

サイバー大学 IT総合学部
専門応用科目
Web応用

第10回
Webページの図形の描画1
～基本
中島 俊治

第10回 学習目標

HTML5の最大の立役者であるCanvas APIを使った図形描画の基礎を学ぶ。

第10回 授業構成

- 第1章 「Canvas API」の概要
- 第2章 記述方法と矩形描画
- 第3章 多角形と直線の描画
- 第4章 円弧・文字・画像の描画

Web応用
第10回 Webページの
図形の描画1～基本

第1章
「Canvas API」の概要

第1章 学習目標

Canvas APIについて実例を交えながら講義し、概要を理解できる。

講義項目

この章の講義項目は次の通り。

1. Canvas APIとは
2. 記述の方法
3. 2章以降の解説

第1章 「Canvas API」の概要

1. Canvas API とは

1-1. Canvas API

Canvas APIは、canvas要素に**JavaScript**
で描画するためのしくみ。

1-2. canvas要素

canvas要素

```
<body>
```

```
<p><canvas id="canvas1" width="300"  
height="200"></canvas></p>
```

- <canvas></canvas> : canvas要素。描画するための範囲を用意する。
- id : id名を指定。ここでは"canvas1"。
- width属性 : 横幅。height属性 : 縦幅。

第1章 「Canvas API」の概要

2. 記述の方法

2-1. 描画範囲を指定

1. canvas要素を記述する

```
<body>  
  <p><canvas id="canvas1" width="300"  
height="200"></canvas></p>
```

- idが「canvas1」 横幅が300px、高さが200pxのcanvas要素を配置している。

2-2. 要素の取得

2. JavaScriptで扱えるようにする。

```
<body>  
  <p><canvas id="canvas1" width="300"  
height="200"></canvas></p>  
  <script>  
    var canvas1 = document.getElementById("canvas1");  
  
  </script>
```

- canvas要素(idは「canvas1」)の取得を行い、JavaScript内で扱えるよう新たに「canvas1」というオブジェクトを生成した。

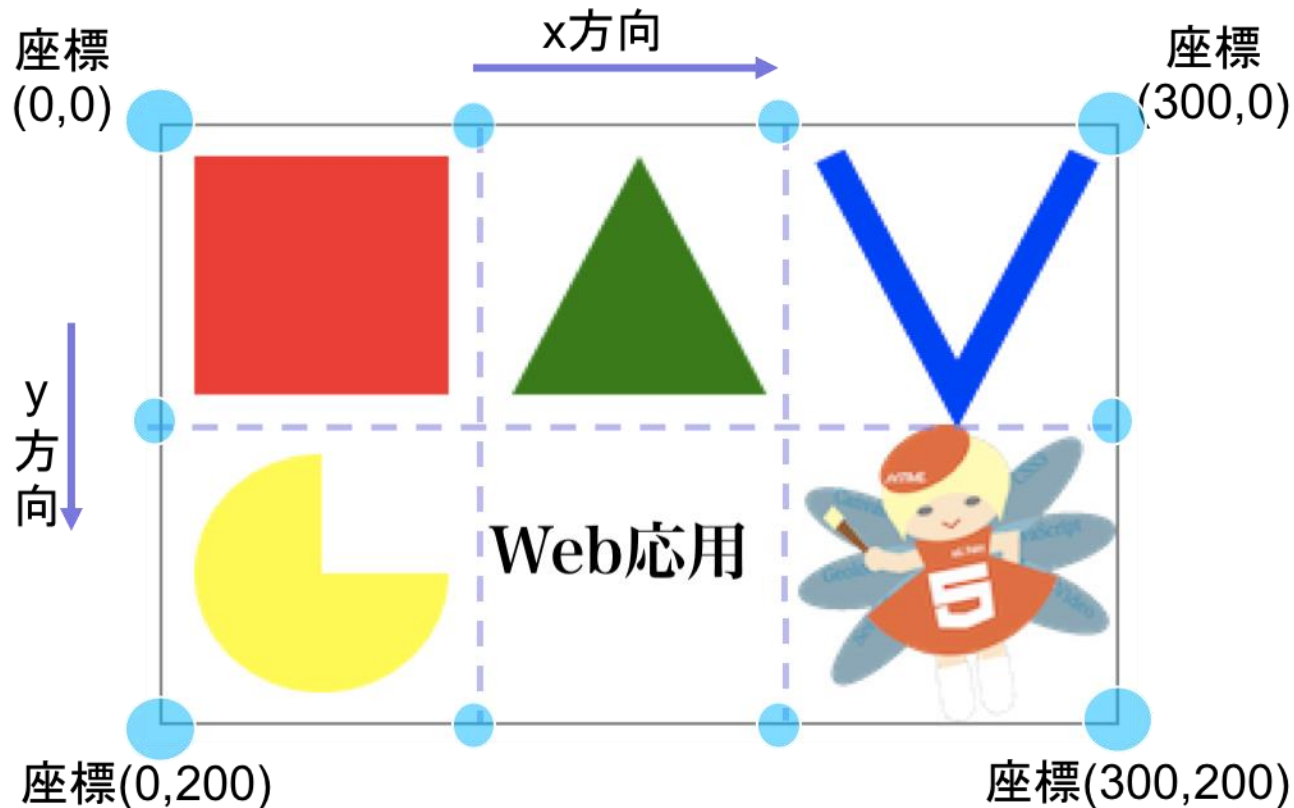
2-3. コンテキストの取得

3. コンテキスト(道具)を得る

```
<body>
  <p><canvas id="canvas1" width="300"
height="200"></canvas></p>
  <script>
    var canvas1 = document.getElementById("canvas1");
    var ctx1 = canvas1.getContext("2d");
  </script>
```

