

コンテンツメディア プログラミング実習Ⅱ jQuery

中村 宮下 齊藤 菊池

Chromeでウェブサイト
いたずらしていたのを思い出してw

「HTMLやCSSの要素を書き換える」

これをプログラム (JavaScript) から行うことによってより動的な体験を提供していく

【今日のおしながき】

- ・jQuery
- ・Chrome拡張入門

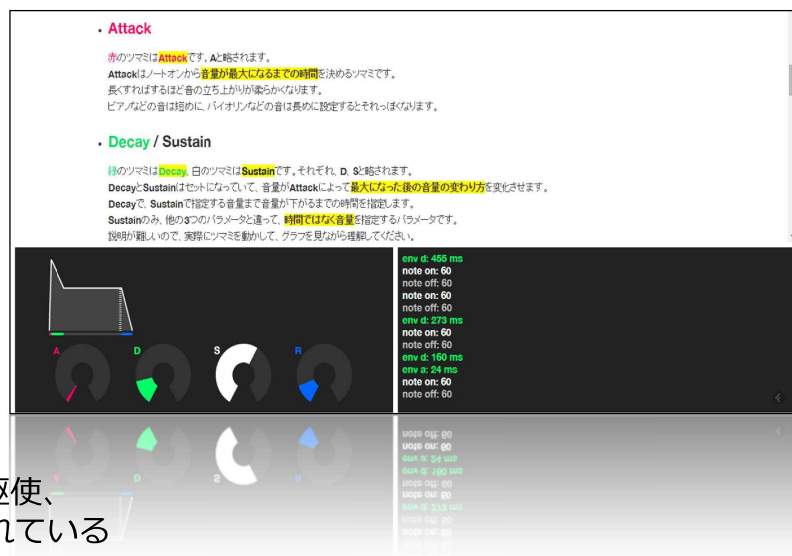
両方ともに「JavaScriptによる書き換え」の考え方でつながっている



前回および前々回とのつながりやオーバーラップも多めにしている
ので、ぜひ過去の資料も見返しながら進もう！

Web上で動く簡単なシンセ (FMS1年 小淵 豊さん制作 @FMS_Cat)

<http://fms-cat.github.io/synth>

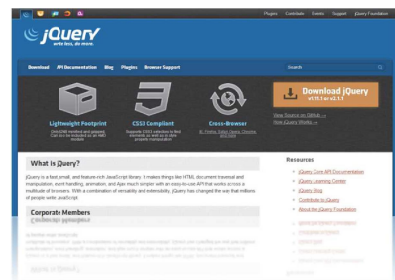


jQueryやJQuery Knobを駆使、
独学で1週間ほどで制作されている

CMP実習2は「課題」という呼び方をしているが、
実装に対するモチベーションや姿勢はCE概論やプログラム実習1・2と同じでOK!

jQuery (復習)

- アプリケーションフレームワークのひとつ
- <http://jquery.com/> にアクセスして
最新版のファイルをダウンロード



- www に直接置いても良いが、できれば機能で整理した
方が良かったため、新たにlib というフォルダ (他のフォルダ
名でも良い) を作成し、そこに入れたほうが良い

lib はライブラリの意味

```
<script src="lib/jquery-2.1.1.min.js"></script>
```

jQueryの基本(復習)

「\$()」は「jQuery()」の略

- 本来は

- jQuery(function(){ ... });
- jQuery('#hoge').on(...);
- jQuery('#hoge').css(...);

