

웹게임 오브젝트

선 & 원 & 사각형

ctx.save & ctx.restore

Math.random



오늘 학습 내용

- 선, 원, 사각형 기초
- 원 그리기 심화
 - 각도와 그리는 방향에 따른 결과의 차이
 - 태극 그리기
- 오브젝트 이동 및 회전
 - Translate
 - rotate
- 랜덤 컬러 생성
 - Math.floor
 - Math.round

웹 프로그래밍 기본 파일

- HTML 파일
 - 확장자 : .html
 - Ex) myfirst.html
- JavaScript 파일
 - 확장자 : .js
 - Ex) myfirst.js

myfirst.html ★★

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Draw Lines</title>
  <script src="myfirst.js"></script>
  <style>
    canvas {
      border: 1px solid black;
    }
  </style>
</head>

<body onload="init()">
  <canvas id="myCanvas" width="400" height="400">
    이 브라우저는 HTML5 canvas 태그를 제공하지 않습니다.
  </canvas>
</body>
</html>
```

myfirst.js ★ ★ ★

```
var vcanvas, ctx;
```

```
function init() {
```

```
    vcanvas = document.getElementById("myCanvas");
```

```
    ctx = vcanvas.getContext("2d");
```

```
}
```

선, 원, 사각형 기초

선, 원, 사각형 함수 만들기

1 Draw Line

```
ctx.beginPath();  
ctx.moveTo(100, 100);  
ctx.lineTo(200, 100);  
ctx.lineTo(100, 200);  
ctx.lineTo(200, 200);  
ctx.closePath();  
ctx.stroke();
```

2 Draw Circle

```
ctx.beginPath();
```

```
ctx.arc(200, 200, 100, 0, 2*Math.PI, true);
```

```
// x좌표, y좌표, 반지름, 시작각도, 끝각도, 방향
```

```
ctx.stroke();
```


3 Draw Circle-fill

```
ctx.beginPath();  
ctx.lineWidth=10;  
ctx.arc(200, 200, 100, 0, 2*Math.PI, false);  
    // x좌표, y좌표, 반지름, 시작각도, 끝각도, 그리는 방향  
ctx.strokeStyle="yellow";  
ctx.stroke();  
ctx.fillStyle="blue";  
ctx.fill();
```

4 Draw Rect

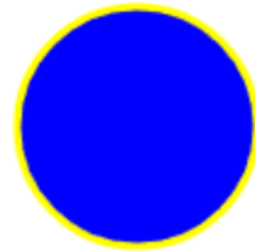
```
ctx.lineWidth=3;
```

```
ctx.strokeStyle="rgb(0, 200, 0)";
```

```
ctx.strokeRect(100, 100, 200, 200);
```

```
// x좌표, y좌표, 사각형너비, 높이
```

미션 1



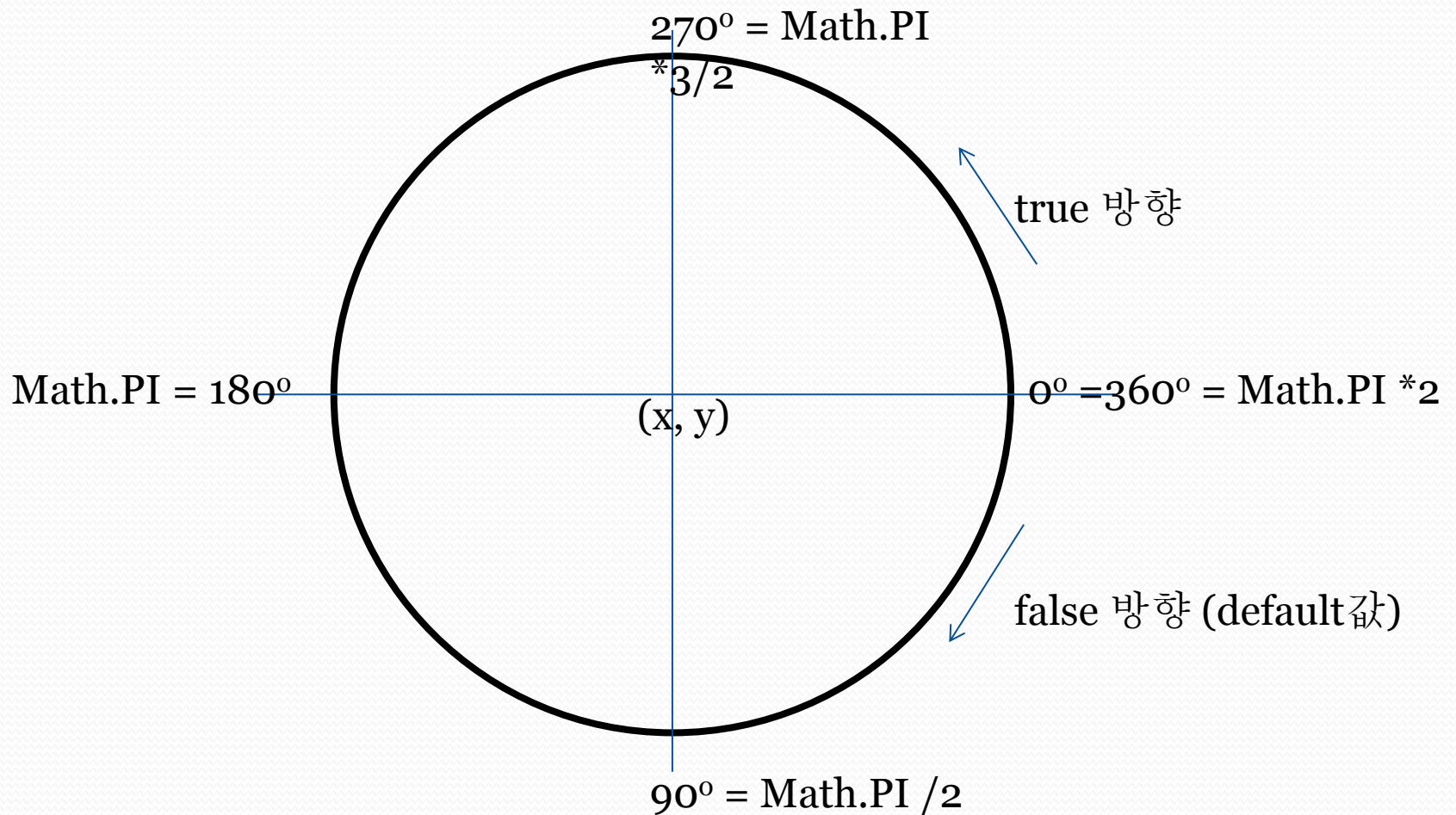
미션 1_{bis} : 함수화 하기