- コンテキスト(Context) : ここでは「道具」「手段」と解釈するとわかりやすい。2dは二次元(平面)のこと。
- ctx1は「道具のオブジェクト」となる。

2-4. canvasの座標




2-4-1. 矩形描画

矩形(くけい)を描画する



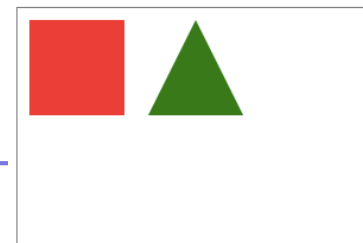
...

```
ctx1.fillStyle="red";  
ctx1.fillRect(10,10,80,80);  
</script>
```

- .ctx1 : 道具のオブジェクト
- .fillStyle : 塗り色を指定するプロパティ
- .fillRect() : 塗りつぶした矩形を描画するメソッド。
値は(開始点のx座標,開始点のy座標,横幅,高さ)を指定
- 開始点は形の左上になる

2-4-2. 多角形描画

多角形を描画する



...

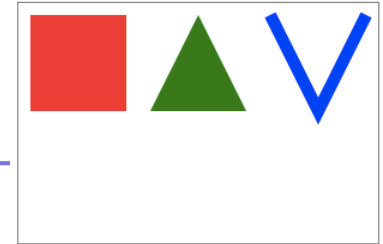
```
ctx1.beginPath();  
ctx1.moveTo(150,10);  
ctx1.lineTo(110,90);  
ctx1.lineTo(190,90);  
ctx1.closePath();  
ctx1.fillStyle="green";  
ctx1.fill();  
</script>
```

```
//パス開始  
//開始点の座標を指定  
//頂点の座標を指定  
//頂点の座標を指定  
//パスを閉じる  
//塗り色を指定  
//塗る
```

- パスは「今から描こうとする図形の頂点を指定」すること

2-4-3. 直線描画

直線を描画する



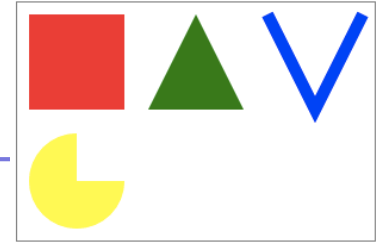
...

```
ctx1.beginPath();  
ctx1.moveTo(210,10);  
ctx1.lineTo(250,90);  
ctx1.lineTo(290,10);  
ctx1.lineWidth=10;  
ctx1.strokeStyle="blue";  
ctx1.stroke();  
</script>
```

```
//パス開始  
//開始点の座標を指定  
//頂点の座標を指定  
//終点の座標を指定  
//線の太さを指定  
//線の色を指定  
//線をひく
```

