JavaScript DOM coding Lessons

Simple HTML page with innerHTML update

```
<!DOCTYPE html>
<html>
<head>
 <title>JavaScript DOM</title>
</head>
<body>
  <h1>Laurence Svekis</h1>
  <div class="output"></div>
  <div id="one"></div>
  <script src="/js/app1.js"></script>
</body>
</html>
```

```
console.dir(document);
console.dir(document.body);
console.log(document.URL);
console.log(document.title);
console.log(document.body.innerHTML);
document.body.innerHTML = '<h1>Laurence Svekis New</h1>';
```

Page Element Selection

```
<!DOCTYPE html>
<html>
<head>
```

```
 <title>JavaScript DOM</title>
</head>
<body>
  <h1>Laurence Svekis</h1>
  <div class="output"></div>
  <div id="one"></div>    <div class="output"></div>    <div
class="output"></div>
  <script src="/js/app1.js"></script>
</body>
</html>
```

```
//document.write('one');
//document.write('two');
const oneElement = document.getElementById('one');
oneElement.innerHTML = '<h1>Test</h1>';
oneElement.innerHTML = '<h1>New</h1>';

const output1 = document.getElementsByClassName('output');
console.log(output1);
for(let i=0;i<output1.length;i++){
  console.log(output1[i]);
  output1[i].innerHTML = `<h1>Index : ${i}</h1>`;
}

const divs = document.getElementsByTagName('div');
console.log(divs);

const el1 = document.querySelector('#one');
console.log(el1);
const el2 = document.querySelector('.output');
```

```
console.log(el2);
const el3 = document.querySelector('h1');
console.log(el3);

const el1s = document.querySelectorAll('#one');
console.log(el1s);
const el2s = document.querySelectorAll('.output');
console.log(el2s);
const el3s = document.querySelectorAll('div');
console.log(el3s);

for(let i=0;i<el3s.length;i++){
  el3s[i].innerHTML = `<h1>Div : ${i}</h1>`;
}

el3s.forEach((e,index)=>{
  e.innerHTML = `<h2>ELE ${index}</h2>`;
})
```

Select and update only divs with a class of output

```
<!DOCTYPE html>
<html>
<head>
 <title>JavaScript DOM</title>
</head>
<body>
  <h1 class="output">Laurence Svekis</h1>
```

```
  <div class="output"></div>
  <div id="one"></div>
  <div class="output"></div>
  <div class="output"></div>
  <script src="/js/app2.js"></script>
</body>
</html>
```

Laurence Svekis

Index : 0
Index : 1
Index : 2

```
const eles = document.querySelectorAll('div.output');
console.log(eles);
eles.forEach((el,ind) =>{
  el.innerHTML = `Index : ${ind}`;
})
```

Create Elements and Remove Elements


```
<!DOCTYPE html>
<html>
<head>
 <title>JavaScript DOM</title>
</head>
<body>
  <h1>Laurence Svekis</h1>
  <div class="output"></div>
  <script src="/js/app3.js"></script>
</body>
</html>
```

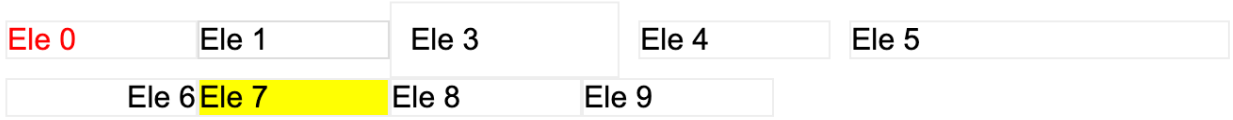
```
const output1 = document.querySelector('.output');
const output2 = document.createElement('div');
console.log(output1);
console.log(output2);

output1.innerHTML = 'Test 1';
```

```
output2.innerHTML = 'Test 2';

document.body.append(output2);
const res1 = output1.append(output2);
document.body.prepend(output2);

const res2 = output1.appendChild(output2);
console.log(res1);
console.log(res2);

output1.removeChild(output2);
console.log(output2);
output1.append(output2);

console.log(output2.parentNode);
output2.parentNode.removeChild(output2);



for(let i=0;i<10;i++){
  const ele = document.createElement('div');
  ele.innerHTML = `Index ${i}`;
  output1.append(ele);
}
/*
let html = '';
for(let i=0;i<10;i++){
  html += `<div>HTML ${i}</div>`;
}
console.log(html);
```

```
output1.innerHTML = html;
*/
```

