

1. jQuery とは

jQuery は Javascript のライブラリのひとつです。

ライブラリとは、あるスクリプトのよく使われる機能などを部品化して、簡単に使えるようにまとめたものです。Javascript のライブラリである jQuery を使えば、複雑になりがちな Javascript のコードを書かなくても、もっと簡単なコードで Javascript を実行することができるようになります。

例えば、CSS で <p> タグ中のテキストカラーを赤くしたいとします。ネイティブな JavaScript でコードを書くと、以下のようになると思います。

```
<script type="text/javascript">
var tags = document.getElementsByTagName("p");
for(var i=0; tags.length; i++){
    tags.style.color = "red";
}
</script>
```

上記のコードでは、まず、HTML 内に記述されている<p>タグを、document.getElementsByTagName("p") で取得して、見つかった<p>タグのカラーを red にします。これを jQuery で書くと、1 行で書けてしまいます。

```
$("p").css( "color", "red" );
```

今後、各項目での(課題)を実際に作成しながら学習していきましょう。

2. jQuery 設定準備

jQuery は MIT と GPL のデュアルライセンスなので、ライブラリー中の直鎖件表示さえ消さなければ、商用非商用を問わず誰でも自由に利用できます。

jQuery のソースコードは、変数名を短くし、コメントや空白、改行を削除した方法で圧縮されています。通常はこの圧縮された Minified 版を使います。ファイル名が jquery-1.4.2.min.js のように min がついてるのがそうです。ダウンロードしたらサーバーにアップし、script 要素を書いて読み込めます。

```
<script type="text/javascript" src="js/jquery-1.6.4.min.js"></script>
```

script を読み込ませる位置は head 要素内が一般的ですが、body 要素を閉じる直前に読み込ませる方法もあります。これは HTML の描画を早くしたいときに行われます。

2.1. Google から直接呼ぶ方法

ダウンロードせずに Google のサービスを使う方法もあります。リンクを貼るだけで利用することができます。

Google Libraries API – Developer’s Guide – Google Libraries API – Google Code
<http://code.google.com/intl/ja/apis/libraries/devguide.html#jquery>

Script に直接以下の URL をリンク指定することで読み込めます。
<https://ajax.googleapis.com/ajax/libs/jquery/1.4.2/jquery.min.js>

3. jQuery の書き方

それでは実際に、jQuery のコードを見てみます。下記のコードは先ほど紹介した、1 行のコードと基本的には同じです。

```
$("#p").css( "color", "red" );
```

まず（“p”）の部分は、要素を指定するところ。上記のコードでは、HTML 上の全ての <p>タグが対象となります。

それに続く .css(“color”, “red”) の部分では、先ほど指定した要素に対して、何らかの処理を加える部分です。命令と呼ばれています。

(課題) 例の p タグで色が赤になることを確認 js_basic.html

JavaScript の実行タイミングと jQuery の ready 関数

jQuery は script 要素を書いて読み込みます。Web ブラウザはファイルの先頭から 1 行ずつ読み込んでおり、script が読み込まれた段階で、script 要素内に記述された命令を実行します。

Script 要素を head 要素内に記述した場合、HTML ファイルを全部読み込んでいない状態で script が実行されてしまいます。

そこでスクリプトの実行タイミングを HTML 読み込んだ後に制御するのが ready()関数です。

```
$(document).ready(function(){  
    ここに jQuery の処理を書きます  
});
```

これは以下の省略形でも使えます。

```
$(function(){  
    ここに jQuery の処理を書きます  
});
```

(課題) Ready 文で最後に制御されることを、body 部の最初、最後、ready 内、script の最後などに alert 文で表示順を確認 jq_ready.html

3.1. jQuery の基本的な書き方

```
$("#p").css( "color", "red" );
```

```
$(function(){  
    $("#セレクト") . 命令(パラメータ);  
});
```

jQuery の基本的な書き方は、セクタを指定して、メソッドをドット(.)でつなげます。メソッドにはパラメータ(引数)を記述する事もあります。これが jQuery のコードの書き方の基本形です。

4. セクタ

前述した通り、jQuery で指定できるセクタは、CSS のセクタで馴染みのあるものばかりです。jQuery には CSS のセクタエンジンがついてるので、当然と言えば当然ですけど … Javascript の苦手な人にとっては、とっても嬉しい仕様です。

