

Ajax の初心者用オンライン教材の作成に
ついて

中川 幸平

目 次

1 はじめに	1
1 製作の理由	1
2 Ajax とは	2
3 Ajax のメリット・デメリット	3
2 製作物について	4
1 現在の Ajax の教材の状況	4
2 教材の目標	5
3 教材の対象	6
3 教材作成の計画	7
1 勉強方法	7
2 全体の構成	8
3 教材製作における問題点	8
4 オンライン教材の内容	9
1 TOP	10
2 1-1 HTTP 通信の流れ	10
3 1-2 ブラウザの互換性	11
4 1-3 タイムアウト処理	12
5 1-4 エラー表示	12
6 1-5 DOM	13
7 1-6 イベント	14
8 1-7 マウスイベント	15
9 1-8 キーイベント	15
10 2-1 Ajax	15
11 2-2 ラジオボタン	16

12	2-3 テキストボックスの入力補助	17
5	まとめ	17
1	アンケートの結果と問題点	17
2	今後の課題	20
3	自己評価	21

1 はじめに

私は卒業制作に『Ajax の初心者用オンライン教材』というものを製作することにした。これはネット上で Ajax について書かれた教材であり、Ajax を使用したサイトを作成したいと考えている人や Ajax を勉強したいと考えている人の支援となるように製作した教材である。

(1) 製作の理由

なぜ私は『Ajax の初心者用オンライン教材』を製作しようと思ったのか。それは、Ajax という技術が、大手のサイトなどで使用され、大勢の人が活用しているにもかかわらず、その名前があまり知られていないということだ。Ajax は Yahoo!の地図⁽¹⁾や Google Maps⁽²⁾、Amazon の商品検索⁽³⁾などでも使われており、大概の人は Ajax と知らず、ただ便利なものとして使用している。Ajax はインターネットを利用する人の身近に多く存在しているのに、その名前は知れていない。

そこで、誰でも見られるオンラインサイトで、代表的な Ajax を使用したサイトを紹介し、簡単で分かりやすい説明がされているサイトがあれば便利なのではないかと考えた。オンラインサイトならどこでも見ることができ、書籍を買うようなお金もかからない。さらに、誰でも知っているような Ajax の例を挙げれば、Ajax がどういうものかわかりやすい。説明も簡単であればどんな技術なのか理解もしやすい。このようなサイトがあれば、Ajax の認知度も上がり、Ajax を勉強したいと考えている人の役にも立つと考え、製作に及んだ。

(2) Ajax とは

Ajax とは『Asynchronous JavaScript + XML』の略で「非同期な JavaScript と XML」という意味だ。

この言葉は 2005 年 2 月 18 日に、Adaptive Path 社の Jesse James Garrett 氏が書いたブログのコラム「Ajax: A New Approach to Web Applications」が初出で、その後世の中に知られるようになった。つまり、Ajax という言葉はここで提唱された⁽⁴⁾。それ以前から Google Maps のように Ajax を実装しているサイトは存在していたが、Ajax と呼ばれるようになったのは、このときからだ。

Garrett 氏のエッセイによると、Ajax とは以下の要素を取り入れたものと言っている。

- ・ XHTML と CSS を使った標準ベースの表現
- ・ DOM(Document Object Model) を使った動的な表示および対話
- ・ XML と XSLT を使ったデータ交換および操作
- ・ XMLHttpRequest を使った非同期なデータ抽出
- ・ すべてをひとつに束ねる JavaScript⁽⁵⁾

Ajax は特別新しい技術ではなく、古い技術の集合体である。以前からある技術である HTML、JavaScript、XML、CSS などを使用する。つまりは、Ajax は技術というより手法だ。Ajax は構成している技術は、どれも少しプログラミングを勉強したことがある人なら、知っている技術ばかりなので、Ajax を使ったサイトを製作したいと思えば簡単に始められる。

(3) Ajax のメリット・デメリット

Ajax のメリットとして、まず非同期通信がある。これはサーバと通信を行う際に、サーバからのレスポンスを待たずに、連続的に作業し続けることができるものだ。他には、JavaScript の HTTP 通信機能を使い、必要最低限のデータのみをサーバとやり取りし、早く効率よく必要な部分のみを書き換えることができる。