Page Element Style Property

Laurence Svekis




```
const main = document.querySelector('.output');

for(let i = 0;i<10;i++){
  const ele = document.createElement('div');
  ele.textContent = `Ele ${i}`;
  ele.classList.add('box');
  ele.style.display = 'inline-block';
  ele.style.border = '1px solid #eee';
  ele.style.width = '100px';
  ele.style.fontFamily = 'Arial';
  main.append(ele);
}

const eles = document.querySelectorAll('.box');
console.log(eles);

eles[0].style.color = 'red';
```

```
eles[1].style.border = '1px solid #ddd';
eles[2].style.display = 'none';
eles[3].style.padding = '10px';
eles[4].style.margin = '10px';
eles[5].style.width = '200px';
eles[6].style.textAlign = 'right';
eles[7].style.backgroundColor = 'yellow';
```

Interactive Page elements Events

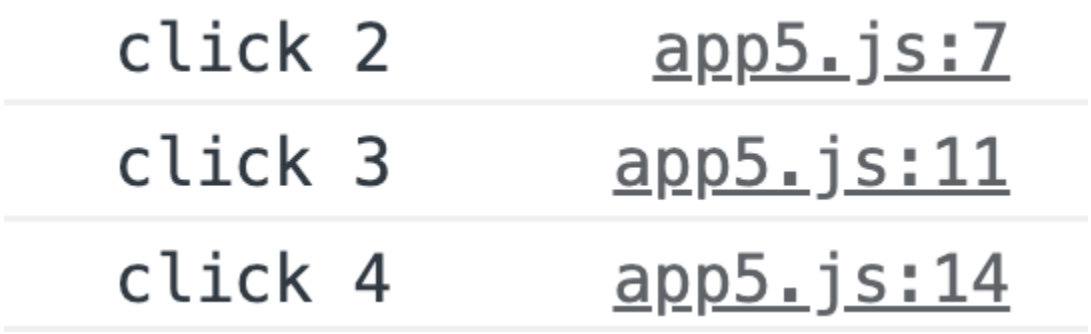
```
const ele1 = document.querySelector('h1');
const ele2 = document.querySelector('.output');
document.body.onload = ()=>{
  console.log('ready');
}
let counter = 0;
ele1.onclick = ()=>{
  adder('clicked');
};
ele1.onmousedown = ()=>{
  adder('mouse down');
}
ele1.onmouseup = ()=>{
  adder('mouse up');
}
ele1.onmouseover = ()=>{
  adder('mouse over');
```

```
}
ele1.onmouseout = ()=>{
  adder('mouse out');
}
ele2.onclick = output;

function adder(val){
  counter++;
  console.log(val);
  ele2.textContent = `${val} click ${counter}`;
}

function output(){
  counter++;
  ele2.textContent = `click ${counter}`;
}
```

Event Object values

```
const ele3 = document.querySelector('h2');

ele3.onclick = adder1;
ele3.onmousedown = adder1;
ele3.onmouseup = adder1;
function adder1(e){
  counter++;
  console.log(e.type);
  e.target.textContent = `Event Type : ${e.type} ${counter}`;
}
```

Event Listeners vs Event Handlers



Use the event listeners when in doubt, event listeners are not overwritten and there is more control over the event.

```
const h2 = document.querySelector('h2');


h2.onclick = ()=>{
  console.log('click 1');
}
h2.onclick = ()=>{
  console.log('click 2');
}

h2.addEventListener('click',()=>{
  console.log('click 3');
})
h2.addEventListener('click',()=>{
  console.log('click 4');
})
```

UseCapture Event Listener Optional

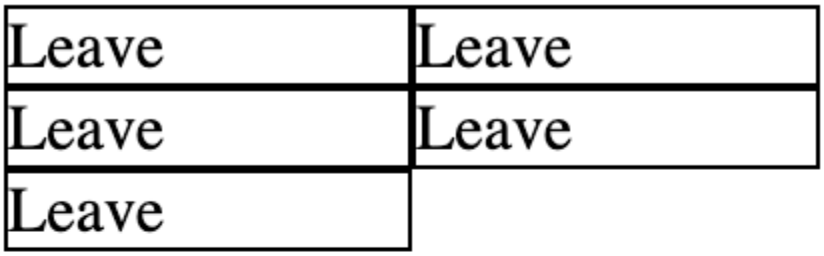
```
<!DOCTYPE html>
<html>
<head>
 <title>JavaScript DOM</title>
 <style>
   .output {
     border:1px solid black;
     width:100px;
     height:50px;
     text-align:center;
     background-color:white;
   }
 </style>
</head>
<body>
   <h1>Laurence Svekis</h1>
   <h2>Click Me</h2>
   <div class="output"></div>
   <script src="/js/app6.js"></script>
</body>
</html>
```

UseCapture Example flows up from the child to parents or true to flow from parent to children.