セクタの例

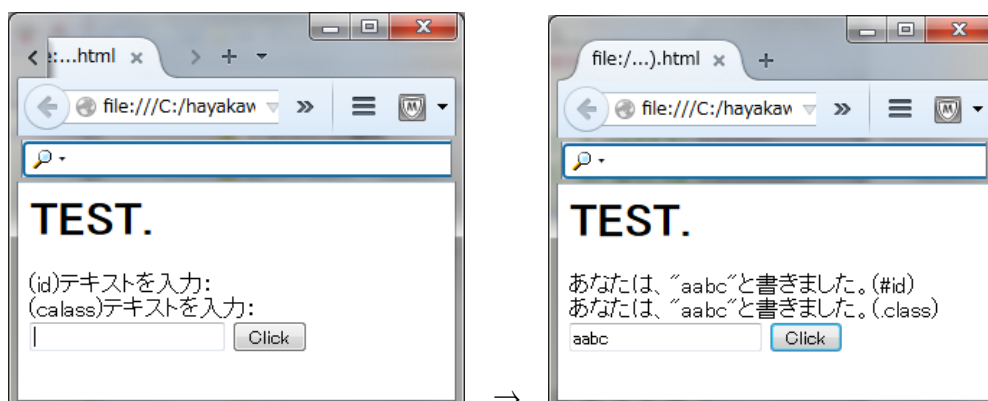
- \$("#navi") … id セクタ
- \$(".navi") … クラスセクタ
- \$("a img") … 子孫セクタ
- \$("p.warnig, p.attention") … グループセクタ

みんな CSS で馴染みのあるものばかりですねー。これならそんなに難くなさそうです。標準の Javascript では、getElementsByTagName や、getElementById で取得していたセクタも、上記のように簡単に指定することができます。

他にも CSS で利用しているセクタがたくさん使えます。IE6 では使用できなかった、隣接セクタや属性セクタも、jQuery では利用できるんです。

(課題)入力した内容をクリックで id、クラスセクタでテキストを表示する
(jq_selector_id_class.html)

クリックした結果は下図のようになる



セレクトアの例

- `$("ul>li")` … チャイルドセレクトア
- `$("ul+li")` … 隣接セレクトア
- `$("h3~h3")` … 間接セレクトア
- `$("a[target='_blank']")` … 属性セレクトア

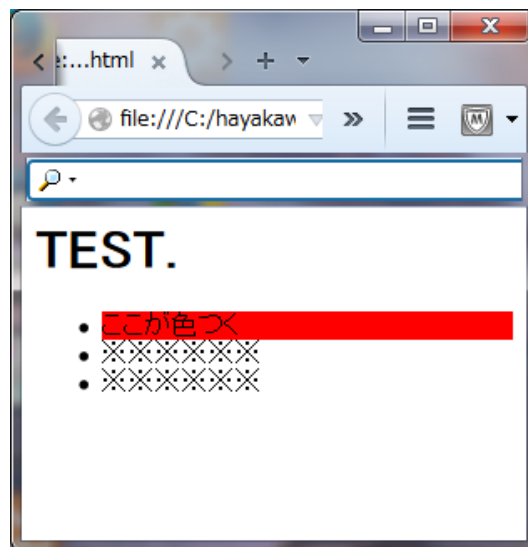
4.1. フィルタ

jQuery 独自のセレクトアとしてフィルタがあります。

フィルタの例

- `$("li:first")` … 一番始めの要素
- `$("li:last")` … 一番最後の要素
- `$("li:even")` … 偶数番目の要素
- `$("li:odd")` … 奇数番目の要素

(課題) `li:first` を使って一番目の要素のバックグラウンドに色を付ける (`jq_filter.html`)



5. 命令

- `$("p").css("color", "red");`
- `$("セレクトア").命令(パラメータ)`

今度は基本的な書き方の後半部分、命令の部分を見ていきます。上記のコード、`.css` の部分が命令、それに続く()内がパラメータ(引数)です。命令は、セレクトで指定した要素に対して、何らかの処理を加える部分です。

また、パラメータは必ずしも必要 … というわけではありません。上記の例では `css` という命令(要素のスタイルシートを変更する命令)にはパラメータが必要ですが、パラメータがなくても OK の命令もあります。

この命令も jQuery ではたくさん用意されています。まずは CSS に慣れている人なら馴染みの深い、CSS 命令、属性命令から見ていきます。

5.1. HTML・CSS の操作

jQuery を利用したスクリプトは

1. どの HTML の要素を操作するか指定する「セクター」
2. 操作する内容(命令)