```
function drawLine(x, y, color) {  
}  
function drawArc(x, y, r) {  
}  
function drawRect(x, y, w, h) {  
}  
function init() {  
    vcanvas = document.getElementById("myCanvas");  
    ctx = vcanvas.getContext("2d");  
  
    drawLine(50, 50, "black");  
    drawLine(250, 250, "gray");  
    drawArc(300, 100, 50);  
    drawRect(50, 250, 100, 100);  
}
```

원 그리기 심화

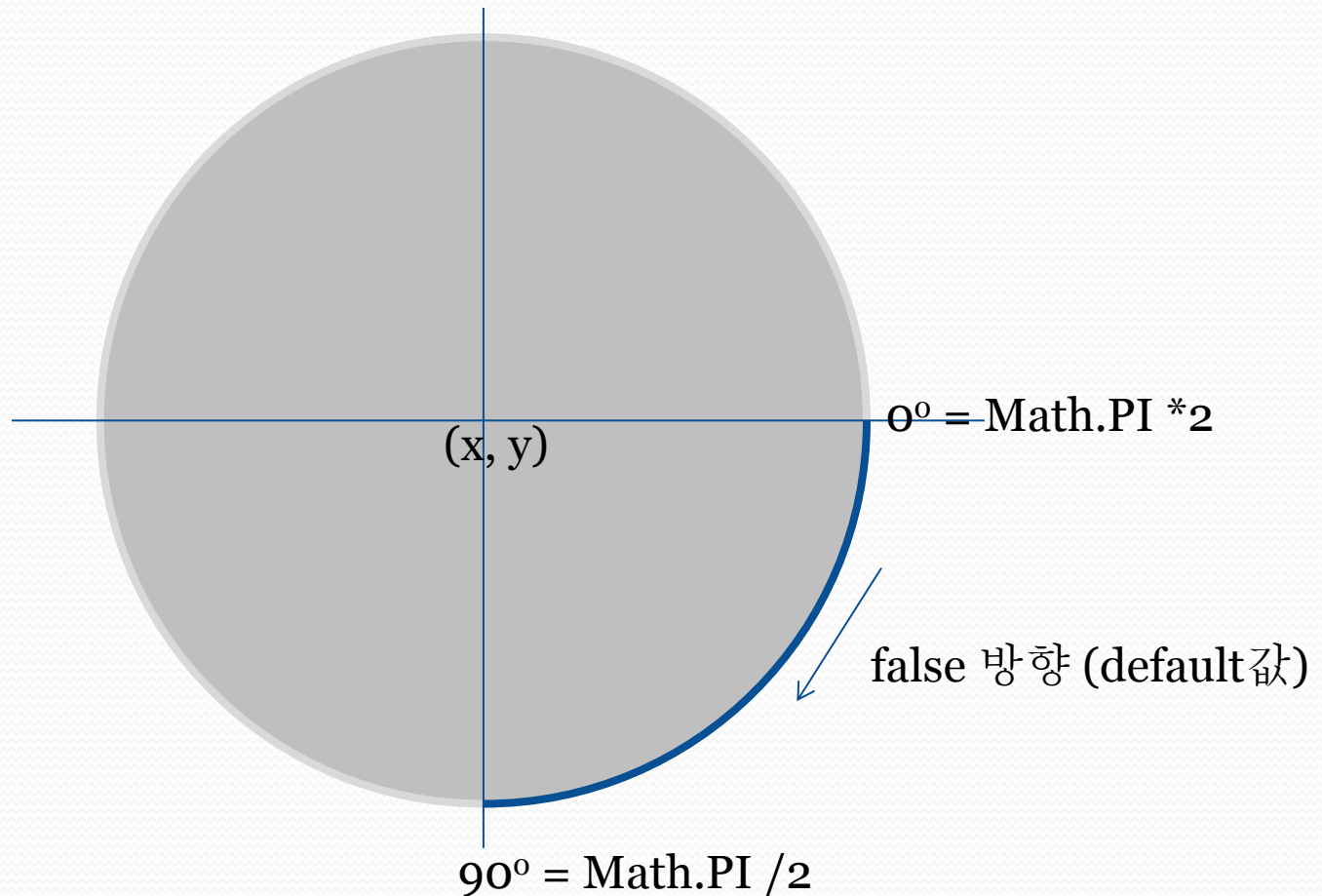
각도와 그리는 방향에 따른 결과의 차이
태극 그리기

원의 성격

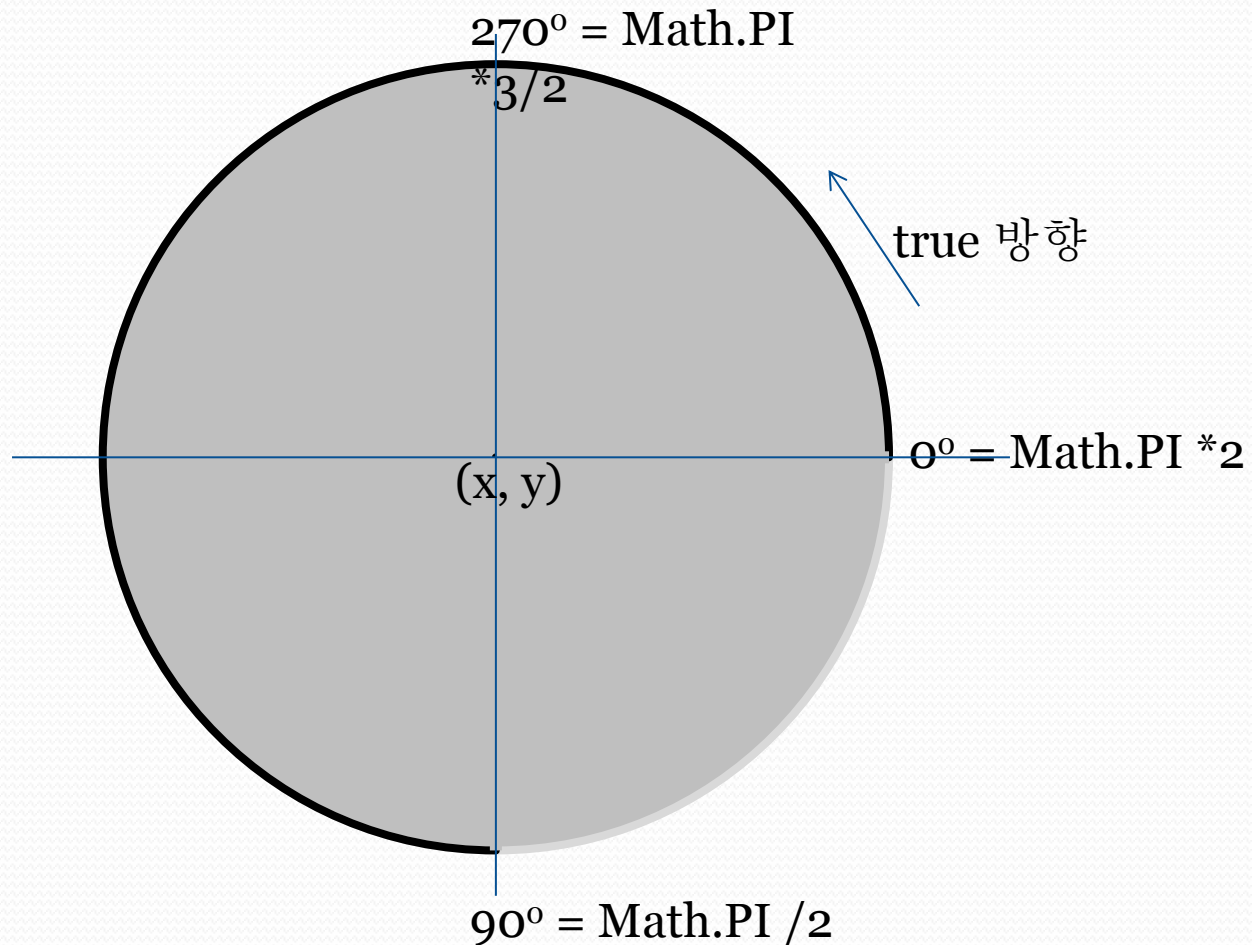