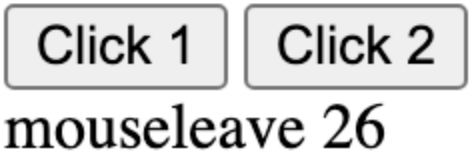
```
const el1 = document.querySelector('h2');
const el2 = document.querySelector('.output');
const el3 = document.querySelector('h1');
el2.textContent = 'Click';
let temp = document.querySelector('.output');
```

```
for(let i=0;i<5;i++){
  const ele = document.createElement('div');
  ele.classList.add('output');
  ele.textContent = `${i+1} Element`;
  ele.id = `Ele${i}`;
  ele.style.height = 300 - i*50 + 'px';
  ele.style.width = 100 + i*50 + 'px';
  temp.append(ele);
  ele.addEventListener('click',()=>{
     console.log(i);
  });
  temp = ele;
}

el1.addEventListener('click',output);
//el2.addEventListener('click',output);
el3.addEventListener('click',output);

function output(e){
  console.log(e.target.id);
  //console.log(e.type);
}



/*
h2.addEventListener('click',(e)=>{
  console.log(e.type);
});
*/
```

Mouse Movement Events



Laurence Svekis



mouseleave 26

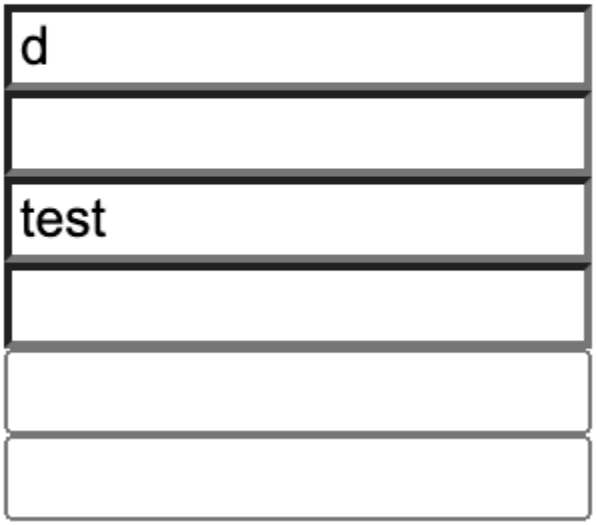
```
const btns = document.querySelectorAll('button');
const main = document.querySelector('.output');
const main1 = document.createElement('div');
document.body.prepend(main1);

for(let i=0;i<5;i++){
  const ele = document.createElement('div');
  ele.style.width = '100px';
  ele.style.display = 'inline-block';
  ele.style.border = '1px solid black';
  ele.textContent = `Element ${i}`;
  ele.addEventListener('mouseenter',()=>{
     ele.style.color = 'red';
     ele.textContent = 'Enter';
```

```
  })
  ele.addEventListener('mouseleave',()=>{
     ele.style.color = 'black';
     ele.textContent = 'Leave';
  })
  ele.addEventListener('mouseover',()=>{
     ele.textContent = 'Over';
  })
  ele.addEventListener('mouseout',()=>{
     ele.textContent = 'Out';
  })
  main1.append(ele);
}


let counter = 0;
btns[0].addEventListener('click',output);
btns[1].addEventListener('click',output);
btns[0].addEventListener('mousedown',show1);
btns[0].addEventListener('mouseup',show1);
btns[0].addEventListener('mouseenter',show1);
btns[0].addEventListener('mouseleave',show1);



function show1(e){
  counter++;
  main.textContent = `${e.type} ${counter}`;
}
```

```
function output(e){
  counter++;
  e.target.removeEventListener('click',output);
  main.textContent = `${e.type} ${counter}`;
}
```