私は Ajax のメリットの中でも、この二つの機能に注目している。なぜなら、必要最低限のデータで通信を行うため、無駄がなく、サーバの負荷も減らせる。そのため、通信速度が速くなり、ストレスなく Web サイトを見ることができる。それでも通信が重く、遅いときでも非同期通信なら、他の作業をすることができる。この無駄のなさが Ajax の魅力だと私は考える。

Ajax を使用しているサイトのデメリットでは、ブラウザの「戻る」の機能が使用できないということだ。これはページ遷移せずに部分的に内容を書き換えているため、「戻る」を使用すると、ページ全体を読み込む前のページに戻ってしまう。つまり、Ajax を使用したサイトを製作するときに、ブラウザの「戻る」の機能に代わるものを設置しておかないと、使いづらいサイトになってしまう。そのほか、ページを遷移せずにサーバと通信を行うということは、ユーザにわからないまま動作するということなので、従来よりも慎重なセキュリティ配慮が必要となるなどが挙げられる。

2 製作物について

(1) 現在の Ajax の教材の状況

Ajax を勉強するために、私は『Ajax 入門』と『AJAX Web アプリケーション アイデアブック』の 2 冊の書籍を参考にさせて頂いた。特に『AJAX Web アプリケーション アイデアブック』は、ネット上にサンプルファイルがおいてあるので、実際に動かして学べたのでとても分かりやすかった。しかし、練習問題のようなものは無かったので、教材として利用するにはあまりむいていない。この他にも多くの書籍が存在している。だが多く存在しているがゆえに、その内容は多種多様である。説明が丁寧で、一行ずつプログラムの説明されていて、しっかり JavaScript や XML の説明も書かれているものもあれば、入門と書かれているにもかかわらず、専門用語が多く、文章のみで説明が書いてあるので、ある程度の知識がある人でなければ理解が難しいものや、プログラムのどの部分のことを説明しているのかわからないものもあった。お金を払って購入する以上、自分にあった書籍でなければ困る。購入者のレビューなどを見れば、ある程度はどんな内容か知ることができるが、やはり実際に見てみなければ判断しにくい。これでは書籍を探すだけで時間がかかってしまう。

一方、インターネット上の Ajax の入門サイトは、説明が見にくく読みにくいものが多かった (図 1・2・3 参照)。資料の図の例で挙げると、図 1 のサイトは、ページの中央に大きく広告があり、メインの部分も広告も同じスタイルで表示されているためとても見にくい。図 2 のサイトは、文章が長く読みにくく、途中で広告が割り込んでいて見栄えが悪い。このサイトは先に説明をしてからプログラムを表示しているが、長い文章を読んでから、プログラムを見て、わからない部分があるとき

に、もう一度長い文章から必要な部分を探し出すのは面倒だ。図 3 のサイトは、単純に使い方がわかりにくい。入門サイトのはずなのに、どの項目から見ればいいのかわかりにくく、また項目ごとに詳細リンクをクリックしないと、説明が短すぎてわからないし、手間がかかる。また、次にどの項目をみればいいのかの目安もなかった。参考書替わりに見るのなら有効なサイトかもしれないが、入門にはむいていない。

(2) 教材の目標

様々な書籍や Web サイトを見た結果、先に述べたような欠点や不満点があったので、それを踏まえて、製作するオンライン教材は、見やすく理解しやすいサイトという漠然的な目標を立てて製作に臨んだ。大まかな目標だが、私自身が Ajax を勉強しようと思って、書籍や Web サイトを見て回ったときに、内容が濃いか薄いかより、見にくさやわかりにくさが気になったので、この目標が自分にあっていると考えた。

では見やすい教材とはどのようにすれば良いのか。まずは、文字の量が必要最低限であること。先にも述べたとおり、長い文章はパソコン画面では見にくい。補足などは文字にリンクを貼るか、サイドメニューに補足事項のリンクを作成しておく。

次は視覚的な問題である。当たり前だが、文字だけで説明しても当然わかりにくい。そのため、説明するためのサンプルプログラムを表示し、その結果も表示する。プログラムを実行すれば、どのようになるのか視覚的にわかるようにしておく。そのためには、教材のページにサンプルプログラムを組み込んでおくなどをする必要があると考える。1つのページにどれだけの量を表示するかも重要だ。1つのページにサンプルプログラムを表示して、その解説を述べて、練習問題を用意すると、

