
***** Mobile Simulation 중간고사. 2017. 10. 17(화) *****

---

1. 브라우저에 의해 제공되는 BOM 객체들은 계층 관계로 이루어진다.
계층의 최상위에 있는 객체는 무엇인가?

A. document B. location
C. window D. navigator

2. 다음 중 브라우저에 어떤 플러그인이 설치되었는지 알아내는데 사용되는 객체는?

A. location B. navigator
C. browser D. window

3. 캔버스 객체가 canvas이고 컨텍스트가 ctx 일 때, 캔버스를 깨끗이 지우는
메소드는?

A. ctx.clearBackground(0, 0, canvas.width, canvas.height);
B. ctx.clearText(0, 0, canvas.width, canvas.height);
C. ctx.clearPath(0, 0, canvas.width, canvas.height);
D. ctx.clearRect(0, 0, canvas.width, canvas.height);

4. 다음 중 HTML5 캔버스 기능에 대한 설명 중 틀린 것은?

A. 캔버스는 HTML5 표준이다.
B. 캔버스는 마우스 이벤트를 처리할 수 없다.
C. 캔버스는 2차원 그래픽을 지원하고, 3차원 그래픽도 지원한다.
D. 하나의 웹페이지에는 여러 개의 켄버스를 둘 수 있다.

5. 컨텍스트 객체의 경로와 메소드에 대해 잘못 설명한 것은?

A. beginPath()는 캔버스에 그릴 도형의 경로를 새로 시작한다.
B. 직선, 원호, 사각형 등의 도형은 경로에 먼저 삽입된 후 캔버스에 한 번에 그려진다.
C. stroke()는 경로에 있는 도형을 모두 캔버스에 그리고, 그린 도형은 경로에서 삭제한다.
D. beginPath()는 이전에 만들어진 경로를 모두 지운다.

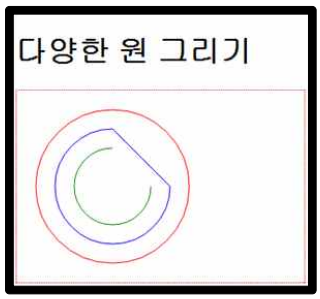
[6-7]. 이미지 위에 마우스를 올린 상태로 5초가 지나면 네이버에 연결하며, 5초 전에 이미지를 벗어나면 타이머를 해제하는 코드.
Complete the blanks in script.

```
<!DOCTYPE html>
<html>
<head>
<title>setTimeout()으로 웹 페이지 자동 연결</title>
</head>
<body>
<h3>이미지에 마우스를 올리고 5초간 그대로 있을 때 사이트로 이동합니다</h3>
<hr>
<img id="img" src="media/naver.gif"
                onmouseover="startTimer(5000)"
                onmouseout="cancelTimer()">
<script>
        var timerID=null;
        function startTimer(time) {
                // 타이머 시작
                timerID = setTimeout("load('http://www.naver.com')", time);

                // 이미지에 마우스 올리면 나타내는 툴팁 메시지
                document.getElementById("img").title = "타이머 작동 시작...";
        }

        function cancelTimer() {

                if(timerID != ___[6]__null)
                        clearTimeout(timerID); // 타이머 중단
        }

        function load(url) {
                window.___[7]__location = url; // 현재 윈도우에  url 사이트 로드
        }

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

(Hint to [7]: window, history, location, screen, navigator, address, homepage)

[8-9]. Answer two questions about the code snippet.

```
<body>
    <canvas id="myCanvas" width="300" height="200"></canvas>
     <script>
          var canvas = document.getElementById('myCanvas');
          var context = canvas.getContext('2d');
          // 1st path
          context.beginPath();
          context.arc(100, 100, 80, 0, 2.0 * Math.PI, false);
          context.strokeStyle = "red";
          context.stroke();
          // 2nd path
          context.beginPath();
          context.arc(100, 100, 60, 0, ___[8]___ * Math.PI, false);
          context.___[9]___;
          context.strokeStyle = "blue";
          context.stroke();
          // 3rd path
          context.beginPath();
          context.arc(100, 100, 40, 0, ___[8]___ * Math.PI, false);
          context.strokeStyle = "green";
          context.stroke();

