) JavaScriptist (http://javascriptist.net/docs/js_ref.html)

本サイトでは JavaScript の入門から応用までの解説をはじめ、リファレンスや逆引きサンプル集なども用意されている。項目が丁寧に分けられているため、情報も見やすく見つけやすい。プログラミングの作成についての解説だけでなく、JavaScript の関連ニュースや情報交換のフォーラムのコンテンツも設置されており、JavaScript の幅広い情報を参考にできるサイトだといえる。

(iv) とほほの JavaScript リファレンス (<http://www.tohoho-web.com/js/index.htm>)

本サイトでは記述方法や文法の説明だけでなく、JavaScript の概念についても詳しく紹介されている。また、シンプルなサイトデザイン

で、説明も簡潔でわかりやすく、初心者にも親切なサイトでもある。JavaScript との学習や、サイトの構成を考える際にも、本サイトを参考とした。

(v) WebAjent (http://www.geocities.co.jp/SiliconValley/4895/jscript/mz_jscript_1.html)

本サイトは JavaScript の基本的な技術を中心に解説している Web ページである。コンテンツの配置や解説のページなどの表現も簡潔で、参考にしやすい。また、プログラミングのサンプル集もあり、実行結果とソースコードが紹介されている。ソースコードは<script>～<script>までの全文が見られ、ソースコードの詳細に合わせ、実際の動作も見られるようになっている。

サンプル集では、時刻によって異なったあいさつが表示されるものや、ポップアップウィンドウで結果が表示されるものなど、目を引くようなプログラミングが紹介されている。JavaScript の特徴を生かしたコンテンツがあり、プログラミングの初心者が関心を持ちながら学べる Web ページとなっている。

(6) 作業の見通しと計画

解説のみでなく、問題も作り、利用者に解いてもらう教材を目指した。解説文とともに実際のプログラムや表を用意し、概要の説明はできるだけ簡潔に作ることを心がけた。Python との比較を紹介する部分では、JavaScript が実際にどのように表示されるかが見られるように、実行結果のサンプルを用意した。問題は、解説を見ながら解けるものを用意し、基礎技術を着実に学習出来ていく形式にしようと計画した。

教材の制作に取り掛かる前に、自身も JavaScript について学ぶ必要

があった。まず、入門書を読み進め、JavaScript の概念やプログラミングの歴史などについても目を通した。また、入門書で紹介されているプログラミングを実際に制作し、記述方法の基本や、規則なども学んでいった。基礎技術の学習と並行し、できることや、プログラミングの有効な点など、JavaScript の現状や可能性などについても知る事となった。入門書での学習を一通り終えてからは、サイトの構造について考える事となった。オンライン教材としての使いやすさや、オンラインの特徴を生かしたコンテンツなどについて構想を練り、他の教材との差別化を図ったウェブサイトの構築を進めていった。

3 本論

(1) 教材の詳細

本教材にはプログラムの歴史や、他のプログラミングとの違いの詳細など JavaScript の概念についての紹介は掲載していない。本教材では、プログラムの作成についての解説に重点を置き、利用者には実際にプログラムを使ってもらうことで、JavaScript でどういったことができ、どのように動くのかということを知ってもらいたいと考え、概要の簡単な説明にとどめている。

「対象者について」の項目でも述べているが、本教材は HTML についての知識がある人を対象として制作したマニュアルである。利用者には、基本的なウェブページが制作できる程度の知識があることを前提としている。よって、ソースコードの紹介では HTML のファイルを使用している部分もあるが、HTML の記述方法や解説などの内容は本教材では含まないものとした。

(2) 教材の内容と構成

本教材は JavaScript についての解説部分と Python との比較部分、例題といったように、大まかに 3つの項目に分けて制作している。(図 1 参照) 解説部分は、プログラミングの制約についても触れており記述方法以外の説明もしている。よって、詳細についても学ぶことができる項目となっているが、プログラミングに触れたことがある人なら、Python との比較を見ればより理解しやすいと思われる。練習問題は、解説や比較例と似たものを用意しているので、それぞれが理解しやすい順序で教材を読み進め、学習を進めていける構成にした。3つの項目の内容は以下のようにになっている。

(i) JavaScript について

この項目は、以下の解説で構成されている。

1. JavaScript とは
2. 記述方法の基本
3. オブジェクト
4. 変数
5. 変数の型
6. 演算子
7. 関数
8. イベントハンドラ
9. 制御構文

1 では、JavaScript の特徴と記入時の注意点を紹介している。大まかにどういった言語なのかということの説明し、次の項目を読み進めていきやすくするための簡単な紹介にした。2 は、1 の内容を踏まえた、更に詳しい記述方法についての解説となっている。HTML ドキュメント

