

正規表現のオンライン教材の作成につ  
いて

田中 翔

## 目 次

1	<b>はじめに</b>	1
1	1 制作の動機 . . . . .	1
2	2 誰のために作るのか . . . . .	2
3	3 類似の教材はあったのか . . . . .	3
2	<b>制作に際しての計画</b>	4
1	1 構想 . . . . .	4
2	2 必要な技術 . . . . .	5
3	3 活用した資料、ソフト . . . . .	6
4	4 制作過程 . . . . .	7
3	<b>制作物についての解説</b>	8
1	1 全体の構成 . . . . .	8
2	2 ページ、システムの概観 . . . . .	8
4	<b>個々のページの解説</b>	12
1	1 ユーザのための工夫 . . . . .	12
2	2 メタ文字解説ページについて . . . . .	12
3	3 練習問題ページについて . . . . .	14
4	4 制作過程での問題点 . . . . .	15
5	<b>評価</b>	16
1	1 ベータテストおよびアンケートの実施 . . . . .	16
2	2 アンケートの結果 . . . . .	17
3	3 アンケートを受け改良した点 . . . . .	18
4	4 自己評価 . . . . .	20
5	5 今後の課題 . . . . .	20

## 1 はじめに

私は卒業制作として「正規表現の初心者向けオンラインマニュアル」を制作した。これは、理解するのが難しいと言われている正規表現について解説する、前提知識のない初心者向けマニュアルである。構成は大きく正規表現の部品となるメタ文字の解説と、基礎的な練習問題のページに分かれている。環境やプログラミング言語に依存しない基礎知識が学習できるように制作した。

解説ページは HTML<sup>(1)</sup>、練習問題のページは HTML と JavaScript を使用しており、サーバなどに頼らずインターネットに接続できる環境があればどこでも閲覧が可能である。また、主要な練習問題には解説をつけ、理解が進むよう工夫した。

### (1) 制作の動機

私が所属する福田ゼミのテーマは「人の役に立つウェブサイトを製作する」というものである。ゼミ関連で聴講した授業の中で、Python の日本語化を考えるというものがあつた。Python はオブジェクト指向型のプログラミング言語で、習得が比較的簡単なことから、福田ゼミでもプログラミングの入門として使用してきた。また、欧米では Perl と同じくらいの人気を誇る言語でもある。

初心者にもわかりやすいとされる Python であるが、当然ながらプログラムの記述は英語で行わなくてはならず、どうしても敷居が高くなってしまふ。これを解決するために、教育用言語として Python の記述をできるだけ簡易で自然な日本語で記述できるようにするというものがあつた。

この授業の中で取り上げられたのが、正規表現という技術である。日

本語で入力されたコマンドを本来の英語コマンドに戻す際に正規表現を用いた検索、置換が使用されていた。

私の年度の授業では使用しなかったが、以前のゼミではプログラミングの授業でも使用していた。しかしながら、福田洋一先生によると、正規表現は概念自体が捕らえにくく学生も理解するのがむずかしいという。私自身、正規表現という技術を知ったのは、その授業がはじめてであり、授業の中で話を聞いただけではどのような技術なのかがよくわからなかった。

正規表現の最大の利点は強力な文章処理力である。正規表現を使用すれば、膨大な数のファイルから目的のファイルのみを検索する、プログラミングの際に置換機能を活かしてソースの修正や変更を楽におこなう等が可能である。

ならば、正規表現を基礎から学べる初心者向けの教材があれば、福田ゼミで学ぶ人や、プログラミングを行う人の役に立つのではないかと考え、オンラインマニュアルを制作することにした。

## (2) 誰のために作るのか

いままで正規表現を知らなかった人、正規表現に興味がある初心者に向けたものを制作する。正規表現ではメタ文字という特殊な概念を使用するため、基礎的なプログラミングの知識があればよりいっそう理解が進む物と思われるが、特定のプログラミング言語の知識を必要とはしない。正規表現やプログラミング、文章処理の知識および経験のない初心者でも、教材を一通り学習してもらえば正規表現の基礎的な知識を習得してもらえるようなものを作っていきたい。