    </script>
</body>
```



8. What is the correct value to draw inner two paths?

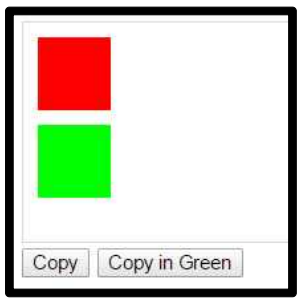
A. 0.5 **B. 1.0**
C. 1.5 **D. 2.0**

9. What is the proper method to draw the second path?

A. createPath() **B. closePath()**
C. openPath() **D. strokePath()**

10-11. The below code snippet manipulates pixels of an square image. The function **copyGreen() duplicates the upper red square in green.**

```
<body>
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.fillStyle = "red";
ctx.fillRect(10, 10, 50, 50);

function copy() {
    var imgData = ctx.getImageData(10, 10, 50, 50);
    ctx.putImageData(imgData, 10, 70);
}

function copyGreen() {
    var imgData = ctx.getImageData(10, 10, 50, 50);
    for (i = 0; i < imgData.data.length; ___[10]___) {
      imgData.data[i+0] = A; // [11]
      imgData.data[i+1] = B; // [11]
      imgData.data[i+2] = C; // [11]
      imgData.data[i+3] = D; // [11]
    }
    ctx.putImageData(imgData, 10, 70);
}
</script>
<br>
<button onclick="copy()">Copy</button>
<button onclick="copyGreen()">Copy in Green</button>
</body>
```



10. What is the correct code to be inserted here?

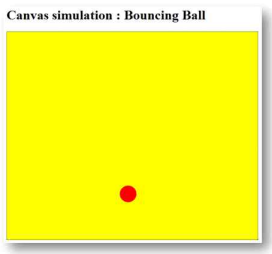
A. i++ **B.** i+=2
C. i+=4 **D.** i+=8

11. What are the proper values in [A, B, C, D] to set a green square?

A. [255,0,255,0] **B.** [0,0,255,255]
C. [255,255,0,0] **D.** [0,255,0,255]

12-14. The below code snippet simulates the motion of a ball bouncing from a box.

```
<script>
var canvas = null; var context = null;
var dx = 10; // velocity in the x-direction
var dy = 10; // velocity in the y-direction
var x = 100; var y = 100;
var r = 20;
var x_max = 0; var y_max = 0;

function init() {
        canvas = document.getElementById('myCanvas');
        context = canvas.getContext("2d");
        x_max = context.canvas.width;
        y_max = context.canvas.height;
        blank();
        context.beginPath();
        context.fillStyle = "red";
        context.arc(x, y, r, 0, Math.PI * 2, true);
        context.closePath();
        context.fill();
       // start animation
        setInterval(___[12]___, 50);
}
function blank() {  // clear screen in yellow
        context.fillStyle = "yellow";
        context.fillRect(___[13]___);  //
}
function draw() {
        blank();
        if (x < (0 + r) || x > (x_max ___[14.a]___))
                dx = -dx;
        if (y < (0 + r) || y > (y_max ___[14.b]___))
                dy = -dy;
        x += dx;
        y += dy;

        context.beginPath();
        context.fillStyle = "red";
        context.arc(x, y, r, 0, Math.PI * 2, true);
        context.closePath();
        context.fill();
}
</script>
```



12. What is the correct code to be inserted here? --- [draw]

13. What are the correct parameters to clear screen in yellow?

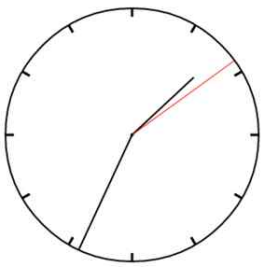
A. 0, 0, x, y
B. 0, 0, x_max, y_max
C. 0, 0, dx, dy
D. 0, 0, x+dx, y+dy

14. What are the correct codes to be inserted in [14.a] and [14.b] for the ball to bounce well?

A. -r + dx , -r + dy
B. -r - dx , - r - dy
C. +r + dx , +r + dy
D. r - dx , r - dy

15-16. Draw hour hand, minute hand, and second hand respectively. What are the proper values to synchronize all hands in a clock?

