

[13차시 교안]

<자바스크립트와 캔버스로 게임만들기(1)>

1. 폼의 유효성 검증

가. HTML 폼 안의 데이터를 서버로 보내기 전에 검증하는 데 사용

나. 입력 필드에서의 잘못을 검증하는 작업

정해진 문자 수 초과, 이메일 주소, 전화번호 등

라. 검증할 내용

- ① 필수적인 필드를 채웠는지?
- ② 유효한 길이의 텍스트를 입력했는지?
- ③ 유효한 이메일 주소를 입력했는지?
- ④ 유효한 날짜를 입력했는지?
- ⑤ 숫자 필드에 텍스트를 입력하지 않았는지?

마. 폼 데이터 접근

- ① 사용자가 입력한 폼 데이터에 접근하기 위해서 폼 안의 id나 name 속성 이용

a. id 속성 이용

- ㉠ 페이지의 요소 식별
- ㉡ getElementById() 사용
- ㉢ name 속성 이용

b. 필드 식별 : form 객체에서 name을 찾음

바. 공백 검증

- ① 필드가 비어있는지 체크
- ② 사용자가 아무것도 입력하지 않은 경우 체크하는 프로그램

```

<script>
    function checkNotEmpty(field) {
        if (field.value.length == 0) {
            alert("필드가 비어있네요!");
            field.focus();
            return false;
        }
        return true;
    }
</script>
<form>
    이름: <input type='text' id='user' >
    <input type='button'
        onclick="checkNotEmpty(document.getElementById('user'))"
        value='확인' />
</form>

```

사. 데이터 길이 검증 : 정해진 개수의 문자만 입력하도록 하는 경우

```

<script>
    function checkLength(elem, min, max) {
        var s = elem.value;
        if (s.length >= min && s.length <= max) {
            return true;
        } else {
            alert(min + " 문자와 " + max + " 문자 사이로 입력해주세요!");
            elem.focus();
            return false;
        }
    }
</script>
<form>
    이름(6-8 문자): <input type='text' id='name' >
    <input type='button'
        onclick="checkLength(document.getElementById('name'), 6, 8)" value='확인' />
</form>

```

아. 정규식

① 특정한 규칙을 가지고 있는 문자열들을 표현하는 수식

② 정규식은 /와 / 내부에 위치

③ (예)

$^{[0-9]+}abc\$$: 숫자가 + 한번이상 반복하고 문자열 "abc"

$/\backslash d \backslash d \backslash d/$: 3자리 숫자 "123", "249"

식	기능	설명
^	시작	문자열의 시작을 표시
\$	끝	문자열의 끝을 표시
.	문자	한 개의 문자와 일치
\d	숫자	한 개의 숫자와 일치
\w	문자와 숫자	한 개의 문자나 숫자와 일치
\s	공백문자	공백, 탭, 줄바꿈, 캐리지리턴 문자와 일치
[]	문자 종류, 문자 범위	[abc]는 a 또는 b 또는 c를 나타낸다. [a-z]는 a부터 z까지 중의 하나, [1-9]는 1부터 9까지 중의 하나를 나타낸다.

④ 수량한정자(몇 번 반복되느냐) 사용

수량 한정자	기능	설명
*	0회 이상 반복	"a*" : "", "a", "aa", "aaa", "aaaa" ..
+	1회 이상	"a+" : "a", "aa", "aaa"
?	0또는 1회	"a?" : "", "a"
{m}	m회	"a{3}" : "aaa"
{m, n}	m회 이상 n회 이하	"a{1, 3}" : "a", "aa", "aaa" 만 가능
(ab)	그룹화	(ab)* : "", "ab", "abab" 등

자. 숫자 검증 예제

숫자로만 된 데이터인지 검증(전화번호, 신용카드 번호, 우편번호 등)

정규 표현식 사용

```

<script>
function checkNumeric(elem, msg) {
    var exp = /^[0-9]+$/;
    if (elem.value.match(exp)) {
        return true;
    } else {
        alert(msg);
        elem.focus();
        return false;
    }
}
</script>
<form>
전화번호(-없이 입력): <input type='text' id='phone'>
<input type='button'
onclick="checkNumeric(document.getElementById('phone'), '숫자만 입력하세요!)"
value='확인' />
</form>