**\$(function(){
プログラム**

}); でjQueryに関するプログラムを書く

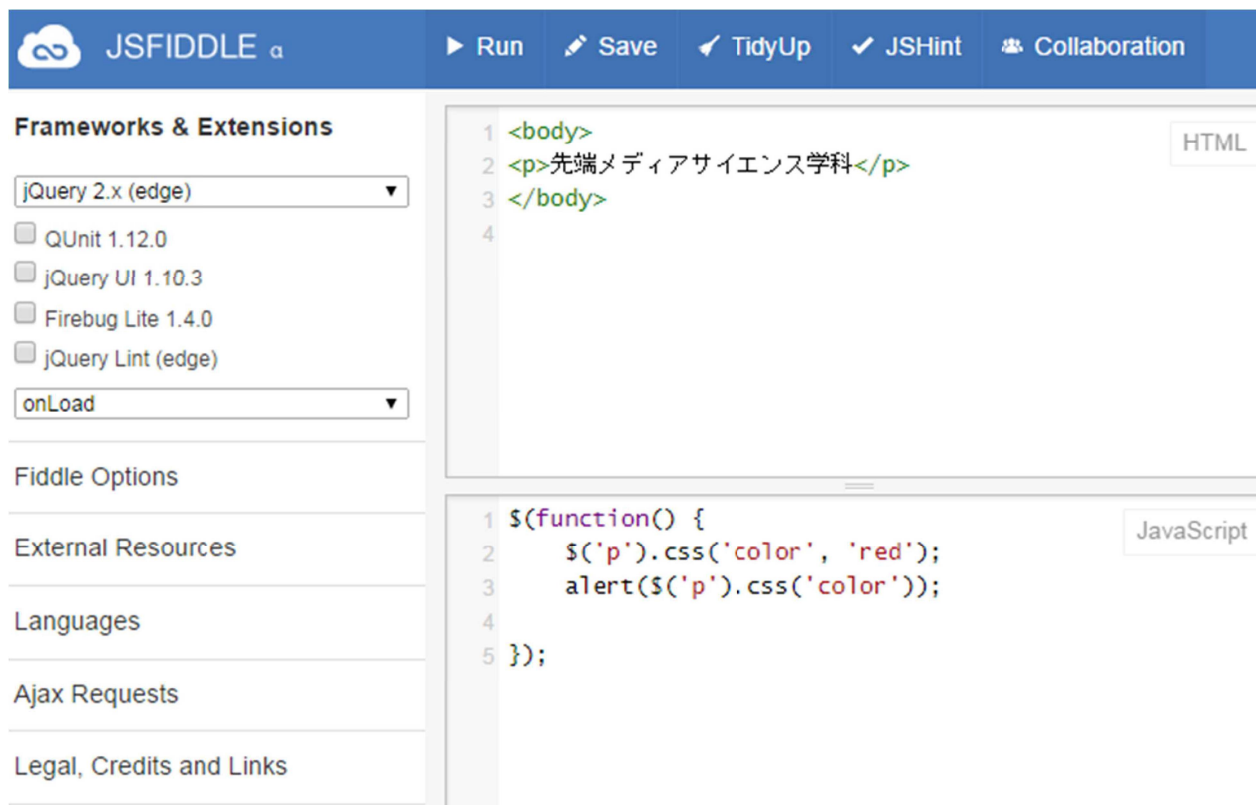
<http://jsfiddle.net/> で軽く復習 (1ファイルにまとめる編)

The screenshot shows the JSFiddle.net interface. The HTML editor contains the following code:

```
1 <body>
2 <p>先端メディアサイエンス学科</p>
3 <script>
4 $(function() {
5   $('p').css('color', 'red');
6 });
7 </script>
8 </body>
9
10
```

The CSS editor is empty. The result preview shows the text "先端メディアサイエンス学科" in red. A blue callout box highlights the jQuery code, with annotations: "←セレクタ" (selector) pointing to '\$('p')', "←メソッド" (method) pointing to ".css()", and "←セレクタ 'と' に違いはない" (no difference in selector ' and ') pointing to the space between the selector and the method. A teal callout box at the bottom right explains: "\$('p').css('color', 'red').css('background', 'yellow');"のように長くつないでも書けるよ (You can also write it by chaining like '\$('p').css('color', 'red').css('background', 'yellow');') and "\$('p').addClass('myStyle');"でまとめてもいいけど (It's also good to combine it with '\$('p').addClass('myStyle');').

