

第2章 記述方法と矩形描画

Web応用

第10回 Webページの図形の描画1～基本

第2章

記述方法と矩形描画

第2章 学習目標

要素の取得、コンテキストの取得と、矩形描画が理解できる。

1. canvas要素

「Canvas API」を学んでいきましょう。

ユーティリティーやゲームなど様々なアプリケーションを制作できるCanvas APIは、「HTML5普及の立役者」とも言われています。

Canvas APIを使えば、Web上に自由に図形や画像、文字を描画することができ、それらをJavaScriptでコントロールすることができます。アニメーションも可能です。

1. ファイルを用意する

演習するために次のhtmlファイルを用意しましょう。(sample10-2-1.html)

■ ソース

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>sample10-2-1</title>
6     <style>
7
8   </style>
9 </head>
10 <body>
11
12
13   <script>
14
15   </script>
16 </body>
17 </html>

```

2. canvas要素

それでは、canvas要素をbody要素内に記述しましょう。idは「canvas1」、横幅100px、高さ100pxとします。

■ ソース

```

10 <body>
11   <p><canvas id="canvas1" width="200" height="200"></canvas></p>

```

3. CSSを適用

canvas要素は透明です。このままでは見ることはできません。演習をやりやすくするために、CSSを使って枠線を適用してcanvasが見えるようにしましょう。

■ ソース

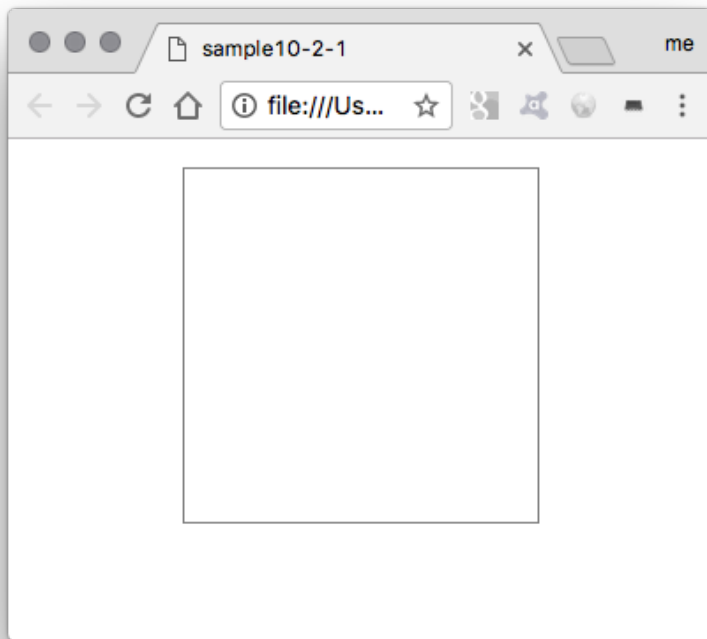
```

6 <style>
7   #canvas1{border:1px solid gray;}
8   p{text-align:center;}
9 </style>

```

解説

サンプルでは、canvas要素の外のp要素に「text-align:center;」を指定して、canvas要素を真ん中揃えにしています。



2. Canvas APIの準備

canvas要素の設置が終わったので、次はJavaScriptで扱うために「要素の取得」を行い、コンテキストというものを生成しましょう。

1. 要素の取得

canvas要素(idは「canvas1」)をJavaScriptで扱うため「要素の取得」を行います。

■ ソース

```
11 <body>
12   <p><canvas id="canvas1" width="200" height="200"></canvas></p>
13
14   <script>
15     var canvas1=document.getElementById("canvas1");
16
17   </script>
18 </body>
```

解説

「document.getElementById()」で要素を取得し、変数「canvas1」に代入することでオブジェクト「canvas1」を生成しています。

2. コンテキスト(工具箱)の生成

次に図形を描くため、コンテキストをオブジェクトのcanvas1から生成します。本講義ではコンテキストを「工具箱」とします。

■ ソース