```
arc(x, y, r, 0, 0.5 * Math.PI); //
```

기본 방향, false 방향

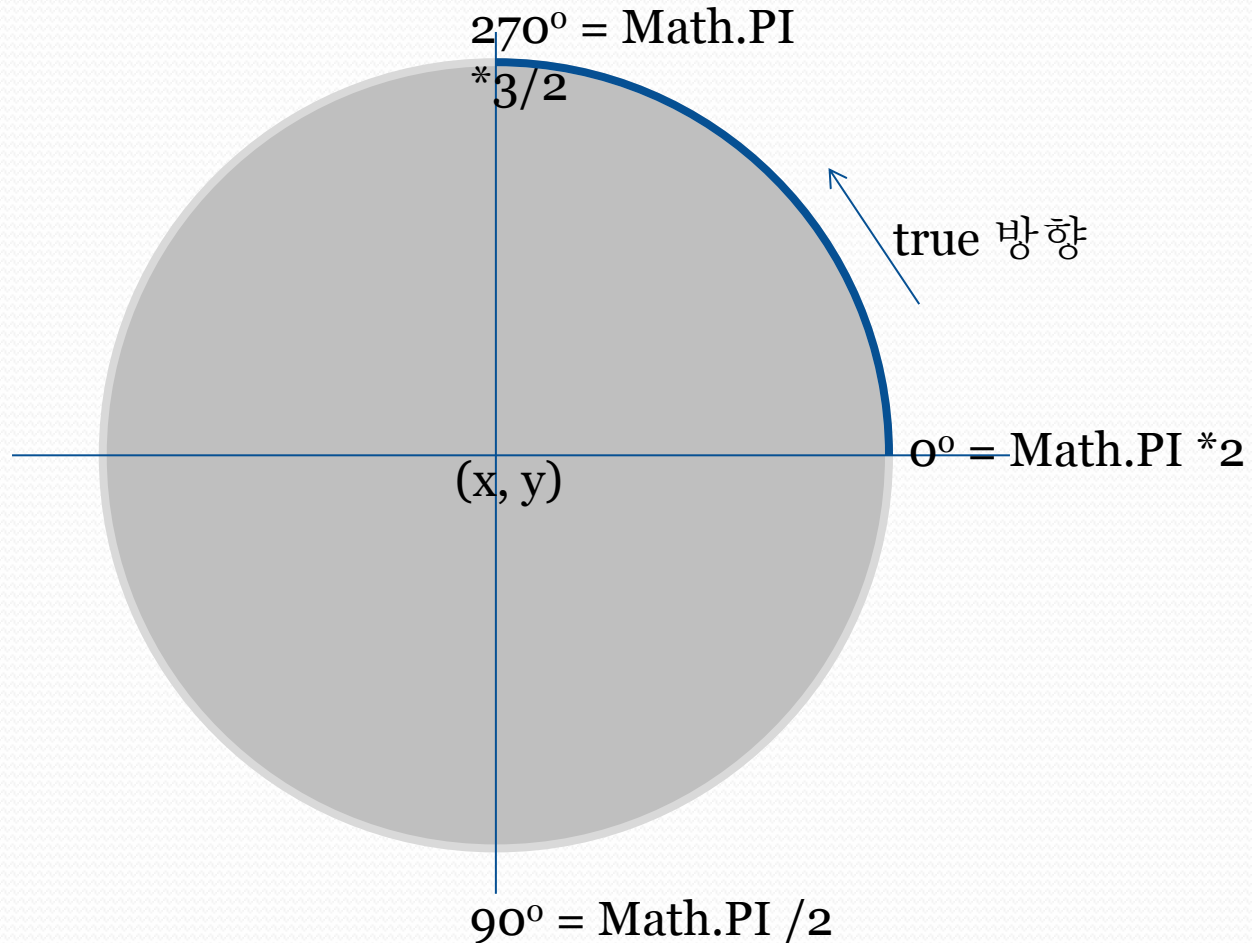


`arc(x, y, r, 0, 0.5 * Math.PI); //`
true 방향




