

JavaScript のオンライン学習システムの 構築について

西迫 絵梨

目 次

1	はじめに	1
1	現状認識	1
2	JavaScript のオンライン学習用教材について	1
3	それによってどのような効果が見込めるのか	3
4	他の教材との比較	3
5	比較結果	6
2	本論	6
1	動作環境	6
2	教材の内容	6
3	教材の詳細	7
4	教材の構成	8
5	ファイルの構成	12
6	重要箇所の強調	15
3	結果と考察	16
1	例題の解説	16
2	紹介する例題について	16
3	間違いの指摘	16
4	練習問題の出題方法	17
5	紹介していないものについて	17
4	評価	18
1	ユーザー評価	18
2	自己評価	19
5	今後の課題	20
6	最後に	20

1 はじめに

JavaScript をこれから勉強しようとしている人、勉強中の人のためにオンラインで学習できるためのサイトにする。また、このサイトをきっかけに更に JavaScript に興味や関心を持ってもらえるようなサイトを作成する。例題のソースを見ながら勉強することで、自らの学習に役立ててもらおう。

(1) 現状認識

JavaScript について解説している本やサイトはたくさん存在している。しかし、それらをただ読んでいるだけでは頭に入りにくく、飽きが出てしまう。それに加え、ただ文章を読んでいるだけではなぜそうなるのかが分かりにくく、プログラムの単語や仕組みを間違えて覚えていたとしても気付きにくい。いくつか解答例を表示し正しいものを選択する問題があれば、細かいところまで見るので間違えて覚えることが減る。また、そのつど解説を表示すればなぜその解答になるのかが分かり、仕組みも覚えやすくなる。そうして、JavaScript のプログラムを繰り返し覚えていけば、JavaScript の学習用として有用なものになると思う。

(2) JavaScript のオンライン学習用教材について

(i) 環境

JavaScript は HTML ファイル内に書き込むか、JavaScript 単体のファイルを用意し外部スクリプトとして定義して利用する。いずれの場合でも、JavaScript はテキスト形式なので、テキストエディタがあればプログラムを作ることができる。やはり JavaScript に適したテキストエディタが好ましい。例えば、Windows に付属しているものとして、

「メモ帳」でも開発はできる。しかし、メモ帳では機能的に貧弱で、保存容量が少ないため、快適に開発を進めることはできない。

また、HTML ファイル内に直接書き込む場合、HTML と JavaScript と色分けして表示できるようなソフトを使用すると便利だろう。なぜなら、色分けしていないと、どこからどこまでが JavaScript なのかが分かりにくくなってしまうからだ。特に、Adobe 社の Dream Weaver や、ジャストシステム社の IBM ホームページ・ビルダーなど、Web ページ作成用のソフトを使えばより便利である。しかし、Web 作成用ソフトは値段が高いため、フリーソフトであるサクラエディタを推奨する。

(ii) オンライン化について

なぜオンライン化にするのかということであるが、実際に JavaScript を学びたいのであれば、ユーザーが環境を用意すればよい。上記にも記してあるが、JavaScript の解説している本やサイトに必要な環境等は説明しているものがある。それに、実際にソースを見ながら自分で考えプログラムを作っていくのが上達への一番の近道であろう。

しかし、JavaScript を勉強しようとしている人にはソフトをインストールしたり自分で環境を用意するのが面倒だという人もいるだろう。特に、Dream Weaver 等は値段が高く手が届きにくい。また、専門書も 3000 円前後と高めなので、初心者にとって上達の壁になっている。そこで、Web 上で解説と問題を用意し、その場で解答と解説が見れば、急いで環境を整える必要が無くなる。インターネットに接続できれば、すぐに勉強にとりかかれる。

(3) それによってどのような効果が見込めるのか

まず、JavaScript の敷居が低くなり、初心者でも勉強にとりくみやすくなる。JavaScript によって Web サイトでできることの可能性が広がるので、興味・関心が広がることが見込まれる。また、それによって JavaScript は奥が深いため、中級、上級へと徐々にレベルアップし、それによって JavaScript 全体の発展につながると思う。

