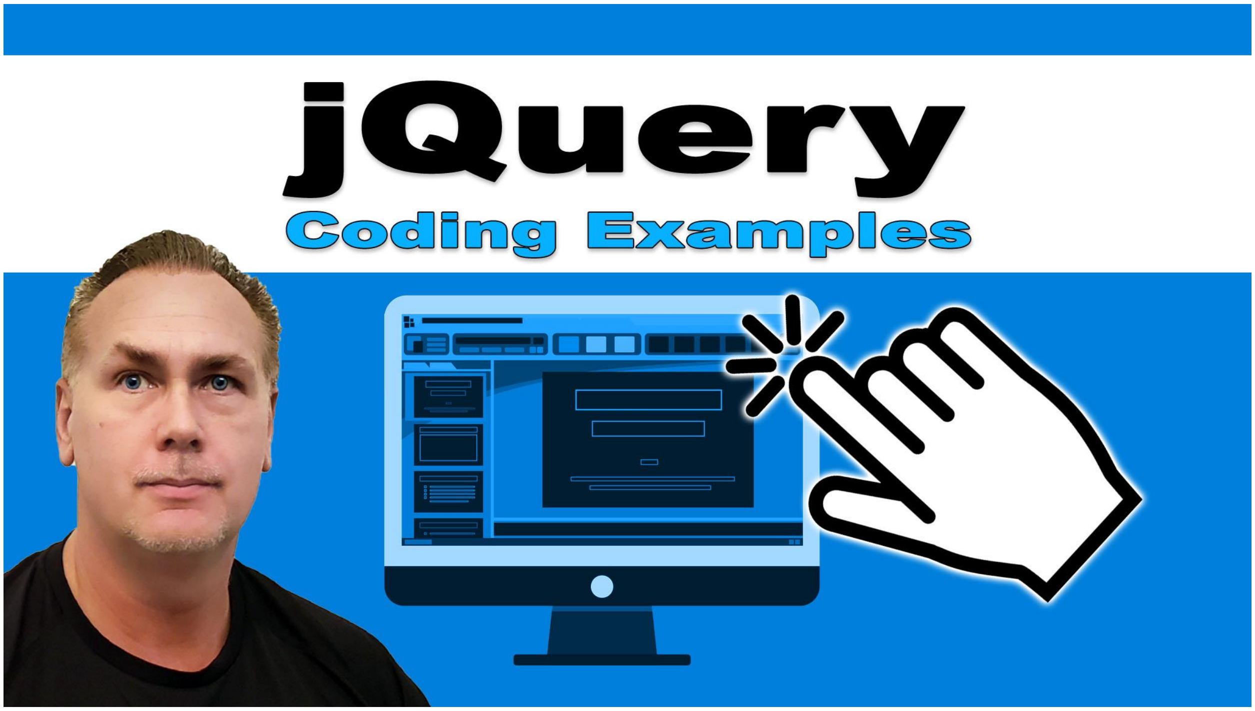
jQuery for beginners to advanced

Create animations, handle page events for interactions, dynamically update page elements using AJAX made easy and more.



Getting started with jQuery adding it to your HTML 3

Selection of Page Elements and DOM Element selection jQuery 4

Element hide and Show Methods and Animation effects 6

Fading Effects with jQuery Elements 8

Sliding Effects on Page Elements create Slide 10

How to create Custom Animations with jQuery 12

How to Insert Elements and Contents into and outside elements 15

Element DOM manipulation and Selection 18

Classes DOM manipulation adding and removing element class 20

jQuery Dynamic List Project Interactive elements with jQuery 23

Laurence Svekis https://basescripts.com/

How to get and set CSS properties with jQuery 26

Getting and setting Element Attributes with jQuery 29

Element Dimensions and properties values with jQuery Methods 34

Traversing Descendants of Page Elements jQuery Selection 37

Traversing Ancestors of Page Elements parent methods with jQuery 39

Traversing Siblings Page Elements and other selections with jQuery 42

Traversing Filtering Page Elements to select them using jQuery Methods 46

How to create a Dynamic List with jQuery project 50

jQuery Data Method save values into the element object 54

jQuery get Method to get the index value of a page element 57

jQuery index of page element index Method 59

jQuery toArray Method to get the jQuery elements as DOM elements 62

jQuery Mouse Move Events Listeners and hover Events 65

jQuery to listen for keyboard Events and get values from the event object 69

jQuery form Events on submit and more 72

Attach events with the on Method with more powerful events 75

jQuery scroll event on Browser Events and window events 78

How to create a jQuery Mole hit game 80

Load a file with jQuery load method 88

Get JSON data with jQuery get method 90

POST data with jQuery using post method 93

GET shorthand Methods jQuery getScript getJSON methods 96

jQuery AJAX method and callback options 99

Introduction to jQuery

jQuery makes it easy to get started with building interactive and dynamic web content. With basic knowledge of HTML and CSS you can start creating interaction.
jQuery is just a JavaScript library and knowledge of JavaScript is helpful for the understanding and debugging of code.

jQuery provides an easy to use solution for web developers to add smooth animations, handle web page interactions and events, to navigate page elements for selection, update and manipulate element properties and contents, and make AJAX requests for data.

Designed for anyone who wants to learn jQuery. Prior HTML and CSS experience is recommended. Learn by example, each lesson has its own challenge to help you get more familiar with specific coding objectives. Source code is included that will help guide you through the lesson content with helpful tips and resources.

jQuery is used because of its ease of use, making it simple to create amazing animations and web page experiences. There is a large community for documentation and tutorials. It works across browsers and standardizes the experience so that all the browsers display the same way. There are a vast number of plugins which can help create even more wonderful things with code. The code is also easier to read as the functions used in jQuery are simple and have semantic meaning.