の 2 つが基本です。jQuery には HTML に含まれるテキストや要素を書き換えたり、CSS プロパティを変更する命令が多数用意されています。

例えばクリックをきっかけ(イベント)に、あるブロック要素の `display` プロパティを `none` から `block` に切り替えることができれば、ページを移動せずとも新しいコンテンツを表示させることができます。

命令の例

- `css()`
`$("#navi a").css("color");`
カラーを取得します。
- `css()`
`$("#navi a").css("color", "red");`
カラーをセットします。
- `css()`
`$("#navi a").css({ "color": "red", "text-decoration": "none" });`
スタイルは複数指定することができます。
- `width()`
`$("a img").width();`
幅を取得します。
- `width()`
`$("a img").width("150px");`
画像の幅に 150px をセットしています。

- `addClass()`
`$("#navi a").addClass("current");`
クラスを追加します。例では `current` というクラスをつけます。
- `attr()`
`$("#a").attr("target","_blank");`
アンカー(<a>タグ)に `target` 属性を追加します。

5.2. テキストの変更と習得

テキストと取得

- `$("#p#first").text("変更後");`
- `<p id="first">変更前</p>`

テキストの取得

- `$("#p#first").text();`

`Text()`の括弧内に何も記述しない場合は、セレクターで指定した要素に含まれるテキストを取得できます。

下記のように組合せて使用出来ます。

- `$("#p#second").text($("#p#first").text("変更後"));`
- `<p id="first">取得する文字</p>`
`<p id="second">変更前</p>`

5.3. HTML の変更と習得

HTML の変更

- `$("#p#first").html("変更後");`
- `<p id="first">変更前</p>`

HTML の取得

- `$("#p#first").html();`

5.4. HTML の挿入

`html()`を使うともともと存在していた要素の内容がすべて上書きされてしまいます。要素の内容を残したまま HTML を追加したいときは、要素を挿入する命令を使います。

- `prepend()` 要素内の先頭に HTML を挿入する
- `append()` 要素内の最後に HTML を挿入する
- `efore()` 要素の前に HTML を挿入する
- `after()` 要素の後に HTML を挿入する

5.5.HTML の移動

HTML の要素を文書内に新たに追加するのではなく、もともと存在していた要素の文書内の位置を変更することも出来ます。

- `prependTo()` 他の要素内の先頭に要素を移動する
- `appendTo()` 他の要素内の最後に要素を移動する
- `insertBefore()` 他の要素の前に要素を移動する
- `insertAfter()` 他の要素の後に要素を移動する

5.6.CSS の制御

CSS の設定

`css()`を使って CSS プロパティを設定するには、括弧内にプロパティ名と値をカンマ区切りで記述します。

- `$("#p").css("color", "red");`

複数の CSS プロパティを同時に設定したいときには以下のように記述します。

CSS のプロパティが複数ある場合

- `$(セレクトター).css({
 プロパティ名:プロパティの値,
 プロパティ名:プロパティの値,
 プロパティ名:プロパティの値
});`

まず、複数指定する場合は、プロパティ名と値はコロン(:)で分けます。CSS の書き方と同じです。それをカンマ(,)で分けて、複数指定する事ができます。基本的に、ブレース({})で囲みます。

- `$("#p").css({
 background-color:"yellow",
 fontWeight:"bold",
 color:"red"
});`

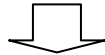
気をつける点は、CSS のプロパティを記述するときとは違い、「background-color」は「background-color」とハイフンに続く文字を大文字にします。

5.7. メソッドチェーン

jQuery のメソッドは、メソッドチェーンといって、ドット(.) でつなげて、複数指定することができます。例えば同じ要素にふたつのメソッドを使いたいとき、別々に書かずに、一度に書く事ができます。

メソッドチェーンの一例

```
/* ふたつのメソッド */  
$("a img").addClass("selected");  
$("a img").width("150px");
```