### (3) 類似の教材はあったのか

正規表現に関する入門書は何冊かあるものの、今回制作したようなオンライン教材で初心者向けのものはない。また、Java や Perl にも実装されている技術のため、それらのプログラミング言語を解説する書籍やウェブサイトにおいて項目のひとつとして解説されていることが多いが、プログラムの知識がどうしても必要となっており、正規表現の概念自体を解説するものは少ない。

正規表現の入門書で、基礎的な学習の教科書としたのが『入門 正規表現』と『正規表現入門』である。両方とも正規表現をいままで知らなかった初心者向けに作られているため、正規表現の概念やメタ文字の基礎からしっかりと記述しており、Amazon.com の書評での評判も良い。

しかしながら、問題点として後半のページでは特定のプログラミング言語や環境に限定した記述がなされているところあげられる。『入門 正規表現』では Java、『正規表現入門』では Java、Perl、Unix、Windows と、環境ごとに異なった解説が成されている。たしかに、正規表現はプログラミングと組み合わせるとテキストエディタで作業を行うよりも便利に使用できるのであるが、正規表現だけを学習したい人にとっては環境を整えたり、一からプログラミングを習得するのは大変な労力が必要である。

練習問題作成の参考とした「正規表現パズル」(資料1)というページは、良い問題が多いのであるが問題と解説をウェブページに掲載し、実際の回答作業はユーザがテキストエディタ上で行うというものであった。このスタイルでは、ユーザが別途テキストエディタを用意しなくてはならないし、問題を解こうとするたびにウェブページから問題文をテキストエディタにコピー&ペーストしなければならないため手間がかか

る。また、後半になるとかなり難易度の高い問題が多く、初心者には難しいと思われる。

上記のような問題を解決できるように

- ・ 初心者を対象としたものであること
- ・ プログラミング言語やテキストディタなど環境、ツールに依存しないものであること

これらの注意点を踏まえた教材を制作してゆくことを目標とした。

## 2 制作に際しての計画

### (1) 構想

この教材は正規表現のことをまったく知らない人に基礎的な知識をつけてもらうためのものである。そこで、基本的なメタ文字の解説と簡単な練習問題からなる教材を制作することにした。

解説文は項目ごとに分け、とくに正規表現の基礎となるメタ文字をわかりやすく解説する。

解説文を読み進めるだけでは頭に入りづらく、面白くないのでユーザに実際に手を動かして問題を解いてもらえるようなページを製作することにした。問題と解説をウェブページで、回答作業はユーザに用意してもらったテキストエディタで行ってもらう「正規表現パズル」のようなスタイルも考えたが、上で考察したように従来の正規表現教材の欠点は、その多くが環境に依存した作りになっていることが要因である。そのため、ウェブページ上ですべての作業を行えるものにした。

Java や Perl の学習の一項目として正規表現を解説するような場合なら開発環境をユーザが用意できるが、本教材はプログラミングの知識を前提としないためウェブ上で完結するシステムが必要である。当初、

問題の形式は数問の中からラジオボタン等で正解の選択肢を選んでもらう形式も考えたが、実際に手を動かしてもらったほうが学習が進むと考え、ユーザが「回答フォーム」に正規表現を入力すると、「結果フォーム」に正規表現が反映された結果が表示されるというテキスト入力形式を採用した。

また、教材はプログラミング言語や特定のテキストエディタに依存しない、汎用的な基礎知識を学べるものとした。たとえば、補足グループという機能は環境によって使用できない場合があり、教材の問題システムでも使用ができなかったので解説は省略した。