Downloading jQuery : https://jquery.com/download/

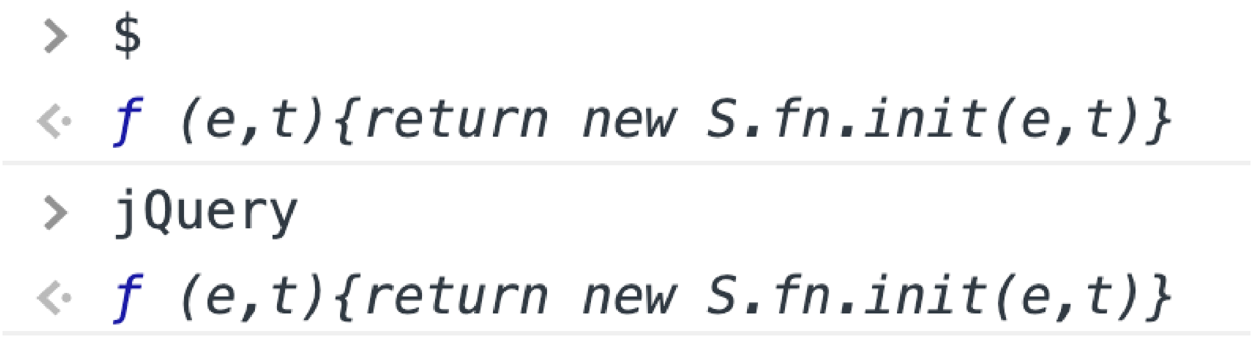
Use the compressed version, as it saves bandwidth and is the smaller file size with whitespace removed. The uncompressed would only be for debugging.

There are many versions of jQuery suggested to use the latest 3.* version

jQuery is designed to simplify the HTML DOM tree traversal and manipulation of page elements. It provides an excellent way to create CSS animations, handle events and AJAX requests.

CDN source https://developers.google.com/speed/libraries#jquery
Code Editor https://code.visualstudio.com/

Use either $ or jQuery in the browser console to check if jQuery is ready



```
> $
<· f (e,t){return new S.fn.init(e,t)}
> jQuery
<· f (e,t){return new S.fn.init(e,t)}
```

Getting started with jQuery adding it to your HTML

How to add jQuery to your HTML and create a simple element selection, event and manipulation of element content with jQuery.

Exercise : Create a file add jQuery and write some jQuery Code

1. Create an HTML file add a page element
2. Add the script tag link to the jQuery library
3. Add a second script tag link to a new js file named app1.js
4. Open the HTML in the browser check if jQuery is on the page
5. In the app1.js select the element, you can use the same syntax as you would with CSS. $()
6. Add a jQuery method named .html()
7. Add a string value within .html()
8. Add the on(‘click’) to the element and nest the update of the html inside
9. Add a second html() chained to the on click method with another string value

```
INDEX.HTML
```

```
<!DOCTYPE html>
<html>
   <head><title>jQuery Course</title></head>
   <body>
       <div>Hello World</div>
       <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.m
in.js"></script>
       <script src="app1.js"></script>
   </body>
</html>
```

APP1.JS

```
$('div').on('click',()=>{
   $('div').html('Laurence Svekis');
}).html('Click Me');
```

Selection of Page Elements and DOM Element selection jQuery

1. Create an HTML file to interact with, including elements with ids and classes.
2. Make a selection of the element using the tag, class and id. Output the jQuery element object into the console.
3. Create variables and assign the jQuery element objects and output them to the page.

```
<!DOCTYPE html>
<html>
   <head><title>jQuery Course</title></head>
   <body>
       <div>Hello World 1</div>
       <div class="div2">Hello World 2</div>
       <div id="div3">Hello World 3</div>
       <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.m
in.js"></script>
       <script src="app2.js"></script>
   </body>
</html>
```

APP2.JS

```
console.log($('div'));
$('div').html('Hello');
console.log($('.div2'));
console.log($('#div3'));
console.log($('#div3, .div2'));

const $ele1 = $('div');
console.log($ele1);
$ele1.html('Hello 2');
const $ele2 = $('.div2');
console.log($ele2);
```

```
const $ele3 = $('#div3');
console.log($ele3);
const $ele4 = $('#div3, .div2');
console.log($ele4);
```

Element hide and Show Methods and Animation effects

Click elements, hide the clicked elements. Add a button that will show the elements.

1. Select all the div's from the page
2. Select the first one using first() and the last one using last()
3. Add a click event to the div's. When clicked it should select the element from the click event that was clicked. Be careful as this will select the nested elements. Update the html ro say clicked. Update the element with text or ready. Then apply the hide() function to the element
4. Add an event to the button which shows all the divs, and the H1

```
<!DOCTYPE html>
<html>
   <head><title>jQuery Course</title>
   </head>
   <body>
       <div>Hello World 1</div>
       <div class="div2">Hello World 2</div>
       <div id="div3">Hello World 3</div>
       <button>Show</button>
```

```
        <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
        <script src="app3.js"></script>
    </body>
</html>

console.log($('div').first());
const val = '<h1>Laurence Svekis</h1>';
$('div').first().html(val);
$('div').last().text(val);

$('div').click((e)=>{
    const $el = $(e.target);
    console.log($el);
    $el.html('clicked');
    $(e.target).text('Ready');
    $(e.target).hide("slow");
});

$('button').click((e)=>{
    $('div').show("fast");
    $('h1').show(4000);
});

function eleClicker(e){
    console.log(e);
}
```

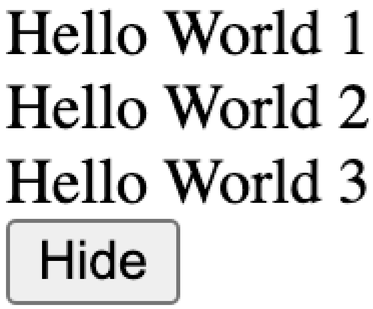
Fading Effects with jQuery Elements

1. Make all the page divs clickable
2. Create variables to track the number of times the elements were clicked, and the number of elements that are currently hidden.
3. Update the text or html of the element after its clicked
4. Add a click event to the button, once clicked shows, toggles and applies different fading effects to the elements.
5. Create a variable that will toggle the opacity of the button element, every time its clicked it should change the opacity from 0.5 to 1 and back.

```
<!DOCTYPE html>
<html>
   <head><title>jQuery Course</title>
   </head>
   <body>
       <div>Hello World 1</div>
       <div class="div2">Hello World 2</div>
       <div id="div3">Hello World 3</div>
       <button>Show</button>
       <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
       <script src="app4.js"></script>
   </body>
</html>

const val = '<h3>Laurence Svekis</h3>';
```

```
$('div').first().html(val);
let counter = 0;
let hiddenEle = 0;
let fader = 0.5;
$('div').click(function(e){
    //$(e.target).fadeOut(1000);
    console.log($(this));
    console.log($(this).text());
    $(this).fadeOut(1000,()=>{
        counter++;
        hiddenEle++;
        const temp = `<h3>Counter ${counter}</h3>`;
        $(this).html(temp);
        $('button').text(`Show (${hiddenEle})`);
    });
})

$('button').click((e)=>{
    //$('div').fadeIn("slow");
    /*
    $('div').fadeToggle('slow',function(){
        $(this).text('Was toggled');
    })
*/
    $('div').toggle(1500);
    hiddenEle = 0;
    $('button').text(`All Showing`);
    $('button').fadeTo('slow',fader,()=>{
        if(fader == 0.5){
            fader = 1;
        }else{
            fader = 0.5;
        }
    })
})
```

