

Ajax 入門サイトの構築について

青木 仁海

目 次

1	はじめに	1
1	1 製作の理由	1
2	2 対象とする人物	2
2	製作物について	3
1	1 既存の入門サイトとの差異	3
2	2 構想と目標	5
3	3 動作環境	6
4	4 製作物の全体像	7
3	Ajax 入門の内容	8
1	1 はじめに	8
2	2 Lesson 1 Ajax とは	9
3	3 Lesson 2 XMLHttpRequest	10
4	4 Lesson 3 サーバーサイドプログラム	11
5	5 Lesson 4 クロスブラウザ対応	13
6	6 Lesson 5 JSON の利用	14
7	7 Lesson 6 XML の利用	15
8	8 Lesson 7 補足	15
9	9 Lesson 8 ライブラリの利用	16
10	10 リンク集	17
11	11 Google 検索バー	17
4	まとめ	18
1	1 アンケート評価	18
2	2 今後の課題	20
3	3 自己評価	21

1 はじめに

私は「人の役に立つものをつくる」というゼミのテーマに沿って、『初心者向けの Ajax 入門サイト』を制作することにした。Ajax は、JavaScript や PHP のような単一の技術ではなく、それらを組み合わせた複合的な技術である。したがって、Ajax の周辺技術である JavaScript や PHP に触れたことのないユーザーには、理解し難いため、JavaScript・PHP の基本に一通り目を通したレベル以上のユーザーを対象とすることにした。

私自身、Ajax の学習を始めた当時はまさに JavaScript と PHP の基礎を終えたところであった。複合的な技術の特性上、覚えることが多々あり学習には時間を要したが、その分、JavaScript・PHP の知識がより深まり web プログラミングに多少強くなったように感じている。すなわち Ajax を学習することで、JavaScript や PHP といった web 制作に必須ともいえる技術を習得することが可能なのである。Ajax は web 制作の初心者にとっての教材であるため、Ajax の入門サイトを卒業制作とした。このサイトが初心者の web 制作に対する理解を深める一助になると信じている。

(1) 製作の理由

Ajax 入門サイトを制作した理由は、Ajax を学習することで web 制作に必須の技術が身につくからである。私は web 制作に興味があり、常日頃から利用していた GoogleMaps⁽¹⁾ や Amazon⁽²⁾ がどのように制作されているか調べたところ、Ajax という技術が用いられていることを知った。これがきっかけで Ajax の学習を始め、学習を通して JavaScript・PHP・XML といった Ajax を取り巻く周辺技術も学ぶこ

とができた。このように Ajax を学ぶことで web 制作において重要な技術を習得することが可能なのである。

先に述べたものに限らず、例えば bing の地図⁽³⁾やライブドアの livedoor Reader⁽⁴⁾など様々な web サイトで Ajax が用いられており、いまや JavaScript と同じくらい必須の技術といっても過言ではないだろう。これから web 制作を始める上で、遅かれ早かれ Ajax に触れることになるので、初心者のうちから勉強を兼ねて Ajax を学習することが web 制作の近道だと私は考えている。

したがって、私は web 制作に興味がある初心者こそ Ajax を学習して欲しいと考え、web 制作の入門となるようなサイトの構築を卒業制作のテーマとした。

(2) 対象とする人物

Ajax 入門サイトは、JavaScript と PHP の基本に一通り目を通したレベルかつそれ以外のプログラミング言語を学習したことの無いユーザーを対象としている。

なぜなら Ajax は実質的には JavaScript の一機能であるからである。すなわち、Ajax を学ぶ際には JavaScript の知識が必要不可欠なのである。また、サーバーと通信するため、サーバーサイドで利用するプログラム言語の知識も必要である。ただし、本入門では PHP を利用しているが、PHP でないといけないという決まりはない。Python や Perl でも構わないのである。PHP は最もメジャーな言語である点・初心者にも扱い易い簡単な言語である点などから本入門では PHP を採用することにした。

JavaScript や PHP の基本を一から説明しては、キリがないた

め、本入門は JavaScript・PHP の構文やルールを学習済みという前提で制作した。基礎が危うい人や周辺技術に全く触れていない人でも理解できるような多少の工夫は施してあるが、基本的には事前に学習することを推奨している。具体的な工夫内容については後述する。

これらの言語の基本に一通り目を通したレベルというのは、その言語で用いられる構文や規則を知識として知っているという意味である。知識としては知っているが具体的な使い方がまだわかっていない人物が最もこの入門に適した対象である。