(4) 他の教材との比較

(i) 書物：David Flanagan 著・村上 列 訳 「JavaScript 第5版」

この書は、主に言語としての JavaScript の解説書である。プログラミング言語を習得するのに必要な基礎知識やコア JavaScript についての知識、クライアントサイド JavaScript の基本機能と高度なトピックを紹介している。また、この本は村上氏による翻訳本であるが、直訳しただけの文章は無く、読んでいてストレスを感じない。

しかし、様々な応用テクニックが掲載されているが、理論的説明が多く、サンプルコードを反復練習して覚える人には難しいだろう。JavaScript は奥が深く、自由自在に操るには本書ぐらいの内容が必要だが、まったくの初心者の場合は、別の本を買って基礎知識を蓄えた上で読むほうが良いだろう。

(ii) 書物：オーム社 出版 大津 真 著 「JavaScript プログラミング入門 第2版」

JavaScript によるプログラミングをしっかりと身につけたいという初心者ユーザー向けな解説書である。JavaScript を言語として捉えた解説から、ちょっと難しめのものも含めたサンプルがたくさん掲載されている。また、触り程度ではあるが、Ajax についても記述があるので、

基礎を身につけたい初心者から中級者までを対象としている。しかし、誤字や内容の誤りも多少あり、表などをノートに書き写して覚えるユーザーには注意が必要となる。

(iii) 書物：技術評論社 出版 「JavaScript 中級講座 Ajax を学ぶ前の基礎知識」について

JavaScript の概要や、必要な環境などの初歩的なことから、Ajax や prototype.js などまで、図を交えて丁寧に解説している。時折、図や表、ブラウザの写真が載せてあり、なぜそうなになるのかなど初心者にも分かりやすくなっている。また、書の中で紹介した例題のサンプルファイルをインターネットの専用サイトからダウンロードすることができる。

しかし、説明が長々と続く感が否めなく、読んでいるほうは退屈になってしまうかもしれない。説明の中で、重要なところに下線を引いたり、太字にして強調するなど文に強弱を持たせたほうが、読者も飽きずに、より理解しやすくなる。また、あまり深い内容には言及していない。この書籍が解説しているのはあくまでも「基礎知識」であって、本格的な内容は他を当る方が良いだろう。

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

JavaScript の概要や、書き方、文法、構文など基本的なことから、正規表現などの基本オブジェクト編や、応用編としてブラウザオブジェクト編まで多岐に渡って解説している。

しかし、延々と文章による解説が続いていて、ただ読んでいては飽きがきてしまう。文法解説となっているので、基本として別に教材を利用し、分からない点がでてきたら当サイトで確認するというサブ的で要素の方が合っているだろう。また、練習問題等があるというわけでもな

いので、初心者には向いているとは考えにくい。

(v) Web: JavaScript basic samples (<http://www.sumnet.ne.jp/domp/jsbs/>)

JavaScript の概要から、実際に動的 Web サイトを作成する上で使用できそうなサンプルがたくさん掲載されている。各項目のページで実行例を見れるのも、初心者にとっては実行結果を確認できるので理解しやすい。

しかし、各項目のページに表示されているのは、各項目のソースのみなので、それに対しての解説はない。サイトを動的なものにしたいという人にはいいと思われるが、JavaScript を学習したいという人には向かないサイトである。

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

基本から、応用まで様々なサンプルとともに、各項目でポイントとして重要なところ、解説、サンプルソース、実行例が表示されていて、デザインも全体的にスッキリしていて見やすい。また、動作ブラウザについても各項目毎にはじめに表示されているのは分かりやすく親切である。

ただ、解説文の文字が小さめで強調等もなく書かれているだけなので、多少見難い。また、最新ブラウザに対応したサイト「新 JavaScript 例文辞典 (<http://www.openspc2.org/reibun/javascript2/index.html>)」もある。こちらでも基本から応用まで多岐に渡ってサンプルがあるのだが、各項目毎に表示されるのは、簡単な解説とサンプルプログラム、実行例のみとなっていて、詳しく知りたい場合は、やはり別の教材を用意した方がよい。そして、こちら 2つのサイトでも練習問題は用意されておらず、サンプルの提供と解説のみとなっている。