ページが縦に長くなり過ぎて見づらくなる。なので項目ごとにページを分ける必要がある。

プログラムの説明は、あえてあまり詳しく書かず、使用するのに必要な説明と重要な注意点のみを書くことで、簡潔で読みやすいものにしようと考えた。やはり、知らない用語で説明されていると読む気がなくなってしまうし、このほうが文字の量も少なくてすむと考えた。

(3) 教材の対象

オンライン教材の対象者は福田ゼミの生徒たちとした。当初は誰でも理解できるような教材にしようと考えていたが、HTML や JavaScript の説明まで製作していたら時間が足りないと思い、授業で HTML と JavaScript の両方を勉強している福田ゼミの生徒に利用者を絞って製作することにした。Ajax の勉強は HTML や JavaScript を必然的に使用することになり、ゼミで学んだことを忘れないための復習にもなると考えた。

ゼミの生徒たちを対象とするのであれば、専門用語などを極力使わずにプログラムを説明しなければならない。専門用語を多用すると、その用語の意味がわからない生徒がいる可能性がある。これではわかりやすい教材といえない。今回のオンライン教材の目標は理解しやすさを重視しているので、そのような問題が起きないように製作しなければいけない。

3 教材作成の計画

(1) 勉強方法

まずはユーザの勉強方法を考えた。これは古典的だが実際にユーザが一からプログラムを打ち込んで覚えていく方法が良いと考えた。

なぜなら、私自身が実際に打ち込んで勉強する方法が一番覚えやすかったからだ。サンプルプログラムを見て、説明を読むだけでは、そのときは理解できていても、すぐ忘れてしまうことが多い。楽をしようとコピー&ペーストで、サンプルをそのまま貼り付けてプログラムを実行し、成功すればすぐ次の問題に取り組む人もがいる。そのため、最初のうちは、ペースが速く、どんどん問題を進めていくが、前に勉強したことの応用問題などになると、しっかり理解ができていないためエラーが起き、それに対処できなくなるのだ。これは私自身が4年生になり、プログラミングの授業のアシスタントやっていて、実際に思ったことだ。そのため、製作するオンライン教材では、サンプルプログラムのコピーをできないようにしようと考えた。そうすれば、コピー&ペーストすることができないため、自分で打ち込むしかなくなる。面倒くさくて、やる気のなくなる人もいるかもしれないが、この方法がプログラムを覚えるのに一番効果的と考える。

この方法では、プログラムの打ち間違いによるエラーが多発するかもしれないが、プログラムを正確に打ち込む練習にもつながると考えた。このプログラムを正確に打ち込むというのは大切なことで、特に「,」や「.」などは間違えて入力した場合、後から見直しても発見しにくいので、間違えないように最初から正確打ち込まなければいけない。そのため、やはりこの方法が最適だと考える。

(2) 全体の構成

オンライン教材の内容は、まず TOP ページに何をやるサイトなのか、どのように使うかを示しておく。TOP ページは一番始めに見るページなので、見栄えが悪くならないように、文章は簡潔に記述する。

章ごとの内容は基本編の 1 章、実用編の 2 章という形にする。各章には、いくつかの項目を置き、項目ごとにプログラムの説明をしていく。項目へのリンクは、サイドメニューに一覧で設けておく。これで見たい項目にすぐアクセスできるようになり、教材が使いやすくなる (図 4 参照)。リンク先にはユーザの作成用プログラムを表示しておき、同時に完成図を表示しておく。ページ下部には前後の項目へのリンクと現在の項目の解説ページと練習問題のページへのリンクを設置しておく (図 5 参照)。こうすることで、一つのページが長くなり過ぎず、使いやすくなると思った。解説のページは、プログラムのどこの部分を解説しているかを明確にして、分かりやすいようにする (図 6 参照)。

(3) 教材製作における問題点

計画で最初に問題になったのが、サンプルプログラムをページ上に表示するうえで、コピーを防止する方法であった。初めは HTML の機能でコピーを防止しようと考えたが、方法がわからず、指導教員の福田洋一先生に相談した。そして先生から、「プログラムを画像にして貼り付ければいい」とアドバイスもらった。画像は 3201 教室にあるマックの中のアプリケーション「プレビュー」を使い、Jedit で作成したサンプルプログラムをそのまま画像にした (図 7 参照)。