```
arc(x, y, r, 0, 1.5 * Math.PI); //
```

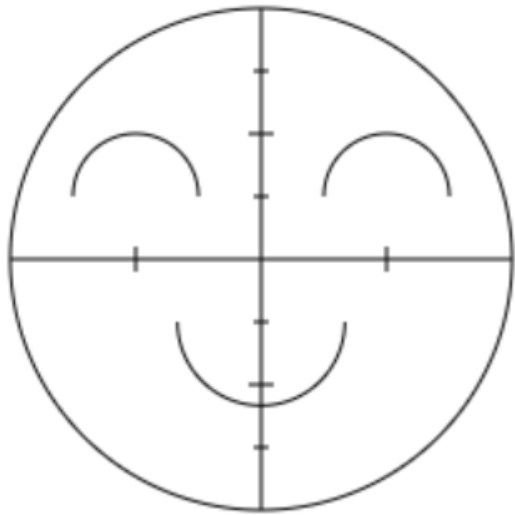
true 방향



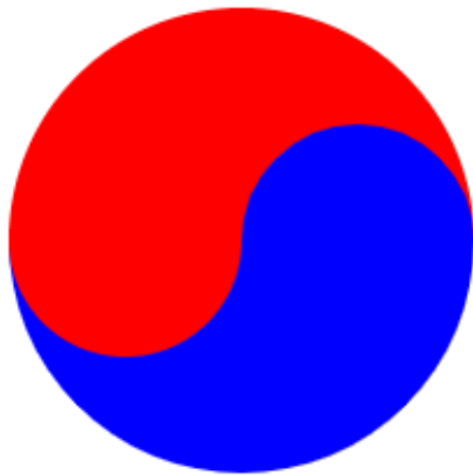
미션 2



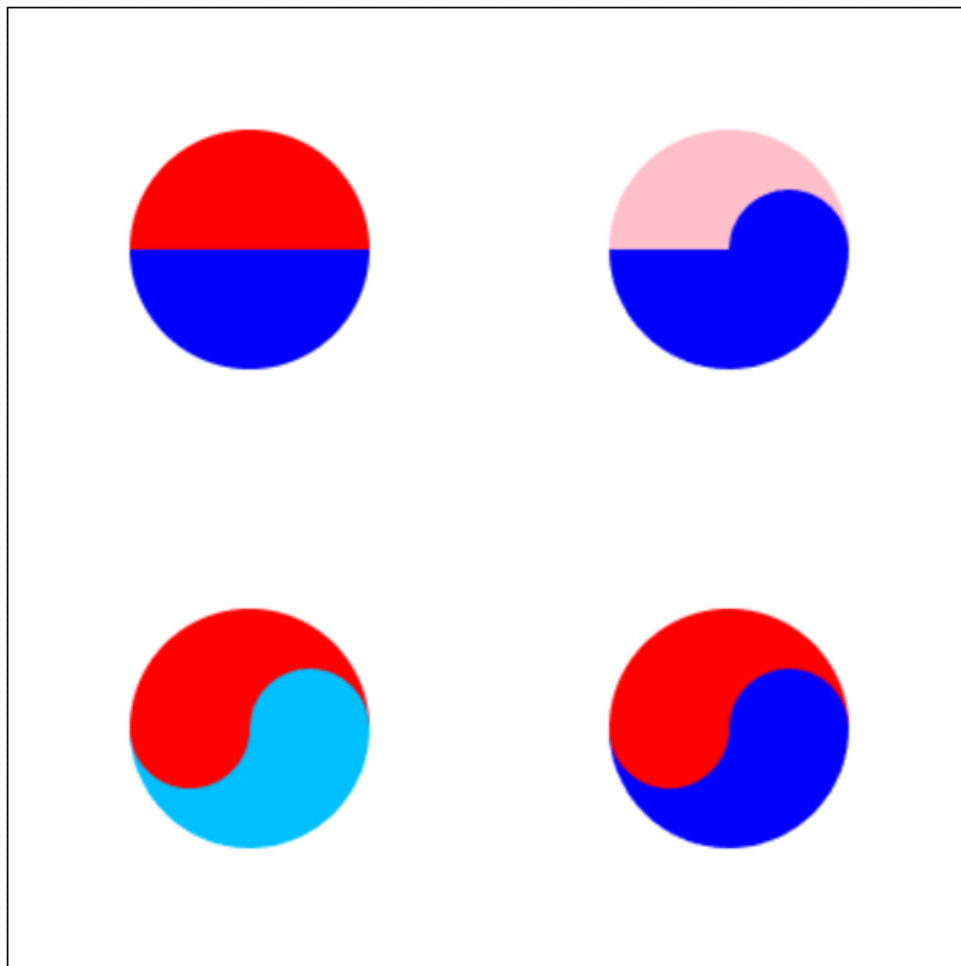
하루



미션 3



이름



태극 함수