```

```

<script>
function checkSelection(elem, msg) {
    if (elem.value == 0) {
        alert(msg);
        elem.focus();
        return false;
    } else {
        return true;
    }
}
</script>
<form>
과일선택 <select id="fruits" class="required">
<option value="0">선택하세요</option>
<option value="1">사과</option>
<option value="2">배</option>
<option value="3">바나나</option>
</select>
<input type='button'
    onclick="checkSelection(document.getElementById('fruits'), '하나를 선택하여야 합니다.')"
    value='확인' />
</form>

```

2. 캔버스

가. HTML5에서는 웹 페이지 위에 직접 그림을 그릴 수 있는 요소를 가짐

나. <canvas> 요소

자바스크립트와 함께 사용하면 정교한 그림을 웹 페이지 위에 그릴 수 있음

다. 캔버스 생성

- ① HTML 페이지 상에서 사각 형태의 영역
- ② <canvas>요소로 생성
 - a. 기본값으로 경계선을 가지지 않음
 - b. 단순히 그래픽을 위한 컨테이너
 - c. 실제 그림은 자바스크립트를 통하여 코드로 그려야 함
- ③ 항상 id 속성을 지정해야 함

id 속성이 있어야 자바스크립트에서 참조 가능
- ④ 가로와 세로 크기 지정
- ⑤ 경계를 추가하기 위해서는 style 속성 사용

```

<canvas id = "myCanvas" width="300" height="100"
    style ="border:1px dotted red;" >
</canvas>

```

라. 컨텍스트 객체

- ① 캔버스 : 도화지 역할
- ② 컨텍스트(context) 객체
 - a. 자바스크립트에서 물감과 붓의 역할
 - b. 물감 붓을 모아놓은 객체

```
var canvas = document.getElementById("myCanvas");  
  
var context = canvas.getContext("2d");
```

마. 간단한 그림 그리기

- ① getContext()
 - a. 컨텍스트 객체를 반환
 - b. HTML5에 내장된 객체로 그림을 그리는 많은 메서드를 가지고 있음
- ② context.fillStyle : 채우기 속성
색상, 그라디언트, 패턴
- ③ context.fillRect(x, y, width, height)
 - a. 채우기 속성을 가지고 사각형을 화면에 그림
 - b. x, y : 시작 위치
 - c. width, height : 너비, 높이

```
<!DOCTYPE html>  
<html>  
<body>  
  <canvas id="myCanvas" width="200" height="100"  
    style="border: 1px dotted red"></canvas>  
  
  <script>  
    var canvas = document.getElementById("myCanvas");  
    var context = canvas.getContext("2d");  
    context.fillStyle = " #00FF00 " ;  
    context.fillRect(0, 0, 100, 50);  
  </script>  
</body>  
</html>
```

바. 직선 그리기

- ① beginPath() : 패스 지정 초기화
- ② moveTo(x, y) : 직선의 시작점 정의
- ③ lineTo(x, y) : 직선의 끝점 정의
- ④ stroke() : 선 출력
- ⑤ fill() : 색 채우기
- ⑥ closePath() : 패스를 닫음

```
<html>
<head>
  <style>
    body {
      margin: 0px;
      padding: 0px;
    }
  </style>
</head>
<body>
  <canvas id="myCanvas" width="300" height="200"></canvas>
  <script>
    var canvas = document.getElementById('myCanvas');
    var context = canvas.getContext('2d');
    context.beginPath();
    context.moveTo(0, 0);
    context.lineTo(100, 100);
    context.lineTo(150, 50);
    context.lineTo(200, 100);
    context.stroke();
  </script>
</body>
</html>
```

사. 사각형 그리기

- ① rect(x, y, w, h) : (x, y)를 왼쪽 모서리로 하고 가로와 세로가 w, h인 사각형
- ② strokeRect(x, y, w, h) : 테두리만 있는 사각형
- ③ fillRect(x, y, w, h) : 색이 채워진 사각형
- ④ clearRect(x, y, w, h) : 특정 영역을 지움

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

    context.beginPath();
    context.rect(10, 10, 100, 100);
    context.rect(50, 50, 100, 100);

    context.stroke();
  </script>
</body>
</html>
```

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

    context.beginPath();
    context.rect(10, 10, 100, 100);
    context.fillStyle = "yellow";

    context.fill();
  </script>
</body>
</html>
```