次の問題は、ユーザが作成したファイルを実行するためにサーバが必要ということだ。Ajax はサーバと HTTP 通信を行ってテキストファ

イルなどを読み込むため、どうしてもサーバが必要となる。この問題は私の所属している福田ゼミの授業で使用している XAMPP という擬似サーバソフトがあるので、これを使用することで解決した。自宅などで勉強する場合は、プロバイダが提供している無料サーバを使用するか、擬似サーバソフトをインストールすることでファイルを実行することができる。

完成図を表示するのにも問題があった。当初は表示しているサンプルプログラムのファイルをページに組み込み、実際に動かせるようにしようと考えていたが、それではブラウザでソースを表示できるため、せっかく考えた、先に述べたコピー防止機能の意味がなくなってしまうため、完成図も画像で表示するのみとした (図 8 参照)。本当ならエラーチェック機能のようなものがあつたほうが便利なのだが、やり方もわからず、難しいそうだったので断念した。その結果、エラーが起きた際は、自分でチェックするしかないのが、この教材の難点となってしまった。

残す問題は、各項目の練習問題の作成だ。私は教職などの勉強をしたことがないので、問題を作るという行為をしたことがない。問題を作成するのは、問題を解くことより難解な作業だった。考えてはみたが、サンプルプログラムを多少変更すればできる程度の簡単な問題を作成するのが限界だった (図 9 参照)。

4 オンライン教材の内容

このオンライン教材を作りにあたって、私自身 Ajax を勉強するのが初めてで、JavaScript の知識を乏しかつたので、Ajax を使う上で必要となる JavaScript の勉強から始めた。この JavaScript の勉強に時間が掛かったため、実際に Ajax の勉強を始めるのが遅れてしまった。

そのため Ajax の勉強をしながら、理解したところのオンライン教材を作成するという、勉強と製作の平行作業となった。このため、膨大な Ajax の手法の中から基本と少しの応用のところまでしか教材を製作することができなかった。これは一意に私の力不足だったと反省している。

オンライン教材の構成は、1 章に基本となる仕組みや技術を説明して、Ajax を使う上で基礎を学べるうようにしている。2 章は、前章で学んだ基礎を元にした実用的なものを学べるようにした。

(1) TOP

TOP ページにこの教材を使う上で、前提として HTML と JavaScript の知識が必要となることや使用するソフト、環境設定などをユーザに指定している。Ajax についての説明は、すでに多くのサイトや参考書で述べてあるので、簡単にどういうものかということだけを説明するようにした。私の経験上、TOP ページに長く説明が書いてあっても読まないことが多く、サイトを見る気がなくなるのでなるべく簡素にしようと考えた (図 10 参照)。

(2) 1-1 HTTP 通信の流れ

1-1 では Ajax を使う上で必ず必要となる HTTP 通信について解説している。HTTP 通信の流れを理解してもらうためにできるだけ単純なプログラムになっている。ここでは HTTP 通信を行い sample.txt の内容を読み込み、それをテキストボックスに自動的に挿入するプログラムになっている。1-1 では HTTP 通信を理解してもらうことのみを考えているのでクロスブラウザ対策はしていない。そのため FireFox な

どの一部のブラウザでしか動作しなくなっている。そのほかの必要な処理も省いてあるので、プログラムとしてはかなり不親切なものになっている。

初めから変数に XMLHttpRequest オブジェクトをセットし、open や send メソッドを使用して、HTTP 通信を行うというのは、理解が難しいかと思ったが、これが Ajax の非同期通信の基幹となるものなので、理解してもらうしかないと考えた。これからずっと使う基礎の部分なので、今理解できなくても項目を進めるうちに理解できると考えた。実際、私自身も最初のうちはなんとなくしか理解できていなかったが、何度もプログラムを作成しているうちに理解できるようになった。

(3) 1-2 ブラウザの互換性

1-2 はほとんど 1-1 と変わらないプログラムだが、前回のプログラムでは動作しなかった Internet Explorer でも動作するようにプログラムを改良してある。if 文を使い条件分けして、様々なブラウザで動作するようにできることが Ajax の利点でもある。

このオンライン教材では、代表的なブラウザである FireFox、Internet Explorer、Safari を対象としているので、そのほかのブラウザを使用した場合のクロスブラウザ対策はせず、エラーを表示するようにした。