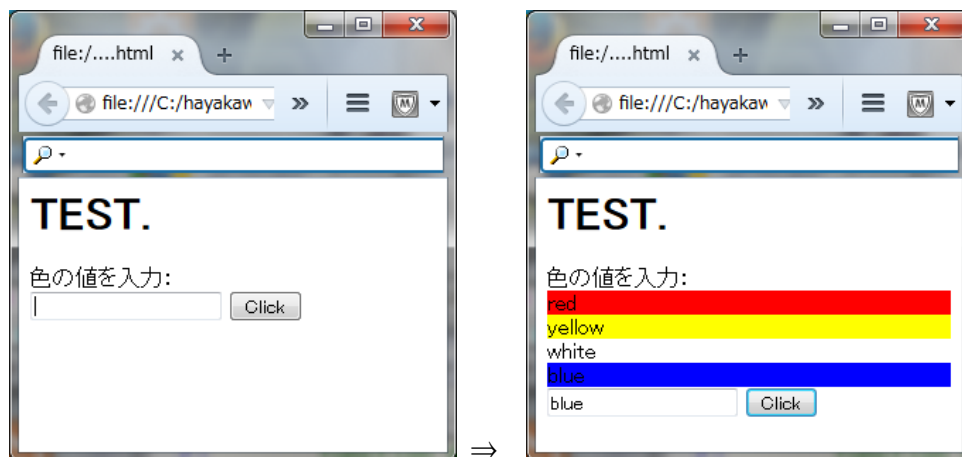
```
/* メソッドチェーン */  
$("a img").addClass("selected").width("150px");
```

上記の例では、クラスを追加する `addClass` メソッドと、幅を指定(または取得)する `width` メソッドを、メソッドチェーンで一度に複数指定しています。

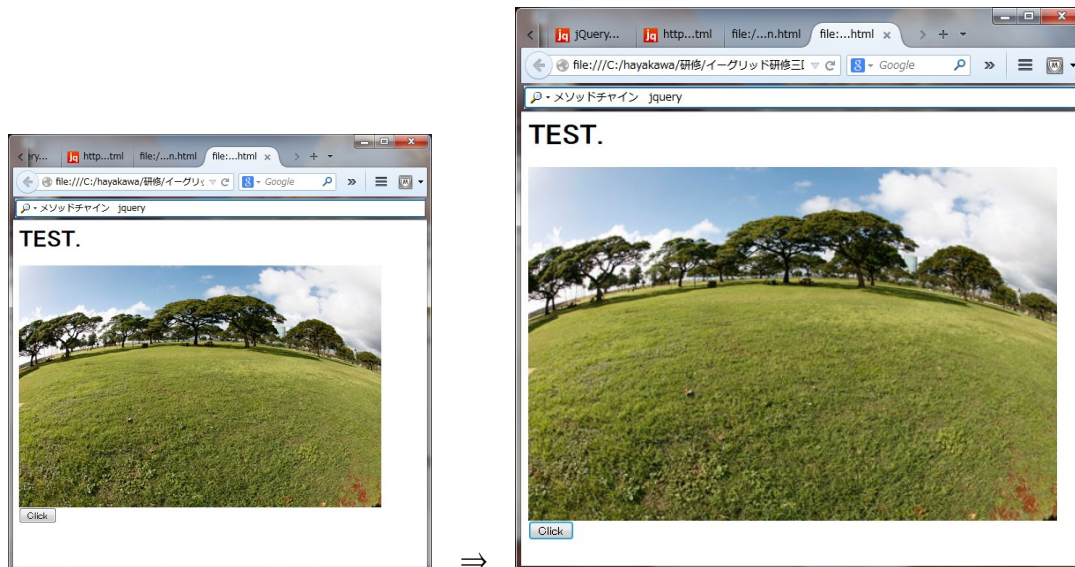
メソッドチェーンを使うと、要素を検索する処理が 1 度で済むので、高速化に繋がります。

(課題) divでの要素の後に `append` で入力された色で文字、バックグラウンド色を追加する

(jq_html_append.html)



(課題) <a img>で表示された画像がクリックで画像が大きく表示される<a img>でのメソッドチェーンで表現 (jq_method_chain.html)



画像の大きさを原画より 100px 分大きくする

(課題) 色々な種類の HTML、CSS、テキストをクリックで変更されることをサンプルで確認する

6. 処理のタイミングを決めるイベント

今までいろいろな命令を見てきましたが、これらを実行するタイミング、きっかけを与えてあげる必要があることがほとんどです。例えばクリックしたときとか、マウスを乗せたとき。そのようなきっかけの事をイベントと言います。

6.1.HTML が読み込まれたタイミングで実行する ready()