2 製作物について

(1) 既存の入門サイトとの差異

Ajax の教材は既にいくつか存在している。私も Ajax 学習時にいくつかの書籍を参考にさせていただいた。参考にした書籍は『10日で学ぶ Ajax 入門』『10日で学ぶ XML 入門』『パーフェクト JavaScript』の三冊である。Ajax に限らず、教材には書籍と web サイトの二種類があり、入門サイトをつくるに際してそれぞれの評価を行なった。

書籍は携帯できるメリットがあるため、移動やちょっとした休憩時などに読むことができ、空いた時間を有効活用することができる。ただし、実際にコンピューターに入力してみないと意味が無いので簡単な予習にしかない。さらに、その時点では理解できないような難しい専門用語を多用している書籍もあった。また、書籍という形式上、ソースコードがページをまたいでしまったり、前のページを確認するたびにページをめくる作業が必要になる。そういった点では書籍は非常に見難く、無用なストレスや時間を要するきらいがある。

入門サイトの場合は書籍と違い携帯することはできないが、近年は情

報端末が発達しており、ipad やスマホといった機器でデスクトップ PC と比べて遜色なく利用することができる。また、書籍と違って動的な表現をすることが可能であるため、より見やすく分かりやすい教材にすることができる。しかし、既存の入門サイトでは、web ならではの特性を生かしているようには思えない。例えば資料編 図 1 のサイトでは、広告が目立ち、肝心の教材部分を圧迫してしまっている。また、どこにならがあるのかがわかりづらく、初心者は混乱してしまうかもしれない。さらに課題を出しているサイトが無く、ユーザーは自身の学習到達度を実感できない可能性がある。ユーザーの意欲を高め、モチベーションを維持させることも教材に求められる重要なテーマではないだろうか。

これらの問題点は、本学の卒業生である中川幸平の 2009 年度卒業論文でも指摘されている。しかし、まだ改善の余地が多いに残されていると考えられる。見やすいレイアウトや課題の作成がなされており、先に述べたような従来の教材の問題点をいくつか改善しているが、サイトの色調が強いため見づらくなっている (資料編 図 2)。また、課題が用意されているが、答えはなくユーザーは自身の回答が正しいのかを判断することができない。さらに、ソースコードと解説が別ページになっているのも問題である。ソースの全体像を確認するたびにページを戻さないといけないため、操作性が非常に悪い。

以上のことから私は、以下の四つの事柄をテーマとして Ajax 入門を制作することにした。

- ・ 専門用語を避けた、簡潔な文章
- ・ 情報の所在が明確でシンプルなデザイン
- ・ 課題を作成し、ユーザーに学習到達度を確認させる
- ・ web サイトならではの動的な表現

プログラミング経験がまだまだ浅い層を対象としているため、専門用語は極力避けることにした。また、簡潔な文章を心がけることでユーザーのモチベーションの維持を図る。さらに情報の所在を明確にする。ユーザーが知りたい情報がどこにあるのかを直感的に把握できるようにすることで、空いた時間に気軽に学習を進められるようにする。そして課題やサンプルなど、web サイト上で完結できるようなシステムを構築する。

上記のテーマを実現することで、既存の教材と差別化を図ることにした。

(2) 構想と目標

Ajax を学習しながら周辺技術である JavaScript・PHP も学べるようなサイトにしようと構想を練った。Ajax の周辺技術は JavaScript・サーバーサイドプログラム (PHP・Perl・Python など)・XML・DOM・HTML・CSS が挙げられる。全て網羅するとなると、膨大な時間がかかるため、HTML・CSS・DOM については割愛することにした。これらの技術は左程難しいものではなく、Ajax 学習と同時進行で行なったとしても問題はないと考えたからである。また、サーバーサイドプログラムは、PHP を採用した。比較的簡単な言語で初心者にも扱い易いためである。

そして、本入門サイトの目標を Ajax 特有の記述である XMLHttpRequest を用いた非同期通信を主軸にした「JavaScript・PHP・XML の利用方法を学習すること」とした。ただし、JavaScript・PHP・XML の三つを理解している人を対象にすると、あまりに限定的すぎるという点から、XML については深く言及せず付録的な立ち位置とした。Ajax

の名前の一部にもなっている XML だが、必須のフォーマットではない。やり取りするファイルは HTML でもプレーンテキストでも構わないのである。それでも十分 Ajax の機能を生かせるし、他のフォーマットを通信する方法がわかれば XML の通信方法も簡単に理解できるからである。