内に埋め込んで使用する際の記述の仕方に加え、大文字と小文字を区別することや、コメントの方法、ステートメントの終わりなど、記述する際の決まりを紹介している。また、古いブラウザや認識できないブラウザへの対処法についても解説している。さらに、外部ファイルの作成方法や、エラーの表示方法などの紹介もしている。1と2では、基本的な概要や記述方法に加え、ファイルを個々に制作する上で、あると便利な知識までを解説している。

3では、記述したプログラムがどのように表示されるかを、document オブジェクトを使用した場合を例に挙げて紹介している。また、オブジェクトの記述方法の規則の説明には、階層図を用意した。(図2参照) 4と5では、変数の使い方をはじめ、宣言や型の種類などについて説明している。変数名のつけ方の規則と、使えない、もしくは使用を控えた方がよい予約語や、型の概念などについても、図を用いて紹介している。(図3-1、3-2参照)

6はプログラミングに使用する演算子の概要と、種類の説明の項目となっている。演算子の大きな種類の紹介に続き、更に細かい区別について、それぞれの演算子がこういったデータに使われるものなのかを種類別に紹介している。

7では、関数の使い方について具体的な記述の方法を説明している。定義することや呼び出すことなど、関数の作成方法から実行結果までを、実際のプログラミングを挙げながら説明した。(図4参照) 8も、7と同様の形でプログラムの全文を掲載して説明している。

9では、それぞれの制御構文が何を行い、こういった場面で使うものなのかということを紹介している。解説には、使用例も紹介し、構文の記述の仕方もわかるようにした。(図5参照)

(ii) JavaScript と Python

本項目は、JavaScript と Python の記述の違いを見てもらうための項目である。プログラミングの知識がある人は、具体的な比較例を見ることで、関数の使い方や演算子の違いなどに気付くことができる。よって、細かな解説を学習の補助として役立てるといった使い方もできると考える。

初めてプログラミングを勉強する人には、解説のページを参考にしながら読んでもらいたい。例に挙げているプログラミングは、他のページでも紹介しているものなので、Python やプログラミングの経験のない人でも、JavaScript のプログラミングの参考として利用できる項目になっている。まずプログラミング自体の動きを見たい人にも、基礎知識を学んでから学習を進めたい人にも、個人の能力や学び易い順序に合わせて利用してもらいたいと考え、作成した項目でもある。

(iii) 問題

練習問題の項目では利用者にプログラミングを記述してもらう形の問題を作成した。問題は、白紙の状態から作成してもらうのではなく、プログラムの和訳にあたる日本語で書かれたものを正しいプログラミングに書き換えるといった出題形式にした。

```
<script type="text/javascript">  
JavaScript とドキュメントに書き込む;  
</script>
```

使い方は、以下のようにになっている。

それぞれの問題の下部にテキストエリアを用意しているので、まず問題部分をすべてコピーし、テキストエリアに張り付けてもらう。次にテ

テキストエリア内で、`<scripttype="text/javascript">~</script>`の間の日本語部分を正しいプログラムに書き直してもらう。書き直したプログラムの結果は、テキストエリア下部の「実行」ボタンで表示することができ、解答画面と同じように表示されれば正解となる。(図6参照)

問題には、解説を読み返しながらかけるような内容のものを選び、記述や計算といった初歩的なものも含めて出題した。

(3) ファイルの構成

- ・ HTML ファイル
- ・ CSS ファイル

HTML ファイルは、プログラムの実行例の表示や、テーブル、解説に使うテキストエリアなどを表示する為に使っている。また、解説では JavaScript を外部ファイルとして作成する方法も紹介しているが、教材でプログラミングを紹介する際は HTML 内に JavaScript を埋め込んだ形のものを用いている。よって、利用者がマニュアルのソースを見る場合にも理解しやすいと思われることから、本サイトの制作に使用した JavaScript は HTML ファイルに埋め込んで使用した。

CSS ファイルは、サイトのデザインに使用している。文字の大きさや色をはじめ、項目のレイアウトなど、あらゆる外観についての設定は CSS ファイルで指定している。

(4) 制作過程について

本教材を制作するにあたり、自身もはじめて JavaScript に触れることになった。よって、マニュアルサイトの作成に取り掛かる前に、概要やプログラミングの作成方法などについて学ぶ必要があった。学習は