```
14 <script>
15     var canvas1=document.getElementById("canvas1");
16     var ctx1=canvas1.getContext("2d");
17
18 </script>
```

解説

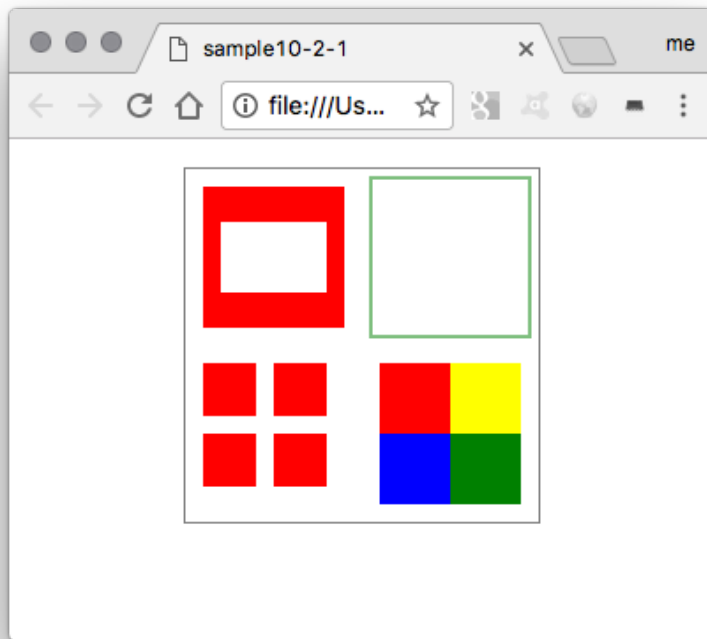
オブジェクト「canvas1」から2次元("2d")のコンテキストを取得し、変数「ctx1」に代入することでオブジェクト「ctx1」を生成します。このオブジェクト「ctx1」で様々な描画が可能になります。

3. 確認

念のために、開発者ツールでエラーを探してみましょう。間違いがあると赤字で内容と行番号を教えてくれます。

3. 矩形を描画

まだ画面は、canvasの枠線だけのはずですね。でははいよいよ図形を描画してみましょう。
最初に描くのは「矩形」です。矩形(くけい)は、四角形のことです。頻繁に使用するためか、記述は比較的簡単です。



1. 座標

canvasには座標があります。ただし通常のグラフ用紙とは座標が違い、**左上が原点(0,0)**です。次の画像で確認しておいてください。

- canvasの座標は左上が原点になります。
- x方向は左から右に向います。
- y方向は上から下に向かいます。

この科目の演習をとおして、しっかり座標の感覚を取得してください。

2. 矩形を描画

それではいよいよ矩形を描画しましょう。以下のコードをJavaScriptに記述します。

■ ソース

```
14 <script>
15   var canvas1=document.getElementById("canvas1");
16   var ctx1=canvas1.getContext("2d");
17
18   //矩形
19   ctx1.fillStyle="red";
20   ctx1.fillRect(10,10,80,80);
21
22
23 </script>
```

- ctx1 : 道具のオブジェクト
- .fillStyle : この行から下で描画される図形の塗り色を指定するプロパティ。ここでは赤色を指定している。またプロパティの場合は「=」を使用することに注意。図形毎に異なった色にしたければ、描画を実行する前に再度.fillStyleを記述して色を指定する。
- .fillRect(開始点のx座標,開始点のy座標,横幅,高さ) : 矩形を描画する
- 開始点の座標(x,y)は図形の左上になるので、注意すること。

3. 消去

「.fillRect()」によく似たものに「.clearRect()」があります。「.clearRect()」は指定した範囲の描画を消去します。

■ ソース

```
18 //矩形
19 ctx1.fillStyle="red";
20 ctx1.fillRect(10,10,80,80);
21
22 //消去
23 ctx1.clearRect(20,30,60,40);
24
25
26 </script>
```

解説

- 開始点座標(20,30)から横幅60px高さ40pxの範囲の描画を消去します。

4. 枠の矩形

strokeRect()は、枠線の矩形を描画することができます。

■ ソース

```
25 //枠の矩形
26 ctx1.strokeStyle="green";
27 ctx1.strokeRect(105,5,90,90);
28
29 </script>
```

解説

- strokeは「線を引く」という意味です。

5. 応用

応用として、以下のような図形を描画してみましょう。

1. 格子状

■ ソース

```
29 //格子状
30 ctx1.fillStyle="red";
31 ctx1.fillRect(10,110,30,30);
32 ctx1.fillRect(10,150,30,30);
33 ctx1.fillRect(50,110,30,30);
34 ctx1.fillRect(50,150,30,30);
```

2. 色違いの格子状

■ ソース

```
36 //色違い
37 ctx1.fillStyle="red";
38 ctx1.fillRect(110,110,40,40);
39 ctx1.fillStyle="blue";
40 ctx1.fillRect(110,150,40,40);
41 ctx1.fillStyle="yellow";
42 ctx1.fillRect(150,110,40,40);
43 ctx1.fillStyle="green";
44 ctx1.fillRect(150,150,40,40);
```

練習問題1

問題

【クイズ】 択一選択（即解答表示）

本講義で解説した`ctx1.clearRect(10,10,80,80);`の働きで正しいものはどれか。

- ☐ 指定した範囲の描画を消去する
- ☐ 矩形を描画する
- ☐ コンテキスト（工具箱）を生成する

練習問題1の解説

正解は、 **指定した範囲の描画を消去する** です。

.clearRect()は、指定した範囲の描画を削除します。

矩形を描くのは、fillRect()やstrokeRect()、コンテキスト（道具箱）を生成するのはcanvas1.getContext("2d") でしたね。

第2章 まとめ

要素の取得、コンテキストの取得と、矩形描画を理解した。

第2章 終わり

Web応用

第10回 Webページの図形の描画1～基本

第2章

記述方法と矩形描画

終わり