Sliding Effects on Page Elements create Slide



Hello World 1
Hello World 2
Hello World 3

Exercise : Slide effect with jQuery

1. Hide the show button
2. Add a click event on the hide button which will hide all the page divs using the slideUp() and invoke a function that will toggle both buttons
3. The second show button should now be visible, add a click event on the button to show all the divs using the slideDown() function.
4. Add a click event to each div that does slideToggle() which should hide the div when clicked.

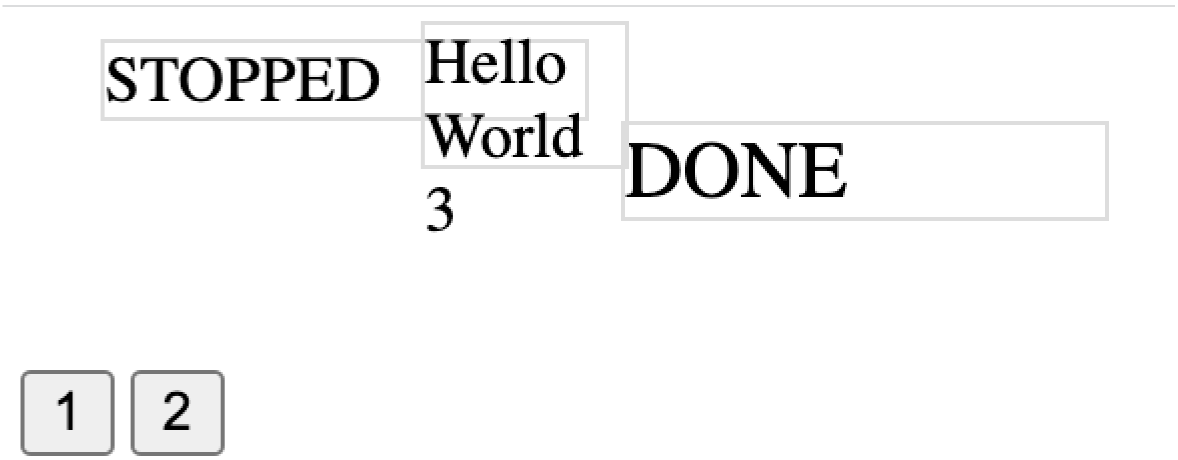
```
<!DOCTYPE html>
```

```
<html>
   <head><title>jQuery Course</title>
   </head>
   <body>
       <div>Hello World 1</div>
       <div class="div2">Hello World 2</div>
       <div id="div3">Hello World 3</div>
       <button id="btn1">Hide</button>
       <button id="btn2">Show</button>
       <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
       <script src="app5.js"></script>
   </body>
</html>
```

```
$('#btn2').hide();
$('#btn1').click(()=>{
   $('div').slideUp('slow',doneFun);
})
$('#btn2').click(()=>{
   $('div').slideDown('fast',doneFun);
})
$('div').click(function(){
   $(this).slideToggle();
})
function doneFun(){
   console.log('slide done');
```

```
    $('#btn1').toggle();
    $('#btn2').toggle();
}
```

How to create Custom Animations with jQuery



STOPPED

Hello World

3

DONE

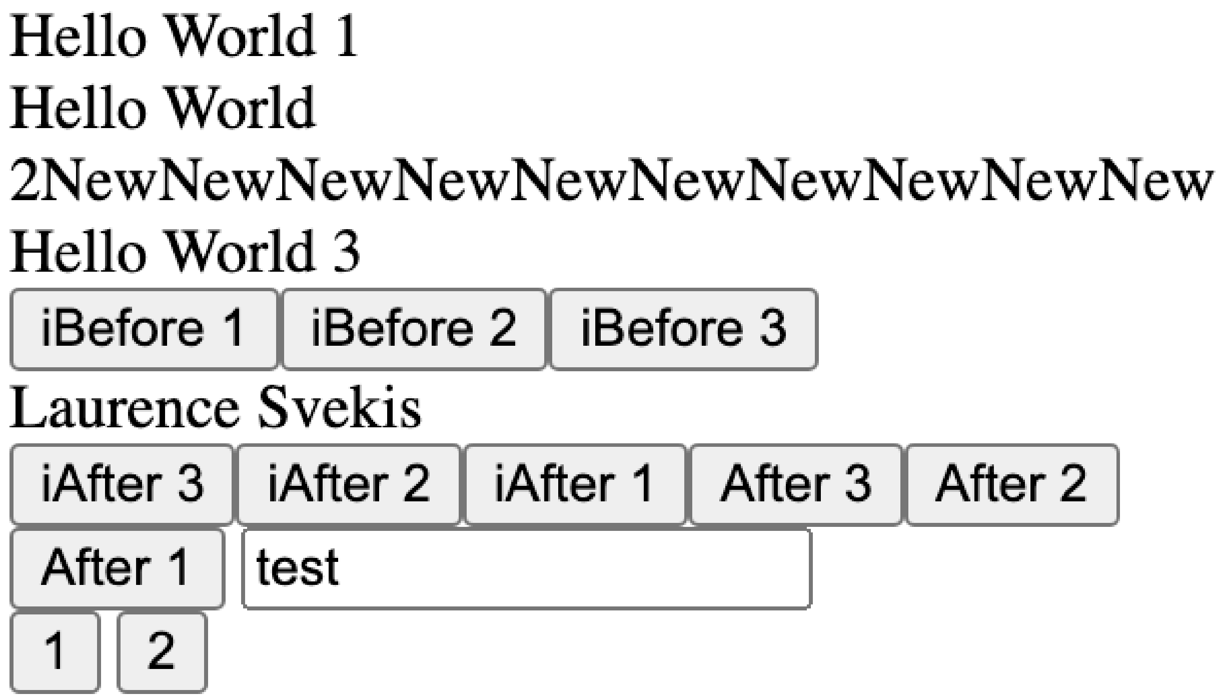
Exercise : select and move page elements and apply custom animation effects to selected elements.