## (2) 必要な技術

まず、当然ながら正規表現の知識が必要である。学習に使用した資料と学習の過程は後述する。

さらにオンライン教材を作る基礎となる HTML が必要である。さらに、レイアウトを整えるために CSS<sup>(2)</sup>も用いた。

練習問題では、ユーザが正規表現を入力するとフォームにその結果が帰ってくるようなものを作るので、そのための技術が必要である。計画段階ではプログラミングの授業で使ったことのある PHP や SQL を使用する予定であったが、福田先生のアドバイスにより JavaScript を使用することにした。

JavaScript を使用する利点はブラウザのみで処理が可能なことである。PHP などではサーバにアクセスが必要であり、もしユーザが不適切な操作を行った場合、サーバに悪影響を及ぼすことも考えられるが JavaScript ではそのような心配はない。また、ブラウザ上で処理することにより、個々の環境に左右されない動作が可能である。そのほかに

も習得が比較的容易である、正規表現が実装されているなどの理由で JavaScript を選択した。

### (3) 活用した資料、ソフト

サイトの構成や練習問題作成の計画を除き、教材の製作作業は大部分が HTML を使用したウェブページ作成である。テキストエディタは J-editX<sup>(3)</sup> を使用し、インターネットブラウザ Safari で確認し問題点を改良していくという作業を繰り返した。J-editX は HTML 構文のカラーリングに対応しており、授業でも扱いなれていたことから使用することにした。

正規表現の学習およびメタ文字解説の参考にした資料は『入門 正規表現』と『正規表現入門』である。

問題作成の参考として『正規表現 ケーススタディブック』と「正規表現パズル」というウェブサイトも使用した。『正規表現 ケーススタディブック』は「郵便番号への一致」や「IPアドレスへの一致」など実生活でも使用できそうな問題が多く、また難易度もあまり高くないものが多く掲載されており、非常に参考になった。

「正規表現パズル」は「EmEditor (エムエディター)」<sup>(4)</sup> というテキストエディタの使用を前提としたつくりになっているが、練習問題自体は J-editX でも動作可能なものであった。このサイトは難易度の高い問題が掲載されており、いくつかの問題を応用編の参考とした。そのほか、福田ゼミのプログラミング演習用テキストで正規表現を扱っているページも参考とさせてもらった。

JavaScript の学習には『JavaScript プログラミング入門』を使用した。



#### (4) 制作過程

私自身が、正規表現については授業で聞く以前は何も知らないまったくの初心者であったため、3 回生の冬頃より基礎からの学習を始めた。『入門 正規表現』や『正規表現入門』、『正規表現 ケーススタディブック』に掲載されている問題や例文を実際に一つずつ J-editX に書き写して、検索・置換機能で正規表現を使用して正しくマッチするかの検証をしていった。そのときに、初心者の視点からわかりにくかった部分や気づいた点をノートにメモしていった。

J-editX に搭載されている正規表現のエンジンは「鬼車（おにぐるま）」<sup>(5)</sup>という正規表現ライブラリが使用されている。これは K.Kosako 氏が製作した日本製正規表現ライブラリで、様々な文字エンコーディングに対応している。

基礎的な学習が一通り終わると、次に問題システムを製作するために必要な JavaScript の知識を『JavaScript プログラミング入門』を使用して学習していった。JavaScript の学習が終わったところで、サイトの制作に取り組み始めた。まず、サイトの構成を以下のように決定した。

1. トップページ
2. 正規表現とは何か
3. メタ文字の解説
4. 練習問題

製作した順序は基本的に 1. 2. 3. 4 と構成順になったが、レイアウトを変更したり、リンクを追加したりするなど、あとから細かく改良を加えたため、かならずしも順序通りではない。

トップページやメタ文字の解説ページをつくりながら、どのような練

習問題を製作するかを考えていった。当初の予定では、練習問題は検索問題だけであったが、正規表現を実際に使用する場面では置換機能も扱うことが多く、そのための練習も必要であると考えたため、検索問題のシステムを改良し置換が可能なシステムを製作することにした。

