

# jQuery入門

+

## ハマりがちな失敗例を ご紹介しながらのハンズオン

2013.12.11 TAM 松尾浩志

# 自己紹介

株式会社TAM 松尾浩志

(エンジニア / テクニカル・ディレクター)

主に PHP / JavaScript / HTML5 / CSS3



パートナー型Webプロダクション  
大阪・東京・シンガポールに105名  
<http://www.tam-tam.co.jp/>

# 今日のテーマ

- ・ 「jQuery入門 & ハマりがちな失敗例をまじえてのハンズオン」
- ・ 対象は：  
「HTML/CSSは分かるけれどJavaScriptはこれから、または独学」  
という方
- ・ jQuery の基礎と、実際にハンズオンで画像ギャラリーを作ってみよう  
ハンズオンでは、「書いてみたけれどうまく動かない」という  
ハマりがちな失敗例をまじえてご紹介

# もくじ

- ・ jQueryとは？
- ・ jQuery のメリット
- ・ コードの書き方
- ・ 開発環境について （ここまで30分ほど）
- ・ 休憩 (10分)
- ・ ハンズオン：画像ギャラリーのスクリプトを書いてみよう （50分）

jQueryとは？

# jQuery とは?

- ・ JavaScript のオープンソース・ライブラリ (便利パーツ集) です
- ・ jQuery自体、JavaScript で書かれています
- ・ つまり、



(JavaScriptでできないことはjQueryでもできません)