1. Add position relative or absolute to the page elements you want to move
2. Select all the elements, with button one click apply various animations with styling values
3. With button two click add different custom animations to the page elements

```
<!DOCTYPE html>
<html>
    <head><title>jQuery Course</title>
        <style>
            div{
                left:100px;
                position:relative;
                border:1px solid #ddd;
                width:120px;
            }
        </style>
    </head>
    <body>
        <div class="div1">Hello World 1</div>
        <div class="div2">Hello World 2</div>
        <div class="div3" id="div3">Hello World 3</div>
        <button id="btn1">1</button>
        <button id="btn2">2</button>
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
        <script src="app6.js"></script>
    </body>
</html>
```

```
$('#btn1').click(()=>{
    $('.div1').text(`I'm moving`);
    $('.div1').animate({
        left:"+=20",
        opacity: 0.5
    },1000,()=>{
        $('.div1').text(`STOPPED`);
    });
    $('.div2').animate({
        left:"+=50",
        fontSize: "20px"
    },3000,()=>{
        $('.div2').text(`Laurence Svekis`);
    });
    $('.div3').animate({
        top:"+=50",
        width:"150px",
        height:"150px"
    },1000,()=>{
        $('.div2').text(`Laurence Svekis`);
    });
})
$('#btn2').click(()=>{
    $('.div1').text(`I'm moving`).animate({
        left:"-=50",
        opacity: 1
    },1000,()=>{
        $('.div1').text(`STOPPED`);
    });
    $('.div3').animate({
        top:"-=50",
        width:"50px",
        height:"35px"
    },1000,()=>{
        $('.div2').text(`DONE`);
    });
})
```

How to Insert Elements and Contents into and outside elements



Hello World 1
Hello World 2NewNewNewNewNewNewNewNewNewNew
Hello World 3

Laurence Svekis

Exercise : adding new content and elements to the page, both inside and outside the selected element. Getting input values and using them in code. Insert before and after.

1. Create new page elements, both adding it to a variable and also as a jQuery object directly. Create a loop to add clickable buttons that output the button text content into the console.   Try adding before and after page elements.
2. On click of the first button, get the text value of the input element. Try append and prepend of page elements. Try remove method on selected element.
3. On the second button create an element that can be added inside the element try both append and appendTo. Note the difference in the statement.

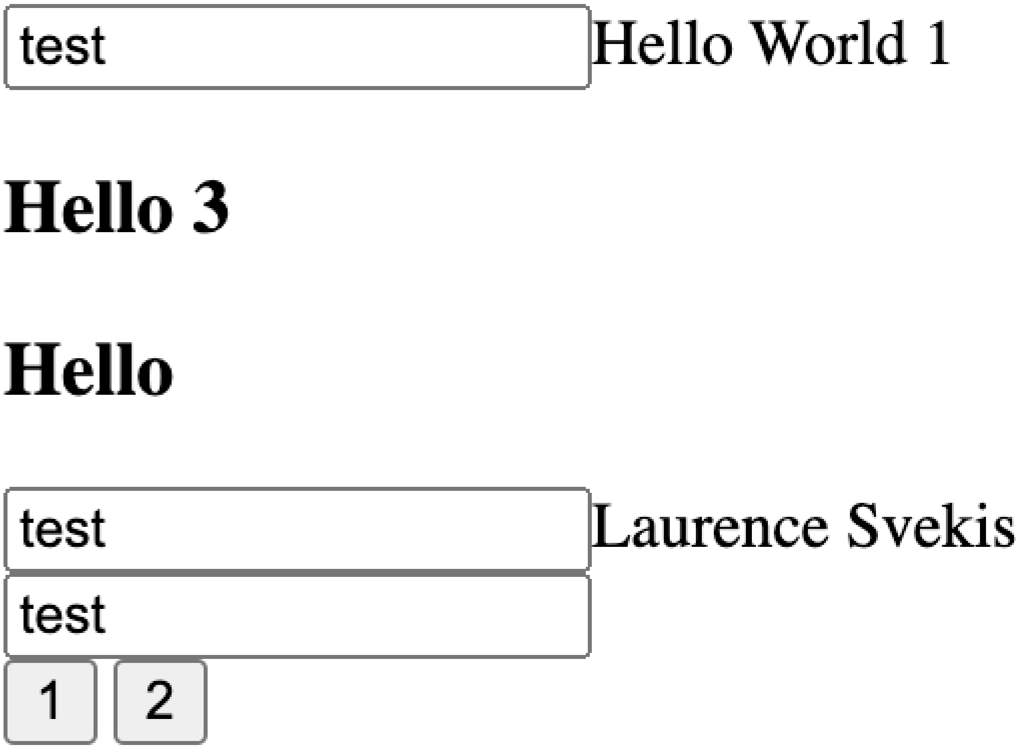
```
<!DOCTYPE html>
```

```
<html>
    <head><title>jQuery Course</title>
    </head>
    <body>
        <div class="div1">Hello World 1</div>
        <div class="div2">Hello World 2</div>
        <div class="div3" id="div3">Hello World 3</div>
        <div class="div4">Laurence Svekis</div>
        <input type="text" value="test"><br>
        <button id="btn1">1</button>
        <button id="btn2">2</button>
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
        <script src="app7.js"></script>
    </body>
</html>
```

```
let counter = 0;
for(let i=0;i<10;i++){
    counter++;
    const $span = $(`<span>Hello :${counter}</span>`);
    $span.text('New');
    $('.div2').append($span);
}
for(let i=0;i<3;i++){
    const $btn = $(`<button>After ${i+1}</button>`);
    $('.div4').after($btn);
    $btn.click(function(){
        console.log($(this).text());
    })
    $(`<button>iBefore ${i+1}</button>`).
    insertBefore('.div4').click(function(){
        console.log($(this).text());
    })
}
for(let i=0;i<3;i++){
    //$('.div4').before($btn);
    $(`<button>iAfter ${i+1}</button>`).insertAfter('.div4').click(function(){
        console.log($(this).text());
    })
}
$('#btn1').click(()=>{
    const val = $('input').val();
    $('.div1').text(val);
    $('.div2').append(val);
    $('.div3').prepend(val);
    $('.div4').remove();
});

$('#btn2').click(()=>{
    counter++;
    const $div = $(`<div>Hello :${counter}</div>`);
    $('.div3').append($(`<div>Hello :${counter}</div>`));
    $(`<div>Hello :${counter}</div>`).appendTo('.div3');
```

```
   console.log($div);

})
```