### 3 制作物についての解説

#### (1) 全体の構成

この章ではサイト全体の構成を解説していく。実際のサイトのキャプチャ画像は資料2、ソースコードは資料3に掲載した。

#### (2) ページ、システムの概観

この教材で使用したファイルは4種類である。以下にその4種類のファイルを示す。

1. 解説用 HTML ファイル
2. CSS ファイル
3. 問題用 HTML、JavaScript ファイル 検索用
4. 問題用 HTML、JavaScript ファイル 置換用

##### (i) 解説用 HTML ファイル

「トップページ」、「正規表現とは」、「メタ文字解説」のページは HTML で記述したファイルである。ソースコードの一例を資料3-1に示す。

##### (ii) CSS ファイル

サイト全体のデザインを整えるために CSS を使用した。製作した CSS ファイルのソースコードは資料3-2である。

### (iii) 問題用 HTML、JavaScript ファイル 検索用

問題のページは HTML と JavaScript で製作した。このファイルでは実際にユーザがフォームに入力した正規表現にあわせ、結果が帰ってくる動的なページである。

検索用のファイルは、問題文の文字列に対し検索のみができるページである。ユーザが問題文を読み、回答フォームに正規表現を入力し回答ボタンを押すと、その正規表現にあわせて検索された後の問題文の文字列が結果フォームに表示される。答えが見たければ、答えのリンクをクリックすると問題の解答と解説が表示される。

この問題システムがどのように作動するのかを解説していく。解説にあわせ、全体のソースコードが一覧できる資料 3-3 も参照していただきたい。

正規表現の入力、及び結果の表示は myForm 内で行われており、正規表現が検索する基となる問題文も myForm 内の content というテキストエリアに格納されている。(資料 3-3 44 行線 49 行)

```
<textarea name="content" rows="4" cols="60">
```

```
ネコ
```

```
犬
```

```
ぞう
```

```
mouse
```

```
</textarea><br>
```

まず、ユーザが文字列を回答フォームである”pattern”に入力し、回答ボタンを押してデータを送信する。(資料 3-3 62 行線 64 行)

```
<input name="pattern" type="text" size="20">
<input name="orgString" type="button" value="回答"
onclick="checkMatch()">
```

問題文の文字列データ `document.myForm.content.value`; は `checkMatch()` で `str1` という変数に変更される。送信されたユーザの回答データは `checkMatch()` ファンクションで受け取られ、`pattern` という変数に変更される。`new RegExp(pattern,"g");` で正規表現オブジェクトを生成する。`g` はフラグと呼ばれ、一番目にマッチした文字列だけではなく、2番目以降にマッチした文字列も検索するという意味である。

`str1.match(reg)` は照合を行い、その結果である文字列を返して変数 `found` に格納している。その後 `found` は変数 `result` へ格納される。(資料3-3 11行線18行)

```
function checkMatch(){
var str1=document.myForm.content.value;
var pattern=document.myForm.pattern.value;
var reg=new RegExp(pattern,"g");
var found = str1.match(reg);
var result = found.join("\n");
document.myForm.kekka.value = result;
}
```

最終的に `result` を `myform` フォーム内のテキストエリア `kekka` の `value` に代入させている。

#### (iv) 問題用 HTML、JavaScript ファイル 置換用

次に置換用の問題ページについて、どのように作動するのかを述べていく。ソースコードは資料 3-4 である。問題文は myForm のテキストエリア content に格納されている。(資料 3-4 46 行繰り S 8 行)

```
<textarea name="content" rows="4" cols="60">
```

近代的なオフィスにはコンピューターは無くてはならない存在です。  
コンピューターの導入は、作業の効率化やペーパーレス化をもたらしました。

```
</textarea>
```

ユーザが回答となる正規表現を pattern フォームに入力したあと、re フォームに置換する文章を入力し、“置換を実行ボタン”を押す。(資料 3-4 68 行繰り V 5 行)

```
[正規表現を入力] <input name="pattern" type="text" size="20">
```

```
[置換する文を入力] <input name="re" type="text" size="20">
```

```
<input name="orgString" type="button" value="置換を実行"  
onclick="tikan1()">
```