```
1. function taegeuk(x, y, r) {  
2.     var r2 = r / 2;  
3.  
4.     ctx.beginPath();  
5.     ctx.fillStyle = "red";  
6.     ctx.arc(x, y, r, 0, Math.PI, true);  
7.     ctx.fill();  
8.     ctx.beginPath();  
9.     ctx.fillStyle = "blue";  
10.    ctx.arc(x, y, r, 0, Math.PI);  
11.    ctx.arc(x + r2, y, r2, 0, Math.PI, true);  
12.    ctx.fill();  
13.    ctx.beginPath();  
14.    ctx.fillStyle = "red";  
15.    ctx.arc(x - r2, y, r2, 0, Math.PI);  
16.    ctx.fill();  
17. }
```

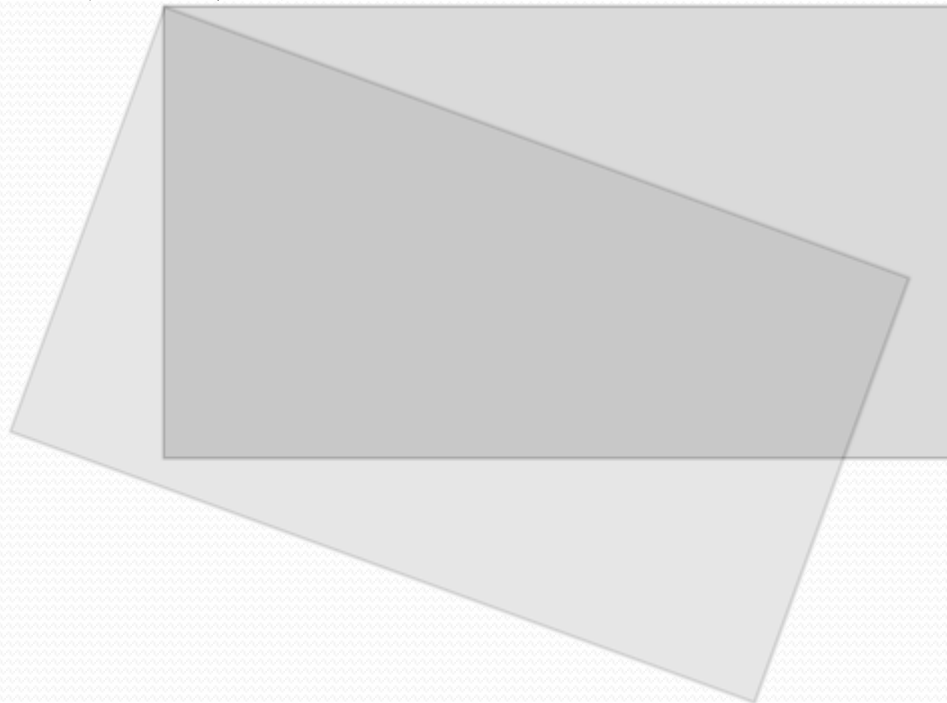
이동 및 회전

Save & restore

Translate & rotate

Canvas에서 회전이란

- 전체 캔버스 회전을 의미
 - canvas 요소에 사용할 수 있는 유일한 회전 방법은 전체 캔버스를 회전시키는 것



1 저장 & 캔버스 이동