Element DOM manipulation and Selection



test Hello World 1

Hello 3

Hello

test Laurence Svekis

test

Exercise : Clone and update with replace page elements

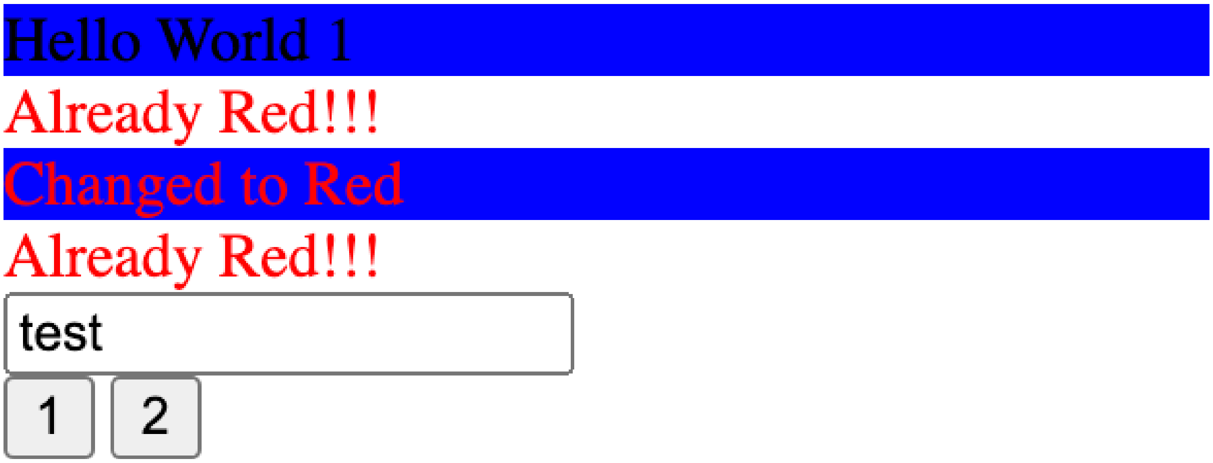
1. For click of button one, using replaceAll and replaceWith update selected elements with new content.
2. Clicking button two using clone() copy a selected element and add the new copied element to an existing element with prependTo()

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

```
   <head><title>jQuery Course</title>
   </head>
   <body>
       <div class="div1">Hello World 1</div>
       <div class="div2">Hello World 2</div>
       <div class="div3" id="div3">Hello World 3</div>
       <div class="div4">Laurence Svekis</div>
       <input type="text" value="test" id="myIn"><br>
       <button id="btn1">1</button>
       <button id="btn2">2</button>
       <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.m
in.js"></script>
       <script src="app8.js"></script>
   </body>
</html>
```

```
$('#btn1').click(()=>{
 $('<h3>Hello</h3>').replaceAll('.div3');
 $('.div2').replaceWith('<h3>Hello 3</h3>');
})

$('#btn2').click(()=>{
   const $ele = $('#myIn').clone();
   console.log($ele);
   const temp = $('#myIn').val();
   $('input').last().clone().prependTo('div');
   //$ele.clone().prependTo('div');
})
```

Classes DOM manipulation adding and removing element class



Hello World 1

Already Red!!!

Changed to Red

Already Red!!!

test

Exercise : Adding, removing and toggling element classes

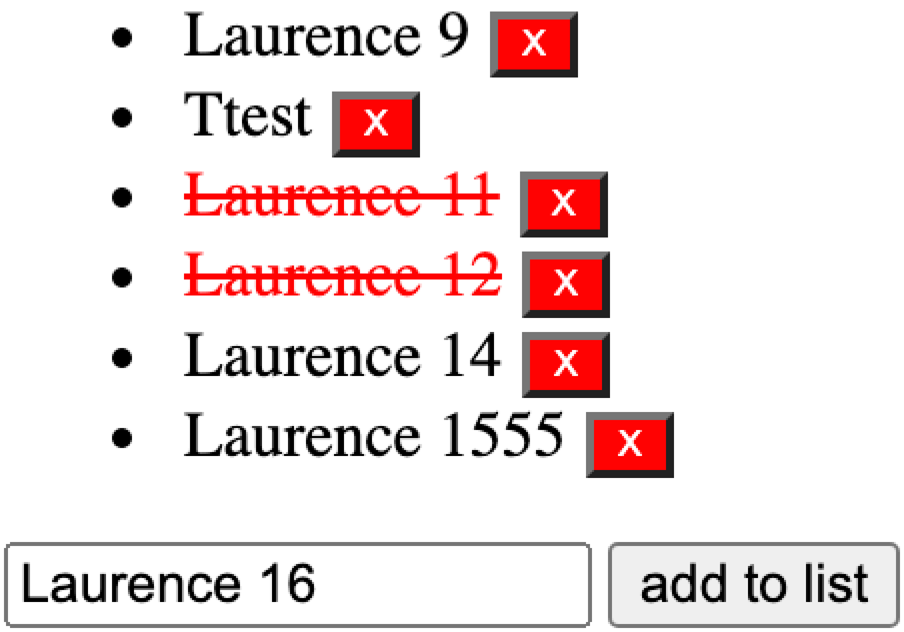
1. Add a click event to all the divs. When clicked, check if the element hasClass() and set the text to a value from the result.
2. Add the class of red and remove the class of blue
3. For the first button add a click event that adds the blue class to all the divs
4. For the second button add a click event that toggles both red and blue classes on all the divs.

```
<!DOCTYPE html>
<html>
   <head><title>jQuery Course</title>
       <style>
           .red{
               color:red;
```

```
            }
            .blue{
                background-color:blue;
            }
        </style>
    </head>
    <body>
        <div class="div1">Hello World 1</div>
        <div class="div2">Hello World 2</div>
        <div class="div3" id="div3">Hello World 3</div>
        <div class="div4">Laurence Svekis</div>
        <input type="text" value="test" id="myIn"><br>
        <button id="btn1">1</button>
        <button id="btn2">2</button>
        <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
        <script src="app9.js"></script>
    </body>
</html>
```

```
$('div').click(function(){
    console.log($(this));
    const boo = $(this).hasClass('red');
    console.log(boo);
    if(boo){
        $(this).text('Already Red!!!');
    }else{
        $(this).text('Changed to Red');
    }
    $(this).addClass('red');
    $(this).removeClass('blue');
 })

 $('#btn1').click(()=>{
    $('div').addClass('blue');
 })

 $('#btn2').click(()=>{
    $('div').toggleClass('red');
    $('div').toggleClass('blue');
 })
```