2-4-4. 円弧描画

円弧を描画する



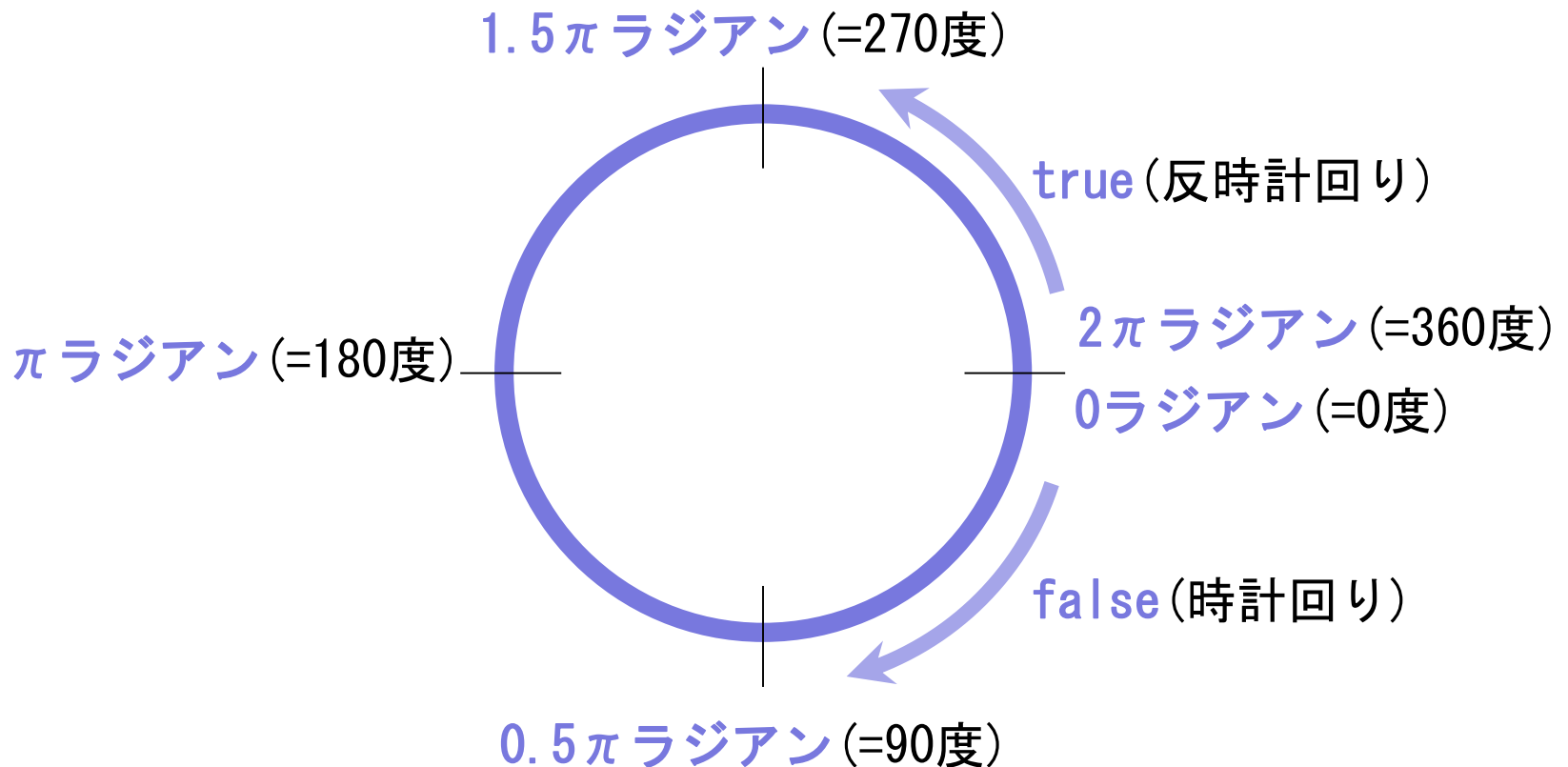
...

```
ctx1.beginPath();           //パス開始
ctx1.moveTo(50,150);        //開始点の座標を指定
ctx1.arc(50,150,40,0,1.5*Math.PI,false); //円弧
ctx1.closePath();          //パスを閉じる
ctx1.fillStyle="yellow";    //塗り色を指定
ctx1.fill();                //塗る
</script>
```

- .arc() : 円弧のメソッド。(中心のx座標,中心のy座標,半径,開始角度,終了角度,方向)を指定する

2-4-5. 角度について

JavaScriptでの角度は「ラジアン(弧度)」



2-4-6. 文字描画

文字を描画する



...

```
ctx1.fillStyle="black";    //塗り色を指定
ctx1.font="20px serif";    //フォントを指定
ctx1.fillText("Web応用",100,150);    //文字
</script>
```

- .fillText() : 文字のメソッド。("表示する文字",x座標,y座標)を指定する
- 開始点の座標は**文字の左下**になる

2-4-7. 画像描画(1)

画像を描画する

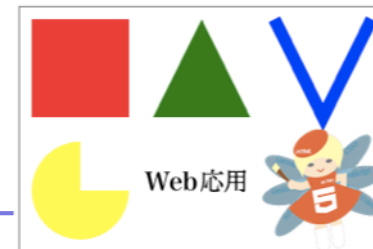


```
...  
var image1 = new Image(); //画像のオブジェクト  
image1.src = "image1.png"; //画像を.srcで指定  
ctx1.drawImage(image1,200,100); //画像  
</script>
```

- Image() : 画像のコンストラクタ(テンプレート)
- .drawImage() : 描画のメソッド。(表示する画像, 開始点のx座標, 開始点のy座標)
- 開始点は**画像の左上**
- 実はこれでは動作しない→(2)へ

2-4-7. 画像描画(2)

画像を描画する(正)



```
...  
var image1 = new Image(); //画像のオブジェクト  
image1.src = "image1.png"; //画像を.srcで指定  
image1.onload = function(){  
    ctx1.drawImage(image1,200,100); //画像  
}  
</script>
```

- 画像は読み込みに時間がかかるので、読み込みが終わったら描画させる
- **`image1.onload=function(){}`** : image1をonload(読み完了)したら、{~}内の処理を行う。

第1章 「Canvas API」の概要

3. 2章以降の解説

3-1. 2章 記述方法と矩形描画

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要素の記述と要素の取得
2. Canvas APIの準備
3. 矩形を描画

3-2. 3章 多角形と直線の描画

多角形・直線の描画の方法が理解できる。

1. 多角形を描画
2. 直線を描画

3-3. 4章 円弧・文字・画像の描画

円弧・文字・画像の描画の方法が理解できる。

1. 円弧を描画
2. 文字を描画
3. 画像を描画

第1章 まとめ

Canvas APIについて実例を通して概要を理解できた。

Web応用

第10回 Webページの
図形の描画1～基本

第1章 「Canvas API」の概要 終わり