(3) 動作環境

Ajax は各ブラウザによって記述を変える必要がある。非同期通信には XMLHttpRequest を用いると前述したが、これは Internet Explorer(以下 IE) では動作しない。IE の場合は ActiveXObject を使用する。二種類の書きかたを毎回表示しては、無駄に情報量が増え、ページを圧迫してしまうため、本入門では XMLHttpRequest に限定することとした。こちらの場合、Firefox・Safari といった複数のメジャーブラウザに対応しているからである。したがって、動作環境としてのブラウザは Firefox と Safari の二種類とした。Safari で動作するため Mac OS X でも利用が可能である。ActiveXObject に関してはクロスブラウザの方法として説明することにした。

入門サイト自体は HTML で制作されているが、サイト内の個々のサンプルプログラムには PHP を使用したものもあるため、事前にローカルサーバーを起動する必要がある。ローカルサーバーは PHP が使えるものであれば何でも良い。しかし、入門サイトの動作確認は XAMPP で行なったため、XAMPP を推奨することにした。Ajax 自体はローカルサーバーがなくとも動作する。PHP を学ぶ必要がない場合は、ローカルサーバーを起動せずに入門サイトのサーバーサイドプログラムの項だけ避ければよい。

入門サイトは HTML・PHP・CSS・XML を利用しているが共に文字コードは UTF-8 に統一している。

(4) 製作物の全体像

入門サイトは HTML で記述されている。ページレイアウトには CSS を用いており、サイトカラーは青を基調としている。総ページ数は 10 ページである。また、別途サンプルファイルを用意してありそれらは 16 ページある。ページ上部には現在開いているページが表示されるヘッダー、左部にはメニュー、ページ中央にメインの入門が表示されるレイアウトになっている。

具体的な内容は Ajax の説明と XMLHttpRequest の使い方、サーバーサイドプログラムの利用から各種ライブラリの使用法まで Ajax とそれを取り巻く様々な技術を 8 つのテーマに分類している。それぞれのテーマを Lesson という形で区切り、それらをさらに細分化して Step という形で区分している。例えば入門サイトの Lesson 3 のテーマはサーバーサイドプログラムとなっていて、その下の Step 1 は PHP 利用の概要で、Step 2 は外部ファイルに書き込む方法となっている。ページ左のメニューには、最初は Step が表示されていないが、Lesson のリンクをクリックすることで、その Lesson の各 Step が展開されるようになっている。Step のリンクからその Step の位置まで飛ぶことができるため、わずらわしいスクロールは必要なく、操作性が向上している。

Step ごとの内容は、「サンプルプログラム・サンプルプログラムのソース・ソースの説明 + 課題」となっている。3 ステップで Ajax を学習するという形式である。サンプルプログラムのソースは Jedit というテキストエディタに記述したもののスクリーンショットを貼り付けてい

る。ソースの説明段階で、テキストエディタに記された行番号を指定して説明することで、参照しやすくしている。課題は、直接テキストエリアに書き込める仕様になっており、ユーザーが自分でテキストエディタを開く必要はない。

各 Step は同一ページで表示しているため、1 ページあたりの情報量は多く、スクロールが必至だが、Lesson 名の書いたヘッダーが常に画面の上部に位置しているため、自分が今どの Lesson を閲覧しているのかわからなくなることはない。ヘッダーの Lesson 名をクリックすることで、Lesson のトップに移動することができる。

説明文は、初心者を対象としているため、極力専門用語の使用を避け、簡潔な文章を心がけ、直感的に扱えるサイトにする予定であった。しかし、アンケート等の結果を受けて、最終的に専門用語の解説を加えることにした。Lesson 8 を終えた頃には Ajax の基本をマスターしていることと思う。

3 Ajax 入門の内容

(1) はじめに

このページのファイル名を index とし、ここが起点であることを明示している。このページでは、本サイトが Ajax 入門であること、JavaScript の基礎を学習した人を対象にしていることなどといった趣旨と対象とする人物を記述している。また、こういった核心的な部分を太字で表しているため、読み流しでも理解できる文章となっている。さらに、Ajax を使ったサンプルゲームを配置している (資料編 図 3)。まずは Ajax というものに触れてもらうことを第一とし、ユーザーの興味をひきつけることを考えた。