# JavaScriptとは？

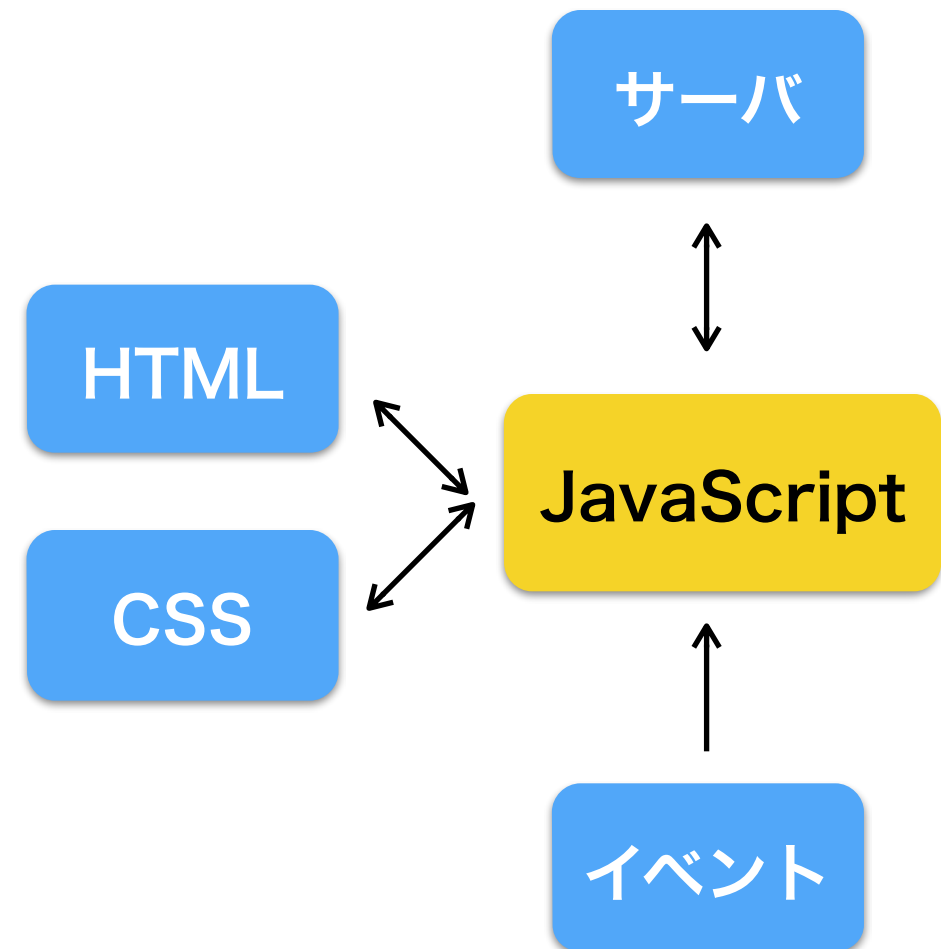
## ブラウザ上で動くスクリプト言語

HTML, CSSを書き換えたり…

マウス操作やキー入力イベントを処理したり…

アニメーションを実現したり…

AJAX でサーバとデータのやりとりをしたり…



HTML + CSS だけではできない、**動的なページ**が作れます

# JavaScriptとは？

- ・ 1996年、Netscape 2.0 で搭載（古い！）
- ・ しかしWebに動きをつけるのは、長らく FLASH が主流でした
- ・ 2005年、**Googleマップ**登場、**AJAX** という名前とともに再注目
- ・ 2008年、**iPhone** 3G発売(**FLASH非対応**)
- ・ 2008年、**HTML5** の登場
- ・ 徐々に FLASH に変わって JavaScript 活躍の機会が増えています





# jQueryのメリット

# jQueryのメリット

- ・ write less, do more.



- ・ 短く書ける！
- ・ CSSっぽく要素を選択できるので、CSS経験があるとわかりやすい
- ・ クロスブラウザ対応
- ・ プラグインが豊富

**ラク！！**

# 短く書ける！

`<img id="logo" />` を非表示にするサンプル

JavaScript の例:

```
document.getElementById('logo').style.display = 'none';
```

jQuery の例1:

```
$('#logo').css('display', 'none');
```

jQueryの 例2:

```
$('#logo').hide();
```

# 短く書ける！！！！！！

`<img id="logo" />` を2秒かけてフェードアウトさせるサンプル

## JavaScript の例:

```
var opacity = 100;
var step = 5;
var duration = 2000;
var fadeOut = function() {
    var element = document.getElementById('logo');
    opacity -= step;
    if (opacity <= 0) {
        element.style.display = 'none';
        return;
    }
    element.style.opacity = opacity / 100;
    element.style.filter = "alpha(opacity:"+opacity+")";
    setTimeout(fadeOut, duration / (100 / step));
}
fadeOut();
```

