

Functions.js

```
// JavaScript Document
function rotatePattern(patternID) {
    var svg = document.getElementById(patternID);
    var svgDoc = svg.contentDocument;
    var wrapper = svgDoc.getElementById("patternWholeWrapper");
    var currentRotation = 0;
    var currentTransform = wrapper.getAttribute("transform");
    var boundBox = wrapper.getBBox();
    var boundBoxCenter = (boundBox.width / 2) + " " + (boundBox.height / 2);
    //Gets current rotation applied to the pattern wrapper.
    if (currentTransform != null) {
        //changes current rotation to the first argument of the transform
        attribute (rotation angle)
        currentRotation += parseInt(currentTransform.substring(7, 10), 10);
    }
    if (currentRotation > 180) {
        wrapper.setAttribute("transform", "rotate(0 " + boundBoxCenter +
        ")");
    } else {
        currentRotation += 90;
        wrapper.setAttribute("transform", "rotate(" +
        currentRotation.toString() + " " + boundBoxCenter + ")");
    }
}

function invertColours() {
    var svg = document.getElementById("bubbles-svg");
    var svgDoc = svg.contentDocument;
    var circPattern = svgDoc.getElementById("waves");
    var outerCircle = svgDoc.getElementById("outerCirc");
    var innerCircle = svgDoc.getElementById("innerCirc");
    var currentFill = circPattern.getAttribute("fill");
    if (currentFill == "black") {
        circPattern.setAttribute("fill", "white");
        outerCircle.setAttribute("fill", "black");
        innerCircle.setAttribute("stroke", "white");
    } else {
        circPattern.setAttribute("fill", "black");
        outerCircle.setAttribute("fill", "none");
        innerCircle.setAttribute("stroke", "black");
    }
}

function spaceOut() {
    var svg = document.getElementById("cubes-svg");
    var svgDoc = svg.contentDocument;
    var linePattern = svgDoc.getElementById("lines");
    var currentHeight = parseInt(linePattern.getAttribute("height"));
    //increases spaces between lines by 2
    linePattern.setAttribute("height", (currentHeight + 1));
}
```

```

function spaceIn() {
    var svg = document.getElementById("cubes-svg");
    var svgDoc = svg.contentDocument;
    var linePattern = svgDoc.getElementById("lines");
    var currentHeight = parseInt(linePattern.getAttribute("height"));
    //increases spaces between lines by 2
    if (currentHeight > 20) {
        linePattern.setAttribute("height", (currentHeight - 1));
    }
}

function enlargePattern(patternID) {
    //Defines pattern to be enlarged
    var elementToEnlarge = document.getElementById(patternID);
    //All divs in document
    var allElements = document.getElementsByTagName("div");
    if (elementToEnlarge.style.position != "absolute") {
        //iterates through divs
        for (i = 0; i < allElements.length; i++) {
            //If the div isn't the pattern's parent, hide it.
            if (allElements[i] != elementToEnlarge.parentElement) {
                allElements[i].style.display = "none";
            }
        }
        //Fills the page with the enlarged element
        elementToEnlarge.style.position = "absolute";
        elementToEnlarge.style.height = "95%";
        elementToEnlarge.style.margin = "0 -16%";
    } else { //Acts as a toggle mechanism, reverses the code in the if above.
        //If the div isn't the pattern's parent, show it.
        for (i = 0; i < allElements.length; i++) {
            //If the div isn't the pattern's parent, hide it.
            allElements[i].style.display = "block";
        }
        //Make the enlarged element normal again
        elementToEnlarge.style.position = "static";
        elementToEnlarge.style.height = "45%";
        elementToEnlarge.style.margin = "0 auto";
    }
}

```

PatternBubbleGlobes.svg

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<svg version="1.1" baseProfile="full" viewBox="0 0 1000 1000"
    xmlns="http://www.w3.org/2000/svg">
    <defs>
        <pattern id="hemi" x="0" y="0" width="50" height="50"
            patternUnits="userSpaceOnUse">
            <path id="waves" d="M0, 25
                C0, 0
                25, 0
                25, 25
                S50, 50

```

```

                    50, 25"
                    stroke="black" fill="black" stroke-width="2" />
                <line x1="0" y1="25" x2="50" y2="25" stroke="black"
stroke-width="2" />
            </pattern>
            <pattern id="circles" x="0" y="0" width="200" height="200"
patternUnits="userSpaceOnUse">
                <g stroke="black" stroke-width="2">
                    <circle id="outerCirc" cx="100" cy="100" r="95" stroke="black"
fill="none" />
                    <circle id="innerCirc" cx="100" cy="100" r="80" stroke="black"
fill="url(#hemi)" />
                    <line x1="40" y1="0" x2="0" y2="40" />
                    <line x1="160" y1="0" x2="200" y2="40" />
                    <line x1="0" y1="160" x2="40" y2="200" />
                    <line x1="200" y1="160" x2="160" y2="200" />
                </g>
            </pattern>
        </defs>
        <g id="patternWholeWrapper">
            <rect x="0" y="0" width="100%" height="100%" stroke="black"
fill="url(#circles)" />
        </g>
    </svg>

```

PatternCubes.svg

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<svg version="1.1" baseProfile="full" viewBox="0 0 1000 1000"
xmlns="http://www.w3.org/2000/svg">
    <defs stroke="black">
        <pattern id="lines" x="0" y="0" width="20" height="20" fill="white"
patternUnits="userSpaceOnUse">
            <line x1="0" x2="20" y1="20" y2="20" stroke-width="2" />
        </pattern>
        <pattern id="boxes" x="0" y="0" width="200" height="200"
patternUnits="userSpaceOnUse" patternTransform="rotate(45)">
            <g>
                <line x1="175" y1="175" x2="200" y2="200" />
                <line x1="0" y1="175" x2="25" y2="200" />
                <line x1="175" y1="0" x2="200" y2="25" />
                <rect x="0" y="0" height="175" width="175" fill="url(#lines)"
/>
            </g>
        </pattern>
    </defs>
    <g id="patternWholeWrapper">
        <rect x="0" y="0" width="100%" height="100%" stroke="black"
fill="url(#boxes)" />
    </g>

```

```
</svg>
```

Styling.css

```
body {
    margin-top: 0;
    margin-bottom: 0;
}
div {
    height: 33%;
    width: 33%;
    display: block;
    float: center;
    border: 1px solid white;
    margin: 0 auto;
    overflow: hidden;
}
```

PatternCreator.html

```
<!DOCTYPE HTML>
<html>

<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <link rel="stylesheet" type="text/css" href="styling.css">
    <title>
        SVG/JS/DOM Pattern Creator
    </title>
</head>

<body>
    <script type="text/javascript" src="functions.js"></script>
    <div id="patternAContainer">
        <button onclick="rotatePattern('cubes-svg') ">Rotate 90°</button>
        <button onclick="spaceIn() ">Decrease Spaces</button>
        <button onclick="spaceOut() ">Increase Spaces</button>
        <button
onclick="enlargePattern('cubes-svg') ">Enlarge/Shrink</button>
        <br>
        <object id="cubes-svg" type="image/svg+xml"
data="PatternCubes.svg"></object>
    </div>
    <div id="patternBContainer">
        <button onclick="rotatePattern('bubbles-svg') ">Rotate 90°</button>
        <button onclick="invertColours() ">Invert Colours</button>
        <button
onclick="enlargePattern('bubbles-svg') ">Enlarge/Shrink</button>
        <br>
```

```
        <object id="bubbles-svg" type="image/svg+xml"
data="PatternBubbleGlobes.svg"></object>
    </div>
</body>

</html>
```