データは tikan1 (資料 3-4 11 行繰り 18 行) に送られる。まず、変数 str に問題文である content が代入される。つぎに、変数 pattern にユーザが入力した正規表現である pattern が、置換後の文章である re は変数 re にそれぞれ代入される。RegExp() は正規表現を新たに生成するオブジェクトであり、検索問題と同じである。

Replace() は置換を実行するためのオブジェクトである。Str 中、reg にマッチする部分を re に置換したものを返す機能があり、置換結果を変数 result に代入している。最終的に、result を myform フォーム内のテキストエリア kekka の value に代入させている。

## 4 個々のページの解説

### (1) ユーザのための工夫

この教材は、正規表現初心者のものである。そのため、初心者のための工夫を多く施すことにした。

「正規表現 (Regular expression) とは何か」のページでは、これから学習を始める人向けに、正規表現とは何かということを簡単に解説している

### (2) メタ文字解説ページについて

メタ文字解説の最初のページは正規表現の基本として、問題ページで作成したシステムを使った簡単な問題を掲載している。これは実際に作業することで正規表現の考え方を漠然とでも良いので理解してもらうために作ったものである。正規表現は概念自体が難しく、理解や学習が進まないという問題を回避する狙いもある。(資料2-1)

メタ文字の解説は大きく4つに種類分けしたが、『正規表現入門』や『正規表現 ケーススタディブック』のメタ文字解説ページを見ると種類分けしていない、またはさらに細かい分類を行っているなど、4つの種類は絶対的な概念ではない。しかし、メタ文字という初めて触れる概念を理解してもらうため、使用した入門書の中で最もわかりやすいと思われる『入門 正規表現』の分類を導入した。これは、メタ文字を以下

の4種類に機能別に種類分けしたものである。

1. 文字を指定するメタ文字
2. 量を指定するメタ文字
3. 位置を指定するメタ文字
4. その他の特殊な指定を行うメタ文字

また、正規表現を使用する上での注意点などを最後のページに掲載した。

解説ページではユーザが見やすいように色の指定を行っている。これから解説するメタ文字およびそのメタ文字の機能の解説は緑色、メタ文字を適用する文字列のリストは水色、メタ文字を使用した正規表現はオレンジ色と太文字を使っている。また、正規表現を適用した結果を表示するときはピンク色枠に、マッチした文字は赤色で示している。(資料2-3)

ひとつのページで複数の正規表現を扱う場合はひとつの正規表現ごとに横線で区切り、わかり易くした。また、正規表現の解説ではできるだけ簡潔な表記を心がけ、練習問題ページで関連性の問題があるメタ文字についてはリンクを張って練習が行えるようにし、理解が促進するようにした。解説ページの左横には各解説ページへのリンクを張り、すばやく他の解説ページに移ることができるようにした。

正規表現の基本である、文字を指定するメタ文字では任意の一文字を指定するメタ文字、`[]`を使って具体的な文字種を指定するメタ文字、単語を指定するメタ文字とその逆指定、数字を指定するメタ文字とその逆指定、空白文字を指定するメタ文字とその逆指定を解説した。文字を指定するメタ文字として、Unicodeで任意の文字を指定するメタ文字や、タブ文字を指定するメタ文字なども存在するが、使用頻度が低いと思わ

れるため教材では省略した。

量を指定するメタ文字は三つのページに分けて解説した。一ページ目は文字の繰り返しを指定するメタ文字として0回以上の繰り返しを指定するメタ文字、一回以上の繰り返しを指定するメタ文字、0回または1回を指定するメタ文字を解説した。二ページ目はを使用して具体的な数を指定するメタ文字を解説した。三ページ目は最短マッチについて解説した。

位置を指定するメタ文字も三ページに分けて解説を行った。一ページ目は行頭と行末を指定するメタ文字を解説した。二ページ目は単語境界を指定するメタ文字を、三ページ目は前方参照の指定とその否定、後方参照の指定とその否定を解説した。