こう書いたほうがわかりやすいかも



The screenshot shows the JSFiddle web editor interface. The top navigation bar includes 'Run', 'Save', 'TidyUp', 'JSHint', and 'Collaboration' buttons. On the left, the 'Frameworks & Extensions' panel is visible, with 'jQuery 2.x (edge)' selected in the dropdown menu. Below it, several checkboxes are present for 'QUnit 1.12.0', 'jQuery UI 1.10.3', 'Firebug Lite 1.4.0', and 'jQuery Lint (edge)'. The 'onLoad' dropdown is also visible. The main editor area is split into two panes. The top pane, labeled 'HTML', contains the following code:

```
1 <body>
2 <p>先端メディアサイエンス学科</p>
3 </body>
4
```

The bottom pane, labeled 'JavaScript', contains the following code:

```
1 $(function() {
2     $('p').css('color', 'red');
3     alert($('p').css('color'));
4
5 });
```

セレクトタ(復習)

elementを選択するには

- tag : \$("tagname")
- id : \$("#idname")
- class : \$(".classname")

カンマで複数選択もできる

<p id="one">, <p class="one"> などと書いたタグに適用して実験してみよう

エフェクト！（括弧内はミリ秒）

- `$('#p').show(1000);`
- `$('#p').hide(1000);`

- `$('#p').fadeIn(1000);`
- `$('#p').fadeOut(1000);`

- `$('#p').toggle(1000);`
- 表示状態にあるものを非表示にし、非表示状態にあるものは表示状態に

ほかにもいろいろ指定方法がある

```
$("#idname > .classname").css('color', 'red');
```

→ **idnameの直下のclassname**だけを指定

半角スペースだとそれ以下の要素を全部指定になる

```
$("#idname").children().css('color', 'red');
```

→ **idnameの子要素すべて**を赤くする

属性セレクタ

`<p>リンク</p>`
があるとき、

```
$('a[href="http://www.meiji.ac.jp/"]').css('color', 'red');
```

で適用できる。

このときの=は等号なので、!=で≠の意味になったりする。
*=「含まれる」という意味になるので、「.ac.jp」ドメインへの
リンクだけまとめて赤くするとかできる！

値の取得

```
alert($('p').css('color'));
```

pタグの色を取得して出力

jsfiddleのプログラムに追加して確認してみよう

※出力の仕方はいろいろあったよね...

- `console.log("出力したい内容");`
- `alert("出力したい内容");`
- `document.write("出力したい内容");`

属性の取得・変更

`<p>リンク</p>`
があるとき、

```
alert($('a').attr('href'));
```

のように取り出したり、

```
$('a').attr('href', 'http://miyashita.com/');
```

のように変更したりできる

タグの中身の変更

```
$('p').text('中身のテキスト変更');
```

```
$('p').html('<HTMLタグ>');
```

`.empty()`、`.remove()` で、
要素を空にしたり削除することもできる

コールバック関数

第二引数として関数を書くと、処理が終わったときにそれを実行してくれる

```
$('#p').fadeOut(1000, function() {  
    alert("Goodbye");  
});
```