Form Field Event listeners

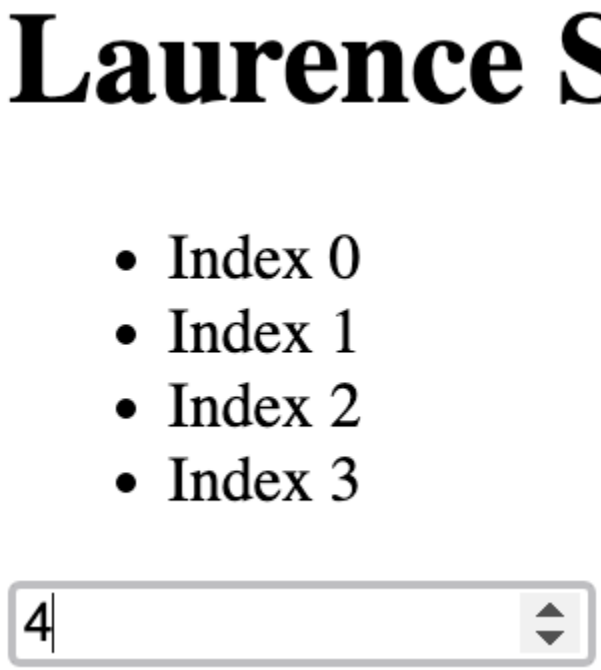
Laurence Svekis




```
const myInput = document.querySelector('input');
const output = document.querySelector('.output');
for(let i=0;i<5;i++){
  const ele = document.createElement('input');
  output.append(ele);
  ele.addEventListener('focus',()=>{
```

```
    ele.style.backgroundColor = 'red';
  })
  ele.addEventListener('focusout',()=>{
    ele.style.backgroundColor = 'white';
  })
  ele.onchange = ()=>{
    console.log(ele.value);
  }
  ele.addEventListener('keydown',adder);
  ele.addEventListener('keyup',adder);
}

function adder(e){
  console.log(e.type);
  console.log(e.key);
}


myInput.onfocus = ()=>{
  console.log('focused');
}
```

List Exercise on events




```
<!DOCTYPE html>
<html>
<head>
 <title>JavaScript DOM</title>
</head>
<body>
  <h1>Laurence Svekis</h1>
  <div class="output"></div>
  <input type="number">
  <script src="/js/app9.js"></script>
</body>
</html>
```

```
const myVal = document.querySelector('input');
const main = document.querySelector('.output');
```

```
myVal.addEventListener('change',()=>{
  main.innerHTML = '';
  const ul = document.createElement('ul');
  main.append(ul);
 if(!isNaN(myVal.value)){
    for(let i=0;i<myVal.value;i++){
        const li = document.createElement('li');
        li.textContent = `Index ${i}`;
        ul.append(li);
    }
 }
})
```

Window Object WOM

```
console.dir(window);
const main = document.querySelector('.output');
main.style.height = '2000px';
window.addEventListener('resize',output);
window.addEventListener('scroll',()=>{
  console.log(window.scrollY);
})


function output(e){
  console.log(window.innerWidth);
}
```

DOM element attributes

Using the set Attribute, element attributes can be added. Elements like input fields, images and hyperlinks commonly have attributes.

Example of elements with attributes added by JavaScript code

```
    <img src="//via.placeholder.com/150" id="img2">
    <a href="http://www.google.com" target="_blank" class="box box1 box2">Click
    me</a>
  </div>
  <input type="number" min="1" max="10" required>
```

```
const main = document.querySelector('.output');



const myInput = document.querySelector('input');
myInput.setAttribute('type','number');
myInput.setAttribute('min','1');
myInput.setAttribute('max','10');
myInput.setAttribute('required','');
myInput.value = '5';

const img1 = document.createElement('img');
main.append(img1);
img1.setAttribute('src','//via.placeholder.com/150');
img1.setAttribute('id','img2');

img1.id = 'img1';
console.log(img1.src);

const hyperL = document.createElement('a');
hyperL.textContent = 'Click me';
hyperL.setAttribute('href','http://www.google.com');
hyperL.setAttribute('target','_blank');
```

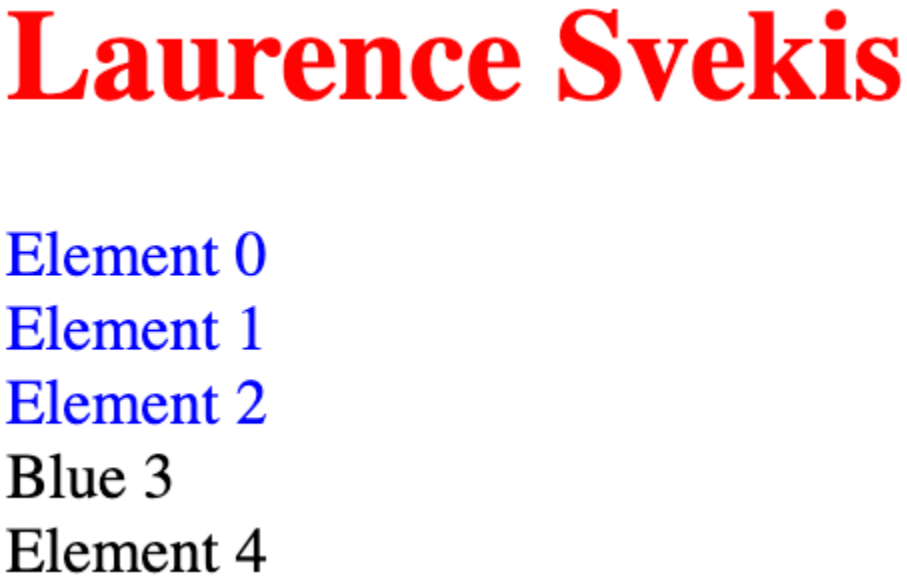
```
hyperL.setAttribute('class','box box1 box2');
main.append(hyperL);
```