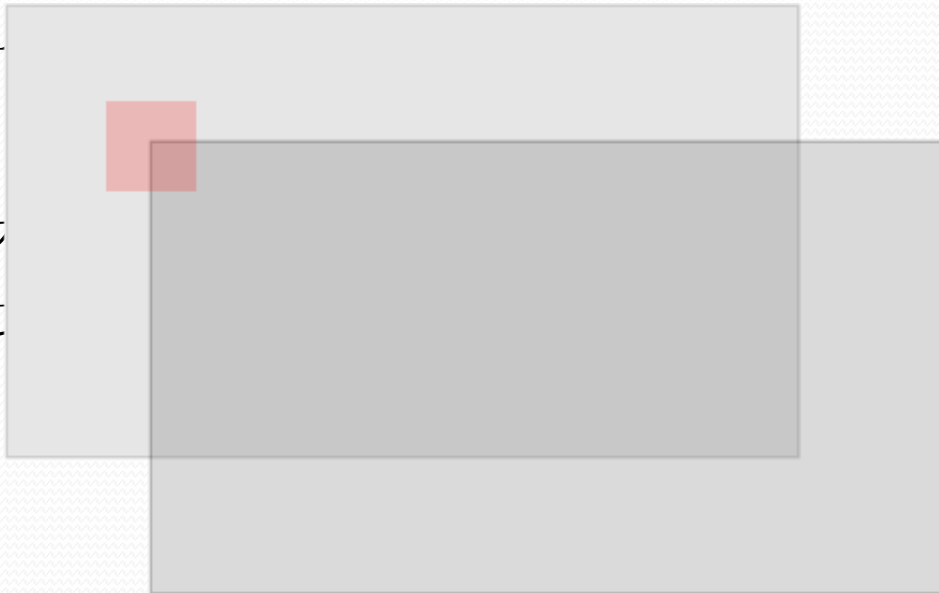
- 캔버스에서 회전 시 해당 요소뿐만 아니라 모든 요소가 회전됨.

1. 현재 캔버스의 모든 컨텍스트 개체(요소)들을 저장한다

- `ctx.save ();`

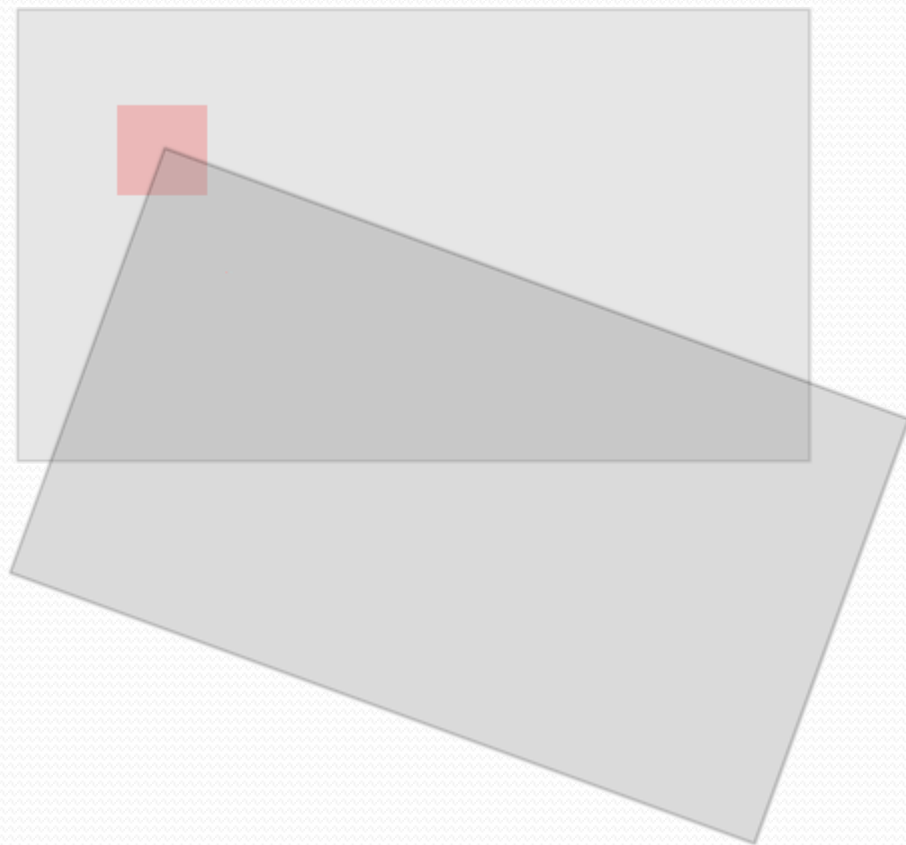
2. 전체 캔버스를 회전하고 요소의 가운데로 이동한다

- `ctx.translate (x, y);`



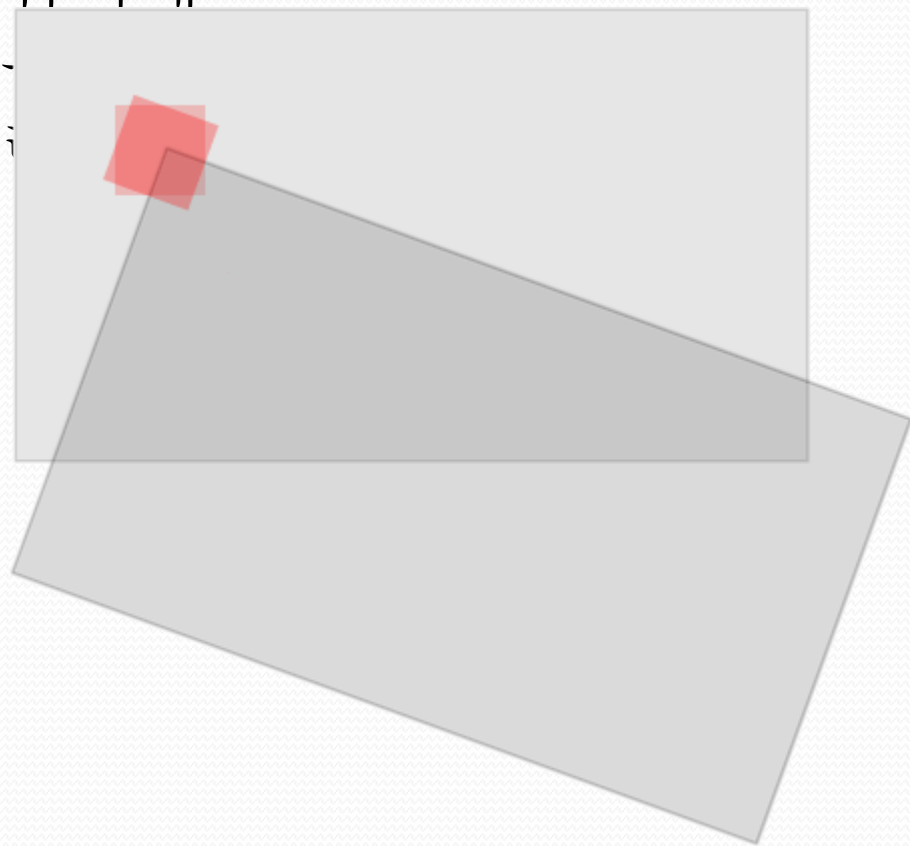
2 회전

- rotate () 메서드를 사용하여 원하는 회전을 수행한다
- ctx.rotate (angle);



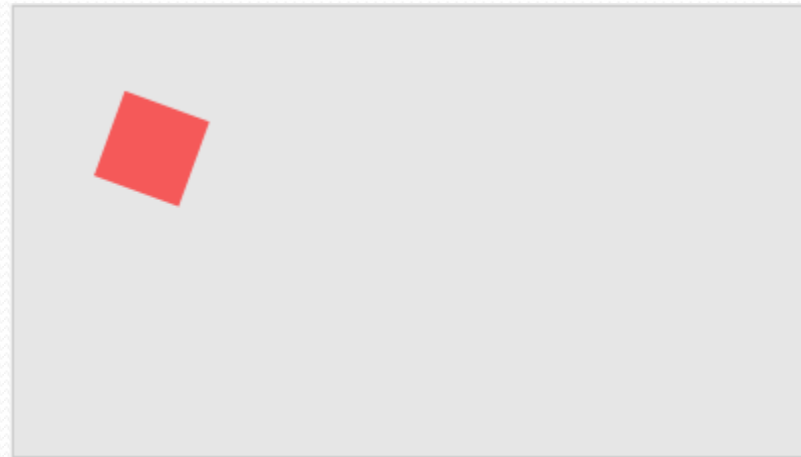
3 그리기

- 이제 캔버스에 컴포넌트를 그릴 준비가 완료됨
- 회전 된 캔버스의 좌표 0,0 위치에
그리고자 하는 요소의 중심
위치시키고 해당 요소를 그
- `ctx.fillRect (x, y, w, h);`
 - X: 너비 / -2
 - Y: 높이 / -2
 - W: 너비
 - H: 높이



4 복원

- 회전 작업이 끝나면 `restore` 메소드를 사용하여 컨텍스트 개체(요소)들을 처음 저장되었던 위치로 복원
 - `ctx.restore ();`
 - 고로 해당 요소 만 회전 된 것이 됨

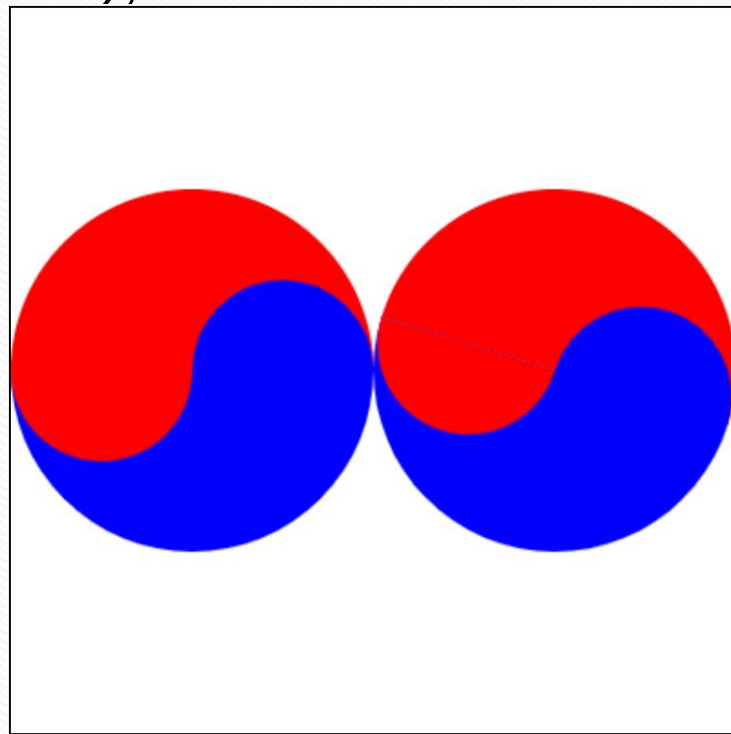


태극 함수