イベント

```
$('#p').click(function() {  
    alert("hello!");  
});
```

のように、

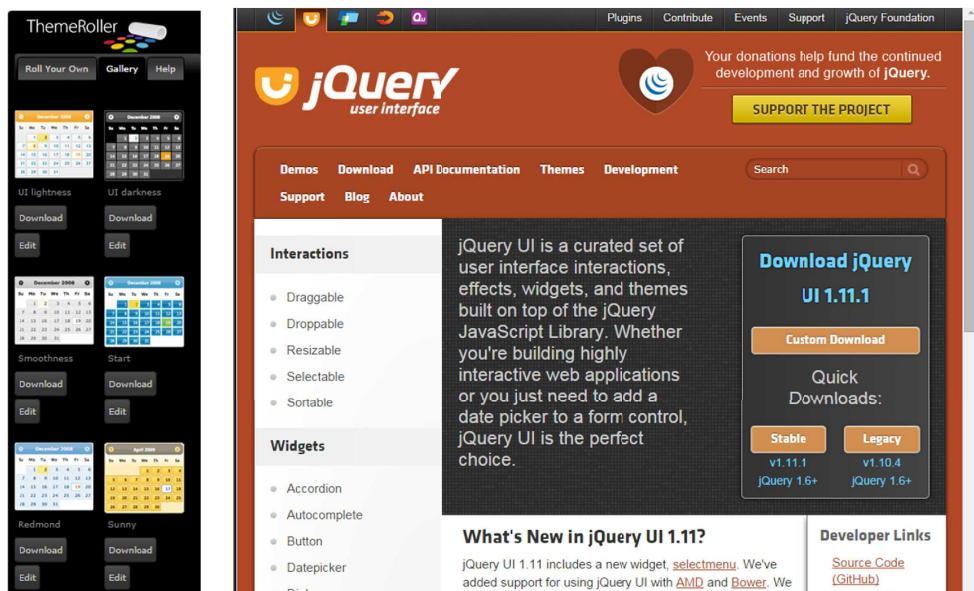
イベントが起こったら実行してほしいことをfunctionとして括弧内に書く

イベント一覧(復習)

- mouseover: 要素にマウスが乗った時
- mouseout: 要素からマウスが離れた時
- mousedown: 要素上でクリックボタンが押された時
- mouseup: 要素上でクリックボタンが離れた時
- mousemove: 要素上でマウスが動かされた時
- click: 要素がクリックされた時
- dblclick: 要素がダブルクリックされた時
- keydown: 要素にフォーカスした状態で, キーボードのキーが押された時
- keyup: 要素にフォーカスした状態で, キーボードのキーが離された時
- focus: 要素にフォーカスが当たった時
- blur: 要素からフォーカスが外れた時
- change: 入力内容が変更された時
- resize: 要素がリサイズされた時
- scroll: 要素がスクロールされた時

jQuery UI (jqueryui.com)

- ダウンロード時にテーマを選ぶ



The screenshot shows the jQuery UI website interface. On the left, there is a 'ThemeRoller' sidebar with various theme options like 'UI lightness', 'UI darkness', 'Smoothness', 'Start', 'Redmond', and 'Sunny', each with 'Download' and 'Edit' buttons. The main content area features the jQuery logo and navigation links for 'Demos', 'Download', 'API Documentation', 'Themes', and 'Development'. A 'Download jQuery UI 1.11.1' section offers a 'Custom Download' button and 'Quick Downloads' for 'Stable' (v1.11.1) and 'Legacy' (v1.10.4). Below this, there are sections for 'Interactions' (Draggable, Droppable, Resizable, Selectable, Sortable) and 'Widgets' (Accordion, Autocomplete, Button, Datepicker, Dialog).

