

Canvas

노트북: 뉴미디어전송

만든 날짜: 2018-08-16 오후 3:16

수정한 날짜: 2018-10-15 오후 4:52

작성자: yurilee9696@gmail.com

URL: https://www.w3schools.com/tags/canvas_measuretext.asp

예제랑 예시랑 결과 보기 |

<canvas 생성하기>

-canvas : 외부 플러그인 없이 웹페이지 위에 직접 그림을 그릴 수 있도록 해주는 요소

-자바스크립트를 활용하면 정교한 그림을 그릴 수 있다.

<canvas **id="myCanvas" width=" " height=" "** style="border:1px dotted red"></canvas> //필수 속성

<주의사항>

-id, width, height를 설정해야 한다.

-한 페이지 내에서 여러 개의 <canvas>를 삽입할 수 있다.

-경계를 추가하기 위해 style 속성을 사용한다.

-<canvas>에 그려지는 모든 것들은 자바스크립트내에서 작성한다.

-기본적으로 <canvas> 요소는 경계선과 내용을 갖지 않는다

-canvas에 그려지는 모든 것들은 자바스크립트 내에서 행해져야 한다.

<canvas 좌표계>

-2차원 격자 모양으로 왼쪽 위 모서리 좌표는(0,0)

크기는 width, height에 따라 결정된다.

(0,0)

(width,height)

<자바스크립트와 연결>

-컨텍스트 객체 : 자바스크립트의 getContext()메서드를 통해 html5의 "2d"객체를 호출한다. 그림을 그리는데 필요한 속성과 값을 갖고 있는 객체이다.

```
var canvas = document.getElementById("canvas");
```

```
var context = canvas.getContext("2d"); //내장형 오브젝트로 경로, 사각형, 원, 글자, 그림 등을 그리는데 많은 속성과 값 가짐
```

*****캔버스는 가장 최근 속성 값을 기준으로 삼기 때문에 아무 지정도 하지 않으면 가장 최근에 된 것으로 된다*****

사각형 그리기

함수	설명
fillRect(x,y,w,h)	속을 채운 사각형 (현재의 fillstyle으로 채워진 사각형을 그림)
strokeRect(x,y,w,h)	사각형 테두리만
clearRect(x,y,w,h)	지정한 사각영역을 지움
rect(x,y,w,h)	윤곽선만 있는 사각형을 그리는 함수

Canvas Element 색상

속성	설명
fillStyle	도형의 내부 색상 결정하기 (기본값은 검정색) 예) cvs.fillStyle="#ff0000";
strokeStyle	선 색상 결정하기 ex) ctx.strokeStyle='blue';
css 색상형식	값
16진	#FF0000 #F00
RGB	rgb(255,0,0) rgb(100%, 0%, 0%)
RGBA	rgba(255,0,0,1.0) rgba(100%, 0%, 0%, 1.0) 0 완전 투명 ~ 1 완전 불투명
HSL	rgb(255,0,0) rgb(100%,0%,0%) h(색조), s(채도), L(명도)
HSLA	rgba(255,0,0,1.0) rgba(100%, 0%, 0%, 1.0)

<선 그리기(path의 이해)>

1.새 경로 초기화

beginPath();

2.기하 도형의 형태 정의

moveTo(); //시작점 이동

lineTo(); 경로를 그리며 좌표 이동

3.도형을 실제로 그리거나 채우기

stroke(), fill()

4.선의 종류

line (직선)

bezier (곡선)

arc (원)

5.선 닫기

closePath();

<canvas 선의 다양한 효과>

1.선의 굵기

context.lineWidth=20 (픽셀)

2.선 끝의 모양

context.lineCap=[butt (기본), round, square] //끝 모양은 butt랑 square랑 똑같은데 square가 좀 더 길이가 김
순서대로 butt, round, square임



3.선의 접합부

context.lineJoin = [bevel, round, *miter]

순서대로 bevel, round, miter임



*는 default 값

<원 그리기>

arc : 부채꼴 그리는 함수

1.(x,y)가 중심이고 반지름이 radius인 원

2.시작각도와 마지막 각도를 연결하면 만나는 점이 2개 있다.

3.Anticlockwise가 0이며 시계방향, 1이면 반시계 방향이다.

4.실선의 원호가 생긴다.

5.새로운 현재점은 (epx,epy)가 된다

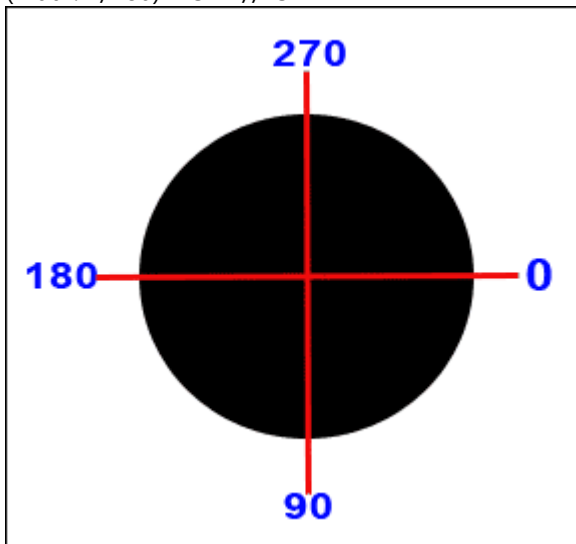
arc(x,y,radius,startAngle,endAngle,anticlockwise) : 원 또는 원호 그리는 함수 //anticlockwise 지정 안하면 0
시작 x좌표, 시작 y 좌표, 반지름, 시작 각도, 끝 각도, 방향