(5) 比較結果

比較の結果、練習問題があったとしても、すぐに結果と解説が返されるというものは殆どない。

Web 上に学習システムがあれば OS やブラウザに囚われることなく学習ができ、インターネットに接続さえできれば学習できるので、場所に制限されるということも少なくなる。更に、問題も解け、かつ解説も一緒になっているので、教材を持ち歩かなくても学習できる。JavaScript をこれから学習していくような初心者にも使ってもらえるような、分かりやすい教材を作成したい。

2 本論

(1) 動作環境

製作には、Web プログラミング言語の一つである PHP を使う。PHP を用いて、ユーザーに問題を解いてもらい、その結果を Web サイト上に返して表示する。また、PHP で作成するにあたって、xampp(<http://www.apachefriends.org/jp/xampp-windows.html>) というフリーソフトをダウンロードして使用する。xampp には、Apache、PHP、MySQL が使える環境が用意されており、xampp を動かすだけで PHP を用いて Web サイトを作成することができる。

(2) 教材の内容

教材の内容を考える。ただ文章で解説するだけでは、読んでいる方は飽きてしまうので、図や表を交えて解説する。また、随所に選択式の簡単な練習問題を設け、その場で結果と解説を表示することでユーザーの学習につなげたい。

そのため、解説の文章が長くなりすぎることは避け、できるだけ簡潔に分かりやすくしたい。サンプルがある場合には、必ずソースを表示し、実行例を用意し、どのように動くのかをユーザーに見せる。また、練習問題を用意し、サンプル以外での利用パターンも解説する。

(3) 教材の詳細

JavaScript を学習するにあたって、はじめに JavaScript とは何か概要を理解していなければならない。最初に、概要を説明する。基本的な単語などの説明は随時解説を取り入れることにする。はじめにまとめて表示してもいいのだが、解説のページを見ながら更に補足説明も見るとするのは面倒臭いだろうと感じたからだ。

次の内容の JavaScript をユーザーに教える。

- ・ JavaScript から Web ブラウザを操作する方法

- ・ DOM を使って Web ページやスタイルシートを動的に操作する方法

一般的に JavaScript は Web ブラウザや Web ページの動作を動的に変更するために使われる。例えば、ポップアップウィンドウを開いたり、オンマウスで色や画像を変えたりといったことを実行している。こういった一般的なものから紹介した方が、初心者には勉強しやすいだろう。その後で、更に詳しく勉強していけばより理解が深まる。

DOM の部分では、はじめに DOM(Document Object Model) の概要を説明する必要がある。

DOM は Web 標準化団体の W3C(World Wide Web Consortium、<http://www.w3.org>) で標準化が行われている。DOM は段階的に拡張が行われており、「Level 1」から「Level 3」までがある。しかし、Web

ブラウザ側は Level 3 には殆ど対応ができていない。今回は Level 2 までの対応のモジュールを内容に取り入れることにする。

また、今回は JavaScript の説明なので、HTML に関する知識についての内容も含めない。Web 上で JavaScript を動かす上で必要になる知識ではあるが、HTML について説明をしていけばきりが無く、HTML の説明が長くなつては、本題の JavaScript を勉強する前に疲れてしまう場合もある。

そのため、今回はリンクを貼り、HTML に関しての学習はユーザーにまかせることにする。

その他にも、今回は、「フレームの操作」、「インラインフレームの操作」についての解説は内容に含めていない。

かつては、Web ページの内部を複数のフレームに分けて、メニューとコンテンツを分けて表示されているといったことがよく行われていた。しかし、将来的にはフレームは廃止される方向にある。まだ完全に廃止されたわけではなく、今も使っているユーザーは多いのだが、初心者ユーザーにも教えるにあたって、廃止される方向にあるものは解説する必要はないと思った。廃止されるものを教えると、かえってユーザーが混乱するかもしれない。教えるのであれば、推奨とされているものの方が良いだろう。しかし、それでは JavaScript から離れてしまうため、フレームは内容から外すことにした。