아. 사각형 영역을 지우는 함수

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

    context.beginPath();
    context.strokeRect(20, 20, 100, 100);
    context.fillRect(120, 120, 100, 100);
    context.clearRect(70, 70, 100, 100);
  </script>
</body>
</html>
```

3. 원그리기

가. 원을 그리기 위해 arc() 메서드 사용

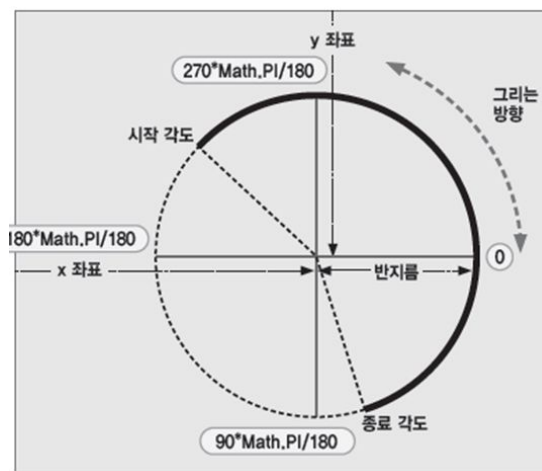
arc(시작 x, y 좌표, 반지름, 시작 각도, 종료 각도, 그리는 방향)

나. 시작 각도와 종료 각도의 단위는 도(degree)를 사용하지 않고 라디안(radian) 사용.

라디안 단위 = 도(degree) * Math.PI/180

다. 그리는 방향

- ① true : 시계 반대 방향
- ② false : 시계 방향




```

<body>
  <canvas id="myCanvas" width="300" height="200"></canvas>
  <script>
    var canvas = document.getElementById('myCanvas');
    var context = canvas.getContext('2d');

    context.beginPath();
    context.arc(100, 100, 80, 0, 2.0 * Math.PI, false);           // 360*Math.PI/180
    context.strokeStyle = "red";
    context.stroke();

    context.beginPath();
    context.arc(100, 100, 60, 0, 1.5 * Math.PI, false);
    context.closePath();
    context.strokeStyle = "blue";
    context.stroke();
  </script>
</body>
</html>

```

```

    context.beginPath();
    context.arc(100, 100, 40, 0, 1.5 * Math.PI, false);
    context.strokeStyle = "green";
    context.stroke();

  </script>
</body>
</html>

```

4. 부채꼴 예제

가. moveTo()메서드 : 시작 지점 지정

나. arc()메서드 : 호를 그림

다. closePath() 메서드로 패스를 종료

자동으로 시작 지점과 호의 양끝점이 연결되어 부채꼴로 완성

```

context.moveTo(50, 50)
context.arc(50, 50, 100, 60*Math.PI/180, 100* Math.PI/180, false);
context.closePath();

context.strokeStyle = "green";
context.stroke();

</script>
</body>
</html>

```

5. 곡선 그리기

(1) 베지어 곡선

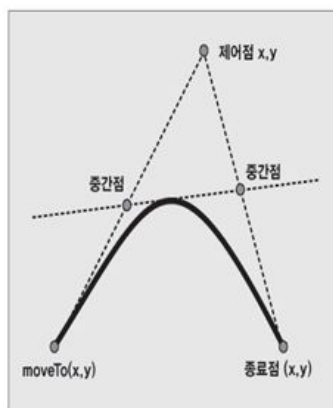
가. n 개의 점을 기준으로 만들 수 있는 $(n-1)$ 차 곡선

나. 2차 베지어 곡선 : 시작점, 종료점, 하나의 제어점 존재

다. `quadraticCurveTo()` 메서드 사용.

라. 하나의 제어점만을 필요로 하기 때문에 : 제어점 x , 제어점 y , 종료점 x , 종료점 y

마. 시작점 : `moveTo()` 메서드를 사용하여 지정



```

<body>

  <canvas id="myCanvas" width="400" height="500"></canvas>

  <script>

    var canvas = document.getElementById('myCanvas');

    var context = canvas.getContext('2d');

    context.moveTo(50,50);

    context.quadraticCurveTo(200, 200, 400, 50)

    context.stroke( );