```
<script>
var date = new Date();

// Get current hour, minutes, seconds
var hours = date.getHours();
var minutes = date.getMinutes();
var seconds = date.getSeconds();

// Draw hours
ctx.strokeStyle = "black";
ctx.lineWidth = 3;
drawHand(clockWidth / 3, hours*30 + minutes*(30/60) + seconds*(30/3600));

// Draw minutes
ctx.strokeStyle = "yellow";
ctx.lineWidth = 6;
drawHand(clockWidth / 2, minutes*6 + seconds*(___[15]___); // 6/60

// Draw seconds
ctx.strokeStyle = "red";
ctx.lineWidth = 1;
drawHand(clockWidth / 2, seconds*(___[16]___));   // 6

function drawHand(length, angle) {
        ctx.save();
        ctx.beginPath();
        ctx.translate(centerX, centerY);
        ctx.rotate(-180 * Math.PI / 180);
        // Correct for top left origin
        ctx.rotate(angle * Math.PI / 180);
        ctx.moveTo(0, 0);
        ctx.lineTo(0, length);
        ctx.stroke();
        ctx.closePath();
        ctx.restore();
}
</script>
```



17-18. What are the proper codes to make X-mas card using 200 snowflakes?

```
<script>
// global variables
var canvas = null;
var context = null;
var bufferCanvas = null;
var bufferCanvasCtx = null;
var flakeArray = [];
var flakeTimer = null;
var maxFlakes = 200;
// prepare image
var imgA = new Image();
imgA.src = "media/bridge.png";

function init() {
        canvas = document.getElementById('myCanvas');
        context = canvas.getContext("2d");

        bufferCanvas = document.createElement("canvas");
        bufferCanvasCtx = bufferCanvas.getContext("2d");
        bufferCanvasCtx.canvas.width = context.canvas.width;
        bufferCanvasCtx.canvas.height = context.canvas.height;

        flakeTimer = setInterval(addFlake, 200);

        Draw();

        setInterval(animate, 30);
}

// Properties of snowflakes
function Flake() {
    this.x = Math.round(Math.random() * context.canvas.width);
    this.y = -10;
    this.drift = Math.random();
    this.speed = Math.round(Math.random() * 5) + 1;
    this.width = (Math.random() * 3) + 2 ;
    this.height = this.width;
}
// make snowflakes
function addFlake() {
        flakeArray[flakeArray.length] = new Flake();
        if (flakeArray.length == _____[17]_____maxFlakes)
                clearInterval(flakeTimer);
}
```

```
function blank() {    // Clear buffer canvas
        // draw image on bufferCanvas
        bufferCanvasCtx.drawImage(imgA, 0, 0, bufferCanvasCtx.canvas.width, bufferCanvasCtx.canvas.height);
        // draw text on bufferCanvas
        var msg = "Merry Chrismas";
        bufferCanvasCtx.font = "50px 'Nanum Gothic'";
        bufferCanvasCtx.fillText(msg, 145, 150);

}
function animate() {    // animate snowflakes
        Update();
        Draw();
}
function Update() {     // set position and speed of snowflakes
        for (var i = 0; i < flakeArray.length; i++) {
                if (flakeArray[i].y < context.canvas.height) {
                        flakeArray[i].y += flakeArray[i].speed;
                        if (flakeArray[i].y > context.canvas.height)
                                flakeArray[i].y = -5;
                                flakeArray[i].x += flakeArray[i].drift;
                                if (flakeArray[i].x > context.canvas.width)
                                        flakeArray[i].x = 0;
                }
        }
}
function Draw() {
        context.save();
        // create a clipping region on buffer canvas
        bufferCanvasCtx.beginPath();
        bufferCanvasCtx.fillStyle="black";
        bufferCanvasCtx.fillRect(0,0,bufferCanvas.width,bufferCanvas.height);
        bufferCanvasCtx.fillStyle="white";
        bufferCanvasCtx.arc(bufferCanvas.width/2, bufferCanvas.height/2,bufferCanvas.height/2,0,2*Math.PI);
        bufferCanvasCtx.clip();
        blank();
        // draw all snowflakes on buffer canvas
        for (var i = 0; i < flakeArray.length; i++) {
                bufferCanvasCtx.beginPath();
                bufferCanvasCtx.fillStyle = "skypink";
                bufferCanvasCtx.arc(flakeArray[i].x, flakeArray[i].y, flakeArray[i].width, 0, 2 * Math.PI);
                bufferCanvasCtx.fill();
        }
        // Double buffering
        context.drawImage(_____[18]_____bufferCanvas, 0, 0, bufferCanvas.width, bufferCanvas.height);
        // copy the entire rendered image from the buffer canvas to the visible one
        context.restore();
}
</script>
```

19-20. What are the proper codes to make cannonball game work?