(2) Lesson 1 Ajax とは

Lesson 1 は 3 つの Step で構成されている。Step 1 では本入門で前提としている開発環境を提示している。開発環境はブラウザ・文字コード・サーバーソフト・サーバーサイドプログラムの 4 項目あり、それぞれ UTF-8・FireFox、Safari・XAMPP、AN HTTPD・PHP としている。当初は使用するブラウザやソフトの名前だけを載せていたが、それのみだと誤解やトラブルを招く可能性があるため、それぞれが何故必要なのか、何故本入門はそれを選択したのかの説明文を付け加えた(資料編 図 4)。Step 2 では Ajax の概要を述べている。まず Ajax は「Asynchronous JavaScript + XML」の略称であることを述べた。厳密な定義を知らなくても、Ajax は利用できるのもので、必要最低限の部分だけを記述している。次に、従来の方を使ったプログラムと Ajax を使ったプログラム(資料編 図 5)を比較して、Ajax を使用するメリットを説明した。

Step 3 では Ajax の基幹となる XMLHttpRequest が行なっている HTTP 通信の解説をしている。HTTP 通信とは通信プロトコルの一種であることを例えを交えながら説明した。また、Ajax は複数の技術を孕んでいるということを理解してもらうために、図(資料編 図 6)を用意した。さらに、HTTP 通信の用語であるリクエスト・レスポンスの解説もしている。

HTTP 通信方法には同期通信と非同期通信の 2 種類があるが、それに関しては Lesson 7 の Step 1 にて詳しく解説すると記述している。

(3) Lesson 2 XMLHttpRequest

Lesson 2 は 4 つの Step で構成されている。Step 1 はサンプル (資料編 図 7) を基に XMLHttpRequest の基本的な使い方を解説している。XMLHttpRequest オブジェクトの生成からレスポンス受信時の処理・リクエストの送信方法までを解説している。JavaScript の文法についても一緒に解説していたが、文章量が必要以上に長くなってしまったため、ボタンを押すと解説が展開される形にした (資料編 図 8)。これによりページのシンプルさを保つことができた。これ以降の解説でも同様のシステムを導入している。サンプルはボタンを押すとリクエストが送信され、ボタン下部の p 要素のテキストがレスポンスに書き換わる。という単純なものにしている。シンプルのほうが理解もし易いからである。リクエストで取得するファイルはテキストファイルとした。XML ファイルについては Lesson 6 で解説するので、これ以降も特別な理由がない限りテキストファイルを扱うようにしている。この形式ならば余計な知識が必要ないからである。

Step 2 では HTML ファイルにリクエストを送信する方法を解説している。リクエストの送信先が変わるだけで送信方法は前回と同様である。サンプルはテーブル要素が書かれた HTML ファイルを取得し、それを div 要素に書き込む仕組みになっている。最初はテーブル要素が書かれたテキストファイルにリクエストを送信していたが、それでは様々なファイルを取得できることを理解してもらえない可能性があるため、HTML ファイルに書き換えた。このほうがより直感的に理解できるだろうという考えもあった。

Step 3・4 では XMLHttpRequest の使い方ではなく、JavaScript のテクニックを紹介している。Step 3 では引数の使い方を解説している。

リクエスト送信用の関数に引数を用意し、それをリクエスト送信先に指定することで動的に送信先を変更することができる。という解説を行った。Step 4 では前 Step の応用編のようなもので、引数と JavaScript の特徴である関数を変数のように扱えるという仕様を組み合わせたテクニックを解説している。関数を変数に代入し、その変数を引数として関数に渡すことでより動的なプログラムを作成できるという内容である。

Step 3・4 は最初は作成する予定ではなかった。しかし、サイトを制作する上で実際に活用している実用的なテクニックなので、初心者の役に立つだろうと考え、急遽作成することにした。

(4) Lesson 3 サーバーサイドプログラム

Lesson 3 は 2 つの Step で構成されている。Step 1 では PHP にパラメータを渡す方法を解説している。サンプルはボタンを押すとリクエストが PHP に送信され、PHP の中でパラメータを受け取り、処理される仕様となっている。複数のボタンがあり、ボタンによってレスポンスが変わる内容となっている (資料編 図 9)。クライアント側の処理は前 Lesson で行ったやり方とほとんど変わらないので、この Step では主に PHP 側の処理について詳しく解説している。具体的にはパラメータの受け取り方や条件分岐に使う switch 文の説明を行っている。また、この Step では課題を用意した。同 Step のサンプルと同じようなものを作るという内容である。パラメータの送信も XMLHttpRequest で行なうため、前レッスンで学んだ記述とほぼ同じではあるが、繰り返し入力することを通じて記述方法を暗記してもらうために、課題では一から XMLHttpRequest を記述してもらう仕様にした。