jQuery Dynamic List Project Interactive elements with jQuery



- Laurence 9
- Ttest
- ~~Laurence 11~~
- ~~Laurence 12~~
- Laurence 14
- Laurence 1555

Laurence 16

Exercise : Create an interactive list that can add new items, the items in the list can be crossed out and also removed with the push of the button.

1. Create the stylign for the list elements, add an UL list and input and button to the HTML page
2. Add an event to the button when clicked will create the list item
3. Create a list item, append it to the UL element.
4. Append a span to the LI with the text coming from the input value.
5. Append a button to the LI
6. When the span is clicked toggle the class of red on the parent LI
7. When the button is clicked remove the parent LI

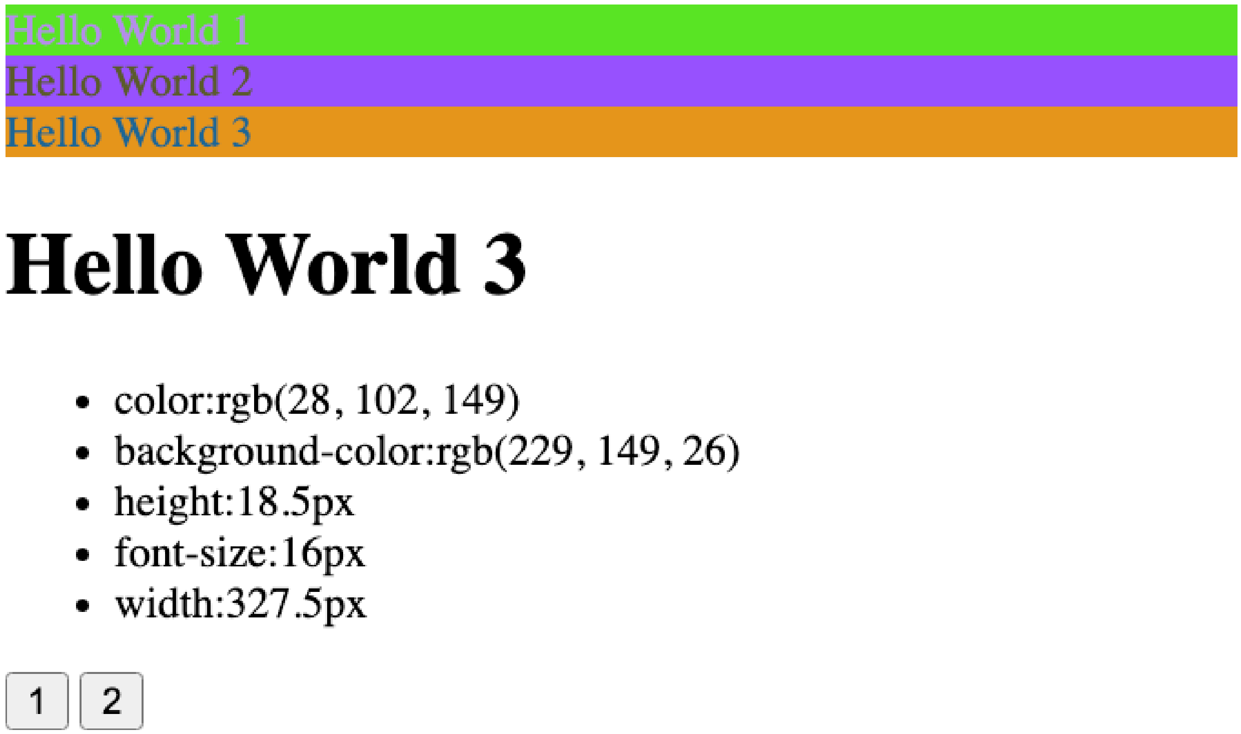
```
<!DOCTYPE html>
<html>
   <head><title>jQuery Course</title>
```

```
        <style>
            li span{
                padding:5px;
            }

            .red span{
                color:red;
                text-decoration: line-through;

            }
            li button{
                font-size:0.5em;
                background-color:red;
                color:white;
            }
        </style>
    </head>
    <body>
        <ul class="output"></ul>
        <input type="text" value="test" id="myIn">
        <button id="btn1">add to list</button>
        <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
        <script src="exe1.js"></script>
    </body>
</html>
```

```
$('#btn1').click(adder);
let counter = 0;
function adder(){
   const val = $('input').val();
   const $li = $('<li>');
   const $el = $('<span>').text(val);
   const $btn = $('<button>').text('X');
   $el.click(function(){
       $li.toggleClass('red');
   })
   $btn.click(function(){
       $li.remove();
   })
   $li.append($el).append($btn);
   $('.output').append($li);
   counter++;
   const temp = `Laurence ${counter}`;
   $('input').val(temp);
   console.log(val);
}
```

jQuery Element Traversing and selection of specific elements update elements

How to get and set CSS properties with jQuery



Hello World 1

Hello World 2

Hello World 3

Hello World 3

- color:rgb(28, 102, 149)
- background-color:rgb(229, 149, 26)
- height:18.5px
- font-size:16px
- width:327.5px

Exercise : Get the style properties of the element

1. Add events to both buttons that update the classes on the divs as well update the style attribute with the CSS method.
2. Add an event to all the page divs. When clicked it should generate a random color and background color for the element. You can use JavaScript code for a random HEX value '#'+Math.random().toString(16).substring(2,8);
3. Using the CSS method add the random colors to the page element style properties.

