
卒業研究 A テキスト

椋山女学園大学現代マネジメント学部 三木 邦弘
令和3年7月2日版

1	スクレイピング	2	2	卒論の書き方	6
1.1	Web ページの取り込み	2	2.1	卒論用フォルダーについて	6
1.2	HTML の解析	3	2.2	タイトルや要旨など	6

1. スクレイピング

スクレイピング (Web scraping) は既存の Web ページから情報を取り込む技術のことです。そのためには、

1. Web ページの内容を取り込む
2. 取り込んだ内容から必要な部分を取り出す

のようなことができる必要があります。その際に問題となるのは、

- URL が「https://」で始まる場合、SSL の設定ができていないと内容を取り込めない。mars の場合は既に SSL が設定されているので問題ありませんが、ちょっと面倒な話です。
- URL が決まっていない場合がある。毎日新しい情報を提供している場合、日付が URL の中に埋め込まれていることがあります。このような場合、大抵入り口となる web ページは固定の URL となっているので、そこにまずアクセスして、今日の URL を求める必要があります。
- 予告なしに URL が変わることがあります。そして従来の URL へのアクセスができなくなるとエラーが出るので気がつくのですが、従来の URL へのアクセスは可能だが内容の更新がされなくなる、と言うような場合は気づきにくいので困ります。
- ページの構造が時々変わる場合がある。コロナのワクチン接種状況のページが、医療関係者だけ、高齢者も追加、一般の人も追加と対象が広がるに従って変わった例があります。さらにその際に URL も変更になってました。
- 必要な情報が HTML のタグの中に埋もれているので、探し出さなければならない。唯一のタグに囲まれているとすぐに見つかりますが、大抵は無数にある<Td> ~ </Td>の一つに囲まれているのが普通です。
- 必要な情報が JavaScript の実行により動的に生成されている場合は、JavaScript のソースが得られても情報自体は得られない。このような場合は、ブラウザを動かしてその表示内容を取り出すような大変高度な技術が必要になります。

このような様々な技術的な問題もありますし、著作権の問題や多くの人がスクレイピングを行ったために、スクレイピングが禁止となったサイトや情報提供をやめてしまったサイトもあります。通常のブラウザによるアクセスと変わらないような頻度や量であれば問題にはなりにくいと思います。

1.1 Web ページの取り込み

PHP で Web ページの取り込みをするのは、file 関数を用いれば簡単にできます。

```
$lines=file("https://www.mgt.sugiyama-u.ac.jp/");
foreach ($lines as $line) {
    echo str_replace("<","&lt;",$line),"<Br>\n";
}
```

オプションとして「FILE_IGNORE_NEW_LINES」を file() で指定すると一つの変数に取り込まれますが、この例のように指定しない場合は、取り込んだ結果は配列型変数になるので、foreach を利用して 1 行ずつ取り出しています。取り出したものをそのまま echo で表示すると、含まれている HTML のタグが働いてしまうので、「<」を全て「<」に置き換えています。

URL のみでアクセスできる Web ページでない場合があります。

- method として POST を利用してデータを送っている。
- cookie を利用している。

このような場合は、これらの情報を付けた上で取り込まないと、エラーメッセージのみが返ってくる場合も少なくありません。リスト 1 はそのような web ページへの対応方法の例です。

リスト 1 POST のデータや cookie を送る例

```
1  $url="https://sample.php";
2  $param=array(
3      'id' => $id,
4      'pass' => $pass);
5  $options = array(
6      'http' => array(
7          'method' => 'POST',
8          'header' => $cookie,
9          'header' => "User-Agent:Mozilla/5.0 (Windows NT 6.2) AppleWebKit/537.1 (KHTML,
              like Gecko) Chrome/21.0.1180.75 Safari/537.1",
10         'header' => 'Content-type: application/x-www-form-urlencoded',
11         'content' => http_build_query($param),
12     )
13 );
14 $page=file_get_contents($url,false,stream_context_create($options));
```

- 3~4 行目で POST で送るデータの用意をしています。「id」と「pass」が名前で、内容は\$idと\$passに入っていると仮定しています。
- 8 行目で\$cookieに入っているものを cookie として設定していますが、\$cookieの内容は、その前のアクセスの際の web サーバーからの応答の中から取り出しておく必要があります。
- 9 行目ではブラウザを偽装しています。これがないとエラーを返してくるサイトがあります。
- 14 行目で\$page 変数に web ページの内容が入ります。