Step2 では PHP を使って外部ファイルにデータを書き込む方法を

解説している。データの書き込みは JavaScript では不可能なので、この方法を学ぶことは非常に重要である。データ書き込みのサンプルは、input 要素のテキストに入力した文字列を外部のテキストファイルに書き込むという内容になっている。また、対象のテキストファイルを参照するボタンも用意しており、データの書き込みに成功したのかその場でわかる内容になっている (資料編 図 10)。データを書き込む方法は PHP で fopen 関数を使い、ファイルを開いた後、fwrite 関数の引数に書き込みたい文字列を記述すればいい。しかし、そこで問題が発生した。データを上書きする際に、新規のデータが既存のデータよりも文字数が少ない場合、文字数分だけ上書きされるが、それ以降はそのまま残されてしまう。例えば「マヨネーズ」という文字列の書かれたテキストファイルに「ソース」という文字列を上書きした場合、「ソースーズ」となってしまう。既存の文字列を削除してから新規の文字列を挿入すればいいのだが、やり方がわからず、結局入力した文字列に空白を加えることで文字数を増やしファイル全体を上書きしたように見せるようにした。

サンプルにおいて、テキストファイルの参照とデータの上書きは同一の PHP ファイルで行なっている。ファイルを読み取るのか書き込むのかを if 文で条件分岐させ、読み取りの場合は fopen 関数の第二引数を r とし、書き込みの場合は w とした。しかし、リファレンスを確認したところ第二引数を r+ とすれば読み書き両方に対応できることがわかり、修正した。これにより、ソースコードが短くなり可読性が向上した。

ところが担当教員から書き込みには r+ を使わず w を使うべきだと指摘を受けた。先に述べた文字列の上書きに関するバグは、r+ が原因だった。当初のやり方どおり、r+ の使用をやめて r と w に分けることでサンプルが正常に動作するようになった。

当初は Step 3 を用意し、そこで PHP を使ったデータベースへのアクセスを解説していたが、Ajax では必須ではない上、データベースの導入や使用に際して無用なトラブルが発生する可能性があるため廃止した。

(5) Lesson 4 クロスブラウザ対応

この Lesson は 2 つの Step で構成されている。この Lesson では複数のブラウザに Ajax を対応させる方法を解説している。代表的なブラウザは FireFox(以下 FX)・Safari・InternetExplorer(以下 IE) などが挙げられるが、前者二つと後者では記述方法を変える必要がある。本入門では FX と Safari に対応した記述をしている。Mac OS X ユーザーでも使用できるからである。また、私が主に Mac OS X を使って開発を行っていたことも背景にある。

Step 1 では IE に Ajax を対応させる方法を解説している。IE では XMLHttpRequest ではなく AcitveXObject を使用する。したがって、この Step では ActiveXObject を生成する・レスポンス受信時に実行する ActiveXObject 特有の処理・レスポンス受信時に呼び出す関数を作成する際の注意点について解説した。

XMLHttpRequest の onload ではリクエストの送信が正常に成功した場合、レスポンス受信時の処理を実行するようになっているのに対して、ActiveXObject では成否の判断による処理はプログラマーが実装しなければならなかった。そこで一つ問題がおきた。条件分岐は `if (request.readyState == 4 && request.status == 200)` という記述を行っており、リクエストが正常に完了した際の `status` の値は 200 になるはずだが、実行してみたところ反応が無かった。リクエスト送信時

の status の値を alert 関数で表示してみたところ、0 となっていた。これはサーバーを立てていないオフラインの環境だと起こる問題であった。ローカルサーバーを立ててもう一度同じ処理を実行した結果、200 の値が得られた。

Step 2 では FX・Safari と IE の両方に対応させる方法を解説している。両方のブラウザに対応させるにはそれぞれに対応した記述を書き、実行する必要がある。それには try-catch 文を使用するため、その解説をこの Step で行なった。また、課題も用意した。この Step での課題はこれまでとは違い、複数のブラウザで動作確認をする必要があるので、テキストエリアではなく、ユーザー自身にテキストファイルを用意してもらうことにした。一からコードを記述しなければならないので、これまでの学習で得た知識をフル活用する必要がある。自身の学習到達度を判断する機会になるだろうと考えからこのような課題を用意した。

(6) Lesson 5 JSON の利用

Lesson 5 は 2 つの Step で構成されている。JSON という JavaScript の特殊な記法は Ajax でよく用いられるフォーマットであるため、Ajax の重要なテーマの一つである。Step 1 ではクライアントからサーバーに JSON 文字列を送信する方法を解説している。サンプルは Lesson 3 で学習した内容の応用で、JSON 文字列の入った変数をパラメーターとしてサーバーに送信している。