(4) 教材の構成

- ・はじめに

 - ページの見方

 - JavaScript の概要の説明

- Window オブジェクト

 - 概要

 - メッセージの表示

 - ステータスバーと各種ウィンドウの操作

- イベント処理

 - 要素上でマウスがクリックされたときに処理を行う

 - 要素上にマウスポインタが合わせたとき/外したときに処理を行う

 - ページの読み込みが終わったとき/閉じる時に処理する

 - 要素本来の動きを止める

 - JavaScript でイベントハンドラを定義する

- フォームを操作する

 - フォームについて

 - テキストの入力欄にアクセスする

 - チェックボックスとラジオボタンにアクセスする

 - セレクトにアクセスする

 - ボタンにアクセスする

 - コントロールの入力禁止/許可を切り替える

 - フォーム関連のイベント

 - 送信とリセット

- 別ウィンドウやフレームを操作する

 - 別ウィンドウの開閉

 - 別ウィンドウと元のウィンドウの間で情報をやり取りする

- ブラウザ関連の各種のプログラミング

 - location オブジェクト

history オブジェクト

タイマー処理

- DOM の概要

 - 概要

 - DOM のレベル

 - ドキュメントツリー

- ノード

 - HTML 文書全体を得る

 - ノードを得る

- ノードの追加／削除

 - ノードの作成

 - ノードの追加

 - 削除と置換

 - 要素ノードに HTML を流し込む

 - ノードの追加/削除の例

- 補足

 - 参考サイトのリンク

(i) 導入部

最初に、JavaScript の概念と HTML に JavaScript を組み込む方法を解説する。その後、JavaScript を動かすために、JavaScript から Web ブラウザを操作する方法として、まず Web ブラウザに対応する Window オブジェクトについて解説する。その後に、実際に何かが起こったときの処理等を解説していく。

なぜ、はじめに Window オブジェクトについて解説するのかというと、JavaScript はオブジェクトを介して様々な対象を操作するのだが、

Web ブラウザ自体も「Window」というオブジェクトで渡されているからだ。

よって、「Window オブジェクト」について学ぶことが、Web ブラウザで動かす上での基本となるのである。

(ii) イベント処理

何かが起こったときに実行される処理のイベント処理を順に教えていく。各イベント処理ごとにページをわけていく。

まず、どのような処理を行うのかということをもとに文と画像を使って解説する。

次に、その処理で使う要素 (HTML 内で書く処理、JavaScript で実行する関数) についてのソースを別窓表示を使って表示し、その要素について解説する。

また、その処理の実行例も用意する。その後、処理の内容に応じて練習問題を選択形式で出題し回答してもらう。

また、似た処理や、解説少ない処理についてはいくつかまとめて同じページ内で解説している。処理についての解説が長い場合は、単独でページを設けているが、似たような処理で解説が短いものはまとめたほうが見やすくなると思ったからだ。

(iii) DOM

Web ページの内容を動的に変化させる際には、JavaScript から HTML 文書にアクセスし、その構造を書き換えるという処理を行っている。HTML 文書にアクセスする際には、「DOM」というシステムを使うので、まず、その DOM についての概要の解説をする。その後で、DOM を使った処理の解説を順に教えていく。各処理の解説の仕方は、イベント処理のときと変わらない。

そうすることで、ページ全体にまとまりが生まれ、ユーザー側にページの変化による混乱を起こさせないようにする。

(iv) 補足

補足として、JavaScript のリファレンスサイトのリンクや、HTML のリファレンスサイトのリンクを貼る。このサイトの中で紹介しきれていない関数等については、ユーザーにリンク先のサイトを利用して勉強してもらおう。

このサイトは、JavaScript をオンラインで解説を読み、問題に答えることで学習を進めていくことを重点においている。

そのため、このサイトで紹介していない関数等の解説はかえって混乱をまねく恐れがあると考えたからだ。

(5) ファイルの構成

作成したファイルは以下に分かれている。

(i) HTML ファイル

HTML ファイルは主にサンプルの実行例を表示する際に使う。この場合、JavaScript は HTML の中に書いている。あとで、ユーザーそのソースを見た際に、分かりやすくするためである。