<각도를 라디언 각도로 변환>

90도 -> $0.5 * \text{Math.PI}$

360도 -> $2 * \text{Math.PI}$

$(\text{Math.PI}/180) * 45$ //45도



<Canvas 문자의 이해>

<Text 모양>

-fillText 함수 (문자열, x, y) //캔버스에 색이 칠해진 문자를 그림 ex) ctx.fillText("Hello World",10,50);

Hello World

strokeText 함수 (문자열, x,y) //캔버스에 문자를 그림(색은 채워지지 않음) ex) ctx.strokeText("Hello World",10,50);

Hello World

ctx.font="30px Arial"; //폰트 정의

<Text 음영>

ctx.fillStyle="rgba(0,0,255,0.8)"; //글자색

ctx.shadowColor="rgba(30,30,30,0.9)"; //그림자색 맨 끝은 투명도

ctx.shadowOffsetX="5"; //그림자 오른쪽으로 이동

ctx.shadowOffsetY="5"; //그림자 아래쪽으로 이동

ctx.shadowBlur="5"; //그림자 블러 범위

<Text 배치>

textAlign : left, right, center, start, end

start

end

left

center

right

center

<Canvas Text 관련 속성 및 메소드>

1.textWidth : pixel

2.textHeight : pixel

3.measureText() //글자의 크기를 구하는 함수 //캔버스에 텍스트를 쓰기 전 텍스트의 너비를 알아야하는 경우

ex) ctx.measureText("문자열").width

4.textBaseline : middle, top, bottom, alphabetic, hanging, ideographic //ctx.textBaseline='alphabetic';

//alphabetic이 ideographic보다 더 아래 있다

middle bottom alphabetic ideographic
top hanging

font-style : 요소의 글꼴 //document.getElementById("myP").style.fontStyle = "italic";

font-variant : 요소를 작은 대문자 글꼴로 설정 //document.getElementById("myP").style.fontVariant = "small-caps";

font-weight : 요소의 글꼴 두께 //document.getElementById("myP").style.fontWeight = "900";

font-size : 요소의 글꼴 크기 //document.getElementById("myP").style.fontSize = "xx-large";

font-family : 요소의 글꼴 //document.getElementById("myP").style.fontFamily = "Impact,Charcoal,sans-serif";

<canvas 그라디언트>

색으로 채워지는 사각형, 원, 직선, 문자등에 사용

캔버스에 그려지는 모양은 solid color에 제한받지 않음

컬러지정

-Red, #ff0000, rgb(255,0,0), rgba(255,0,0,1)등

그라데이션 지정

-선형 그라데이션 : **createLinearGradient**(x0,y0,x1,y1) //앞에 점 2개 시작점, 뒤에 점 2개 끝점

-원형 그래디언트 : **createRadialGradient**(x0, y0, r0, x1, y1, r1) //앞에 점 2개 시작점, 시작 반지름 크기, 점 2개 끝점, 끝나는 반지름 크기

-**addColorStop**(offset, color) : offset 위치에 그래데이션 경계색 지정 //offset이 0이면 시작색 1이면 끝색

<Canvas 선형그래디언트 작업 순서> //순서 그대로!!! 빼먹지 말고!!

1.Canvasgradient 객체 생성

=> context.createLinearGradient(x0, y0, x1, y1)

2.객체 저장

=>var gra=context.createLineatGradient(x0, y0, x1, y1)

3.색상 offset 추가하기

=> gra.addColotStop(offset, 색상)

4.FillStyle 적용하기

=> ctx.fillStyle = gra;

=>ctx.fillRect();

<canvas 이미지>

var img = new Image();

img.src="../ww.jpg";

img.onload=function(){

context.drawImage(img,100,100);

}

함수	설명
drawImage(image, dx, dy)	원하는 위치에 원래 크기로 그림을 삽입
drawImage(image, dx, dy, dw, dh)	원하는 위치에 이미지의 사이즈를 지정해서 삽입
drawImage(image, sx, sy, sw, sh , dx, dy, dw, dh)	원래 이미지로부터 일부분을 잘라내 삽입, 잘라내는 부분은 (sx, sy)를 왼쪽 위 정점으로 하여 넓이 sw, 높이 sh인 사각형 모양임

64p 예시는 무시

<이미지 로딩 순서>

1.Image 요소 동적 생성

var img = new image();

2.이미지 로딩

img.src=" "

3.캔버스 출력

image.onload = function(){

context.drawImage(img,100,100);

}

false만 정해져 있음 true값에 어떤 값을 써도 괜찮음

font 속성을 통해 글자크기 글자체등을 지정할 수 있다