Step 2 は前 Step とは逆にサーバーから JSON 文字列を受け取る方法を解説している。PHP 特有の記述にはなるが、Ajax プログラムを作成する上で使用する頻度が高い重要なテクニックなので詳しく解説することにした。しかし、サンプルプログラムを作成する上で問題点が見つ

かった。PHP で `json_encode` を使って配列を JSON 文字列に変換を試みたのだが、正常に動作しなかった。リファレンスを確認したところ、`json_encode` を呼び出す前に `header('Content-type:application/json')` と記述し、JSON 文字列を使用することを明示する必要があることがわかった。これは知識として知っていないと困ることなので、解説の項目に付け加えた。

(7) Lesson 6 XML の利用

Lesson 6 では XML ファイルの取得方法を解説している。この Lesson には Step が一つしかない。XMLHttpRequest の使い方は基本的には同じだが、レスポンスが格納される変数が異なる。また JavaScript の処理も、XML にあわせた記述をしなければならない。Step 1 のサンプルは取得した XML ファイルの要素を取り出して、それらをテーブル要素のセルに組み込み、表示している。XML は DOM や JavaScript の複雑な知識が必要とされるため、詳しい解説を行なうと、膨大な情報量になる上、時間的な制約もあったため、この Step はおまけ的な立ち位置とした。Ajax で XML を扱う際の注意点と、JavaScript 側の操作を簡単に説明し、後はさらに学習したいユーザーに向けて、必要な知識や技術の方向性を示すだけにとどまった。具体的には XML を扱う上で DOM という概念が必要不可欠であると述べ、学習したい人は別途入門サイト等を利用するべきであるという旨を記述した。

(8) Lesson 7 補足

Lesson 7 は 2 つの Step で構成されている。この Lesson ではこれまで使用しなかった XMLHttpRequest の機能を補足している。Step 1

では、同期通信の方法を解説している。サンプルは Lesson 3 で使用したものと同じ内容だが、通信方法が同期通信となっている。しかし、単純なプログラムであるため、実行速度が速く、同期されていることを実感できないのが難点である。重いプログラムを作成することを考えたが、良い案が浮かばなかったため、同期通信を用いた例を示しながら、詳しく説明するにとどまった。

Step 2 では、POST 送信について解説している。サンプルでは POST を使ったリクエストの送信を行なっているが、Ajax では POST と GET の違いをユーザーが実感することはできないため、これも詳しい解説をするにとどまった。実際に使用する頻度は少なく、重要度が低いテーマなので解説だけで済ませても良いと考えた。

(9) Lesson 8 ライブラリの利用

Lesson 8 は 2 つの Step で構成されている。この Lesson はメジャーな拡張ライブラリを使った Ajax の利用方法を解説している。Step 1 は prototype.js での Ajax の記述方法を解説している。Step 2 では jQuery での Ajax の記述方法を解説している。この 2 つを選んだ理由は、特にメジャーで web サイトを制作する際に用いることが多いからである。現在入門サイトを利用している初心者も、後にこれらを利用することがあるだろうと考え、代表的なライブラリの紹介を行なった。入門サイトではこれらのライブラリを同一のページに適用しているため、仕様が衝突してしまうことがあった。\$ という記号がそれぞれのライブラリで別の意味をもっており、これを使用すると、期待通りにプログラムが動かなかった。\$ を使わなくてもプログラムの実装はできるので、使うのを避けることにした。そして、解説ではこの部分を注意点として

強調することにした。

(10) リンク集

リンク集では、入門サイトで使用している言語やオブジェクトのリファレンスサイトあるいは入門サイトへのリンクを貼っている。入門サイトを利用中は、これらを参照する機会が多いと考え、リンクは別のタブで開くように設定した。当初は、新規ウィンドウで開く仕様を目指したが、最新のブラウザではリンク要素の新規ウィンドウを開く属性である `target="blank"` を設定しても別のタブで開く仕様となっており実装できなかった。その解決策として JavaScript の `open` 関数を使ったが、ページを参照するたびにウィンドウをアクティブにしなければならず、別のタブを開いた場合の手間と大差ないことから、最終的に別のタブで開く仕様にした。

(11) Google 検索バー

メニューの最下部に Google 検索バーを設置した。入門サイトは全てを網羅しているわけではないので、サイトを見てもわからないことがある場合に、検索してもらえればと考え、Google 検索バーを設置することにした。しかし、検索結果を別ウィンドウあるいはタブで開くことができず、解決策を模索するもうまくいかず断念した。最新のブラウザは URL 欄や付属の検索バーから検索が行なえるので、Google 検索バーを設置する必要性が薄く、最終的に Google 検索バーは廃止することにした。Google 検索バーを設置した当初よりもメニューの項目が増えており、デザイン的な観点からも Google 検索バーの廃止は最善であった。