特殊な指定をするメタ文字は上記のメタ文字の種類分けには入らない、補助的な働きを行うメタ文字である。この項目は二ページにわたって解説を行った。一ページ目は or 演算子について解説を行った。二ページ目はメタ文字のエスケープと数式の指定について解説を行った。

その他の注意点のページは、直接メタ文字自体を解説するものではないが、正規表現を実際に使っていくうえで使用頻度が高いであろうと思われる事項を、解説ページの最後に掲載した。ここで取り上げたのはカタカナの指定と数字の指定に関する注意点である。

### (3) 練習問題ページについて

練習問題ページは難易度別に 1. 基礎編 2. 基本編 3. 応用編の三つに分かれている。練習問題の問題文には日付への一致や正しい形式の郵便番号への一致など、できるだけ実生活でも役に立ちそうな例を挙げるようにし、ABC や AAA などのあまり意味のない文字列から検索させるよ



うなものは極力避けた。

また、ユーザの便を考え前に戻るやトップへ戻るなどのリンクを充実させた。ページの下部に問題ページなら「前に戻る」「トップへ戻る」「次の問題」へのリンクを、解説ページなら「問題ページへ」「問題トップへ」「次の問題へ」のリンクをそれぞれ設けた。これにより、問題を解いて解答を見たあとすぐに次の問題に移ることが可能である。以前の問題やメタ文字の解説ページで参考にできるページがあれば適宜リンクを張った。(資料2-4、2-5)

基礎編はメタ文字解説ページの正規表現を使えば簡単に解ける問題を11問集めており、メタ文字解説ページの復習的な位置づけである。

基本編は基礎編よりやや難易度の高い問題を7問集めた。複数のメタ文字を組み合わせて回答する問題で、日付や単語を取り出すなど基礎編より実用性が高い問題もある。また、検索の問題だけでなく、単純な置換と、置換を利用しての削除という問題も掲載した。

応用編はこれまでに学習した知識をもとに、通貨表記への一致やファイル名への一致などより難易度の高い11問に挑戦してもらう。そのため、それ以前の問題よりヒントや解説を詳しく記述するように心がけた。

#### (4) 制作過程での問題点

大きな問題となったのが、練習問題ページの動作である。今回教材用に製作した検索・置換システムはテキストエディタや一般的なプログラミング言語で使用されるものに比べて単純なものである。そのため、テキストエディタではきちんとした結果が返ってくるが、練習問題ページで動作の検証を行ってみると意図した結果が帰ってこない問題が

あった。

練習問題を考える際に、テキストエディタではすべての問題について動作するかを検証したのであるが、練習問題ページのシステムでの動作検証を怠ってしまった問題があった。そのため、ページは完成したもののシステムが正常に動かないというトラブルが発生した。問題文を変えたり、JavaScript のコードを書き換えたりして動作が可能になった物もあるが、動作不良が確認された多くの問題が使用できず、問題を削除したり差し替えを行った。

## 5 評価

### (1) ベータテストおよびアンケートの実施

一通りページが完成したところで、ゼミの後輩の3回生8人にベータテストを行ってもらった。3回生の方は、ゼミでプログラミングを勉強してはいる物の、正規表現を扱うのが始めてであったようなので、初心者から見た意見を聞くことが可能であった。ベータテストに際して、いくつかの質問を記したアンケートに答えてもらった。設問は以下の通りである。

1. 一ページあたりの文章量は適切でしたか？
2. 「メタ文字解説」ページの解説はわかりやすかったですか？
3. トップメニューやサイドメニュー、リンクなどはわかりやすかったですか？
4. 「練習問題」ページは使いやすかったですか？
5. 「練習問題」ページの答えの解説はわかり易かったですか？
6. 良かった点
7. 悪かった点

## 8. 改良すべき点

評価形式は「一ページあたりの文章量は適切でしたか？」から「練習問題」ページの答えの解説はわかり易かったですか？までの設問が1（悪い）から5（良い）の五段階評価、それ以降が文章で気づいた点を書いてもらう論述式にした。