## jQuery の例:

```
$('#logo').fadeOut(2000);
```

# いろいろできます

`show()` `hide()`

`text()` `html()` `css()`

`toggle()`

`attr()` `val()`

`fadeIn()` `fadeOut()` `fadeTo()`

`addClass()` `removeClass()`

`slideDown()` `slideUp()`

`append()` `prepend()` `wrap()`

`animate()`

`after()` `before()`

`width()` `height()`

`find()` `parent()` `children()`

などなど..... 便利なメソッドが、覚えきれないほどたくさんあります。

# CSSっぽく書ける！

`$('#id')` ← ID を選択

`$('.class')` ← クラスを選択

`$('ul#id > li.class')` `$('h1, h2')` ← CSSと同じですね

`$('li:first')` `$('li:eq(3)')` ← jQuery独自の記法もあります

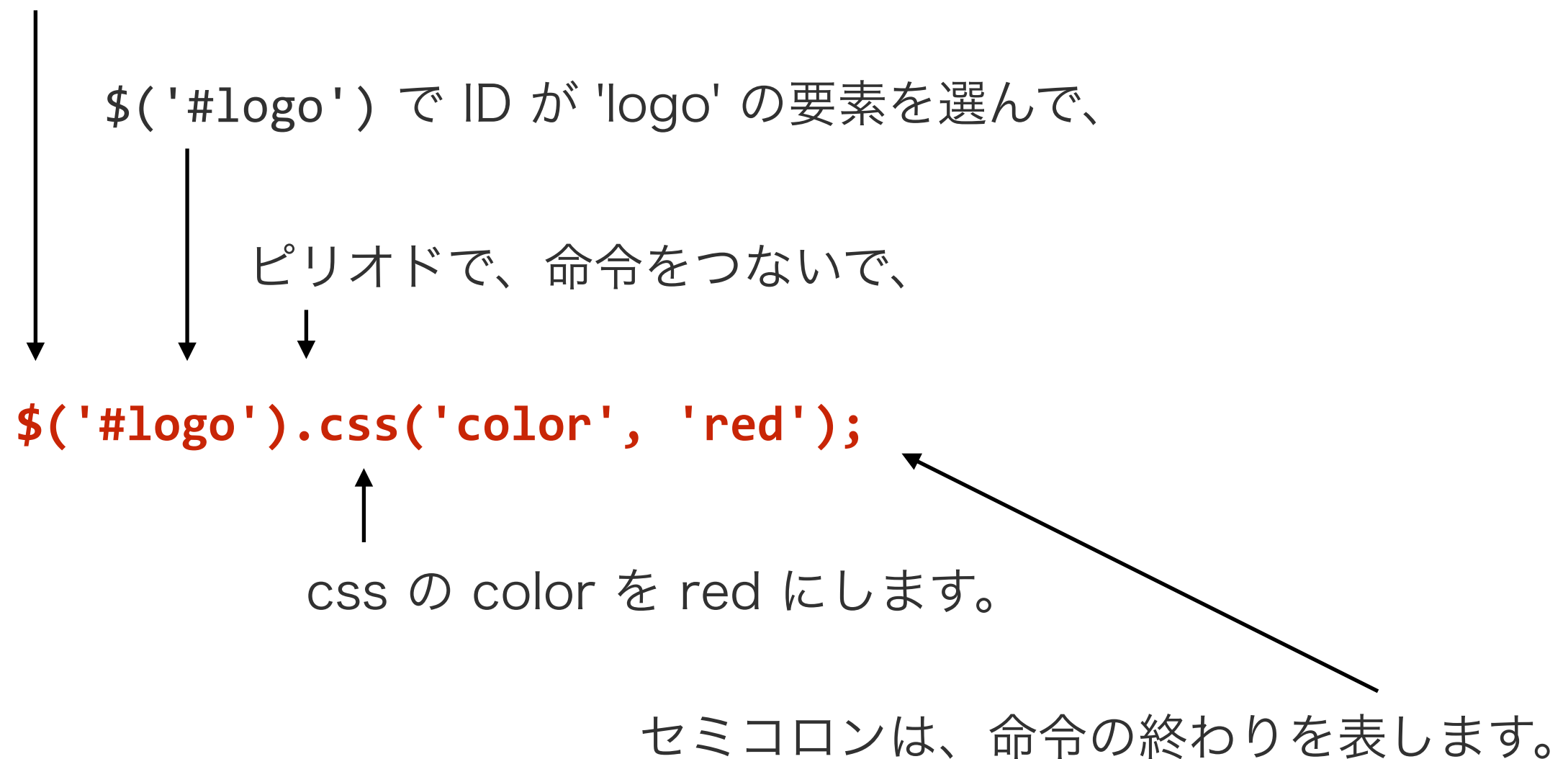
`$('input[type=checkbox]')` `$('input:not(:checked)')`

`$('CSSのセレクタ似')` なので、CSS を知っている人にはなじみやすい！

# コードの書き方

# 使用例 (1)

jQuery の命令は、かならず「\$」ではじまります。





# 使用例 (2)

htmlを書き換える例：

```
$('.sample').text('好きな文字列');
```

```
<p class='sample'>ダミーの文字列</p>
```



```
<p class='sample'>好きな文字列</p>
```