Element Get and Has attribute

update all the page hyperlink elements to have a target of _blank.

```
const a1 = document.querySelector('a');
console.log(a1.hasAttribute('href'));
console.log(a1.getAttribute('href'));

const hypers = document.querySelectorAll('a');
hypers.forEach(ele =>{
  if(ele.hasAttribute('target')){
     if(ele.getAttribute('target') != '_blank'){
        ele.setAttribute('target','_blank');
     }
  }else{
     ele.setAttribute('target','_blank');
  }
})
```

Adding and Removing Classes

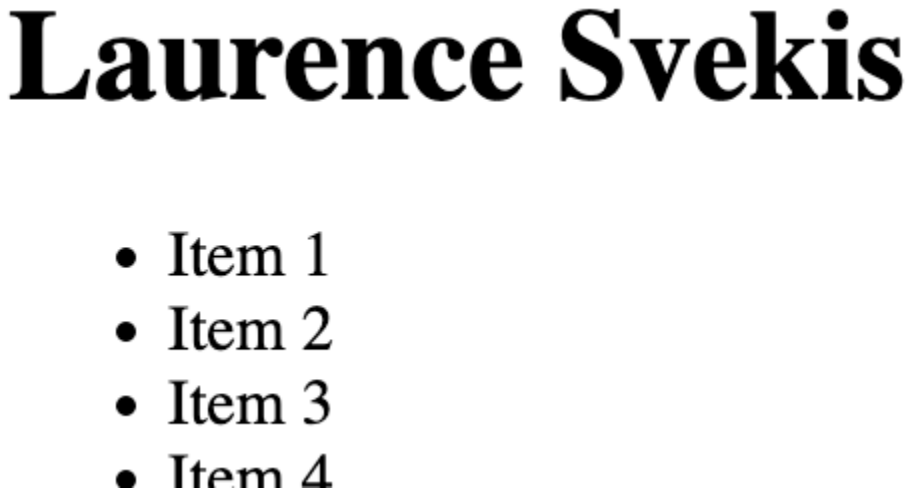



```
<!DOCTYPE html>
<html>
<head>
 <title>JavaScript DOM</title>
 <style>
   .red{
     color:red;
   }
   .blue{
     color:blue;
   }
 </style>
</head>
<body>
   <h1>Laurence Svekis</h1>
   <div class="output"></div>
   <script src="/js/app14.js"></script>
```

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

```
const h1 = document.querySelector('h1');
const main = document.querySelector('.output');

for(let i=0;i<10;i++){
  const ele = document.createElement('div');
  main.append(ele);
  ele.textContent = `Element ${i}`;
  ele.onclick = ()=>{
     console.log(ele);
     if(ele.classList.contains('blue')){
        ele.textContent = `Blue ${i}`;
        ele.classList.remove('blue');
     }else{
        ele.textContent = `Element ${i}`;
        ele.classList.add('blue');
     }
  }
}

h1.onclick = ()=>{
  console.log(h1.classList.contains('red'));
  h1.classList.toggle('red');
}
```