ブラウザが Web ページを読み込み終わったと言うきっかけも立派なイベントです。

```
$(function(){  
  読み込まれたときに実行する処理  
});
```

これは以下の省略形です。

```
$(document).ready(function(){  
  読み込まれたときに実行する処理  
})
```

```
$(セレクト).イベント(function(){  
  $(セレクト).命令  
})
```

6.2.他のイベントでも ready()は必須

これから紹介する他のイベントでも必ず ready()の内側に処理を書きます。例えばクリックされたら処理をするという命令もブラウザが Web ページを読み込み終わっていないとセレクトで指定した要素を見つけれられません。そこで、次のように\$(function(){……})の内側に入れていきます。

```
$(function(){  
  $("button").click(function(){  
    button がクリックされた時の処理  
  });  
});
```

6.3.クリックイベントの書き方

要素がクリックされたら処理を行うもっともよく使われる処理です。

```
$(セクタ).click(function(){  
  セクタで指定された要素がクリックされた時に実行する処理  
})
```

具体的には以下ようになります。

```
$(function(){  
  $("button").click(function(){  
    $("img").attr("src","sea.jpg").attr("alt","海");  
  });  
});
```

```
<button>画像を変更</button>  
<p></p>
```

Button 要素がクリックされると img 要素の src 属性の値を「sea.jpg」に、alt 属性の値を「海」に書き換えるスクリプトです。attr()はカンマ区切りで指定した属性値を変更する命令です。複数の命令をメソッドチェーンを使って src 属性と alt 属性を一度に書き換えます。

※イベントの処理における注意

click()は任意の要素を指定できますが、もともとクリックされる要素である a 要素をクリックすると期待しない動作をしてしまいます。jQuery の click イベントが発生し実行されるのですが、a 要素の href 属性に設定されたリンク先の URL をその後に開いてしまいます。

a 要素の機能を無効化します。

回避方法1

```
<a href="javascript:void(0)">画像を変更</a>  
<p></p>
```

回避方法2

```
$(function(){  
  $("a").click(function(){  
    $("img").attr("src","sea.jpg").attr("alt","海");  
    return false;  
  });  
});
```

(課題) サンプルの確認 (jq_event.html)

6.4. イベントが発生した要素を取得する

イベントが発生した要素を取得するセレクタが\$(this)です。イベントを設定しているclick(function(){……})内で\$(this)を書くとイベントが発生した要素を取ることができます。

HTML

```
<ul>
<li><a href="flower.jpg">花</a></li>
<li><a href="sea.jpg">海</a></li>
<li><a href="dog.jpg">犬</a></li>
<li><a href="building.jpg">建物</a></li>
</ul>
<p></p>
```

script

```
$(function(){
  $("a").click(function(){
    $("img").attr("src",$(this).attr("href"))
    .attr("alt",$(this).text());
  });
});
```

(課題) サンプルの確認 (jq_event1.html)

7.フォーム

jQuery にはフォームに特化した機能があります。

1. フォーム部品の値を取得・変更する命令
2. フォームに関するイベントの処理
3. フォーム部品を選択するセクタ

7.1.value 属性の値を取得する val()

val()は括弧内に何も記述せずに使用すると、フォームに入力されているテキストや選択されている値を取得します。

Script

```
$(function(){
  $("button").click(function(){
    $("p").text($("#input").val()+"にメール送信します。");
    $("button").text("送信");
  });
});
```

HTML

```
<p>メールアドレス:<input type="text" name="name" /></p>
<button>確認</button>
```

確認ボタンを押すと、input 要素の値を取得し、p 要素内に書き出します。同時にボタン要素の文字を「送信」に変更します。

val()は input 要素だけでなく select、option、textarea にも使えます。

```
$("#select").val();
```

7.2.value 属性の値を変更する

フォームの内容を変更するときも val()を使います。

```
$(function(){
  $("button").click(function(){
    $("#input").val("");
  });
});
```

これは input 要素をクリアするときに使えます。

7.3.フォームのイベント

フォーカス focus()

input()要素などのフォーム部品がマウスやタブキーによって選択された状態になったことを感知し、設定された処理をします。

```
$(セクタ).focus(function(){  
    セクタで指定した要素がフォーカスされたときに実行する処理  
});
```

次のスクリプトでは input 要素の value 属性の値を val()で書き換え「入力してください」というテキストを灰色で表示しておきます。ユーザーが input 要素をフォーカスすると value 属性の値が空になります。