今回から使い始めた try、catch 文は説明が難しく、私自身あまり理解できていなかったので説明を省略して、使い方のみを表示しておいた。これは下手に説明するよりも、こうしておけばエラーが起きた時に HttpObjFail 関数が実行されるということがわかればいいと考えた。これならば、混乱することなくこの項目を理解できると考えたからだ。

(4) 1-3 タイムアウト処理

1-3 も前回の改良版で、新たにタイムアウト処理ができるように改良した。HTTP 通信は絶対に正常に動作するとは限らないので、なんらかの原因で応答が返ってこないときに通信を中断してエラーを表示する必要がある。ここではそのための関数を解説している。

新しく二つのメソッドを使い始めたが、`setInterval()` メソッドは、引数に実行する関数名と何ミリ秒毎に繰り返し実行するかを指定するだけで、`clearInterval()` メソッドは実行中の `setInterval()` メソッドをストップさせるという簡単なものなので、詳しい解説でなくとも容易に理解できると考えた。

タイムアウト処理は、サイトを作成して、HTTP 通信を行う際に、必ず必要となってくるものなので、なるべく初期の段階で解説しておく必要があると考えた。

(5) 1-4 エラー表示

1-4 も 1-3 同様、前回までのプログラムの改良版だ。1-4 の改良部分は HTTP 通信を行い、応答が返ってきたときにその応答が正常でなかった場合、エラーを表示するようになっている。エラーの表示方法もただ「エラー」と表示するのでは、なぜエラーが起きたのかわからないので、`HttpObj.status` の中に格納されている数字とテキストを表示して、現在の状態がわかるようにしてある。例としては「404 : Not Found」のように通信を行ったがファイルが存在しない、もしくは見つからないというエラーがある。この場合は通信先の URL が間違っているか、削除されて存在しないなどの問題が考えられる。このようにどういったエラーが起きたのかがわかれば、それに応じて対処できるように

なる。

私は最初「.status」と「.statusText」がどういったものなのか、よくわからなかったが、実際にプログラムを実行し、エラーを表示させて、Web サイトを見ているときに出てくるエラーと同じものだったので理解できるようになった。

Ajax は XMLHttpRequest オブジェクトを使って、HTTP 通信を行うので、通信でエラーが発生しても、ブラウザの機能では何も表示されない。これから複雑なプログラムを作成していく上で、どこがエラーを起しているかわからないと、プログラムの作成に支障が生じる。そのためこのエラー処理も初期の段階で解説しておく必要があると考えた。

(6) 1-5 DOM

1-5 では HTTP 通信から離れ、これから Ajax で操作する要素をどのように特定するか、その方法を理解するためのページとした。内容はボタンをクリックすると、リストが増減するというものだ。

このオンライン教材では、どの要素を操作するかを特定するために、タグに id をつけておき、それを目印に要素を特定するように説明している。これはブラウザの違いによるエラーが起きないようにするためのクロスブラウザ対策だ。この方法以外にも要素を特定する方法はあるが、複数の方法を説明すると、混乱して分かりにくいと考えたため、あえてこの方法のみを説明している。この方法にも欠点があり、タグに id をつけるとき、同じ id をつけるとエラーが起きることがあるが、これは注意するように解説で述べておけば問題はないと考えた。

このセクションでは、DOM の説明を省いている。それはこのオンライン教材が JavaScript を勉強していること前提としており、多少なり

とも DOM のことがわかっていると考えたからだ。DOM を説明しようとするブラウザによる DOM ツリーの構造の違いなどを説明しなければいけなくなるが、この項目で解説している要素の特定方法なら、この違いは関係なくなるため、説明しなくても問題ないと考えた。

ノードを作成したり、結合したりする `createElement` や `appendChild` などの動きが少々わかりにくいので、文章による説明だけでなく、図か何かで説明を追加しようと考えていたが、後回しにしており、結局時間がなく実現できなかった。

(7) 1-6 イベント

1-6 はユーザのアクションに対応して様々な動作ができるように、イベントについて解説している。イベントを使用することでマウスを動かしたり、クリックしたり、ボタンを押したり、文字を入力したりしたときに関数などを実行できるようになる。これは Ajax でページを操作する上で重要なものだ。イベントは種類が多く、覚えるのは大変なので、サイドメニューに一覧を表示するリンクを作成しておいた。このリンクはサンプルプログラムを見ているときも見られるように、クリックすると別ウインドウを開いて表示するようにしてある。