Traversing between elements

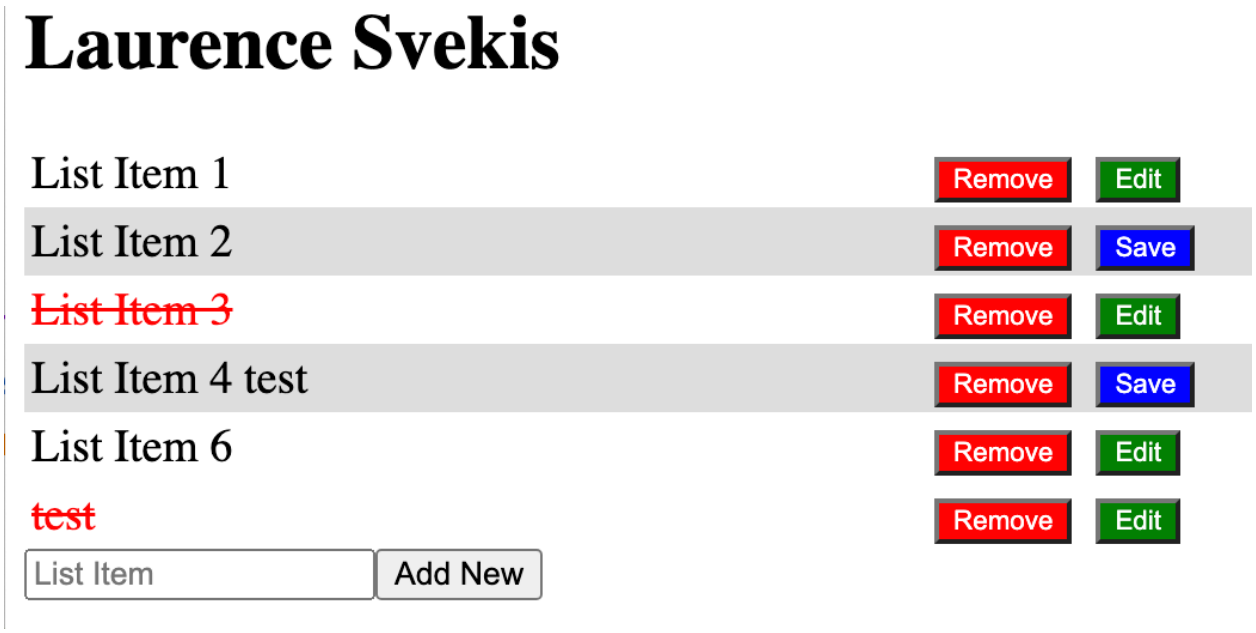



```
<!DOCTYPE html>
<html>
<head>
 <title>JavaScript DOM</title>
</head>
<body>
  <h1>Laurence Svekis</h1>
  <div class="main"></div>
  <script src="/js/app15.js"></script>
</body>
</html>
```

```
const main = document.querySelector('.main');
const ul = document.createElement('ul');
main.append(ul);
for(let i=0;i<9;i++){
  const li = document.createElement('li');
  li.textContent = ` Item ${i+1}`;
  li.onclick = ()=>{
```

```
        const par = li.parentNode;
        console.log('First' + par.firstChild.textContent);
        console.log('Last' + par.lastChild.textContent);
        console.log('CUR' + li.textContent);
        if(li.nextSibling != null){
            console.log('NEXT' + li.nextSibling.textContent);
        }
        if(li.previousSibling != null){
            console.log('PREV' + li.previousSibling.textContent);
        }
    }
    ul.append(li);
}

console.log(ul.childNodes);
console.log(ul.childElementCount);
console.log(ul.children);

ul.childNodes.forEach(el=>{
    console.log(el.textContent);
})
```

List Exercise




```
<!DOCTYPE html>
<html>
<head>
 <title>JavaScript DOM</title>
 <style>
  .striker {
    text-decoration:line-through;
    color:red;
    cursor:grab;
  }
  .btns{
    margin:0 5px;
    font-size:0.7em;
    color:white;
  }
  ul, li{
    margin:0;
```

```
    padding:0;
    list-style-type:none;
  }
  .content{
    font-size:1.2em;
    padding:3px;
    width:50%;
    display:inline-block;
  }
 </style>
</head>
<body>
  <h1>Laurence Svekis</h1>
  <div class="main"></div>
  <script src="/js/app16.js"></script>
</body>
</html>
```

```
const main = document.querySelector('.main');
const ul = adder('ul',main,'ul','' );
const inEle = adder('input',main,'input','' );
const btn = adder('button',main,'btn','Add New' );
inEle.setAttribute('type','text');
inEle.setAttribute('placeholder','List Item');

btn.addEventListener('click',newListItem);

for(let i=0;i<6;i++){
  inEle.value = `List Item ${i+1}`;
  newListItem();
```

```
}



function newListItem(){
  const li = adder('li',ul,'li');
  const ele = adder('span',li,'content',inEle.value);
  const delEle = adder('button',li,'btns','Remove');
  const ediEle = adder('button',li,'btns','Edit');
  ele.setAttribute('contenteditable','false');
  delEle.style.backgroundColor = 'red';
  ediEle.style.backgroundColor = 'green';
  ele.onclick = ()=>{
     if(ele.getAttribute('contenteditable') == 'false'){
        ele.classList.toggle('striker');
     }
  }
  delEle.onclick = ()=>{
     //console.log(delEle.parentNode.firstChild);
     li.remove();
  }
  ediEle.onclick = ()=>{
    if(ele.getAttribute('contenteditable') == 'true'){
        ele.setAttribute('contenteditable','false');
        li.style.backgroundColor = '#fff';
        ediEle.textContent = 'Edit';
        ediEle.style.backgroundColor = 'green';
    }else{
        ele.setAttribute('contenteditable','true');
        ele.classList.remove('striker');
```

```
            li.style.backgroundColor = '#ddd';
            ediEle.textContent = 'Save';
            ediEle.style.backgroundColor = 'blue';
        }
    }
    inEle.value = '';
}


function adder(t,parent,c,tc){
    const ul = document.createElement(t);
    ul.classList.add(c);
    ul.textContent = tc;
    return parent.appendChild(ul);
}
```