Script

```
$(function(){  
    $("input").val("入力してください").css("color","#CCC")  
    .focus(function(){  
        $(this).val("").css("color","#000");  
    });  
});
```

HTML

お名前:<input type="text" value="" />

(課題) サンプルの確認 (jq_formXX.html)

8.Ajax

通常ユーザーが新しい情報を得るためには、a 要素などで設定されたリンクをクリックし、ブラウザは新しい HTML をサーバーから取得し Web ページ全体を描画し直します。

Ajax を使うとページの一部分だけの更新が可能になります。Web サイトの操作性を大幅に向上させることができるようになります。(本格的な\$.ajax について色々と検討ください)

8.1.テキストを挿入する load()

load()は括弧内に記述したファイルを読み込み、セレクトで指定した要素内のテキストを書き換えます。

```
$(セレクト).load(ファイル名);
```

次は button がクリックされると ample.txt が読み込まれ、p 要素のテキストを書き換えるサンプルです。

Script

```
$(function(){  
  $("button").click(function(){  
    $("p").load("./sample.txt");  
  })  
});
```

HTML

```
<button>変更</button>  
<p>変更前</p>
```

sample.txt

sample のテキストです。

Ajax 利用時の注意点

Ajax の仕様上、文字コードは「UTF-8」しか使えません。

また、load()で取得できるファイルの存在する場所は HTML と同一ドメインにあるファイルだけとなります。

```
$("p").load("http://www.hoge.com/sample.txt");
```

上記は同一ドメインなら OK ですが、異なるドメインでは読み込みできません。その場合、PHP などを利用して URL を読み込ませることで対応できます。

8.2. post 命令

jQuery で POST 送信をすることができる命令に ajax と post があります。

次は post のサンプルです。

```
<html>
<head>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(function(){
$('#sw').click(function(){
$.post('post.php',{'pd': 'こんにちは'},function(data){
alert(data);
});
});
});
</script>
</head>
<body>
<a href="javascript: void(0)" id="sw">switch</a><br />
</body>
</html>
```

post.php ファイル

```
print 'your post data is '.$_POST['pd'];
```

(課題)load での ajax の確認 (jq_ajax.html)

※サーバでの処理が必要な php は今後確認されたし！！

9. アニメーションのメソッド

コンテンツをフェードさせて表示させたり、スライドさせたりと HTML と CSS だけではなかなかできなかった動きをつける事ができます。

エフェクトメソッドの例

- toggle()
\$(".hidden").toggle();
表示されてるものを隠したり、非表示のものを表示します。
- fadeIn()
\$(".fade").fadeIn();
フェードインさせて表示します。
- fadeOut()
\$(".fade").fadeOut();
フェードインの逆ですねー。フェードアウトさせて非表示にします。
- show()
\$(".hidden").show();
要素を表示します。
- hide()
\$(".hidden").hide();
これは逆に要素を隠します。
- animate()
\$(".img").animate({ "height" : "0px" });
アニメーションしながら、高さを 0 にしています。

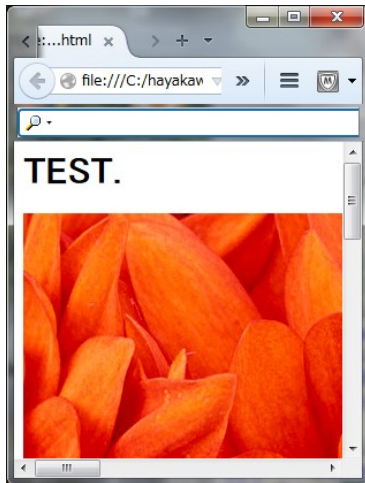
9.1.非表示状態の要素を表示する show()

show()は CSS の display プロパティの値が none、つまり非表示の HTML 要素をアニメーションしながら表示する命令です。

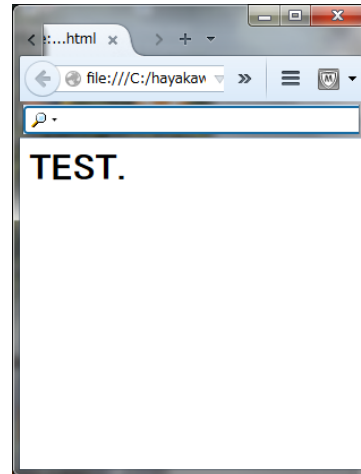
- \$(セレクト).show(スピード);
- div{
width:200px;
height:200px;
display:none;
}
- \$(function(){
\$("button").click(function(){
\$("div").show("slow");
});
});
- <button>表示</button>
<div></div>