# 使用例 (3)

cssを書き換える例：

```
$('.sample').css({  
    color: 'red',  
    display: 'block',  
    width: '100%'  
});
```

```
.sample {  
    color: red;  
    display: block;  
    width: 100%;  
}
```

JavaScript と CSS の文法の違いはありますが、**ほとんど一緒！**

# 使用例 (4)

ロゴをクリックしたら、ナビがフェードアウト

```
$('#logo').click(function() {  
    $('#nav').fadeOut();  
});
```

**function(){ ~~~ }** は処理のひとかたまり、ひと纏まりを表します。

`function(){ }` 内のコードはすぐには実行されず、呼び出されたタイミング（この場合は `#logo` のクリック時）にはじめて実行されます。

# 基本はこれだけ、 あとは組み合わせ（応用）

- ・ **HTML や CSS を書き換える**
- ・ **なにかイベントが起きたら、**  
（例：マウスのクリックやホバー、キー入力、画面のスクロールなど）  
`function(){ ~~~ }` のひとかたまりの  
**処理を実行する**

（他にはサーバとのAJAX通信や、一定時間おきに処理する、  
などのパターンもある）

# JavaScriptの基本的な文法は 知っておく必要があります

- ・ jQuery は JavaScript のライブラリ（便利パーツ集）
- ・ 応用するためには、**結局 JavaScript の文法を知らなければならない**
- ・ jQuery入門のサイトや本などでは、  
JavaScript の文法の基礎は詳しく書かれていないことも多い
- ・ JavaScript の土台の上に jQueryがあります  
「JavaScript 入門」的な本やサイトも見てみてください

# 応用の入門

- ・ プログラミングの基本は、だいたい以下の4つ
  - ・ 変数（数字や文字列の扱い）
  - ・ if 文（条件判定）
  - ・ 配列とループ
  - ・ 関数（function）
- ・ このへんが理解できれば、あとはパズルのように組み合わせればだいたいことができます

# メソッドチェーン

同じものに対する命令を、ピリオドでつなげて一度に書けます

```
$('#sample').css('color', 'red');
```

```
$('#sample').text('テキスト');
```

```
$('#sample').fadeIn();
```



```
$('#sample').css('color', 'red').text('テキスト').fadeIn();
```

書きやすい、処理速度も速い、コードがわかりやすい。

# 開発環境について



# ブラウザとエディタだけ

- ・ JavaScript は、ブラウザ上で動きます
- ・ 他の言語と違い、複雑なサーバ設定やインストール作業はいりません
- ・ エディタはお好みで、ただし JavaScript の文法チェックをしてくれるものがおすすめ（JSHintプラグインなど）