requestAnimationFrame movement

Laurence Svekis

```
<!DOCTYPE html>
<html>
<head>
 <title>JavaScript DOM</title>
 <style>
   .main{
     position:absolute;
     width:100px;
     height:100px;
     left:0;
     top:0;
     background-color:red;
   }
 </style>
</head>
<body>
 <h1>Laurence Svekis</h1>
 <div class="main">Move</div>
 <script src="/js/app17.js"></script>
</body>
</html>
const main = document.querySelector('.main');
const pos = {x:0,y:0};
//const intVal = setInterval(updater,100);
let intAni = requestAnimationFrame(moveEle);

function moveEle(){
   pos.x+=10;
   pos.y+=5;
```

```
  mover();
  if(pos.x > 200){
     cancelAnimationFrame(intAni);
  }else{
     intAni = requestAnimationFrame(moveEle);
  }
}

function updater(){
  pos.x+=10;
  pos.y+=5;
  if(pos.x > 200){
     clearInterval(intVal);
  }
  mover();
}

function mover(){
  main.style.left = pos.x+'px';
  main.style.top = pos.y+'px';
}
```

Bouncing Ball Animation

JavaScript How to Create a Bouncing Ball Animation with Vanilla JavaScript and moving Page elements. This exercise will create a bouncing red ball, within a div. Movement is done by selecting the page element, and updating the style property with JavaScript code. The updated position of the ball is tracked within a variable object that contains all the

values needed for the ball. Use a request animation frame to create a smooth animation on the screen. Once the ball position reaches any of the container element boundaries it will reverse direction on the axis that the boundary was reached. This ensures a continued movement within the boundaries. Speed and ball size can also be dynamically adjusted within the ball object settings.

1. Create an HTML document with one page element div, with the class of main.

```
<div class="main"></div>
```

2. Select the page element into a variable named main.

```
const main = document.querySelector('.main');
```

3. Set the ball object details, as a variable named b. Set the x and y start position, the width and height for the ball, the direction on x axis and direction on y axis. Add a speed value and a blank object for the animation object.

```
const b = {x:50,y:30,w:40,h:40,dx:1,dy:1,speed:5,ani:{}}
```

4. Create a new page element nested within the main page element. Assign the object to a variable named ball, and append it to the main page element.

```
const ball = document.createElement('div');
main.append(ball);
```

5. Setup the style values, apply height and width, update the ball to be round and with the b object property values. *Yes this can also be set with CSS

```
main.style.width = '600px';
main.style.height = '400px';
main.style.backgroundColor = 'black';
ball.style.backgroundColor = 'red';
ball.style.borderRadius = '50%';
ball.style.width = `${b.w}px`
ball.style.height = `${b.h}px`
ball.style.position = 'relative';
ball.style.left = `${b.x}px`;
ball.style.top = `${b.y}px`;
```

6. Create a function named mover() to handle the animation. Within mover update the b.x and b.y position values using the direction multiplied by speed.

```
b.x += b.dx * b.speed;
b.y += b.dy * b.speed;
```

7. Set the ball style, updating the style values with the new x and y position for the ball. This will update it to the page.

```
ball.style.left = `${b.x}px`;
ball.style.top = `${b.y}px`;
```