JavaScript の入門書を読み、ひとつひとつ例題のプログラミングを作成していきながら進めていった。教材の制作過程での問題点と解決策は以下のとおりである。

- ・ 教材の水準について

本教材は、JavaScript の基本的な記述の仕方を身につけることや、応用につなげられる技術の習得を目標として制作しようと考えていた。しかし、利用者がプログラミングに対し、どれほどの知識を持っていることを前提とし、解説を進めていくのかということが問題に挙がった。

プログラミングを初めて学ぶ人には、ファイルの作成方法や用語の説明など、細かい説明が必要である。一方で、プログラミングを作成したことがある人には、プログラミングの概念や用語の詳細の解説は必要ではないと考えられる。だが、前章でも述べたように、初心者に向けてのマニュアルはこれまでの卒業制作の課題としても存在していることもあり、本教材は個々に合った形で、より多くの人が使え教材にしたいと考えた。よって、解説だけでなく他のプログラムとの比較も用意することで、初心者であっても、プログラミングを学んだことのある人には利用し易いようなマニュアルサイトにした。

- ・ JavaScript の説明について

本教材の対象者は、プログラミングの初心者だけではない。よって、解説のページは、プログラミングを学んだことのある人にも向けたものにしようと考え、JavaScript についてどれほどの解説をするかという点も配慮すべき点となった。プログラミングの知識の有無に関わらず参考にしやすい解説にするため、できるだけ簡潔な説明を用意した。またプログラミング言語の概念等は省略し、プログラミングを作成する際に最低限必要とされる情報に的を絞り、紹介することにした。

- ・ 比較例と練習問題について

プログラミングの比較例と練習問題の難易度も問題点に挙がった。経験者にとっては、少し応用の効いた出題は問題ないかもしれないが、JavaScript を初めて学ぶ人にとっては難しく、意欲が減退してしまう可能性がある。よって、紹介するプログラミングの例は種類分けし、なるべく短いものをいくつか用意することにした。また、練習問題は、プログラミングの例や解説の内容と重なるものを出题することにした。次々に新しい技術を紹介するのではなく、似た問題を何度も目にすることで、プログラミングの感覚をつかんでもらい、紹介した例を確実に身につけてもらえる形にした。

(5) 工夫した点

各コンテンツは、目次やページについての説明を必要としないような、項目名に忠実な解説にした。また、1つの事を説明する場合でも、解説ができるだけ長くならないように、「変数1」や「変数2」といったように項目を分け、読みやすいように工夫をした。メニュー部分は、各項目を閲覧する場合でも左側に表示されたままなので、どのページにも簡単に移動できる。さらにページの上部に項目名が記されているので、現在どのページを見ているのかがすぐわかるようにした。(図7参照)

重要な部分や、目にとめてほしい部分は、太字にしたり色をつけたりして他の文章と区別できるようにした。また、記述方法や用語の説明には実際のプログラムを紹介し、文章での説明は簡潔なものにした。

記述の方法を説明する項目でも、`<script>~</script>`内にプログラムを書き込むという説明をしているが、変数や関数などの説明に至っても、`<script>`タグと`</script>`タグまでを省略せずにプログラムは全

文紹介することになっている。<script>タグを使用して記述することは JavaScript の基本であり、最も覚えてもらいたい部分でもあるため、サイト全体を通して目にするのできるように以上の方法をとった。

マニュアルを利用しながら JavaScript の感覚を掴んでほしいため、教材ではなるべく簡単な例を参考として掲載している。HTML の全文を紹介する場合は、以下のように、ソースコードについての説明も加えて紹介している。

```
<INPUT TYPE="Button" //ボタンの指定
      VALUE="ShowAlert" //ボタン上に文字の表示
      onClick="Button()">
```

さらに、紹介したソースコードの実行結果も用意しているので、プログラム動作を見てもらえるようにした。(図8参照) Python との比較を紹介する項目でも、JavaScript については実行結果を別ウィンドウで見られるようにし、ポップアップウィンドウで表示される結果も見やすいように工夫した。

練習問題の項目では、実行結果がポップアップウィンドウで表示されるようなテキストエリアを用意した。ソースコードと詳細は以下のようになっている。

※関数定義部分

```
function dispSubWin(taObj)
{
    sbWin = window.open("blank.html","checkWin",
width=320,height=240");
    sbWin.document.open();
```

—JavaScript のオンライン教材サイトの構築について—

```
with (sbWin.document) {  
  write("<html><head><meta http-equiv=\"content-type\"  
  content=\"text/html; charset=UTF-8\">\n");  
  write("<meta http-equiv=\"content-Script-Type\"  
  content=\"text/javascript\"></head><body>");  
  write(taObj.value)  
  write("</body></html>");  
  sbWin.document.close();  
}  
}
```