今回から使用する `AddListener` 関数と `SetListener` 関数は汎用性の高い関数として作成されており、内容を少し変えるだけで、様々なアクションに対応できる。この 2 つの関数は、これからほとんどのプログラムで使用する重要な関数だ。`SetListener` 関数はそれほど難しくないと思うが、`AddListener` 関数が少々難しいかもしれないと考えた。特に `AddListener` 関数の 4 つ目の引数に `false` を指定しているが、その説明は省略した。ターゲット・フェーズやバブリング・フェーズの説明をし

でも難しいだけであり、今回のオンライン教材では、すべて false を指定しているので、この説明は省略しても問題ないと考えた。

(8) 1-7 マウスイベント

1-7 は 1-6 で解説したイベント中でもマウスに関するイベント、mouseover、mouseup、mousedown、mousemove、mouseout、click イベントで使用することができるイベントオブジェクトプロパティの解説をしている。このマウスイベントはサイトを作成する上で、あまり使用しないものが多い気がしたが、マウスの位置でイベントが起きたときに、その id を取得する target などは使用するかもしれないと思い製作した。

e.type などの e. の意味の説明をしていないが、イベントオブジェクトプロパティすべてについて使用しているので、これは解説なしでも使い方は理解できると考えた。

(9) 1-8 キーイベント

1-8 も 1-6 で解説したイベントの中のキーボードに関するイベント、keyup、keydown、keypress イベントで使用することができるイベントオブジェクトプロパティを解説している。このキーイベントも使用する機会が少ないかと思ったが、入力されたキーのコードを返す keyCode は使えるかもしれないと思い、製作した。Math.random() の解説はしていないが、それほど使用頻度が多いものでもないと考え省略した。

(10) 2-1 Ajax

2-1 からは 1 章のように基本的なことではなく、より実用的な Ajax の手法を解説している。やっていることは 1-1 と同じでテキストを書き

換えるだけのプログラムだが、JavaScript はすべて js ファイルに書き、それを html ファイルで読み込むようにしてある。こうすることでページ本体の html ファイルを簡潔に書くことができる。さらに同じ js ファイルを読み込ませれば、ほかのページでも同じ動作ができ、1 章のときよりも汎用性が高くなった。

2-1 で理解に苦しんだのが無名関数の存在だった。この関数は、サーバから帰ってきたデータの動きを把握するのが難しく理解するのが困難だった。時間をかけ理解できたが、次はそれをどう説明すればわかりやすいかを考えるのに苦労し、わかりづらい言葉を使わずにデータの動きを解説することにした。実際に誰かにプログラムを作ってもらって、わかりやすいかテストできなかつたので、これで本当にわかりやすいのかがわかんないのが難点だった。

(11) 2-2 ラジオボタン

2-2 は今までやっていた、ただテキストを書き換えるだけのプログラムをもっと実用的にした手法を解説している。今までは、テキストファイルの文章をただ書き換え表示していただけだったが、実際はタイトルをつけたり、色や字体を変えたりと、文章を装飾している場合が多い。今までは、タグを書いて文章を書いてもタグも文字として認識されていたため、そのまま表示された。だが innerHTML を使うことにより、タグをタグとして認識するようになり、装飾されたテキストを表示できるようになる。

ここまできると私自身が Ajax に慣れてきたので、詰まることなく理解できた。この innerHTML は便利で、製作中だったオンライン教材も、innerHTML を使って、Ajax で製作し直そうかと考えたが、製作

の期限が迫っており、時間的に余裕がなかったため断念した。

(12) 2-3 テキストボックスの入力補助

2-3 は今まで Ajax を勉強してきて一番理解するのに時間がかかった。内容は Yahoo! の検索機能や Amazon の商品検索のような、テキストボックスに何かが入力されたら、その候補をリスト表示するものだ。今回のオンライン教材ではデータベースを使用していないので、候補リストをテキストファイルで作成している。

このプログラムは関数が多く、まずどの関数がどこで使用されているのか把握するのに時間がかかった。テキストファイルを分割して表示するのは、ほとんど今までの復習だったので理解が早かったが、そのテキストをリスト表示するためのリストサイズの指定が難しかった。offsetTop や offsetHeight など初めて見るものが多く、その理解に苦労した。