8. Run the next repaint with the requestAnimationFrame, you can assign it to a variable in case it needs to be canceled.

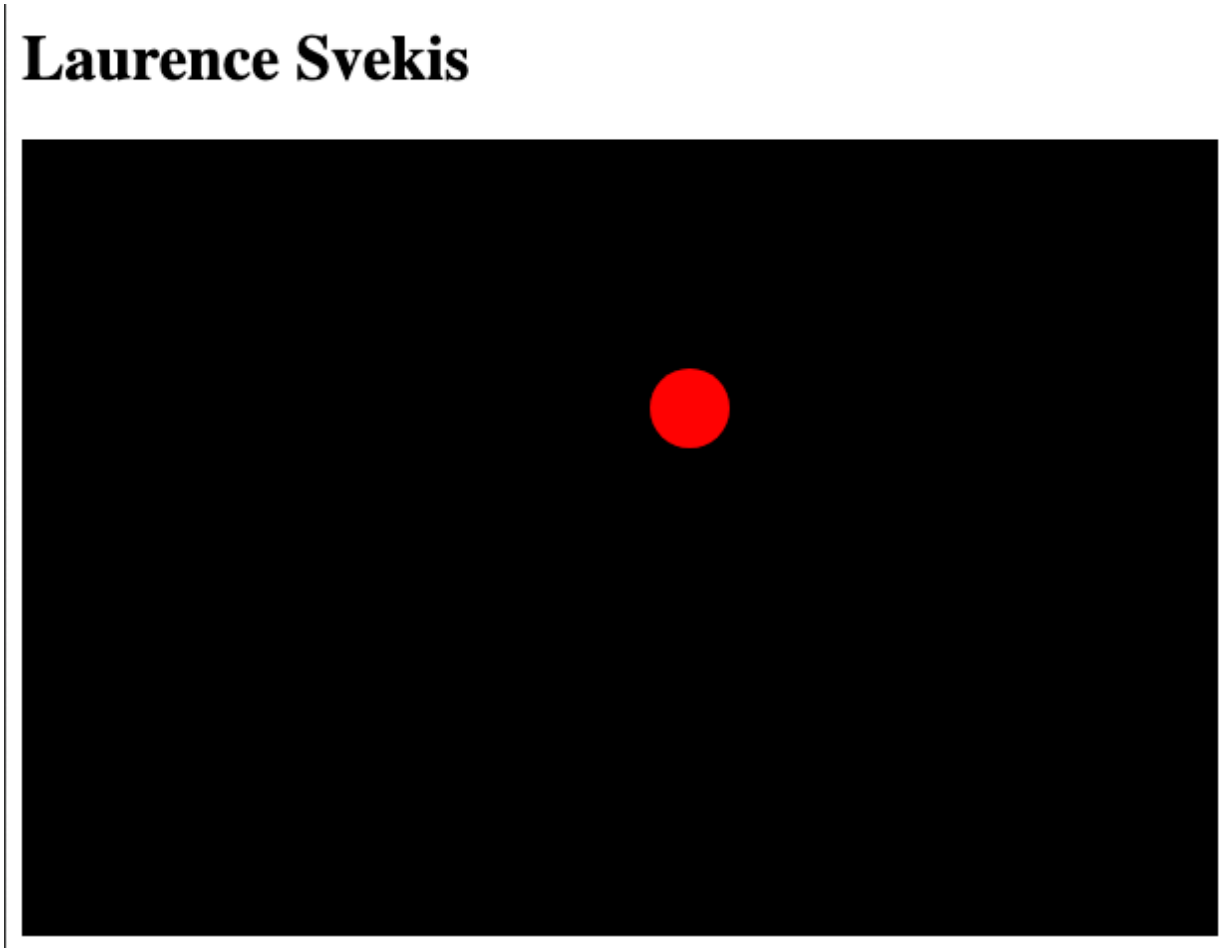
```
b.ani = requestAnimationFrame(mover);
```

9. In the mover() function check for the x and y axis boundaries, if the position is outside the boundaries reverse the direction by multiplying it by -1. 1*-1 = -1 and 1*1=1 so the result will be that the direction will be either negative or positive 1.

```
if(b.x>600-b.w || b.x < 0){
    b.dx *= -1;
}
if(b.y>400-b.h || b.y < 0 ){
    b.dy *= -1;
}
```

10. Globally invoke the mover() function, this can be done within the requestAnimationFrame

```
b.ani = requestAnimationFrame(mover);
```




```
<!DOCTYPE html>
<html>
<head>
 <title>JavaScript DOM</title>
</head>
<body>
 <h1>Laurence Svekis</h1>
 <div class="main"></div>
 <script src="/js/app.js"></script>
</body>
</html>
```

```
const main = document.querySelector('.main');
```

```
const h1 = document.querySelector('h1');
main.style.width = '600px';
main.style.height = '400px';
main.style.backgroundColor = 'black';

const ball = document.createElement('div');
const b = {x:50,y:30,w:40,h:40,dx:1,dy:1,speed:5,ani:{},move:false}
ball.style.backgroundColor = 'red';
ball.style.borderRadius = '50%';
ball.style.width = `${b.w}px`
ball.style.height = `${b.h}px`
ball.style.position = 'relative';
ball.style.left = `${b.x}px`;
ball.style.top = `${b.y}px`;
main.append(ball);

h1.addEventListener('click',()=>{
  if(!b.move){
     b.ani = requestAnimationFrame(mover);
     b.move = true;
  }else{
     cancelAnimationFrame(b.ani);
     b.move = false;
  }

})

function mover(){
  if(b.x>600-b.w || b.x < 0){
```

```
    b.dx *= -1;
  }
  if(b.y>400-b.h || b.y < 0 ){
    b.dy *= -1;
  }
  b.x += b.dx * b.speed;
  b.y += b.dy * b.speed;
  ball.style.left = `${b.x}px`;
  ball.style.top = `${b.y}px`;
  if(b.move){
    b.ani = requestAnimationFrame(mover);
  }
}
```