4 まとめ

(1) アンケート評価

ゼミの4回生2人と3回生5人の計7人にアンケートを行なった。実際に入門サイトを利用してもらい、サイトは見やすいデザインとなっているか。説明文はわかりやすくなっているか。初心者でも理解できる内容になっているか。などの項目に答えてもらった。その結果、主に以下のような回答が見られた。

良い評価

- ・ コードが色づけしてあり見やすい
- ・ リンク集が別のタブで見れて便利だった

悪い評価

- ・ 専門用語が多くてわかりづらいところがある
- ・ 課題が使いにくい

中立的な意見

- ・ アンケート時は時間がなかったが、じっくりサイトを見れば (Ajax を) 理解できると思う

別の入門サイト・リファレンスのリンク集は、入門サイトを閲覧しながら、参照することも多いと考えたため別のタブで開くようにしていた。その結果、良い評価として挙げられ、実用的なものできた満足している。また、コードが見やすいという評価もあり、サイトの扱いやすさという点では優れたものが作れたといえるのではないだろうか。

悪い評価に関しては、専門用語が多いという答えが複数見られた。本入門サイトのテーマに掲げたとおり、専門用語は極力避けたつもりであった。しかしそれはプログラミングの専門用語であり、非同期通信や HTTP 通信といったネットワークなどのプログラミング以外の専門用

語はそのまま使っていた。これは盲点だった。プログラミングの入門サイトなため、プログラミング部分のわかりやすさに気をとられていたのが原因である。アンケート後すぐさま、サイトを見直し、必要に応じて専門用語に解説を加えた。Lesson 1 の Step 3 で行った HTTP 通信の解説や Lesson 7 の Step 1 の同期通信の説明などは、このアンケートを受けて付け加えたものである。修正を加えたため、もう一度アンケートを行ないたかったが、時間が無かったため断念した。しかし、この修正で初心者の理解度が多少なりとも向上するだろうと考えている。

また、本入門独自の実装である課題についても不満の声があった。大きな声としては、ソースを参考にしながら課題に取り組めないというものがあった。入門サイトのサンプルプログラムのソースコードと、課題用のテキストエリアが離れているため、ソースコードを参考にするために画面をスクロールしなければならないのが不便であるという意見であった。確かに、初心者のうちは、何も参照せずにコードを書くのは不可能なので、解答を見ながら課題を書ける仕様にすべきであった。

したがって、このアンケート結果をうけて、課題部分を改善した。具体的には、課題を入力するテキストエリアの下部に、解答ボタンを設置し、そのボタンを押すことで解答が展開されるシステムを実装した。これにより、最初は自力で課題を解き、不明な点や理解が不十分な点があれば、解答を展開し、それを参照しながら課題を進めることができる(資料編 図 11)。

本入門サイトにおける最も重要な部分である Ajax を理解できるかどうか、という点では、概ね理解できるという答えが多かった。アンケートはゼミの時間に行なったため、どうしても時間が限られてしまう。そのため、腰を据えて学習できず、早足になってしまった結果、自

分でも理解できたか、あるいはこれからできるのかの判断が難しいようだった。

(2) 今後の課題

アンケートで指摘された、いくつかのミスや問題点を改善したが、まだ改善すべき部分が2つ残されているのが今後の課題である。一つは入門サイトを表示したウィンドウを縮小すると、レイアウトが崩れて文章を読み取れなくなる問題である。これはメニューの位置がスクロールしても固定されていることが原因である。縦スクロールしてもメニューを利用できる代わりに、横スクロールすると文章の上に重なってしまうのである。これを改善するために、横スクロールをしなくてすむように、要素の幅を % で指定した。しかし、この手法で画面を縮小すると、メニューが縦に伸びてしまい、画面のサイズを越えた位置の内容が見えなくなってしまう。新しい手法を考え縦横スクロールに対応させることが今後の課題である。

もう一つは、課題用のテキストエリア内で TAB を使った空白が使用できない問題である。新しい手法を考えるが思い浮かばず、時間的な制約もあり、拡張ライブラリを探してみたがリンク切れで使用できなかった。

システム部分だけでなく、教材内容においても課題が残った。一つはサンプルがどれもボタンを押したら文字が書きかえられる、alert 関数で表示される、といった画一的なもので広がりがないところである。教材であるからには様々な形に触れられたほうがより知識の幅が広がるのではないだろうか。またそれがユーザーのモチベーション向上にもつながると考えられる。