(ii) CSS ファイル

js.css は、サイト全体のデザインに適用している CSS ファイルである。また、別窓表示している箇所の css ファイルは js2.css とファイルを分け、整理した。

(iii) 教材別の PHP ファイル

今回のサイトはほぼ PHP ファイルで構成されている。

head の部分と、サイドメニュー、footer の部分は、headleft.php と

footer.php とし、それぞれ require "○○.php" と記述し、外部ファイルから読み込んでいる。こうすることで、サイドメニュー等の HTML の長い箇所の記述を簡略化でき手間が省ける。

問題の部分では、HTML の form(ラジオボタン) でユーザーの解答を送り、その解答が正解と合っているかをチェックし、結果を返している。今回は、そのファイルをひとつのファイルの中で実行している。

なぜひとつのファイルで実行しているのかというと、こういったサイトでは紹介する例題が多くなってしまいうので、作る側にとって見やすくするためである。

また、ひとつのファイルにすることで、練習問題がある場合も、正解した場合に表示するページも同じファイルなので作りやすくなると思った。

そのファイルについては、練習問題が出題されているファイルである countup.php というファイルを用いて解説する。ちなみに、この countup.php では、要素上でマウスがクリックされたときに処理を実行する例を紹介している。

(iv) form(練習問題の出題)

問題は、ページの下に表示されており、フォームでユーザーの選択した解答を同じファイル内にあるプログラムに送る。その際、ラジオボタンには問題全体として同じ名前 (name) と、ラジオボタン毎の名前 (value) を指定する必要がある。

問題全体の名前は、ユーザーが選択した問題番号にあたり、ラジオボタンは、ユーザーが問題の中で、選択した解答の番号にあたる。

この場合、name は q 1 で、value は 1、2、3 と数字で指定した。

(v) プログラム (解答の結果を返す)

プログラムは、ファイルの上部にあるのだが、`if (isset($_POST["q1"]))` でまず `q1` という問題が選択されているかを調べている。

その次にある `($_POST["q1"] == 3)` で正解である `3` を選択されているのであれば、「正解です」と表示する。その下に正解である解答の解説をしている。

その後で、`3` 以外の番号が選択されていた場合の処理をしている。今回は、問題の解答が違うというメッセージを表示させている。そのメッセージが表示されるのはページトップとなっているのは、間違っただけでなぜ違うのかそのページで解説している例題を再び見ることで理解しやすくなるだろうと考えたからだ。

しかし、間違っただけで選択をした場合、メッセージが TOP に表示されるようになっているが、解説を理解しているが、手が滑って誤った選を択してしまったという場合もあるかもしれない。そのときは問題までスクロールするというのが手間だと考えられるので、メッセージの下に問題に直接とべるようにリンクをはっておいた。

また、正解した場合、そのページで紹介した解説を表示させると、回答の解説もあるためページ全体が長くなり、見にくくなるため、間違えたときは紹介した解説を表示させ、正解の場合は表示させないようにした。

`3` 以外の番号を選択した場合のところのはじめに、`if($a=' a')` と入力し、表示させたい紹介の部分のを開始に `<<<FEO` と入れ、最後に `FEO` と再び入れる。正解の解説の上に、`$a = "b"` と入れる。FEO で囲む前に、`if($a != 'a')` と入れ、FEO の後に `}` で if 文を閉じる。

正解の場合は、`$a` には `b` が代入され、例題の解説は `$a` が `a` であると

きのみ表示と指定するため、表示されず、間違えた場合は、\$a には a が代入されるため表示されるようにした。

(vi) JavaScript ファイル

JavaScript ファイルで実行しているものは、サイドメニューの折りたたみと、各解説ページで紹介する例題で解説している要素のソースを別窓で表示するためのプログラムを記述している。

サイドメニューの折りたたみは、サイトのメニューの数が増え、ユーザーが利用している際に、どこまで進んでいるのか分かりにくくなる可能性がある。また、メニューが増えた場合、ページ全体が縦に長く表示されてしまうため、デザインとしてもバランスが悪くなる。