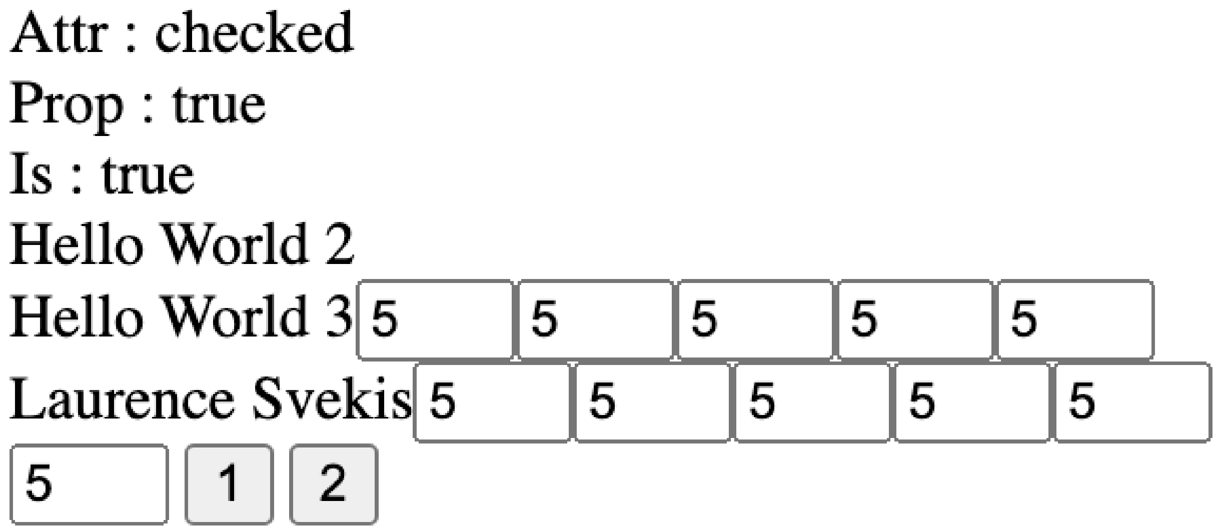
4. Using the CSS method get all the properties, use an array of properties to return an object of values. Output the results into the console and out one of the page elements.
5. Select one of the buttons to increment the width of all the divs by 50px when clicked.

```
<!DOCTYPE html>
<html>
   <head><title>jQuery Course</title>
       <style>
           .red{
               color:red;
               font-size:0.9em;
           }
           .blue{
               background-color:blue;
           }
       </style>
   </head>
   <body>
       <div class="div1">Hello World 1</div>
       <div class="div2">Hello World 2</div>
       <div class="div3" id="div3">Hello World 3</div>
       <div class="div4">Laurence Svekis</div>

       <button id="btn1">1</button>
       <button id="btn2">2</button>
       <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.m
in.js"></script>
       <script src="app11.js"></script>
   </body>
</html>
```

```
$('#btn1').click(()=>{
   $('div').toggleClass('red');
   $('div').css('width','100px');
})
$('#btn2').click(()=>{
   $('div').toggleClass('blue');
   $('div').css('width','+=50');
})
$('div').click(function(){
   console.log($(this).css('color'));
   console.log($(this).css('background-color'));
   const rc = '#'+Math.random().toString(16).substring(2,8);
   const rc2 = '#'+Math.random().toString(16).substring(2,8);
   $(this).css('background-color',rc);
   $(this).css('color',rc2);
   const temp =
$(this).css(['color','background-color','height','font-size','width']);
   let elText = $(this).text();
   let  html = `<h1>${elText}</h1><ul>`;
   for(const prop in temp){
       html += `<li>${prop}:${temp[prop]}</li>`;
   }
   html += '</ul>';
```

```
    output(html);
})
$('.div4').click(()=>{
    output('test');
})

function output(val){
    $('.div4').html(val);
}
```

Getting and setting Element Attributes with jQuery



```
Attr : checked
Prop : true
Is : true
Hello World 2
Hello World 3 5  5  5  5  5
Laurence Svekis 5  5  5  5  5
5  1  2
```

Exercise : Create input checkboxes within the page elements add attributes and set properties of page elements

1. Using a for loop, create checkboxes with attributes. Add a click event on the element that will output the value of the element from the attri of checked, the prop of checked and using is for the checked state. Output the results to the page.

2. Using another loop to create some input checkbox elements, then using the prop() method then set the checked boolean value. Add a click event to track and output the checked results.
3. Add a click event to an element, Update the value counting the clicks on that element.
4. Using detach() remove an element and save it into a global variable. Then on a separate event append the stored detached element to the page again. The events and properties will still be contained within that element.

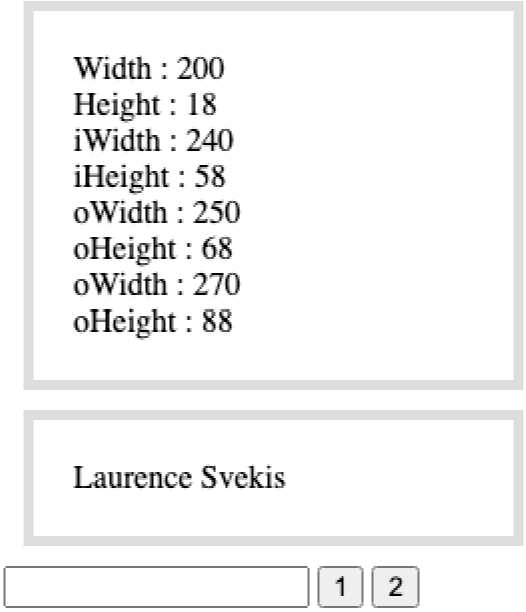
```
<!DOCTYPE html>
<html>
   <head><title>jQuery Course</title>
       <style>
           .red{
               color:red;
               font-size:0.9em;
           }
           .blue{
               background-color:blue;
           }
           .box{
               width:200px;
               height:35px;
               border:1px solid #ddd;
           }
       </style>
   </head>
   <body>
       <div class="div1">Hello World 1</div>
       <div class="div2">Hello World 2</div>
       <div class="div3" id="div3">Hello World 3</div>
```

```
        <div class="div4">Laurence Svekis</div>
        <input >
        <button id="btn1">1</button>
        <button id="btn2">2</button>
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
        <script src="app11.js"></script>
    </body>
</html>
```

```
let counter = 0;
const arr = [];
for(let i=0;i<5;i++){
    //$ele = $('<input>');
    $('<input>').attr({
        type:'checkbox',
        value:i,
        checked:'checked'
    }).appendTo('.div3').click(function(){
        checkerInput($(this));
    });
}
for(let i=0;i<5;i++){
    $ele = $('<input>');
    $ele.attr({
        type:'checkbox',
        value:i
```