加えて、web ならではの動的な表現をもっと活用すべきであった。現状では動的なデザインは課題・解説・サンプルでしか活用されていない。ユーザーが何かを入力する機会を課題だけではなく、入門サイトのいたるところに用意すれば、より良い物ができあがったのではないだろうか。そういったある種の楽しさのような要素も盛り込むべきであった。

(3) 自己評価

入門サイト制作の作業に関する反省点は、無計画に作業を進めすぎたところである。Ajax は必要とされる知識が複数あるため学習に時間を要したが、それでも入門サイトの制作期間は十分にあった。しかし、無計画に制作を進めてしまい、途中で方向性を変えたり、多くの時間を要して制作したサンプルゲームを廃止することになったりして、無駄に時間をとられてしまった。その結果、完成が遅れてしまい、アンケートを省みる時間も無かった。アンケートで得た改善点は急ピッチで作業を行い、間に合わせたが、修正版のアンケートを行なうことができなかったため、どこまで改善されたかは推測するしかなかった。また修正によって新たなバグが発生している可能性もないとは言い切れない。制作物の内容については、最初に設定したテーマを比較して考察をすることにした。

専門用語を避けた、簡潔な文章

アンケートでは「専門用語が多くわかりにくい」といった意見が見られた。文章を簡潔することに気をとられ、初心者に対する用語の配慮が欠けていたことが原因だと考えられる。これを受けて、入門サイトの全ページを改め、改善した(資料編 図 12-13)。その際、専門用語を避けるのではなく、解説を入れることにした。そもそもこれくらいなら理解

できるだろうと考えた上で使用していた専門用語なので、避けるよりも解説を入れて学習させたほうが、利用者のためになると考えたからである。しかし、新たに解説を加えたことで、文章量が増えてしまった。そのため、JavaScript の解説を非表示にし、必要に応じてボタンを押すことで展開されるという形にした。この結果、文章を必要最低限に抑えることができた。再アンケートを行なっていないため、どの程度改善されたかは判断が難しいが、自分としては及第点を越えられたと考えている。

情報の所在が明確でシンプルなデザイン

学習内容を 8 つのレッスンとそれぞれに付随するいくつかの Step に区分した。さらに、それらのリンクを常に画面上に表示されるデザインにしたため、情報の所在が明確なデザインを確立できたと考えている。アンケートにおいても、欲しい情報がどこにあるかわかりにくい。という意見は無かった。参照する頻度が高いと考えられるリファレンス等の他サイトへのリンクを別タブで開くようにした点もアンケートで良い評価として挙げられ、テーマどおりの実装ができたと考えている。このテーマについては、満足できる内容に仕上がった。

課題を作成しユーザーに学習到達度を確認させる

課題の実装は行なったが、アンケートでは解答が欲しい、解説を参考にしながら入力したい。といった問題点の指摘が多かった。それを受けて、解答を用意した。加えて、その解答を参照しながら回答を入力できるようにした。もう一つ問題点の指摘があったが、時間的な制約もあり、それは今後の課題とした。しかし、十分実用に足る仕上がりとなったので、満足している。

—Ajax 入門サイトの構築について—

web ならではの動的な表現

解説の展開や課題の入出力等、web ならではの動的な表現を実装することができた。いくつかの課題は残されているが、これだけでも従来の入門サイトとの差別化されていると考えている。

制作当初に設定したテーマを鑑みてみると、それぞれいくつかの課題と、改良の余地は残されているが及第点は越えていると考えている。したがって、十分に実用に足る完成度であり、自己評価としては非常に満足している。

注

- (1) <https://maps.google.co.jp/>
- (2) <http://www.amazon.co.jp/>
- (3) <http://www.bing.com/maps/>
- (4) <http://reader.livedoor.com/>

文献表

漆尾貴義 2007 『10 日で覚える Ajax 入門』 翔泳社

山田祥寛 2004 『10 日で覚える XML 入門』 翔泳社

井上誠一郎 土江拓郎 浜辺将太 2011 『パーフェクト JavaScript』 技術
評論社

中川幸平 『Ajax の初心者用オンライン教材の作成について』 大谷大学
人文情報学科福田ゼミ 2009 年度卒業論文

http://www2.otani.ac.jp/hi/fukuda/thesis/2009_thesis/nakagawa/index.html

Ajax 入門 -Ajax Tower

<http://www.ajaxtower.jp/ini/>