### (2) アンケートの結果

アンケートの結果は資料4の通りである。5段階評価の結果を表にまとめた物を見てみると、4点が多く概ね高評価であった。しかし、問3のリンクに関しては3点がおおく必ずしも使いやすいサイトではないことが判明した。

問6の良かった点ではサイトのデザインがシンプル、説明が分かりやすいなどの意見があった。また、練習問題のページについては答えや解説が分かりやすい、問題数が豊富であるなどの意見が寄せられた。これらはメタ文字解説ページ、練習問題ページなど、ページごとにフォーマットを使用しデザインを統一したこと、また初心者向け教材と言うことを念頭に置き解説を充実させたことが原因であると考えられる。

問7の悪かった点では大きく分けてサイトのデザインや構成に関する点、教材の難易度に関する点、製作上のミスに関する点が指摘された。

まず、サイトのデザインに関する点ではサイト全体が左に寄っている、メニューなどのレイアウトが見にくい、スクロールが少し長い、スクロールの一番下に「上に戻るボタン」がほしい、仕切り線をもっとはっきり表示する等があげられた。教材の難易度に関しては、初心者には覚えることが多すぎる、問題が難しい、練習問題をもっとシンプルに等の声が寄せられた。製作上のミスに関しては、問題の解説がないペー

ジがあると指摘された。

問8の改良すべき点でも、おもにサイトのデザインや構成に関する点について改良が必要であるという感想が寄せられた。サイトのデザインや構成に関する点では、トップメニューを大きく表示する、サイドメニューの表記を分かりやすくする、「以前やったものを参考に」というような部分にはリンクを張る、数字という表記は算用数字とする、重要な単語を強調する、簡単な問題から順を追って難しくしてゆくなどの声が寄せられた。

### (3) アンケートを受け改良した点

ベータテストのアンケートを受け改良を行うことにした。まず製作上のミスである、問題の解説がないページについてであるが、該当するページに解説を書き加えた。サイト全体のデザイン、見やすさに付いての意見が多かったので、サイト全体を少し横長にして中央によせた。それにあわせ、横幅が短く文字が読みにくい印象があったトップメニューの幅を延長し見やすくした。

練習問題について難易度が高すぎるという意見を反映することにした。たしかに、ベータ版に掲載していた物の中には、初心者には難易度が高すぎる物もあり、初心者向け教材にはふさわしくないとと思われる状態であった。そこで基礎的な練習問題はそのままにし、難易度が高すぎるとと思われる後半の応用問題を削除して、数問簡単な問題を追加した。

以前やった内容にリンクを張ってほしいという意見には、できるだけ文章中にメタ文字解説ページおよび練習問題ページへのリンクを張って対応した。また、メタ文字解説ページのサイドメニューは「2-1 量指定1」から「量を指定する1」のような表記に変更した。もともと前者

の表記は私がページを製作する際に、管理をしやすくするために番号を振った物でユーザにとっては不要な物である。そのため、より分かりやすい表記に変更をした。

ベータテストの段階では、練習問題の目標は文章で表示していただけなので実際にはどのようなものが表示されれば正解かがわかりにくく、「練習問題のページにおいて、正しい回答をすればどのように表示されるのか、実際に書いてほしい」という意見があった。そこで、メタ文字解説ページで行った形式を使い、正解の記述をテーブルで表示することにした。(資料2-4)

逆に改良を行わなかった点もある。「初心者には覚えることが多すぎる」「スクロールが長い」という意見があったが、この教材は実際には数日をかけて学習を行う用に想定されている。また、一ページあたりの解説や文章量も基本的な事項に絞り、かなり抑えたつもりである。時間がない中ですべての解説ページに一通り目を通してもらったため、実際に学習するよりも多く見えたために寄せられた意見だと考えられる。そのため、ページを新たに製作して項目を分割することのようなことは行わず、各メタ文字解説ページの下部に「トップへ戻る」リンクと「ページ一番上へ戻る」リンクを追加することで操作性を向上させる対策を行った。