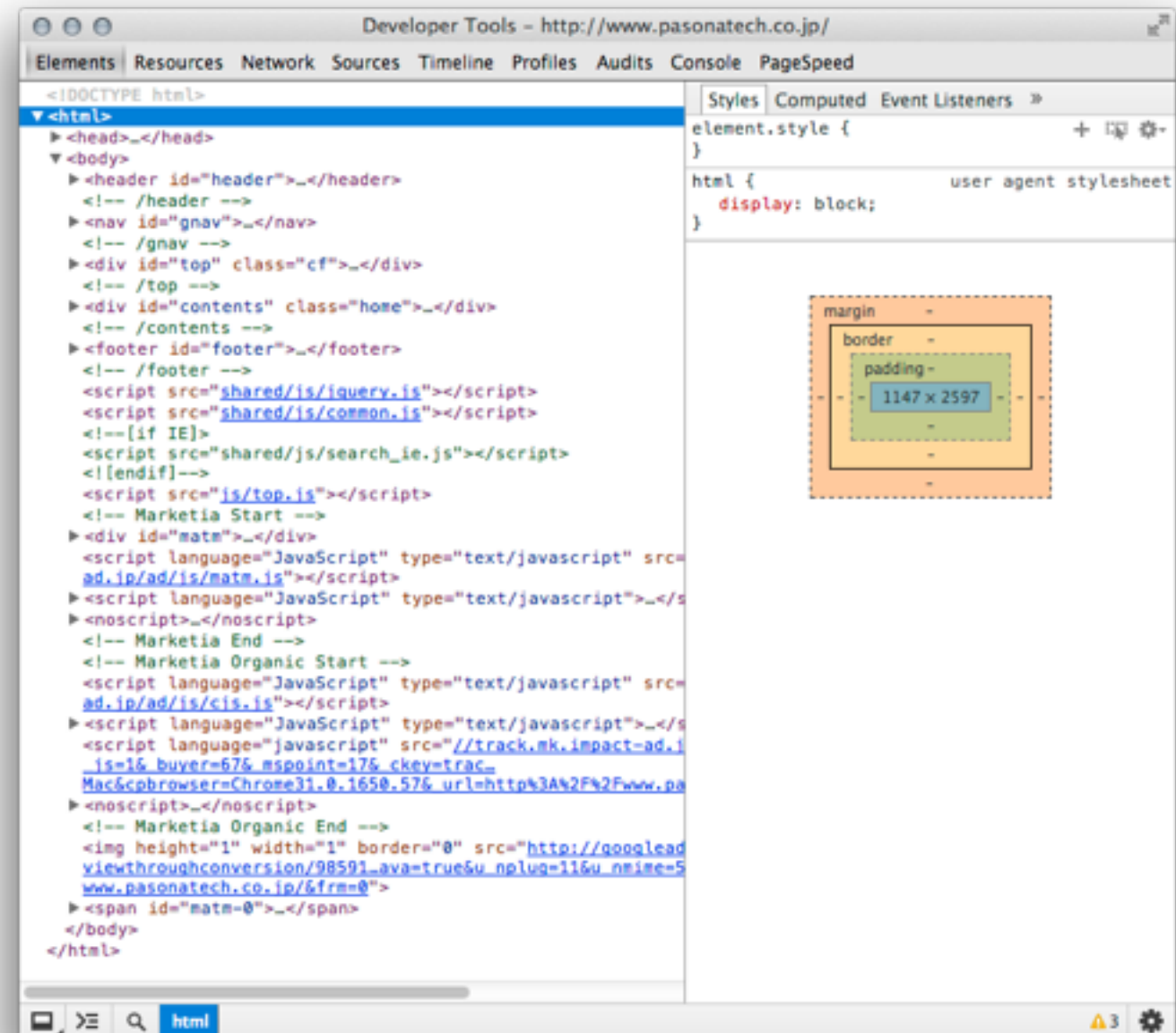
- ・ 動作テストも、ブラウザ内に便利な開発ツールが備わっています

# 開発ツールべんり

## Chrome Developer Tool の例：

Chrome内蔵の開発ツール。  
HTML も CSS もその場で  
表示しながら編集できます。

JavaScript の動作を  
途中で止めたり、  
実際にコードを入力して  
テストできます。

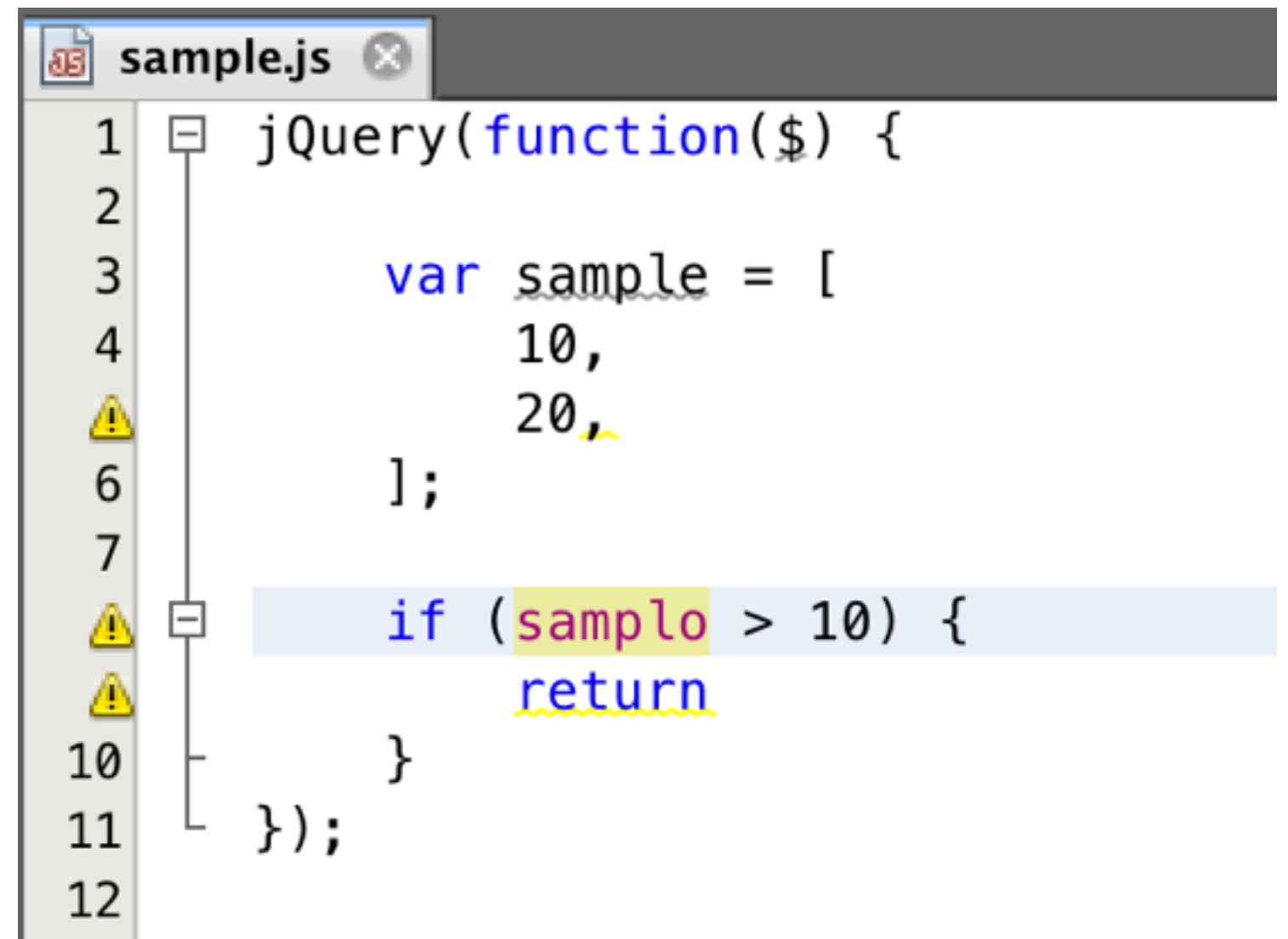


# 文法チェックべんり

Netbeans の例：

エラーが 4つ、  
見つかっています

コードを書いている途中で  
ミスが分かるので  
とても効率的です。



```
1  jQuery(function($) {  
2  
3      var sample = [  
4          10,  
5          20,  
6      ];  
7  
8      if (sampo > 10) {  
9          return  
10      }  
11  });  
12
```

The screenshot shows a NetBeans IDE window titled 'sample.js'. The code is as follows:

```
1  jQuery(function($) {  
2  
3      var sample = [  
4          10,  
5          20,  
6      ];  
7  
8      if (sampo > 10) {  
9          return  
10      }  
11  });  
12
```

Four yellow warning icons are visible in the left margin, indicating syntax errors:

- Line 5: A comma at the end of the array '20,'.
- Line 8: The variable 'sampo' is misspelled (should be 'sample').
- Line 9: The 'return' statement is incomplete.
- Line 10: A closing curly brace '}' for the 'if' statement.