```
1. function taegeuk(x, y, r) {  
2.   var r2 = r / 2;  
3.  
4.   ctx.beginPath();  
5.   ctx.fillStyle = "red";  
6.   ctx.arc(x, y, r, 0, Math.PI, true);  
7.   ctx.fill();  
8.   ctx.beginPath();  
9.   ctx.fillStyle = "blue";  
10.  ctx.arc(x, y, r, 0, Math.PI);  
11.  ctx.arc(x + r2, y, r2, 0, Math.PI, true);  
12.  ctx.fill();  
13.  ctx.beginPath();  
14.  ctx.fillStyle = "red";  
15.  ctx.arc(x - r2, y, r2, 0, Math.PI);  
16.  ctx.fill();  
17. }
```

회전 적용

```
1. function init() {  
2.     vcanvas = document.getElementById("myCanvas");  
3.     ctx = vcanvas.getContext("2d");  
4.  
5.     taegeuk(100, 200, 100);  
6.  
7.     // 회전  
8.     ctx.save();  
9.     ctx.translate(300, 200);  
10.    ctx.rotate(0.3);  
11.    taegeuk(0, 0, 100);  
12.    ctx.restore();  
13. }
```



랜덤 컬러 생성

Math 객체의 주요 메소드

메소드	설명
abs(x)	X의 절대값 리턴
sin(x)	싸인 x 값 리턴
exp(x)	e^x 값 리턴
pow(x, y)	x^y 값 리턴
random()	0 ~ 1 보다 작은 임의의 실수 리턴 (0 ~ 0.999999999...)
floor(x)	X보다 작거나 같은 수 중 가장 큰 정수 리턴. <code>Math.floor(3.2) = 3</code>
round(x)	X를 반올림한 정수 리턴. <code>Math.round(3.7)=4</code> , <code>Math.round(3.2)=3</code>
sqrt(x)	X의 제곱근 리턴

방법 1

```
1. function makeColor1() {  
2.     var x, y, z;  
3.  
4.     x = Math.floor(Math.random() * 256);  
5.     y = Math.round(Math.random() * 255);  
6.     z = Math.floor(Math.random() * 255) + 1;  
7.  
8.     return "rgb(" + x + "," + y + "," + z + ")";  
9. }
```

방법 2

1. `function makeColor2() {`
2. `return "#" + (parseInt(Math.random() * 0xffffffff,`
 `10)).toString(16);`
3. `}`

방법 3