※実行部分

```
<form>  
<textarea cols="40" rows="5" name="aTEXT"></textarea><br>  
<input type="button" value="実行"  
onClick="dispSubWin(this.form.aTEXT)">  
</form>
```

実行ボタンが押されると、オンクリックイベントが起動し、dispSubWin の関数を呼び出すようになっている。

dispSubWin は別ウィンドウにテキストエリアの入力値を利用するために、テキストエリアオブジェクトを引数としている。onClick イベントで dispSubWin の関数を呼び出す際に、dispSubWin はテキストエリアオブジェクトを引数としているので、フォーム内の aTEXT という名前のテキストエリアオブジェクトを dispSubWin に渡している。

```
sbWin = window.open("blank.html","checkWin","width=320,
```

height=240”); はウィンドウオブジェクトで、新たにウィンドウを開くための定義である。url は blank.html を、name は checkWin を、style は width=320,height=240 を引数としてオープン関数に渡している。同時にウィンドウオブジェクトを sbWin に代入している。

sbWin.document.open(); はウィンドウオブジェクトのドキュメントを書き込み可能にしている。

with 関数によって先ほどの document オブジェクト (sbWin.document) に対して処理を行う。write でドキュメント内に、<html>~</html>の内容が書き込まれ、テキストエリア内の入力値がプログラムの実行結果として別ウィンドウによって見られる仕組みになっている。

以上のプログラムによって、テキストエリア内に<script>~</script>を含んだプログラムを記述し、ボタンを押すと 320 × 240 の別ウィンドウが現れ、実行結果として表示される仕組みになっている。(図 9 参照)

問題ごとに解答できるテキストエリアが用意されているので、利用者は問題を解くための環境を用意することなく問題に取り掛かることができる。さらに、練習問題は、出題部分でも、<script>タグを使用することや、文末に「;」をつけるなどといった記述の規則がわかるようにしており、気軽に問題を解いてもらえる形にしている。

4 まとめ

(1) ユーザー評価

教材のマニュアルサイトが一通り完成した時点で、人文情報学科の学生に実際に使用してもらい、アンケートを行った。アンケートは、良かった点と良くなかった点についての感想に加え、改良すべき点や気に

——JavaScript のオンライン教材サイトの構築について——

なった点などの指摘を依頼したものである。見やすさや理解に役立てられたという理由では、以下の点が評価された。

- ・ シンプルなレイアウト
- ・ サンプルや実行結果の例
- ・ Python と JavaScript の比較
- ・ 変数や演算子の説明のわかりやすさ
- ・ 説明のページに用意された図
- ・ 統一感のあるサイトデザイン

一方で、良くなかった点としては以下のような感想が集まった。

- ・ 解説の中で使われている用語の説明がほしい。
- ・ 練習問題の 2 問目以降にもヒントがあったほうが良い。
- ・ 練習問題のページが小さい場合がある。
- ・ アラートなどのプログラムが書きづらい。
- ・ 問題を解く際に、どのページを参考にしたらよいかわかりづらい。
- ・ 解答の記述の方法の詳細がほしかった。

改良すべき点や気になった点についても意見をもらうことができた。

- ・ 練習問題で参考にすべきページのリンクがほしい。
- ・ 文字の変換ミスが少しある。
- ・ 練習問題の解答画面で、何も表示されないものがある。
- ・ 解答を書き込むテキストエリアを大きくしてほしい。

デザインや説明に用意した図などについては、見やすいという評価がもらえたが、プログラミングの用語やページに関する説明が不十分だったことがわかった。良くなかった点や改良すべき点として寄せられた意見としては、解答に使用するテキストボックスや、文字の表示についての不具合を指摘するものが多かった。アンケートを通して、Firefox

以外のブラウザではうまくページが表示されない場合があることもわかった。

改良すべき点で集まった指摘の採用については以下ようになった。

- ・ 練習問題で参考にすべきページのリンクについて

練習問題の項目に関しては、問題数が多くない事に加え、問題にはなるべく短く、基本的なものを用意している。さらに、出題した内容は、紹介したプログラムと重なったものになっている。また、本教材は、JavaScript を気軽に学んでもらえるよう、解説で紹介したソースコードなどのコピーを禁止していない。上記の理由に加え、学習した内容を思い出しながら、繰り返しページを読み返して解答してほしいため、リンクは用意しなかった。

良くなかった点として挙がっていた、ヒントについても、日本語表記されたプログラミングを書き直す際の最低限の注意点は既に用意しているので、利用者には、必要に応じて項目を確認してもらいたいと考える。

