

# Webプログラミング実習1

2017/6/14

Kazuma Sekiguchi

class@cieds.jp

# 課題

- 今までにやったJSの技法を活かしてサイトを作成
  - インタラクシヨンのあるサイトを作成
  - インタクシヨンは効果的に利用すること
  - jQueryの使用OK、jQueryプラグインはそれだけを利用する訳では無いならOK（プラグイン以外も利用する）
- テーマ
  - 梅雨や夏に楽しめるもの、こと、場所の紹介サイト
  - 写真などの素材はネットから取得しても可（商用利用可のもの、CC0ライセンスのもののみ許可）
  - 自分で作成したキャラクターなどは許可
  - トップページのみ作成すればOK

# 課題

- 提出

- デザインカンパ (Ai、Ps、Xd、Sketchなど)
  - AiやPsは画像を埋め込んでください
  - JPEG形式などにしないこと (aiとかpsdのまま)
- HTML、CSS、JS、画像群

- 期限

- 7月5日授業時間までに提出

# jQueryのプラグイン

- jQueryは同じような表現を再利用できるようにしたプラグインが多数存在
  - 1万種類以上はあると思われる
- ほとんどがMITライセンスなので自由に利用可能
  - 商用利用も可
  - ライセンス表記、元の所有者表記すれば改変も可能
- 最近ではプラグインの保守が行われていない傾向が出ているので、最終更新日に注意
  - 2013年以前から更新されていないようなら使用しない方が無難

# jQueryプラグインの使い方

1. 大体はプラグインの説明に書いてある（英語の場合もある）
  - “How to install” , “How to Use” など
2. HTMLをきちんと記述し、指定された必要なJSファイルとCSSファイルを読み込む
  - HTMLも指定されたIDやclassを付ける必要がある場合もある
3. 必要であれば、初期化コードを記述する
4. これで動作する



# jQueryプラグインの初期化（使えるようにする）

古い

```
<script type="text/javascript">
  $(document).ready(function(){
    //何らかのコード
  });
</script>
```



今なら

```
<script>
  $(function(){
    //何らかのコード
  });
</script>
```

# ばらばらに初期化しない

- 初期化コードは一カ所にまとめて記述する

```
<script>
    $(function(){
        //何らかのコード1
    });
</script>
    :
    :
<script>
    $(function(){
        //何らかのコード2
    });
</script>
```



```
<script>
    $(function(){
        //何らかのコード1
        //何らかのコード2
    });
</script>
```

# jQueryを多重読み込み

- プラグインを利用するときにコピペでやろうとする人は要注意（コピペでやるな、ということ）
  - jQueryは1つだけ読み込めば動作する
  - プラグインの説明などでjQueryを読み込むコードを記載している例が多いが、既にjQueryを読み込んでいるならjQueryの読み込みは不要
  - 多重読み込みすると重くなるほか、誤作動の原因
- プラグインを読み込むコードを解釈して利用すること

# jQueryのファイル名に注意

- バージョンやら圧縮形態やらでさまざまな名前が存在する
  - jquery.js jquery.min.js jquery-1.7.2.js
  - jquery-1.7.2.min.js jquery-1.7x.js
- jQueryで利用するプラグインにも似たような名前が多い
  - jquery-easing.js jquery-lightbox.js
- プラグインのjsとjQuery本体は別物
  - プラグインを使いたければ、jQuery本体がいる

# jQueryの判別

- ファイルを開く

```
/*! jQuery v3.2.1 jquery.com | jquery.org/license */
```

- 圧縮されていても先頭にこういう行があれば、jQuery本体
- 無ければ、jQueryのプラグイン

# 組み込み

- プラグインでCSSファイルが渡されて居る場合は、CSSファイルを読み込む
  - 先にプラグインのCSSファイル、後に自分で記述したCSSファイルを読み込む（順番注意！）
  - 必要なJSファイルを読み込み、初期化設定を行う
    - 必要であれば、HTMLに少し改良を加える（スライドショーであれば、画像の追加など）



# 動作確認

- インタラクションの場合、動くことが重要
  - メニュー部分などでプラグインを利用している場合、プラグインがエラーを出したりするとメニューが選択できない不幸が・・・
- 表示や動作がきちんと行えることを念入りに確認する

# オプション

- プラグインによっては、オプションを与えることで変化をもたらせることが可能
  - オプション名と設定できる値はプラグインの説明に書いてある
  - 意外と細かい設定ができるものもある
  - やりたいことがデフォルトで実現できないのであれば、説明を探してみるべし
- 最近は大規模的な関数を通して、外部から変えることができるものも多い

# jQueryプラグインのプラグイン

- いくつかのプラグインを動作させるために別のプラグインが必要なことも
  - jquery.easing.js（良く使う）
    - jQueryのアニメーションにエフェクトの種類を増やすプラグイン
    - 単体では使わない
    - 必要な場合は必ず目的のプラグインの前に読み込ませる

```
<script src="jquery.min.js"></script>  
<script src="jquery.easing.js"></script>  
<script src="jquery.plugin.js"></script>
```

目的のプラグイン

# Canvasを使っているプラグイン

- プラグインによっては、Canvasタグを利用して描画するものが存在
  - Canvasタグを利用している場合、Chromeなどの一部ブラウザではローカル環境では表示されない（サーバーにアップすれば見れる）
    - Crossドメインセキュリティに抵触するため、ブロックされる
- IEとかEdgeを使うとみられるので、そちらで確認すること