「練習問題を簡単な物から徐々に難しい物にしてほしい」という意見があったが、ベータ版でも基本的にそのような順序で問題を載せていた。これもベータテストの時間が30分程度しか無く、3回生の方が十分に内容を理解する前に終わってしまったために寄せられた意見だと考えられ、これに対し改良は行わないことにした。

#### (4) 自己評価

教材を作るうえで、掲げていた目標は初心者の使用を十分に考慮に入れること、エディタやツールに依存しない教材を製作することの2点であった。この目標は達成できたといえるだろうか。

まず、一番目の点についてアンケートの結果を見ると、わかり易い内容であるという感想が多かった。しかし一部の問題が難しすぎるという意見や表示が見にくいという意見もあり、初心者向けでないところもあることが判明した。これらの問題点には、できるだけ改良を施すことにより対応した。2つ目の点について、ページをHTML、システムをJavaScriptで製作することにより多くの環境で使用できるようにしたため、クリアできたといえるだろう。

以上の考察から、当初の目標は概ね達成できたと考える。しかし、教材はまだユーザに実際に使用されたわけではない。そのため、完成版が本当に使いやすいものになっているかはわからない。今後の課題として、実際に多くのユーザに使用していただき、意見を聞きながら改良していく必要があるだろう。

#### (5) 今後の課題

オンライン教材は完成し、問題なく動作するが改良を行うべき点も見えてきた。

まず、問題数の増加である。当初、福田先生からは練習問題を50問は作成するようにと提案されたが、ベータテスト前に出来上がったのは40数問程度であった。さらに、ベータテスト直前に動作不良に気づいたことと、アンケートの結果を受け、問題数を削ったため完成版では30問程度になった。

これは、私の勉強不足と時間の制約によるところが大きい。メタ文字解説の復習的な問題を作るのは簡単であったが、やや難易度の高い、複数のメタ文字を組み合わせたような問題を作ることは難易度が高かった。初心者向け教材であるので、初心者が解けるような難易度であり、かつ製作した問題システムのなかで出題できる問題を考えなくてはならなかったためである。

今回製作した検索問題システムは、ユーザが入力した正規表現に対してマッチした文字列のみを返すものである。しかし、これでは資料5-1のような問題のときに、どこにマッチしたのかわかりにくい。和風料理にマッチさせているつもりが、実際には和風旅館にマッチしているかもしれない。今回の問題では、上記のようなわかりにくい問題はできるだけ避けたが、出題できる問題の幅が狭まることにもなった。

そこで、資料5-2のようにマッチした文字列にのみカラーリングが施されるようなシステムに改良することができれば現在のものよりわかり易く、出題できる問題数も増えるだろう。また、現在の問題システムでは後方参照や補足グループなど正しく動作しない正規表現があり、これを扱えるようにする改良も行うことができればなお良い。

## 注

- (1) HyperText Markup Language
- (2) Cascading Style Sheets
- (3) アートマン 21 が開発、販売するテキストエディタ。  
正規表現による検索、置換機能が充実している。  
[http://www.artman21.com/jp/jedit\\_x/](http://www.artman21.com/jp/jedit_x/)
- (4) 江村豊氏が Microsoft Windows 向けに開発したテキストエディタ。正規表現を用いた検索、置換機能も実装されている。  
<http://jp.emeditor.com/>
- (5) K.Kosako 氏の鬼車のページ <http://www.geocities.jp/kosako3/oniguruma/index.html>

## 文献表

- 大津真 2002 『JavaScript プログラミング入門』 オーム社
- 平田豊 2007 『正規表現入門』 工学社
- ハーシー 2004 『正規表現ケーススタディブック』 毎日コミュニケーションズ
- 岩谷宏 2008 『入門正規表現』 技術評論社
- 明智重蔵 2006 『正規表現パズル』<http://oraclesqlpuzzle.hp.infoseek.co.jp/regex/>