- ・ 練習問題の解答画面について

何も表示されないと指摘を受けた解答画面については、記述の間違いを直し、正しく表示されるように記述し直した。

- ・ 文字の変換ミスについて

文字の変換ミスについては間違っている部分を確認し、書き直した。

- ・ 解答を書き込むテキストエリアについて

解答に使用するテキストエリアは、スクロールバーが使用でき、記述に問題ないと考え、全ての問題においてテキストエリアのサイズを統一していた。しかし、指摘を参考にし、記述の長くなるものに関してはテキストエリアのサイズを大きくした。記述したものをスクロールせずに見られるよう、問題に合わせた十分なサイズのテキストエリアを用意す

ることにした。

(2) 自己評価

本教材は、JavaScript を初めて学ぶ人にも、これから更に学びたいという人にも利用してもらえるものとして制作した。しかし、プログラミングの用語についての詳しい説明や、概念についての解説がないことが、わかりづらさにつながってしまった。より多くの人に使ってもらう事を目標とし、プログラミングについての説明は作成についての解説に重点を置き、簡潔なものを目指した。だが、プログラミングの知識がある人も対象としたことで、JavaScript の初心者にとってはわかりづらい表現になってしまった。

JavaScript と Python の比較例を用意したところや、図を用いての説明したところなどは、利用者の理解に役立てられるように考えた部分であり、アンケートでも評価をもらえたことから、実現できた点であると考えられる。しかし、解説の丁寧さに欠けてしまった点は、JavaScript をこれから学びたい人に向けても制作している教材としては不十分なところだといえる。それぞれの学びたい順序で学習を進められるように、解説部分とプログラミングの紹介部分を用意したが、解説部分は JavaScript の初心者配慮したものである必要があった。

(3) 今後の課題

今後の課題としては、解説とプログラミングの紹介が改良すべき点と考えられる。サンプル集を用意するなど、JavaScript の特徴を生かしたコンテンツを増やすことも、関心を持って学習してもらう為のより工夫すべき点だと考えられる。

プログラミングの知識の有無に関わらず快適に学習してもらうには、コンテンツを更に見直さなくてはならない。また、練習問題の構成についても見直す必要がある。参考ページやヒントなど、解答に導くための工夫が今後の課題であるといえる。さらに、Firefox 以外のブラウザにも対応させる必要がある。オンラインで使用できる教材として様々なブラウザに対応させることで、より多くの人にとって見やすく使いやすい教材にすることができると思う。

(4) 紹介していないものについて

前章でも述べたように、JavaScript は「Ajax」の登場で近年さらに注目を集めている言語である。開発は進み続けており、検索サイト以外でも使われる場面が広がっている。また、徐々に知名度も拡大している技術であることから、今後 JavaScript を学ぶ人にとって、Ajax はより必要とされる知識のひとつとなると考えられる。

(5) 最後に

JavaScript は多くのウェブページで利用されているプログラミング言語であり、規則の柔軟さやファイル作成の手軽さからも、身近な言語であるといえる。また、Ajax の登場によって、見た目が華やかで動きも面白い Web アプリケーションの開発が進んでいる。外観の斬新さに加え、軽快な動作も Ajax が注目されている理由のひとつといえる。本教材では Ajax については割愛したが、近年ウェブページでもよく見られるような目を引く技術を教材内で紹介できたなら、JavaScript について興味を持つ人も増えるだろう。

本教材は解説や、例の挙げ方などにおいて、マニュアルサイトとして

——JavaScript のオンライン教材サイトの構築について——

はまだ不十分なものであるといえる。改良すべき点も見られるが、本教材が、JavaScript に関心を持ち、個々の更なる学習の意欲につながるきっかけになればと考える。

文献表

大津真

2006 『JavaScript プログラミング入門第2版』 オーム社

『JavaScriptist』

<http://javascriptist.net/>

『スクリプト言語の比較』

<http://pub.cozmixng.org/the-rwiki/rw-cgi.rb?cmd=view;name=%A5%B9%A5%AF%A5%EA%A5%D7%A5%C8%B8%C0%B8%EC%A4%CE%C8%E6%B3%D3#footnote-3>

『JavaScript 例文辞典』

<http://www.openspc2.org/reibun/javascript/>

『Python 例文辞典』

<http://www.openspc2.org/reibun/Python/>

『とほほの JavaScript リファレンス』

<http://www.tohoho-web.com/js/index.htm>

『Web Ajent』

<http://www.geocities.co.jp/SiliconValley/4895/jsrpt/mzjsrpt1.html>