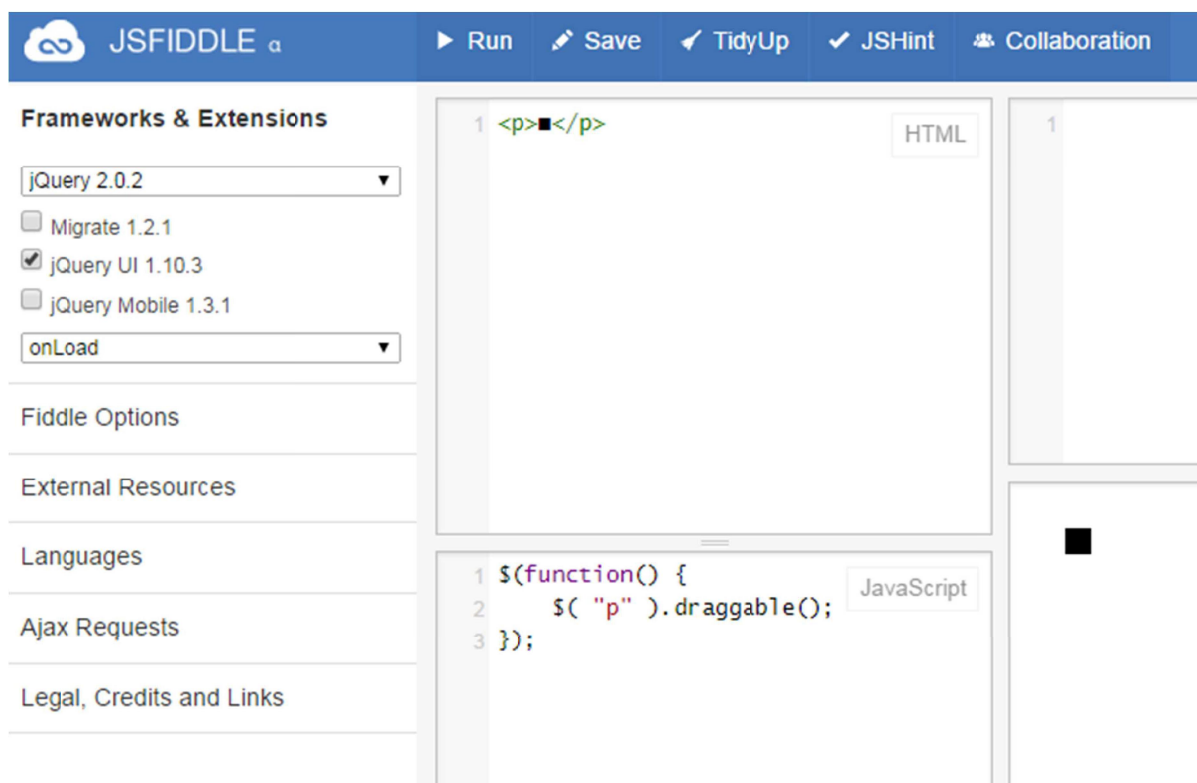
弾むエフェクト

(<http://jqueryui.com/effect/>)



The screenshot shows the JSFiddle interface. The top navigation bar includes 'Run', 'Save', 'TidyUp', 'JSHint', and 'Collaboration'. The left sidebar contains 'Frameworks & Extensions' with 'jQuery 2.0.2' selected, 'jQuery UI 1.10.3' checked, and 'jQuery Mobile 1.3.1' unchecked. The 'onLoad' event is selected. The main editor area has two tabs: 'HTML' and 'JavaScript'. The HTML tab contains the code `1 <p>ぽよん</p>`. The JavaScript tab contains the code `1 $(function() {`, `2 $('p').effect("bounce");`, and `3 });`. The preview area on the right shows the rendered text 'ぽよん'.

段落をドラッグ可能にするw



The screenshot shows the JSFiddle interface. The top navigation bar includes 'Run', 'Save', 'TidyUp', 'JSHint', and 'Collaboration'. The left sidebar contains 'Frameworks & Extensions' with 'jQuery 2.0.2' selected, 'jQuery UI 1.10.3' checked, and 'jQuery Mobile 1.3.1' unchecked. The 'onLoad' event is selected. The main editor area has two tabs: 'HTML' and 'JavaScript'. The HTML tab contains the code `1 <p>■</p>`. The JavaScript tab contains the code `1 $(function() {`, `2 $("p").draggable();`, and `3 });`. The preview area on the right shows a single black square '■'.

簡単に作れちゃうタブメニュー

The screenshot shows a web editor interface with a blue header bar containing 'Run', 'Save', 'TidyUp', 'JSHint', and 'Collaboration' icons. The main workspace is divided into three panes. The top-left pane is labeled 'HTML' and contains the following code:

```
1 <div id="tabmenu">
2
3 <ul>
4   <li><a href="#tab1">タブ1</a></li>
5   <li><a href="#tab2">タブ2</a></li>
6 </ul>
7
8 <div id="tab1">タブ1の中身</div>
9 <div id="tab2">タブ2の中身</div>
10
11 </div>
```

The bottom-left pane is labeled 'JavaScript' and contains the following code:

```
1 $(function() {
2   $('#tabmenu').tabs({
3     selected: 1
4   });
5 });
```

The right-hand pane shows a rendered preview of the tab menu. It features two tabs: 'タブ1' (selected) and 'タブ2'. Below the tabs, the content 'タブ1の中身' is displayed.