解説も関数が多い分長くなってしまい、見にくくなってしまった。どうすればもっと簡潔に、わかりやすく説明できるか考えたが、私の知識不足と時間的余裕がなかったため、改善できなかった。

5 まとめ

(1) アンケートの結果と問題点

想像以上に時間を費やし、妥協点や問題のある部分もあったが、なんとかベータ版を作成し、福田ゼミの3回生の男女9人に、実際にオンライン教材のベータテストをしてもらった。使用してみたの感想やこれからの課題点などをアンケートで答えてもらったが、時間がなかったため、短い時間しかテストしてもらえず、実際にファイルを作成してもら

えなかった。本当なら、じっくり時間をかけて、サンプルプログラム一つ作成してもらい、作成しやすかったか、作成したプログラムを理解できたかなどの意見をもらいたかった。

だが、短い時間でも全体の見やすさや、使いやすさを見てもらい、良かった点や悪かった点、ほかに改善してほしい点などを答えてもらった。自分では使いやすいと思っていたところが、他人にしては使いにくかったり、見落としていた部分への指摘をされたりなど、とても参考になる意見をもらえた。

主な評価は以下の通りであった。

良い評価

- ・ コピー防止機能が教材らしくて良いと思う
- ・ 細かく説明してあるのでわかりやすかった
- ・ 完成図の画像が貼ってあるのでわかりやすい

悪い評価・改善点

- ・ sample ファイルの画像の重要部分にカラーリングしてほしい
- ・ サイドメニューの内容が説明不足でわかりにくい
- ・ スクロールと一緒にサイドメニューも動いてほしい
- ・ 解説へのリンクをもっと強調するか、どこにあるか最初に教えてほしい
- ・ 例題をもっと具体的にしてほしい

評価は、全員が大体同じところが良いと感じ、悪いと感じたようだった。この評価を元に今後の課題点が明確になった。すべての箇所を意見の通りに改善することは、時間的にも知識的にも厳しかったので、出来るところから改善していった。

サンプルプログラムを画像化して、コピーできないようにした、コ

ピー防止機能は思った以上に好評であった。だが、画像が微妙にズレているのが気になるという意見もあった。これはプログラムを画像化する際、分割したプログラムの画像のサイズが微妙にズレていたためだ。「プレビュー」を使ってプログラムを画像化するとき、長いプログラムは分割しないと画像化できなかったのだが、そのときのサイズを指定するときにズレたのが原因である。改善しようかと考えたが、画像の数が多く、時間的に苦しいため断念した。画像はズレてはいるものの、特に支障なく見られるので問題はないと考える。

リンクの強調は、css を変更すれば、すぐに可能だが、ただ大きくするだけや、色を変えるだけでは不十分だと感じるため、TOP ページに説明文を載せるなどして、わかりやすいように改善するほうが、当初の目標にあったものになると考える。

悪い評価のなかで、サイドメニューを画面と共にスクロールさせてほしいというものがあつた。これは方法がわからなかったので、福田洋一先生に質問して教えてもらった。frameset タグを使用すればできるとわかったが、この方法はあまり推奨されていないらしく、作業量も多いため断念した。そのかわり、a タグを使い、ページ下部にページ上部に戻るリンクを設置し、少しでも使いやすいうように改善した (図 11 参照)。

ほかに練習問題をもっと具体的にしてほしいという意見もあつた。この意見はもつともなものだと思う。教材である以上、練習問題は必要不可欠なものだが、各項目に 1 問しか問題がない。しかし、それを考えるには、私のプログラムの知識では思いつかなかつた。これは私の力不足による問題により実現が難しかった。

(2) 今後の課題

アンケートに協力してくれた 3 回生たちから、オンライン教材の内容に関しては、それほど指摘を受けず、初めての Ajax だったが理解しやすかったと言う意見をもらえた。しかし、見易さや利便性に関しては課題が残った。ベータテストをしてみて、すべての人に見やすく、使いやすいというのはやはり無理があり、多くの意見をもらい、その中から妥協点を探していかなければいけないと考えた。