# 開発をはじめる

ブラウザとエディタだけあれば特別なセットアップはいらず、いきなり開発をはじめられます。

1. jQuery 本体を [jquery.com](http://jquery.com) からダウンロード
2. index.htmlファイルから jQuery を呼び出す
3. `<script>` タグ内に書きはじめる
4. ブラウザで実行して動作確認
5. Developer Tool でデバッグ

# jQuery のダウンロード

- ・ <http://jquery.com/> からダウンロード
- ・ jQuery にはバージョン1系 とバージョン2系があります

- ・ 1系：IE6以降に対応

- ・ 2系：IE9以降に対応  
(そのぶん軽量で高速)

- ・ 2系はIE8に対応していないので、案件に応じて選択を



# index.htmlから呼び出す

```
<html>
```

```
<head>
```

```
  ～～
```

```
</head>
```

```
<body>
```

```
  ～～
```

```
    <script src="jquery-1.10.2.min.js" type="text/javascript"
  ></script>
```

```
</body>
```

```
</html>
```

基本的には </body> の直前に記述します。

# 書きはじめる

<body>

～～

```
<script src="jquery-1.10.2.min.js" type="text/javascript" ></script>
```

**<script>**

**\$(function() {**

**ここにコードを書く**

**});**

**</script>**

</body>

DOMの解析（HTMLの要素の展開）が終わった段階でコードが実行開始される

※ css同様、JavaScript ファイルは外部呼び出しにするのがセオリー

ハンズオン



# まずは体験、 サイトを書き換えてみよう

- ・ パソナテックさんのサイトを Google Chrome で開いてみる
- ・ Chrome Developer Tool のコンソールを開く  
**mac: [cmd] + [opt] + [J]**  
**windows: [ctrl] + [shift] + [J]**

```
$( 'body' ).fadeOut();
```

```
$( 'body' ).fadeIn(5000); ← 単位はミリ秒、5000ミリ秒 = 5秒
```

```
$( '.tagline' ).text( 'jQueryの勉強をします!!!' );
```

```
$( '.tagline' ).text( 'こんにちは！' ).css( 'color', 'red' );
```

# 画像ギャラリーを作ってみよう

1 2 3



ボタンを押すと、画像がフェードインで表示されるギャラリー的なスク립トを、みんなで書いてみよう。

# 画像ギャラリーを作ってみよう

- ・ <http://bit.ly/codematch>  
↑ にhtmlとcssがあります。  
右下の “Download ZIP” から一式をダウンロードして、適当な場所に展開
- ・ <http://jquery.com/> から jQuery をダウンロード、assetsフォルダに入れる
- ・ index.html の `</body>` の直前に以下を記述

```
<script src="jquery-1.10.2.min.js" type="text/javascript" ></script>
```

```
<script>  
$(function() {  
});  
</script>
```

and more

# プラグインの探し方

- ・ jQuery は便利なプラグインがたくさんあるので、有効活用しよう
  - ・ jQuery オフィシャルサイト
  - ・ jQuery UI
  - ・ GitHub
  - ・ ググればいっぱいでてくる
  - ・ 単にプラグインを紹介しているサイトよりも、実際に使ってみてデモなどを公開しているサイトの方が頼りになる
  - ・ 実際にサイトのコードを開き、どんなプラグインを使っているか見る



# 人のコードを参考に

- ・ ブラウザの「ソースを見る」から、人の書いた JavaScript のコードは簡単に参照できます。
- ・ プラグインのコードもそうですし、GitHub などにもコードはたくさん公開されています。
- ・ コードの書き方には「これだ！」という正解はないことが多く、他人のコードを見ることはとても参考になります。
- ・ また、jQuery だけでなく、JavaScript の知識も深めてください。
- ・ 次のステップは、プラグインを作って GitHub などで公開してみよう！

ありがとう  
ございました