1.2 HTML の解析

変数の中に取り込んだ web ページの内容から必要な情報を取り出すためには、情報のある場所を示すものを目印にします。例えば「現在のポイント数」と言う文字の後に取得したい情報があるならば、`strpos()`などで「現在のポイント数」の位置を求めます。また web ページの内容には多数の HTML のタグが含まれています。これを目印に取り出すことが考えられます。目的の情報が3つ目の表の1行目にあるならば、3つ目の「<Table>」の後の最初の「<Tr>~</Tr>」にあるはずで、これを単純に「<Tr>」だけを目印に探すのは無理で、「<Table>」を探し、次に「<Tr>」を探すというようなステップを踏む必要があります。

web ページの内容を HTML のタグに分解してくれるライブラリがあります。ここでは「PHP Simple HTML DOM Parser^{*1}」を紹介します。他にも「phpQuery^{*2}」などがあります。これを利用するにはまずこのライブラリをダウンロードする必要があります。「<https://sourceforge.net/projects/simplehtmldom/files/simplehtmldom/>」をブラウザで開くと、一覧が表示され様々なバージョンのものがあることが分かります。2021 年 7 月の時点では「2.0-RC2」が最新のように見えますが、アクセス数を見ると「1.9.1」の方が多いためそちらにします。「1.9.1」をクリックするとこのバージョンのページが表示されるので「simplehtmldom_1.9.1.zip」をクリックしてダウンロードします。Windows の場合ダウンロードしたファイルを右クリックしてメニューで「全て展開」を選択すると解凍することができます。解凍してできたフォルダーの中にある「simple_html_dom.php」をペディターでアップロードします。

*1 <https://simplehtmldom.sourceforge.io/>

*2 <https://code.google.com/archive/p/phpquery/downloads>

「PHP Simple HTML DOM Parser」の簡単な使用例はリスト 2 のようになります。

リスト 2 PHP Simple HTML DOM Parser の簡単な例

```
1 <HTML lang="ja">
2 <Head>
3 <Meta charset="UTF-8">
4 <Title>簡単な例</Title>
5 </Head>
6
7 <Body>
8 <?php
9 include "simple_html_dom.php";
10 $page=file_get_contents("aaa.htm");
11 $html=str_get_html($page);
12 echo $html->find("body",0)->plaintext;
13 ?>
14 </Body>
15 </HTML>
```

- 9 行目でライブラリのファイルを取り込みます。
- 10 行目は web ページの内容を \$page に取り込んでいます。この部分は他のサイトのページを取り込む場合、前章の内容と同様に変更する必要があります。
- 11 行目で web ページの内容を HTML のタグで分解したものを \$html に入れています。
- 12 行目では 1 番目の「Body」タグの内容を取り出し、その中の文字の部分のみを取り出して、echo で表示しています。find() の中のゼロが 1 番目を示しています。

「Body」タグの中の「H1」タグの中身を取り出したい場合は次のようにします。

```
echo $html->find("body",0)->find("h1",0)->plaintext;
```

このように find() を重ねることにより入れ子になった HTML のタグの中の方にあるものを取り出すことができます。また HTML のタグで挟まれたものではなく、タグの中で指定したものを取り出すこともできます。

```
echo $html->find("body",0)->find("a",0)->href;
```

これで A タグで指定した href の値、つまりリンクの URL を取り出すことができます。途中の HTML のタグは、次のように省略することができます。ただしその場合何番目かの数字が変わる可能性があります。

```
echo $html->find("a",0)->href;
```

次のように foreach と組み合わせて、同じタグの内容を全て取り出すことも可能です。

```
foreach($html->find('a') as $e) {
    echo $e->href,"<Br>\n";
}
```

これで web ページ中の A タグに指定された URL が全て表示されます。

同じタグの内容を全て取り出すのではなく、出てきたタグを順番に扱いたい場合はリスト 3 のようにします。

リスト 3 タグを出てきた順に扱う例

```
1 $parent=$html->find('h3',0)->parent();
2 foreach ($parent->children() as $tag) {
3     if ($tag->tag=='h3') {
4         H3 のタグに対する処理
5     }
6     if ($tag->tag=='table') {
7         Table のタグに対する処理
8     }
9 }
```

1. 1 行目では 1 つ目の H3 タグの親となるタグを \$parent に入れています。<X> ~ <Y> ~ </Y> ~ </X> のようにタグ X の中にタグ Y が含まれる時、タグ Y の親はタグ X になります。またタグ X の子はタグ Y になります。表示されているタグを全てという場合は、\$parent に body タグを入れます。
2. 2 行目の繰り返しでは、1 つ目の H3 タグの親タグの子のタグを順番に取り出して \$tag に入れます。
3. 3 行目と 6 行目で取り出したタグが H3 や Table かどうか調べています。

2. 卒論の書き方

この章では卒業論文を L^AT_EX でどのように記述すれば良いかについて説明します。昔卒論を担当していた時は、やはり L^AT_EX で卒業論文を記述してもらいました。卒論用スタイル設定ファイルを用意して、学部指定の形式になるようにし、また全員の原稿を集めて冊子を作成したりしていました。卒業論文は A4 サイズ、冊子の方は B5 サイズでしたが、冊子用スタイルに変更するだけで済みました。L^AT_EX の基本的な使い方は「基礎演習」のテキストを参照してください。ここでは卒論用スタイル設定ファイルによって使える機能などについて説明します。

