Webプログラミング実習1

2017/9/06

Kazuma Sekiguchi
class@cieds.jp

Canvasでアニメーション

- 実のところ、結構大変なので、あまりオススメしない
  - CSSアニメーションの方が速度が出ることもあり、それほど利用されている場面も少ない
- Canvas自体にはアニメーション機能は備わっていない
  - JSでCanvasにオブジェクトを描画する
  - JSでCanvasに描画したオブジェクトを消去する
  - JSでCanvasにオブジェクトを少し動かして描画する

  の繰り返し
  - setIntervalやrequestAnimationFrameを多用する

requestAnimationFrame

- requestAnimationFrameはアニメーションが可能になったら実行される関数
- アニメーションを終了させるときは、cancelAnimationFrame()を利用する
  - 予めrequestAnimationFrameを実行したときに値を変数に格納しておく必要がある→終了時に使用する

```
var req = requestAnimationFrame(anime);
function anime(){
//アニメーション内容
}
cancelAnimationFrame(req);//アニメーションの終了
```

複数の要素を動かす

- 複数の要素を動かす場面が多い
  - パーティクルを利用する場合など
  - すべての要素を管理する配列またはオブジェクトを利用する
  - オブジェクトなどに現在値、移動値などを格納しておいて、取り出して利用する（場合によっては書き戻す）
  - オブジェクトに格納しておくことで、ループ処理が可能
  - 番号で指し示すことが可能であるため、それぞれの要素に名前を付けたのと同じ状態になる

オブジェクト（復習）

```
var obj = {};
obj.x = 200;
obj.y = 400;
obj.name = “object Element”;
var arr = [];//配列を作成
arr.push = obj;//配列内にオブジェクトを格納
arr[0].x;//→200が取得できる
```

```
var arr[];
arr.push({
x = 200;
y = 400;
name = “object Element”;
});
arr[0].x;//→200が取得できる
```

まとめて記述すると右のようにも記述可能

pushを上手く利用

- pushメソッドを使うと配列に要素を追加可能
  - pushメソッドをループさせれば、必要な数だけ配列内に要素を追加することができる
  - インデックス（配列に付けられる番号）を指定しても格納可能だが、pushの方が手軽なことが多い

移動

- 移動とはベクトルである
  - ベクトル＝XとY座標における移動量（厳密にはだいぶ違う）
  - どのときに使うか→移動のときに利用する
  - 計算：元のX座標とY座標と移動後のX座標とY座標の差
  - XとY座標の差が移動量となる（これを何秒掛けて移動するか）
  - 通常この値をオブジェクトに持たせておくことで、ループ処理し移動させる