  </script>

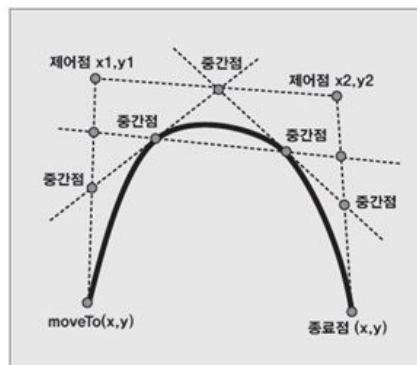
</body>

```

(2) 3차 베지어 곡선

가. 2개의 제어점을 가지므로 총 5개의 보조선이 생성되어 이를 기준으로 곡선이 생성

나. bezierCurveTo(제어점 x1, 제어점 y1, 제어점 x2, 제어점 y2, 종료점 x, 종료점 y) 메서드 사용



```

<body>
  <canvas id="myCanvas" width="300" height="200"></canvas>
  <script>
    var canvas = document.getElementById('myCanvas');
    var context = canvas.getContext('2d');

    context.beginPath();
    context.moveTo(90, 130); (제어점 1, 제어점 1, 제어점 x2, 제어점 y2, 종료점 x, 종료점 y)
    context.bezierCurveTo(140, 10, 288, 10, 288, 130);
    context.lineWidth = 10;

    context.strokeStyle = 'black';
    context.stroke();
  </script>
</body>
</html>

```

6. 일반적인 도형 그리기

(1) 경로를 형성해서 그림

가. beginPath() 호출로 경로 시작

나. closePath() 호출로 경로 종료

다. stroke()이나 fill()을 호출하여 마무리

```

<body>
  <canvas id="myCanvas" width="300" height="200"></canvas>
  <script>
    var canvas = document.getElementById('myCanvas');
    var context = canvas.getContext('2d');

    context.beginPath();
    context.moveTo(50, 100);
    context.lineTo(75, 50);
    context.lineTo(100, 100);
    context.closePath();

    context.fillStyle = "green";
    context.fill();
  </script>
</body>

```

```

<html>
  <head>
    <title> 다각형 그리기 </title>
    <script>
      function polygon()
      {
        var canvas = document.getElementById('canvas');
        context = canvas.getContext('2d');

        context.beginPath();
        context.moveTo(150,10);
        context.lineTo(80,350);
        context.lineTo(350,350);
        context.lineTo(280,10);
        context.closePath();
        context.stroke();
      }
    </script>
  </head>
  <body onload="polygon();">
    <canvas id="canvas" width="700" height="400" style="border:solid 1px #000000">
      canvas 사용하기
    </canvas>
  </body>
</html>

```

```

      context.beginPath();
      context.moveTo(500,10);
      context.lineTo(350,150);
      context.lineTo(430,350);
      context.lineTo(570,350);
      context.lineTo(650,150);
      context.closePath();
      context.fill();
    }
  </script>
</head>
<body onload="polygon();">
  <canvas id="canvas" width="700" height="400" style="border:solid 1px #000000">
    canvas 사용하기
  </canvas>
</body>
</html>

```

7. 텍스트

(1) 캔버스에 텍스트를 그리기 위해 사용하는 속성과 메서드

가. Font : 텍스트 그리는데 사용되는 폰트 속성 정의

나. fillText(text, x, y) : 캔버스에 채워진 텍스트를 그림

다. strokeText(text, x, y) : 캔버스에 텍스트의 외곽선만 그림

```

<body>
  <canvas id="myCanvas" width="300" height="200"></canvas>
  <script>
    var canvas = document.getElementById('myCanvas');
    var context = canvas.getContext('2d');

    context.font = 'italic 38pt Arial'
    context.fillText('Hello World!', 20, 100);
  </script>
</body>
</html>

```

```

<!DOCTYPE HTML>
<html>
<head>
  <style>
    body {
      margin: 0px;
      padding: 0px;
    }
  </style>
</head>
<body>
  <canvas id="myCanvas" width="300" height="200"></canvas>
  <script>
    var canvas = document.getElementById('myCanvas');
    var context = canvas.getContext('2d');

    context.font = 'italic 38pt Arial'
    context.strokeText('Hello World!', 20, 100);
  </script>
</body>
</html>

```

8. 선그리기

- 가. context.lineWidth : 선 두께 지정
- 나. context.strokeStyle : 선 색상 지정
- 다. context.lineCap : 선의 양쪽 끝부분 스타일 지정(butt, round, square)
- 라. lineJoin : 선이 꺾이는 부분 스타일(Bevel, round, miter)

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

    ctx.beginPath();
    ctx.lineWidth = 20;
    ctx.lineCap = "round";

    ctx.moveTo(20, 20);
    ctx.lineTo(200, 20);
    ctx.stroke();
  </script>
</body>
</html>
```

