1. Intro

JavaScript

INDEX.HTML

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <script src="app.js"></script>
</head>
<body onload="greeting()">
    Content of the page
</body>
</html>
```

APP.JS

```javascript
function greeting() {
    document.write('My new greeting!');
}
```

Styles

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>

    <link rel="stylesheet" href="style.css">
</head>
<body>
    <p>
        Paragraph 1
        <span class="last">My span</span>
    </p>
    <p id="par2">Paragraph 2</p>
    <p class="last">Paragraph 3</p>

    <span class="last">My span</span>
</body>
</html>
```

STYLE.CSS

```
#par2,
p.last {
    color: green;
}

p {
    color: blue;
    font-size: 50px;
    font-family: Arial, "Times New Roman";
}
```

2. JavaScript OO

INDEX.HTML

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>JavaScript OO</title>
</head>
<body>

    <script src="app.js"></script>
</body>
</html>
```

APP.JS

```
// Base

var Base = function(value) {
    this.prop1 = value;
}
Base.prototype = {
    func1: function (value) {
        this.prop1 += value;
    }
};

// Derived

var Ext = function() {
    Base.call(this, 10);
    this.prop2 = "two";
}
Ext.prototype = new Base();

Ext.prototype.func2 = function(value) {
    this.prop2 += value;
}


var b = new Base(1);
b.func1(2);

document.writeln(b.prop1);
document.writeln();


var c = new Ext();
c.func1(1);
c.func2("...");

document.writeln(c.prop1);
document.writeln(c.prop2);
```

3. Events

DYNAMIC.HTML

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
</head>
<body>
    <button onclick="add(event)">Add</button>
    <div id="content"></div>

    <script>
        // Adding elements

        var counter = 1;
        var content = document.getElementById('content');

        function add(e) {
            var p = document.createElement('p');
            p.innerText = 'Dynamic paragraph ' + counter++;
            content.appendChild(p);
        }

        // Hover handlers for paragraphs

        document.body.addEventListener('click', remove, true);

        function remove(e) {
            if (e.target.tagName === 'P') {
                e.target.parentNode.removeChild(e.target);
            }
        }

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

8. Canvas

CANVAS-VS-SVG.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
</head>
<body>
    <canvas id="myCanvas" height="100px" width="300px"
style="background-color: lightgray">
        Canvas not supported
    </canvas>

    <script type="text/javascript">
        var canvas = document.getElementById('myCanvas');
        var context = canvas.getContext('2d');

        context.fillStyle = 'orange';
        context.strokeStyle = 'blue';

        context.rect(10, 10, 80, 80);
        context.fill();

        context.beginPath();
        context.moveTo(40, 40);
        context.lineTo(200, 80);
        context.stroke();
    </script>

    <svg id="mySVG"
        xmlns="http://www.w3.org/2000/svg"
        style="height: 100px; width: 300px; background-color:
lightgray">
    </svg>

    <script>
        var svgNS = 'http://www.w3.org/2000/svg';
        var svg = document.getElementById('mySVG');
        var r = document.createElementNS(svgNS, 'rect');
        r.setAttributeNS(null, 'x', 10);
        r.setAttributeNS(null, 'y', 10);
        r.setAttributeNS(null, 'width', 80);
        r.setAttributeNS(null, 'height', 80);
        r.setAttributeNS(null, 'fill', 'orange');
        svg.appendChild(r);

        var l = document.createElementNS(svgNS, 'path');
        l.setAttributeNS(null, 'd', 'M 40 40 L 200 80');
        l.setAttributeNS(null, 'stroke', 'blue');
        svg.appendChild(l);
    </script>

    <svg xmlns="http://www.w3.org/2000/svg"
        style="height: 100px; width: 300px; background-color:
lightgray">
        <rect x="10" y="10" width="80" height="80" fill="orange"/>
        <path d="M 40 40 L 200 80" stroke="blue"/>
    </svg>
</body>
</html>
```

10. Video

VIDEO.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
</head>
<body>
    <script type="text/javascript">
       var video = document.createElement('video');

        video.src = 'SwanseaBay.mp4';
        video.autoplay = true;
        video.controls = true;
        video.loop = true;

        video.playbackRate = 3.0;
        video.volume = 0.5;

        document.body.appendChild(video);

        // Play/Pause Button

        var playButton = document.createElement('button');
        playButton.innerText = 'Play / Pause';
        playButton.onclick = function() {
            var video = document.querySelector('video');
            if (video.paused) {
                video.play();
            } else {
                video.pause();
            }
        }

        document.body.appendChild(playButton);
    </script>
</body>
</html>
```

13. Files and Data

Drag & Drop

DRAGDROP.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
    <script>
        function dragStart(e) {
            e.dataTransfer.setData('text/plain',
e.target.innerText);
        }

        function dragOver(e) {
            e.preventDefault();
        }

        function drop(e) {
            var data = e.dataTransfer.getData('text/plain');
            e.target.innerText = data;
        }
    </script>
</head>
<body>
    <div id="part1" draggable="true"
ondragstart="dragStart(event)">Part 1</div>
    <div id="part2" ondragover="dragOver(event)"
ondrop="drop(event)">Part 2</div>
</body>
</html>
```

Files

FILES.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
    <script>
        function newFile(elem) {
            var list = document.getElementById('list');
            list.innerHTML = '';

            for (var i = 0; i < elem.files.length; i++) {
                var file = elem.files[i];
                var li = document.createElement('li');
                li.innerText = file.name + ' (' + file.type + ') ' + file.size;
                list.appendChild(li);

                if (i === 0) {
                    showContent(file);
                }
            }
        }

        function showContent(file) {
            var fr = new FileReader();
            fr.onload = function (e) {
                var content = document.getElementById('content');
                content.innerText = e.target.result;
            }
            fr.readAsText(file);
        }
    </script>
</head>
<body>
    <input type="file" onchange="newFile(this)" multiple />
    <ul id="list"></ul>
    <div id="content" style="background-color: aqua"></div>
</body>
</html>
```

17. Web Workers

NOWEBWORKER.HTML

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>

    <script src="lib.js"></script>

    <script>
        function setResult(value) {
            var r = document.getElementById('result');
            r.innerText = value;
        }

        function newText(value) {
            sleep(1000);
            setResult(value.toUpperCase());
        }
    </script>
</head>
<body>
    <input type="text" onkeyup="newText(this.value)">
    <div id="result"></div>
</body>
</html>
```

LIB.JS

```javascript
function sleep(delay) {
    var start = new Date().getTime();
    while (new Date().getTime() < start + delay);
}
```

WEB WORKER.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>

    <script>
        function setResult(value) {
            var r = document.getElementById('result');
            r.innerText = value;
        }

        function newText(value) {
            if (typeof (Worker) !== 'undefined') {
                var w = new Worker('worker.js');
                w.onmessage = function (msg) {
                    setResult(msg.data);
                }
                w.postMessage(value);
            } else {
                alert('Worker not defined!');
            }
       }
    </script>
</head>
<body>
    <input type="text" onkeyup="newText(this.value)">
    <div id="result"></div>
</body>
</html>
```

WORKER.JS

```
importScripts('lib.js');

onmessage = function (msg) {
    sleep(1000);
    while (true) {
        postMessage(msg.data.toUpperCase());
    }
}
```