しかし、ページを開いた時点でサイドメニューが折りたたまれていると、何があるのか分かりにくくなるため、ユーザー自身に折りたたみを実行してもらえようにした。

(6) 重要箇所の強調

JavaScript のイベント処理等の解説の文章では、重要な部分がどこなのか、すぐわかるように、HTML で `~` で囲むと、赤字で表示されるようにした。`` とすると、赤の下線を引くようになっている。

赤字は、言葉や単語、簡単なプログラムを、その解説の中でポイントとなる点として強調した。赤字の下線は、主に、赤字で強調したものの解説となっている。色や下線を引くことで、単語やプログラムなどとそれ以外のものとのユーザーが見分けやすいようにした。

3 結果と考察

(1) 例題の解説

上記でも説明したように、本文中には、重要なポイントには赤字や下線で強調しているが、まわりくどい解説になっている箇所がある。ユーザーのアンケートにも、歌詞悦がややまわりくどい箇所があるとの指摘があった。

どう解説すれば初心者にとって分かりやすくなるのか。難しいと思う単語に関しては、ページの下に注意として記載しておいた。しかし、それではそれでは理解できていないユーザーが出てくる可能性がある。もっと解説をいただく必要がある。

(2) 紹介する例題について

基本的な例題を紹介する方が、初心者には勉強を取り組みやすいだろうと考え、あえて複雑な応用問題は数を減らした。基本的なことを勉強する前に複雑な問題を取り組んでしまうと挫折してしまうのではないかと考えたからだ。

しかし、紹介している例題を減らしたことで、実際に利用できるパターンの紹介が減ってしまった。もっと実用的な例題も増やしたほうがよいだろう。また、紹介例が少ないため、紹介している例題でしか理解ができず、応用されたものや、別のパターンのものが出ると分からなくなるという可能性がある。

(3) 間違いの指摘

練習問題でユーザーが選択を間違っていた場合、正確な指摘ができていない。どこが間違っているのかが表示されている方が、ユーザーには

分かりやすいだろう。その後、問題の箇所に戻りよくみてみれば分かる
とは思いますが、間違っていると納得してしまい、すぐ次の例題へ進むユー
ザーがいるかもしれない。その点はユーザーの自主性にまかせるしか
ない。

(4) 練習問題の出題方法

練習問題は、解説の内容に合わせて出題している。選択形式になっ
ているが、問題の順番等是不変。何度問題を解いても、出題形式が
変わらないため、正解の番号だけを覚えて次に進むユーザーが出てくる
可能性がある。

また、ランダムとして出題しても、答えだけを覚えてしまう場合もあ
るだろう。内容を理解した上で、問題を解いているのであれば問題ない
が、そうでない場合は、別の似たような例題などにぶつかったときに、
自分の回答のどこがなぜ間違っているのか気がつかないという可能性も
ある。

(5) 紹介していないものについて

今回は初心者でも勉強できるようにと基本的な問題が多いが、
JavaScript は奥が深く、勉強すべきものが多く存在している。

また、今回は DOM を利用したものもくつか紹介しているが、他に
もよく使われるメソッドやプロパティは存在している。それに加え、
W3C で段階的に標準化が行われており、モジュールもさらに追加され
ていっくだろう。やはり、それらに対応した解説等もいずれ必要になる。

さらに、近年では JavaScript の中でも「Ajax」は非常に応用範囲が
広く、注目されている技術であり、これからも発展を続けていっくだろう。

またその他にも、Ajax の流行と同時進行的に JavaScript のさまざまなライブラリが作られるようになってきた。このライブラリを利用することで、複雑な処理を簡単なスクリプトで行えるようになり、スクリプト作成の幅が一段と広がっている。

上記で紹介した Ajax やライブラリ以外にも、prototype.js など多くの技術がある。今回は、JavaScript の中で一般的で利用しやすいところを紹介するにとどまったが、ユーザーのためにもこれらの技術の紹介は必要になってくるのではないだろうか。

4 評価

(1) ユーザー評価

サイトを 2/3 ほど製作した時点で、製作途中のシステムを、人文情報学科の学生 3 人に利用してもらい、アンケートを行った。全員から、サイトの配置や色は見やすいとの感想を貰うことができた。そのほかにも、例題で使用している要素 (HTML、JavaScript で書いたプログラムのソース) を別窓で見ることができるというのが利用しやすいとの感想も貰えた。