2.1 卒論用フォルダーについて

mars のデスクトップに「卒論 xxx」という名前のフォルダーを作成し、その中に卒論本文のファイルや画像ファイルを入れてください。xxx の部分は自分の学籍番号の下 3 桁です。冊子を作成する際に全員の原稿データを一箇所に集めるため、このような名前にしておいてください。その中にサンプルファイル (soturion18.tex) をコピーして、その内容を書き換える形で卒業論文を作成してください。ファイル名は変更しないでください。

2.2 タイトルや要旨など

卒業論文の形式は「卒業研究の手引」で次のように決まっています。(1) 指定ファイル表紙、(2) 要旨、(3) 中表紙、(4) 目次、(5) 卒論本文、(6) 参考文献リスト、(7) 指定ファイル裏表紙、となっています。(1) と (7) は現代マネジメント学部指定のファイルを学生会館 2 階の売店で購入して、自分で手書きする必要があります。購入は早めに、タイトルはギリギリで変わることもあるので、提出直前に書いた方が良いでしょう。また (2) は Word のファイルで別途 S*map で提出する必要があり、それも卒業論文と同じ日時が締切なので注意してください。

大昔は「卒業研究の手引」に原稿用紙 50 枚分以上ということが書かれていました。400 字 × 50 枚で 20,000 字に相当する量ですが、現在の「卒業研究の手引」でも 1 ページは 1,200 字程度となっているので 16.6 ページ分となります。その後この決まりが無くなったのは、20,000 字も書けないと言う学生が多いためだと思います。実際のところは図などが入りますし、国際コミュニケーション学部では、卒業論文は原稿用紙 100 枚分以上と言う話を聞いたことがあります。よって本文を 17 ページ以上書くことにします。

それ以外の部分は、卒論用スタイル設定ファイルが適切な形にするので、形式の面で特に注意することはないでしょう。サンプルファイル (リスト 4) に従って、例えばサンプルファイルの「ここに 800 字程度の要旨を書け」とあるところに要旨を入力すれば、要旨が学部の指定の場所に指定の形に出てくるようになります。

リスト 4 卒論サンプルファイル

```

1 \documentclass[sotu18]{jsarticle}
2
3 \title{Web ベースの\mbox{タッチタイピング練習ソフトウェア}}
4 % または
5 %\dttitle{タッチタイピング練習ソフトウェア}{ー ブラウザでどこでも練習できる ー}
6 % 使わない方の先頭に % を付ける
7
8 \author[A18EA999]{梶山 花子}
9
10 \begin{document}
11
12 \maketitle % 表題のページの出力

```

```

13
14 \begin{abstract}
15 ここに 800字程度の要旨を書く
16 \end{abstract}
17
18 \tableofcontents % 目次の出力
19
20 \section{はじめに}
21
22 ここに卒論の本文を書く
23
24 \section{参考文献}
25
26 \begin{thebibliography}{99}
27 \bibitem[latex] Leslie Lamport 著、Edgar Cooke・倉沢良一監訳、
28 「文書処理システム\LaTeX」、株式会社アスキー、1990 年
29 \end{thebibliography}
30
31 \section{プログラムリスト}
32
33 \subsection{入り口のweb ページ: index.php}
34
35 \ListIn{../www/index.php}
36
37 \end{document}

```

- 卒業論文のタイトルは`\title{ }`で指定します。長いタイトルの場合、変なところで改行されることがあります。そのような場合は改行されたくない部分を`\mbox{ }`で囲ってください。サブタイトルがある場合は、`\dtitle{ }{ }`を使ってください。
- 参考文献の後にプログラムリストを付けます。プログラムリストは付録の扱いなので、卒業論文のページ数には入りません。`\ListIn{ }`で取り込むファイルを指定します。
- 本文中でプログラムの説明を書く場合、この本文のずっと後ろにあるプログラムリストを参照する形では分かりにくいものになってしまいます。数行であれば、`\begin{FV} ~ \end{FV}`を利用して次のような形で出すことができます。

```

for ($i=0;$i<100;$i++) {
    echo "I love you.<Br>\n";
}

```

周りの枠が不要であれば、`\begin{V} ~ \end{V}`で次のような形にすることができます。

```


for ($i=0;$i<100;$i++) {
    echo "I love you.<Br>\n";
}

```

- 長いプログラムリストの場合は、説明文中で何行目の事を説明しているのが示さないと、似たような行があると分かりにくいことがあります。そのような場合は次のようにしてください。

```
\begin{lstlisting}[caption=卒論サンプルファイル,label=latex2-samp]
\documentclass[sotu18]{jsarticle}
  中略
\title{Web ベースの\mbox{タッチタイピング練習ソフトウェア}}
\end{document}
\end{lstlisting}
```

これでリスト 4 と同じように先頭に行番号が付いた形で出るようになります。また長いリストでページに入り切らない場合も、そのまま出てくるので余分な空きができません。

- `\btn{Enter}` のように周りを囲いたい場合は、`\btn{Enter}` のように書きます。
-  のように文中にボタンなどの画像を出したい場合は、`\gbtn{ファイル名}` のように書きます。大きな画像でもこれぐらいの高さに縮小されて出ます。