(課題)表示されている画像をクリックで hide し TEST の表示をクリックすると show する

(jq_hide&show.html)



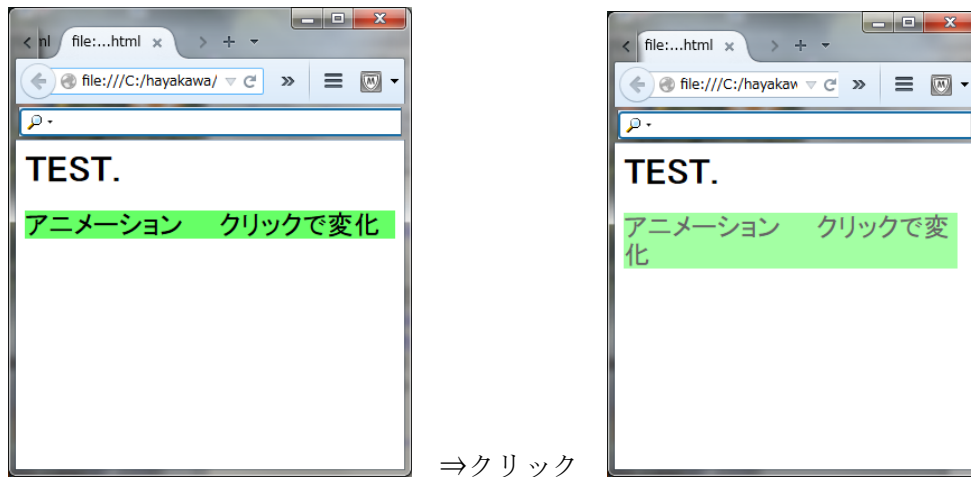
⇒ 画像クリック



TEST クリックで表示

(課題) クリックで `width:"400px",fontSize:"24pt",opacity:1.0` で 3000 速度で

アニメーションされる (jq_animation.html)



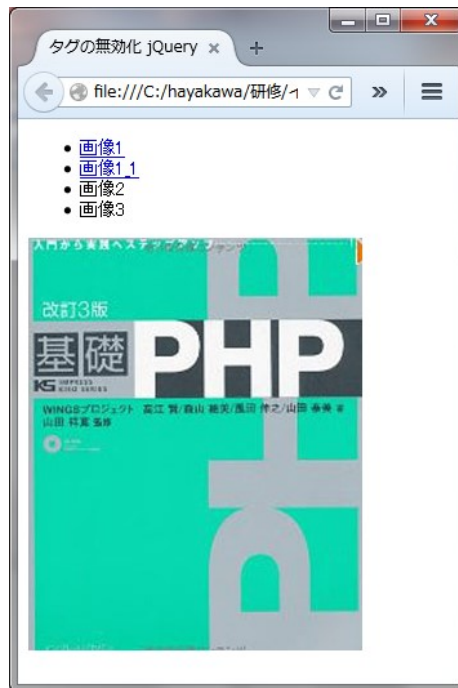
クリックで `width:"400px",fontSize:"24pt",opacity:1.0` される

3000 速度でアニメーションされる

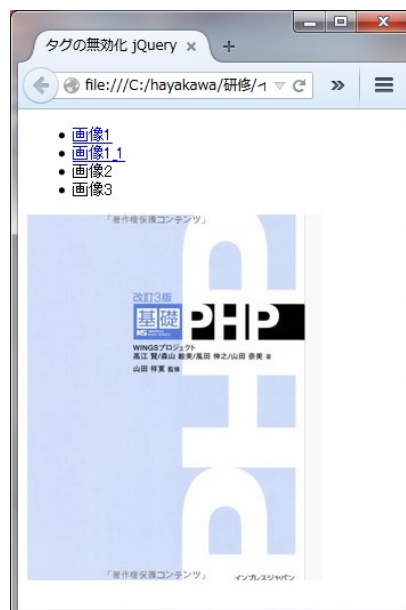
10.課題

下記の画面で示す初期画像とクリックで各画像を表示させる jquery で表現する

初期の表示



画像1をクリックすると



画像 1_1 をクリックすると