しかし、上記の結果と考察のところでも記述していたとおり、解説のまわりくどさが増え取上げられていた。やはり、どのように記述すれば初心者でも理解しやすくなるのか、かみくだいた表現というものを研究しなければならないと感じた。

問題数に関しては、少なすぎず多すぎず丁度よいとの言葉を貰えたが、練習問題のパターンの幅をもう少し広げたほうがよいだろう。また、適当なところでまとめとしてまとめの練習問題を作成するといいいのではという意見があった。確かに、例題を解説を見ながら理解したとし

でも、時間が経過すると記憶が曖昧になってしまう。ある程度内容が進んだところで、まとめとして練習問題を何問か用意しておけば、復習ができ、身につきやすくなる。

また、単語や言葉を注意として解説した箇所は、それらだけをまとめたページを作成するといいいのではという意見もあった。確かに、それらの索引があればさらに便利だと思う。

その他にも、中には FireFox をつかってアンケートに答えてくれたユーザーがおり、FireFox では画像のリンクには枠がついているという報告があった。FireFox の対応が遅れてしまい、私の技術的な問題も重なり、今回はそのままになってしまった。FireFox 以外のブラウザでも対応できるようにしなければならない。

(2) 自己評価

オンライン学習教材を作成してみて、初心者にも理解しやすい解説の配慮が難しいと感じた。基礎である JavaScript の解説は理解しにくいということはないと思うが、例題が応用へと複雑になっていくにつれて、解説も複雑になり、まわりくどい解説になってしまった。ユーザーが増えれば、それだけ理解しにくいと感じるユーザーも増え、JavaScript の勉強を止めてしまうというユーザーが出てくる可能性もある。赤字で表示したり、下線を引いたりと分かりやすいレイアウトを考えたが、内容が難しくては意味がない。理解しやすいような言い回しなど私自身も勉強していかなければならない。

教材の面として、作ると決めた教材部分はすべては作れなかった。それは私の時間配分にミスが生じたためである。また、用意した練習問題が適切であるのかどうか、分からない。そして、この教材を通して

JavaScript についての知識が身についているのかは、分からない。時間が足りずに、練習問題を作れていない箇所もある。

実際に多く人の学習に役立ててもらおうシステムにするにはまだまだ不十分であり、改良が必要であると思う。

5 今後の課題

今後の課題として考えられるものとして、まず解説をいかに理解しやすいものとするか、構成を見直す必要がある。

また、もっと様々なパターンを作り、実際に活用できる例題を増やす。そうすることで、よりユーザーの勉強意欲の向上も見込める。

そして、今回のサイトでは内容に含めていないが、JavaScript の応用として、Ajax や prototype.js についても内容に含めると、初心者だけでなく、より多くのユーザーに利用してもらえるようなものになるだろう。

6 最後に

JavaScript というのは非常に多くの機能がそなわっており、使いこなすのは難しいが、今回の作成したサイトで紹介している Web オブジェクトは、Web ページを動的なものにするというものであり、興味を持っている人は多いと思う。

しかし、JavaScript というスクリプト言語が、難しいと感じてしまい手を出せないユーザーもたくさんいると思う。また、サンプルサイトで掲載されているものを利用するだけで JavaScript については分からないという人も多いと思う。

そんな人のために、練習問題もついたサイトがあれば有用なものにな

—JavaScript のオンライン学習システムの構築について—

るだろう。

初心者にも分かるような解説というものが難しく、まわりくどい箇所もいくつか見えるため、まだシステムとしては不十分ではあるが、このサイトをきっかけに JavaScript の魅力を知ってもらえればと思う。

文献表

藤本 尙

2007 『JavaScript 中級講座 Ajax を学ぶ前の基礎知識』 技術評論社

著：David Flanagan 日本語訳：村上 列

2007 『JavaScript 第5版』 オライリー・ジャパン

大津 真

2006 『JavaScript プログラミング入門 第2版』 オーム社