```
1. function makeColor3() {  
2.     var letters = '0123456789ABCDEF', color = '#', i;  
3.  
4.     for (i = 0; i < 6; i += 1) {  
5.         color += letters[Math.floor(Math.random() *  
6.             16)];  
7.     }  
8.     return color;  
9. }
```

setInterval()

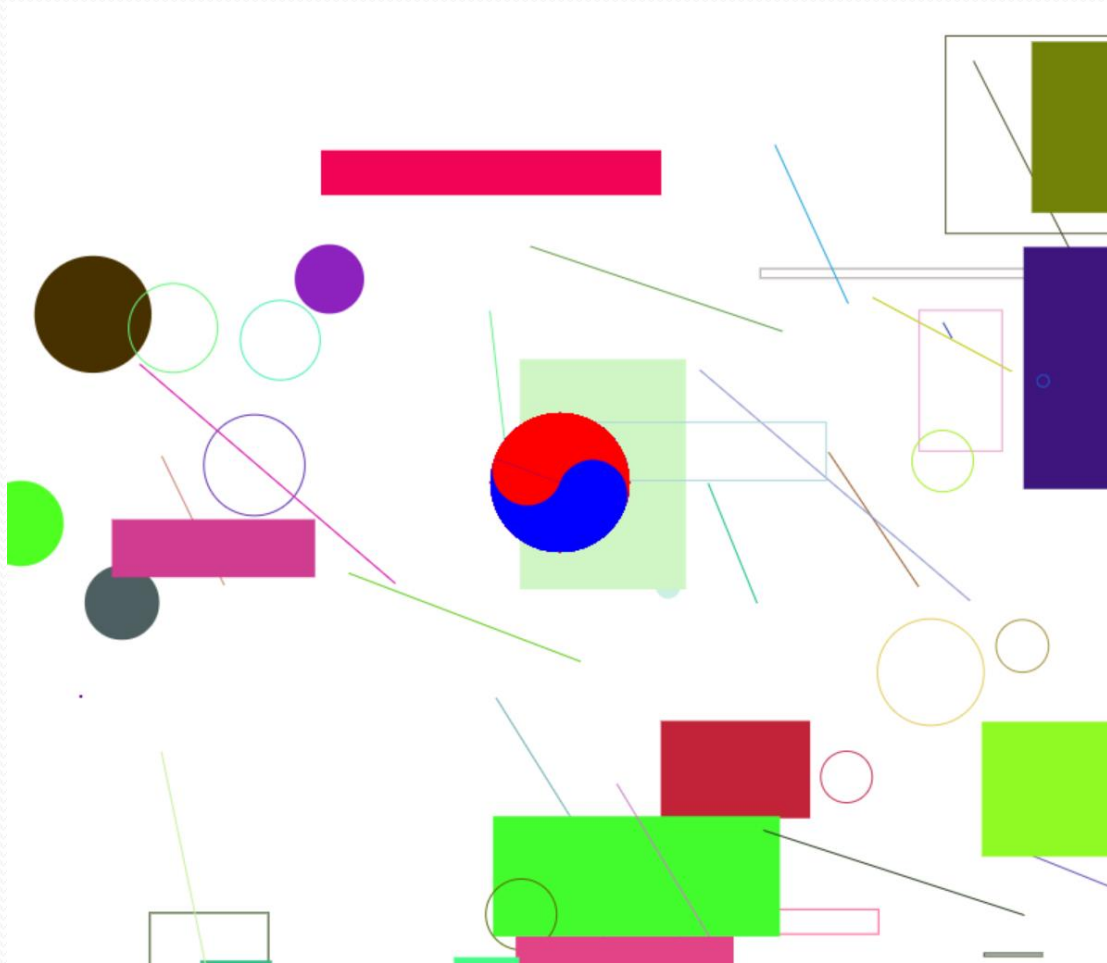
일정 시간마다 반복 실행하는 함수

setInterval()

```
setInterval(drawLine, 1000);
```

- drawLine : 호출할 함수 명
- 1000 : 인터벌 타임 즉 1초마다

미션 4 : 자신만의 작품



도전 학습

희망자에 한함

도전1 : 태극 무한 회전 Rotate Loop

```
1. var vcanvas, ctx;
2. var x, y, rd = 0;
3.
4. function taegeuk(x, y, r) { .... }
5.
6. function gameLoop() {
7.     var v = 0.05;
8.
9.     ctx.save();
10.    ctx.translate(x, y);
11.    ctx.rotate(rd);
12.    taegeuk(0, 0, 100);
13.    ctx.restore();

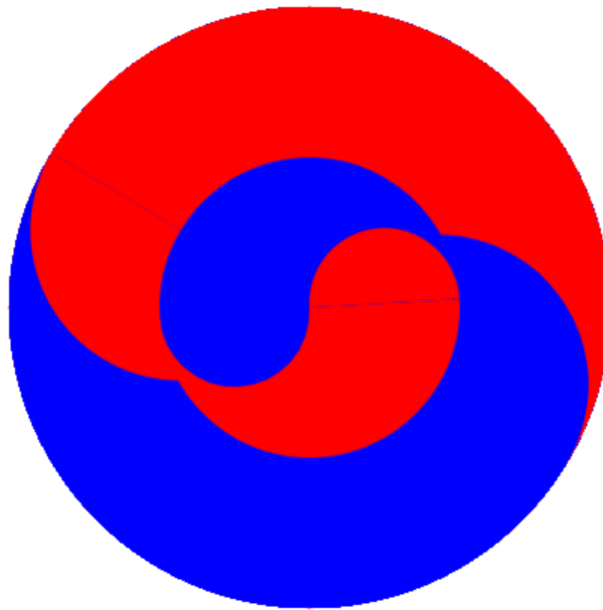
14.    if (rd > Math.PI * 2) {
15.        rd = 0;
16.    } else {
17.        rd += v;
18.    }
19. }
```

```
20. function init() {
21.     vcanvas =
        document.getElementById("myCanvas");
22.     ctx = vcanvas.getContext("2d");

23.     x = vcanvas.width / 2;
24.     y = vcanvas.height / 2;
25.
26.     setInterval(gameLoop, 13);
27. }
```


도전2 : 태극 이중 무한 회전

- 큰 태극은 시계 방향으로 천천히 회전
- 작은 태극은 시계 반대 방향으로 빠르게 회전
- 두 태극 모두 캔버스 정 중앙에 위치



도전3 : 바람개비 회전

