

A close-up photograph of pink cherry blossoms in full bloom, with soft lighting and a blurred background. The flowers are the central focus of the image.

Webプログラミング実習1

2017/7/19

Kazuma Sekiguchi

class@cieds.jp

Canvas

- Canvas自体はHTML5で登場したタグ
 - ドキュメント内に絵を描くための場所を生成するタグ
- JSを利用することで、Canvasに対して任意の画像を作成することが可能
 - JSを使って絵を描く、ということ
 - アニメーションも可能
 - イメージ画像をCanvas上に出力することも可能
 - 画像の上にJSで描いた画像を載せる、ということも可能

Canvasのデフォルト

- 結構使いづらい
 - JSを駆使して記述が必要
 - ブラウザーごとの差異が比較的大きい（特に小数点以下の扱いが異なる）
 - レイヤー機能などは存在しない（自分で作成するか、外部ライブラリを使うのが一般的）
 - 描いたものはすべて同一のものとして扱われる
 - ペイントソフトと同じ扱い
 - 異なるものを描いたとしても同一扱いのため、それぞれに違うことをすることが不可能

Canvas

- Canvasタグを利用する際は、width値とheight値、id名を付与する
 - width値とheight値はそのままCanvasの大きさになる
 - id名はJSからcanvasを識別するために必要
 - canvas自体は複数設置が可能
- JSからcanvasを利用する場合は、id名をキーにしてCanvasオブジェクトを取得し、そこに対して、描画を行っていく

jCanvasの利用

- jQueryをベースにCanvasを利用しやすくするためのjQueryプラグイン
 - 以下が比較的簡単に作成、利用することが可能（これをCanvasで実装しようとするとかかなり大変）
 - レイヤー機能
 - オブジェクトのドラッグ
 - 複数図形をグループ化してのドラッグ
- あんまり得意じゃ無い点
 - アニメーション作成

使い方

```
$ctx = $("#main");  
$ctx.addLayer({  
  type: "image", //描画するものの種類。image:画像、rectangle:長方形、arc:曲線、line:直線など  
  name: "mainImage", //レイヤー名。個別の名前を付けておく  
  source: mainImage.src, //画像ファイル名。typeでimageを付けたときにのみ有効  
  x: 0, //描画位置。Canvasの左上を基点にしたとき  
  y: 0, //描画位置。Canvasの左上を基点にしたとき  
  width: 300, //描画サイズ  
  height: 400, //描画サイズ  
  fromCenter: false, //基点を中心にするかどうか。falseでしない  
  index: 0, //重なり順の指定(何となく指定しても無視されるような・・・)  
  draggable: true, //ドラッグを許可するかどうか。trueで許可  
  groups: ["mainLayer"], //レイヤーをグループ化したときの名前  
  dragGroups: ["mainLayer"], //ドラッグするときのグループ名。同じ名前が付いているオブジェクト  
  が同時にドラッグ対象になる  
}).drawLayers(); //レイヤーをすべて描画する
```

レイヤーで描画したとき

- レイヤー内に描画したときは、最後にdrawLayers();を呼び出さないと描画されない（更新されない）
 - レイヤー内に描画した要素は削除することが可能（レイヤーごと削除）
 - 但し削除がかなり時間が掛かるため、非表示にした方が多い場合が多い
 - visible:falseを設定すれば非表示になる

レイヤー内の要素を更新、変更

- レイヤー内のオブジェクトは個別に変更が可能
 - `getLayer()`メソッドを利用して、レイヤー名を指定し、変更したいプロパティ名を指定
 - `=`で変更したい値を指定すれば更新される

```
$ctx.getLayer("mainImage").source = "newImage.png";
```

- `drawLayers()`を呼び出す必要がある点は注意が必要（更新されない）

外部からの変更

- 外部からレイヤー内の要素を変更することが可能
 - ボタンをクリックして画像を挿入
 - サムネイルをクリックして画像を変更
 - など
- jQueryベースのため、他の要素からデータを取得してCanvasに反映させるのは比較的簡単
 - どのようにして必要なデータを持たせるかの方が難しいかも
 - data-属性を上手く利用すると良い