```
<script>
var canvas = document.getElementById("myCanvas");
var context = canvas.getContext("2d");
/* 변수 초기화 */
var velocity;            // 사용자가 입력한 공의 초기속도
var angle;                        // 사용자가 입력한 공의 초기각도
var net;
var ballV;               // 공의 현재 속도 (2차원 속도)
var ballVx;              // 공의 현재 x방향 속도
var ballVy;              // 공의 현재 y방향 속도
var ballX = 10;          // 공의 현재 x방향 위치
var ballY = 260;         // 공의 현재 y방향 위치 */
var ballRadius = 10;   // 공의 반지름
var score = 0;           // 점수 초기화
var netWidth = 30;
var net_h = 150;
var number = 0;

var image = new Image(); // 이미지 객체 생성
image.src = "media/lawn.png"; // 이미지 파일 이름 설정
var netimage = new Image();
netimage.src = "media/net.png";
var timer;  // 타이머 객체 변수 to control cannonball

/* 공을 화면에 그린다. */
function drawBall() {
            context.beginPath();
            context.arc(ballX, ballY, ballRadius, 0, 2.0 * Math.PI, true);
            context.fillStyle = "red";
            context.fill();
}
function random(n1,n2) {
          return Math.floor(Math.random() * (n2-n1+n1)+n1);
}
/* 배경을 화면에 그린다. */
function drawBackground() {
            context.drawImage(image, 0, 270);
            context.drawImage(netimage, 450, random(10,50), 30,net_h);
}
/* 초기화를 담당하는 함수 */
function init() {
            ballX = 10;
            ballY = 260;
            ballRadius = 10;
            draw();
}
```

Canvas simulation : CannonBall

```
/* 전체 화면을 그리는 함수 */
function draw() {
        context.clearRect(0, 0, 500, 300);  /* 화면을 지운다. */
        drawBall();
        drawBackground();
        var msg = "Score : "+ score;
        context.font = "30px 'Gothic'";
        context.fillText(msg,15,50);
        gameOver();
        _____[19]_____gameClear();
}
/* 사용자가 발사 버튼을 누르면 호출된다. */
function fire() {
        init();
        velocity = Number(document.getElementById("velocity").value);
        angle = Number(document.getElementById("angle").value);
        var angleR = angle * Math.PI / 180;
        ballVx = velocity * Math.cos(angleR);
        ballVy = -velocity * Math.sin(angleR);          // negative y-direction
        net_h = Number(document.getElementById("netHeight").value);
        draw();
        number++;
        timer = setInterval(calculate, 100);  // next position & velocity of cannonball
        document.getElementById("number").innerHTML = "횟수 = " + number;
}
/* 공의 현재 속도와 위치를 업데이트한다. */
function calculate() {
        ballVy = ballVy + 1.98;  // y-방향 속도 계산 (Vy = Vy + g*dt)
        ballX = ballX + ballVx;
        ballY = ballY + ballVy;

        /* Hit test : 공이 목표물에 맞았으면 */
        if ((ballX >= 450) && (ballX <= 480) && (ballY >= 60) && (ballY <= 60+net_h)) {
                score++;
                document.getElementById("score").innerHTML = "점수 = " + score;
                clearInterval(timer);
                init();
        }
        /* 공이 경계를 벗어났으면 */
        if (ballY >= 300 || ballY < 0) {
                clearInterval(timer);
                init();
        }
        draw();
}
```

```
function reStart(){
 //context.clearRect(0,0,500,300);
            if(timer != null)
                        clearInterval(timer);

            ballX = 10;
            ballY = 260;
            ballRadius = 10;
            number = 0;
            score = 0;
            document.getElementById("number").innerHTML = "횟수 = " + number;
            document.getElementById("score").innerHTML = "점수 = " + score;
            document.getElementById("velocity").value = 30;
            document.getElementById("angle").value = 45;
            document.getElementById("netHeight").value = 80;
            ____[20]____draw();
}

function gameOver(){
          if((score < 6)&&(number == 10)){
                    clearInterval(timer);
                    var k = "Game over";
                    var k1 = "Press new start.";
                    context.font = "30px 'Gothic'";
                    context.fillStyle = "red";
                    context.fillText(k,145,150);
                    context.fillText(k1,145,180);
                }
}

function gameClear(){
          if(score == 10){
                 clearInterval(timer);
                 var k = "Game Clear!"
                var k1 = "Press new start.";
                 context.font = "30px 'Gothic'";
                 context.fillStyle = "red";
                 context.fillText(k,145,150);
                context.fillText(k1,145,180);
                }
}
</script>
```