9. 도형 채우기

(1) 단일 색상으로 채우기

- 가. context.fillStyle : 채우기 색 지정
- 나. globalAlpha : 투명도 값 지정(0-1)

(2) 그라디언트로 채우기

- 가. createLinearGradient(x, y, x1, y1) - 선형 그라디언트 생성
- 나. createRadialGradient(x, y, r, x1, y1, r1) - 원형 그라디언트 생성
- 다. addColorStop(offset, color) : 종료 색상과 위치 지정. 그라디언트 위치 0-1사이 실수

(3) 그라디언트로 사용하려면 fillStyle 속성이나 strokeStyle 속성을 그라디언트 객체로 지정하고 도형을 그립니다

(4) 선형 그라디언트 예제

```

<body>
  <canvas id="myCanvas" width="300" height="200"></canvas>
  <script>
    var canvas = document.getElementById('myCanvas');
    var context = canvas.getContext('2d');

    var gradient = context.createLinearGradient(0, 0, 200, 0);
    gradient.addColorStop(0, "white");
    gradient.addColorStop(1, "red");

    context.fillStyle = gradient;
    context.fillRect(10, 10, 180, 90);
  </script>
</body>
</html>

```

(5) 원형 그라디언트 예제

```

<body>
  <canvas id="myCanvas" width="300" height="200"></canvas>
  <script>
    var canvas = document.getElementById('myCanvas');
    var context = canvas.getContext('2d');

    var gradient = context.createRadialGradient(70, 50, 10, 80, 60, 120);
    gradient.addColorStop(0, "white");
    gradient.addColorStop(1, "red");

    context.fillStyle = gradient;
    context.fillRect(10, 10, 180, 90);
  </script>
</body>
</html>

```

(6) 패턴 채우기

가. 패턴으로 채우는 방법

- ① createPattern() 메서드 사용 - 패턴 객체 생성
- ② fillStyle 속성으로 패턴 객체 설정
- ③ fill() 메서드를 호출하여 도형의 내부를 패턴으로 채움

나. createPattern() 메서드의 매개변수 : 이미지 객체, repeat 옵션

repeat 옵션 : repeat(기본값), repeat-x, repeat-y, no-repeat

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

    var image = new Image();
    image.src = "pattern.png";
    image.onload = function () {
      var pattern = context.createPattern(image, "repeat");

      context.rect(0, 0, canvas.width, canvas.height);
      context.fillStyle = pattern;
      context.fill();
    };
  </script>
</body>
</html>
```

10. 이미지 그리기

(1) drawImage(image, dx, dy)

가. 이미지를 원래 크기로 삽입

나. 화면의 dx, dy 위치에 이미지 삽입

(2) drawImage(image, dx, dy, dw, dh) : 이미지를 지정 한 사이즈로 삽입

(3) drawImage(image, sx, sy, sw, sh, dx, dy, dw, dh) : 이미지의 일부분을 잘라내어 삽입

```
<body>
  <canvas id="myCanvas" width="600" height="400"></canvas>
  <script>
    var canvas = document.getElementById("myCanvas");
    var context = canvas.getContext("2d");
    var image = new Image();
    image.src = "html5_logo.png";
    image.onload = function () {
      context.drawImage(image, 0, 0);
    };

  </script>
</body>
```

11. 그림자 스타일

- 가. shadowColor : 그림자 색깔 지정(기본값 : 투명)
- 나. shadowOffsetX : 대상을 기준으로 그림자의 x 좌표 지정(기본값 : 0)
- 다. shadowOffsetY : 대상을 기준으로 그림자의 y 좌표 지정(기본값 : 0)
- 라. shadowBlur : 그림자의 흐림 정도 지정(기본값 : 0)

```
<html>
  <head>
    <title> 그림자 </title>
    <script>
      function rect()
      {
        var canvas = document.getElementById('canvas');
        context = canvas.getContext('2d');

        context.shadowOffsetX = 10;
        context.shadowOffsetY = 10;
        context.shadowColor = 'gray';
        context.shadowBlur = 1;
        context.fillRect(120, 120, 100, 100);
      }
    </script>
  </head>
```

```
<body onload="rect();">
  <canvas id="canvas" width="500" height="400">
    canvas 사용하기
  </canvas>
</body>
</html>
```