一番指摘を受けたサイドメニューは、すべてのファイルを html で作成していたため、更新する際にすべてのファイルを書き換えなければいけないという問題があった。製作を始めた当初は、形だけのサイドメニューにしていたので問題なかったが、後半になるにつれ、更新するのが大変な作業量となった。これの改善策としては php を使い、一つのサイドメニューファイルを、require() で全ページに読み込ませる方法が最適と考える。また、サイドメニューのスクロールに関する問題も考えなければいけない。frameset タグを使用せずにサイドメニューとメインページを独立させる。もしくはメインページのスクロールに合わせてサイドメニューをスクロールさせる必要があると考える。

練習問題については、問題の追加は勿論のこと、解答例が必要な問題には、その画像も表示して問題に取り組みやすくする必要がある。

各項目の解説は、もっとわかりやすくする方法がないか、考える必要がある。図を表示して視覚的にわかりやすくする、新しい関数やメソッドなどにリンクを貼り、詳しい解説も見ることができるようになるなど、より良い解説方法を考えなければいけない。

今回製作したオンライン教材の内容は Ajax でできることのほんの一部だけなので、この先さらに内容を増やすとしたら、膨大な時間が必要

となってくる。なぜなら、対応ブラウザを増やすだけでも、そのクロスブラウザ対策が必要になってくる場合があるし、Ajax で可能なことが大量にあるからだ。すべてを解説しているオンライン教材にできれば一番だが、私の知識ではできそうにないため、妥協するしかなかった。理想はあるが、オンライン教材のタイトル通り「Ajax の基本」を追求し、よりわかりやすいオンライン教材にすることのほうが、現実的と考える。

(3) 自己評価

オンライン教材を製作しての反省点は、最初に Ajax をテーマにしようと決めたときに、基本的な部分は HTML と JavaScript なのですから理解できると軽く考えて始めてしまったことである。いろいろなサイトや参考書など読んで勉強しているうちに徐々に理解が難しいところが出てきた。そのたびに作業が中断し、想定外の時間がかかった。そのため、途中から教材のページに載せるためのサンプルプログラムが正常動くことを重視して製作するようになった。だがそのかわり、プログラムに対する理解度が低下し、解説の部分が後回しになることが多くなった。当初の予定より製作状況に遅れが生じてきたため、規模を縮小して完成を急ぐかたちとなってしまった。

このようになったのは、私自身の認識の甘さや計画性の低さが原因として起こった問題である。Ajax を卒業製作に選んだときか、最低でも製作を始めた段階でもっと下調べをおこなっておけば、防げた事態であった。加えて私の知識不足もあり、計画通り実現できない部分もあった。これから先、今回のようなことがあったときは、残り時間をしっかりと頭にいれて計画をねり、実行することが重要と考える。

—Ajaxの初心者用オンライン教材の作成について—

またプログラムを解説を述べるにあたって、どこまで解説すれば十分なのか認識するのが難しかった。他者と自分の知識の差があるので、どれだけ解説すれば相手にとって適量なのか、他人の目線にたつことが困難であった。また、それも不特定多数が見るものなので、その中でも妥協点を探さなければいけなかった。前半では自分も初心者だったので、ほぼ同じ目線で解説できたと思うが、後半は気を抜くと必要な説明を記述し忘れていたときがあった。他人のために何かを製作するの大変なことだと、改めて実感できた。

製作物の完成度は、後に自分が見ても理解でき、なんとか使用できるレベルにはなっていると考える。ブラウザによって見た目が変わるが、サンプルプログラムや解説のページは問題なく見えるので、内容自体には問題はない。初めて教材というものを製作したが、自分用としては満足できるものができたと考える。

注

- (1) <http://map.yahoo.co.jp/>
- (2) <http://maps.google.co.jp/>
- (3) <http://www.amazon.co.jp/>
- (4) 入門 Ajax 増補改訂版 p4 より一部引用
- (5) AJAX Web アプリケーション アイデアブック p12 より一部引用

文献表

高橋登史朗 2004 『Ajax 入門』ソフトバンク クリエイティブ

高橋登史朗 2006 『Ajax 入門 増補改訂版』ソフトバンク クリエイティブ

羽田野太已 2005 『AJAX Web アプリケーション アイデアブック』秀和システム

Ajax を勉強しよう

http://www.openspc2.org/JavaScript/Ajax/Ajax_study/index.html

Ajax 初心者入門講座

<http://ajax.pgtop.net/>

Ajax 入門 | Ajax の基礎と応用の学習 StudyNet

<http://ajax.studynet.jp/>