簡単に作れちゃうアコーディオン

The screenshot shows a web editor interface with a blue header bar containing 'Run', 'Save', 'TidyUp', 'JSHint', and 'Collaboration' icons. The main workspace is divided into three panes. The top-left pane is labeled 'HTML' and contains the following code:

```
1 <div id="menu">
2
3   <h1>見出し1</h1>
4   <p>中身1</p>
5
6   <h1>見出し2</h1>
7   <p>中身2</p>
8
9 </div>
```

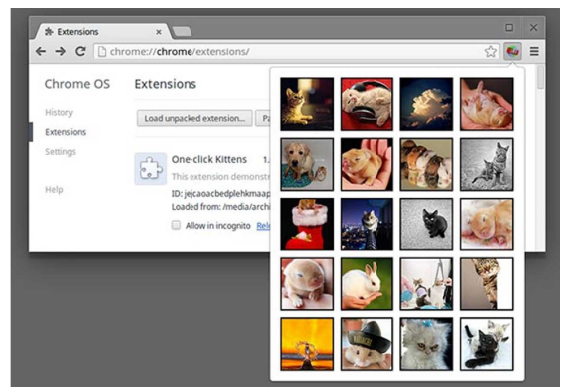
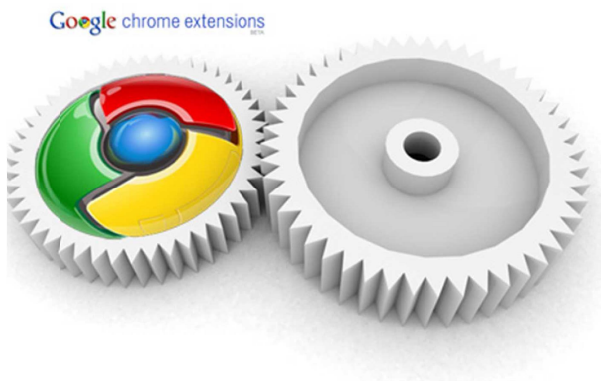
The bottom-left pane is labeled 'JavaScript' and contains the following code:

```
1 $(function() {
2   $('#menu').accordion();
3 });
```

The right-hand pane shows a rendered preview of the accordion. It displays two sections. The first section is expanded, showing the heading '見出し1' and the content '中身1'. The second section is collapsed, showing the heading '見出し2' with a downward-pointing arrow.

課題

- jQueryおよびjQuery UIのAPI Documentationを見て、テキストに載っていない仕様を試してください！！
- jQueryおよびjQuery UIを組み合わせて、派手でインタラクティブなウェブサイトを作ってください！！



Chrome拡張 入門

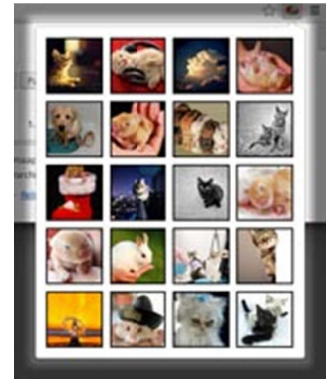
<https://developer.chrome.com/extensions/getstarted>

必要なファイル

- アイコンファイル (19×19ピクセルサイズ)
- manifest.jsonファイル
拡張機能のメタ情報を記す
JavaScriptのオブジェクト形式で書く



このほか、必要に応じてポップアップメニュー用のhtmlファイルなど



manifest.json (UTF-8で保存してね)

```
{  
  "name": "Meiji Menu",  
  "version": "0.1",  
  "manifest_version": 2,  
  "description": "明治大学学生にとって便利なメニュー",  
  
  "browser_action": {  
    "default_icon": "icon.png",  
    "default_title": "明治大学便利メニュー",  
    "default_popup": "pop.html"  
  }  
}
```

pop.html

```
<!DOCTYPE html>
<html lang="ja">
<head>
  <meta charset="utf-8">
</head>
<body style="min-width:100px">
<p><a href="http://www.meiji.ac.jp/" target="_blank" >明治大学</a></p>
<p><a href="http://www.meiji.ac.jp/isc/meijimail-info.html" target="_blank" >Meiji Mail</a></p>
<p><a href="https://oh-o2.meiji.ac.jp/portal/oh-o_meiji/" target="_blank" >Oh-o!Meiji</a></p>

</body>
</html>
```

