

第2章 アニメーション機能の実装

Web応用

第12回 Webページの図形の描画3～アニメーション

第2章

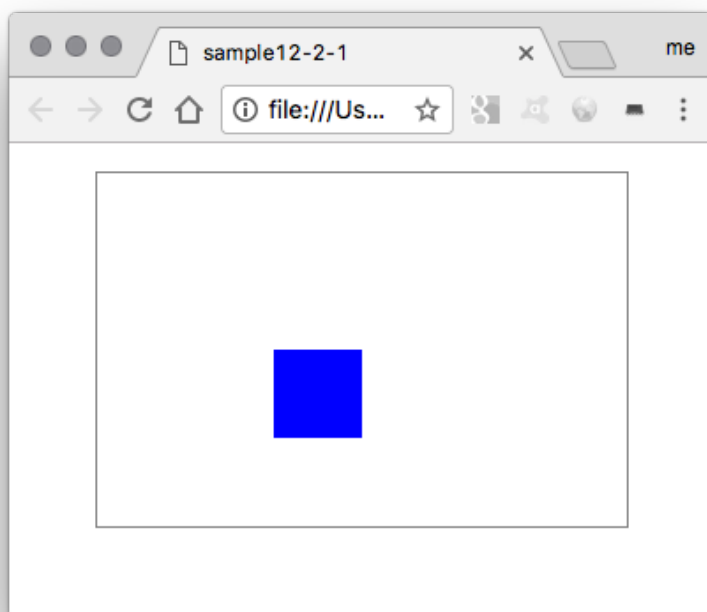
アニメーション機能の実装

第2章 学習目標

アニメーションの機能の実装について理解できる。

1. ファイルの用意

私たちは、描画について学んできました。矩形、円弧、多角形、直線、文字、画像等です。
この回では、それらを動かしてみましょう。



ファイルを用意する

次のコードを記述しましょう(sample12-2-1.html)

■ サンプル

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>sample12-2-1</title>
6      <style>
7        #canvas1{border:1px solid gray;}
8        p{text-align:center;}
9      </style>
10   </head>
11   <body>
12     <p><canvas id="canvas1" width="300" height="200"></canvas></p>
13
14     <script>
15       var canvas1=document.getElementById("canvas1");
16       var ctx1=canvas1.getContext("2d");
17
18
19     </script>
20   </body>
21 </html>
```

2. 関数の定義と実行

1. 変数の初期化

変数の初期化（変数を宣言して初期値を設定する）を行います。

■ サンプル

```
18  //変数の設定と初期化
19  var target = {x:100, y:100, dir:1};    //xy位置と移動方向
```

解説

- ここでは、アニメーションするものを「target」とし、その位置座標をx,y、進行方向をdirとして、それぞれ値を代入して初期化しています。
- このように記述すると、targetのxの値は「target.x」、yの値は「target.y」、移動方向は「target.dir」と記述でき、とても分かりやすくなります。

- target.dirは、target.xの値に毎回追加していく増分です。値が1なので、100,101,102,103…と増えていきます。
もしcanvasの右端に到達したときも1だと突き抜けてしまうので、canvasの右端に到達したら値を-1にするプログラムも取り付けましょう。

2. 関数の定義と実行

描画する関数を「game1()」として定義と実行を記述します。

■ サンプル

```
14 <script>
15   var canvas1=document.getElementById("canvas1");
16   var ctx1=canvas1.getContext("2d");
17
18   //変数の設定と初期化
19   var target = {x:100, y:100, dir:1};           //xy位置と移動方向
20
21   //関数の定義と実行
22   game1();           //関数の実行
23
24   function game1(){           //関数の定義
25       //① 画面をクリア
26       ctx1.clearRect(0,0,300,200);
27
28       //② 描画
29       ctx1.fillStyle = "blue";
30       ctx1.fillRect(target.x,target.y,50,50);
31
32       //③-1 移動計算
33       target.x += target.dir;
34
35       //③-2 壁衝突計算
36       if(target.x < 0 || 250 < target.x){
37           target.dir *= -1;
38       }
39   }
40 </script>
```

解説

- 関数「game1()」の処理内容は大きく3つでそれぞれ ①画面をクリア ②図形を描画 ③次の位置を計算 です。
- ③-1では、targetのxの値(初期値は300px)にdirの値(1)を足してtargetを移動させています。
- ③-2では、targetのxの値(targetの左上の位置になる)が0よりも下になるか、250よりも上になったら(canvasの幅が300、targetの大きさは50なのでxが250で右端にぶつかる)、移動方向に-1を掛けて方向を反転させている。
- 縦棒の記号「|」（バーチカル）は、シフトキーを押しながらキーボードの「¥」を押してください。バーチカル2つ「||」は「又は」という演算子です。

3. 動作確認

動作を確認しましょう。表示されますが、アニメーションにはなっていません。次のページでアニメーションさせましょう。

3. アニメーション化

アニメーションさせるために、関数の実行を変更します。

1. setInterval()

■ サンプル

```
21 //関数の定義と実行
22 game1(); //関数の実行
```

を次のように書き換えます。

■ サンプル

```
21 //関数の定義と実行
22 setInterval(game1,50); //関数の実行
```

- setInterval()は一定時間毎に関数を起動させます。
- ここではgame1()というプログラムを50ミリ秒毎に起動させる式です。
- setInterval()の中の関数名には丸括弧は付けません。
- 1秒は1,000ミリ秒です。50ミリ秒は1秒間に20コマのアニメーションを実行することができます。

2. 動作の確認

動作を確認してみましょう。正しく入力していれば矩形は動くはずです！
動かない場合は、開発者ツールでエラーを見つけましょう。

練習問題1

問題

【クイズ】 択一選択（即解答表示）

次の文は、1秒間にgame関数が何回起動しますか。

```
setInterval(game,1000);
```

- ☐ 1000回
- ☐ 10回
- ☒ 1回

練習問題1の解説

正解は

1回

です。

setIntervalの間隔はミリ秒単位で指定しますので、1000ミリ秒は1秒になります。
ですから1秒間に1回起動します。

第2章 まとめ

アニメーションの機能の実装について理解できた。

第2章 終わり

Web応用

第12回 Webページの図形の描画3～アニメーション

第2章

アニメーション機能の実装

終わり