インストールしてみよう

- 拡張機能メニュー、デベロッパーモードの「パッケージ化されていない拡張機能を読み込む...」からフォルダを指定するだけ
- リロードボタンが超便利！



背景色を変えるChrome拡張に挑戦

```
{  
  "name": "background change",  
  "version": "0.1",  
  "manifest_version": 2,  
  "description": "背景色変更",  
  
  "permissions": ["tabs", "http://*/*"],  
  "browser_action": {  
    "default_icon": "icon.png",  
    "default_title": "background change",  
    "default_popup": "pop.html"  
  }  
}
```

manifest.json

```
<!DOCTYPE html>  
<html lang="ja">  
<head>  
<meta charset="utf-8">  
</head>  
<body style="min-width:100px">  
<p id="change">色変更</p>  
<script src="change.js"></script>  
</body>  
</html>
```

pop.html

change.js

```
document.getElementById('change').onclick = function() {  
  chrome.tabs.executeScript(null, {  
    "code": "document.body.style.backgroundColor='red'"  
  });  
}
```

課題

- 自分が使いやすくてビジュアル的にも楽しいオリジナルメニューを作ろう
- JavaScript等と組み合わせて便利なツールを作ってみよう

Chrome 拡張サンプル

□例1: gooahoo!

■manifest.json

```
{
  "name": "Google to Yahoo",
  "version": "1.0",
  "manifest_version": 2,

  "description": "Google がヤフーに",
  "content_scripts": [
    {
      "run_at": "document_end",
      "matches": [
        "https://www.google.co.jp/*",
        "http://www.google.co.jp/*"
      ],
      "js": [
        "content.js"
      ]
    }
  ],
  "permissions": [
  ]
}
```



■content.js

```
console.log('start')

logo = document.getElementById('hplogo')
logo.style.background = "url(http://k.yimg.jp/images/top/sp/logo.gif) no-repeat"

searchBox = document.getElementById('gbqfsa')
searchBox.innerHTML = "Yahoo 検索"

console.log('end')
```

□例 2: timer

■manifest.json

```
{
  "name": "Timer",
  "version": "1.0",
  "manifest_version": 2,
  "description": "時計です",
  "browser_action": {
    "default_icon": "icon.png",
    "popup": "popup.html"
  }
}
```



■popup.html

```
<html>
  <head>
    <script type="text/javascript" src="popup.js"></script>
  </head>
  <body onLoad="disp()">
    <div id="time"></div>
  </body>
</html>
```

■popup.js

```
function disp() {
  var now = new Date();
  // 時間取得

  var nowh = now.getHours()
  if(nowh < 10)
    nowh = "0" + nowh

  var nowm = now.getMinutes()
  if(nowm < 10)
    nowm = "0" + nowm

  var nows = now.getSeconds()
  if(nows < 10)
    nows = "0" + nows

  // 時間を書き込む
  var time = document.getElementById("time")
  time.innerHTML = nowh + ":" + nowm + ":" + nows
  setTimeout("disp()", 1000)
}
```

□ 例3 meiji_delete

■ manifest.json

```
{
  "name": "Meiji photo delete",
  "version": "1.0",
  "manifest_version": 2,

  "description": "あの顔を消します",
  "content_scripts": [
    {
      "run_at": "document_end",
      "matches": [
        "http://www.meiji.ac.jp/*"
      ],
      "js": [
        "content.js"
      ]
    }
  ],
  "permissions": [
  ]
}
```



■ content.js

```
head = document.getElementById('HEADER')
photo = head.children[0]
head.removeChild(photo)
```