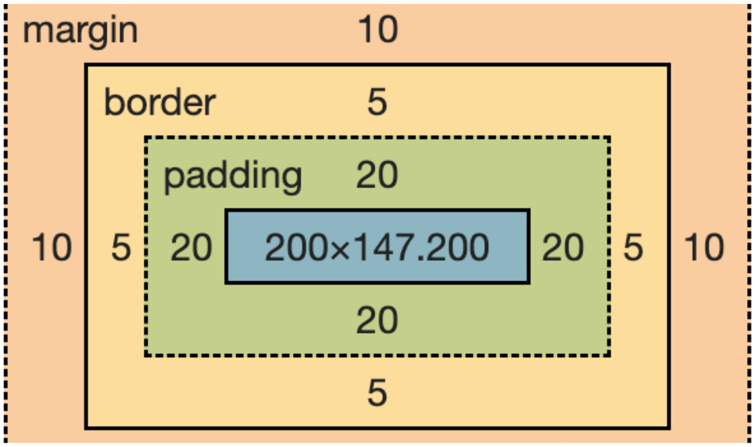
```
   }).appendTo('.div4');
   $ele.prop('checked',true);
   $ele.click(function(){
       checkerInput($(this));
   })
}

function checkerInput($el){
   let temp = '';
   temp += `<div>Attr : ${$el.attr('checked')}</div>`;
   temp += `<div>Prop : ${$el.prop('checked')}</div>`;
   temp += `<div>Is : ${$el.is(':checked')}</div>`;
   $('.div1').html(temp);
}


$('.div2').click(function(){
   counter++;
   $('.div1').html(`<div>Counter ${counter}</div>`);
})
$('div2').click(function(){
   console.log($(this).attr('id'));
   const ran = Math.floor(Math.random()*100);
   $(this).attr('id',`id${ran}`);
})


$('#btn1').click(()=>{
   //$('.div1').addClass('box').empty();
   if(arr.length > 0){
       $ele = arr.shift();
       $('.div4').append($ele);
   }
   $('input').attr({
       type:'number',
       min:'0',
       max:'10',
       value:'5'
   });
})
$('#btn2').click(()=>{
   //$ele = $('.div1').detach();
   //arr.push($ele);
})
```

Element Dimensions and properties values with jQuery Methods



```
Width : 200
Height : 18
iWidth : 240
iHeight : 58
oWidth : 250
oHeight : 68
oWidth : 270
oHeight : 88
```

Laurence Svekis



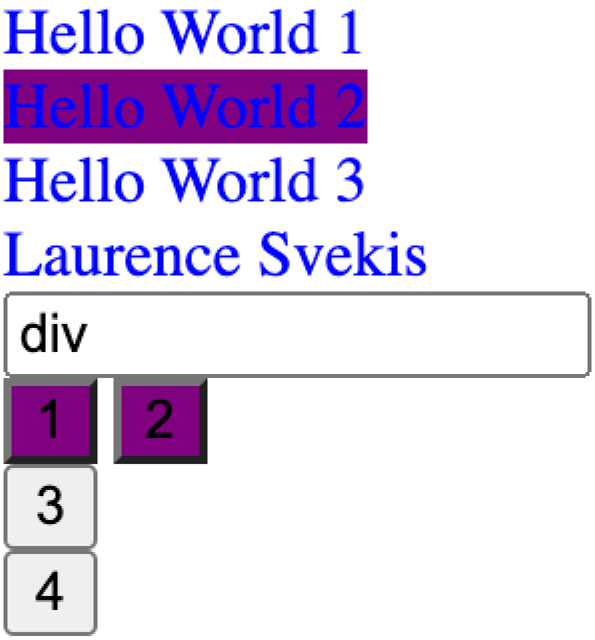
Exercise : Get the dimensions of page elements

1. Add a class to the page elements
2. Add a click to the divs on the page. When the element is clicked show within the element the heights and widths including inner and outer values.
3. Add a click to the button to display in the console the window and document height and width values.

```
<!DOCTYPE html>
<html>
   <head><title>jQuery Course</title>
       <style>
           .box{
               width:200px;
               border:5px solid #ddd;
               padding:20px;
               margin:10px;
           }
       </style>
   </head>
   <body>
       <div class="div1">Hello World 1</div>
       <div class="div2">Hello World 2</div>
       <div class="div3" id="div3">Hello World 3</div>
       <div class="div4">Laurence Svekis</div>
       <input >
       <button id="btn1">1</button>
       <button id="btn2">2</button>
       <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
       <script src="app12.js"></script>
   </body>
</html>
```

```
$('div').addClass('box').click(function(){
    let html = `<div>Width : ${$(this).width()}</div>`;
    html += `<div>Height : ${$(this).height()}</div>`;
    html += `<div>iWidth : ${$(this).innerWidth()}</div>`;
    html += `<div>iHeight : ${$(this).innerHeight()}</div>`;
    html += `<div>oWidth : ${$(this).outerWidth()}</div>`;
    html += `<div>oHeight : ${$(this).outerHeight()}</div>`;
    html += `<div>oWidth : ${$(this).outerWidth(true)}</div>`;
    html += `<div>oHeight : ${$(this).outerHeight(true)}</div>`;
    $(this).html(html);
})

$('#btn1').click(()=>{
    console.log($(document).width());
    console.log($(window).width());
    console.log($(document).height());
    console.log($(window).height());
})
```

Traversing Descendants of Page Elements jQuery Selection



Hello World 1

Hello World 2

Hello World 3

Laurence Svekis

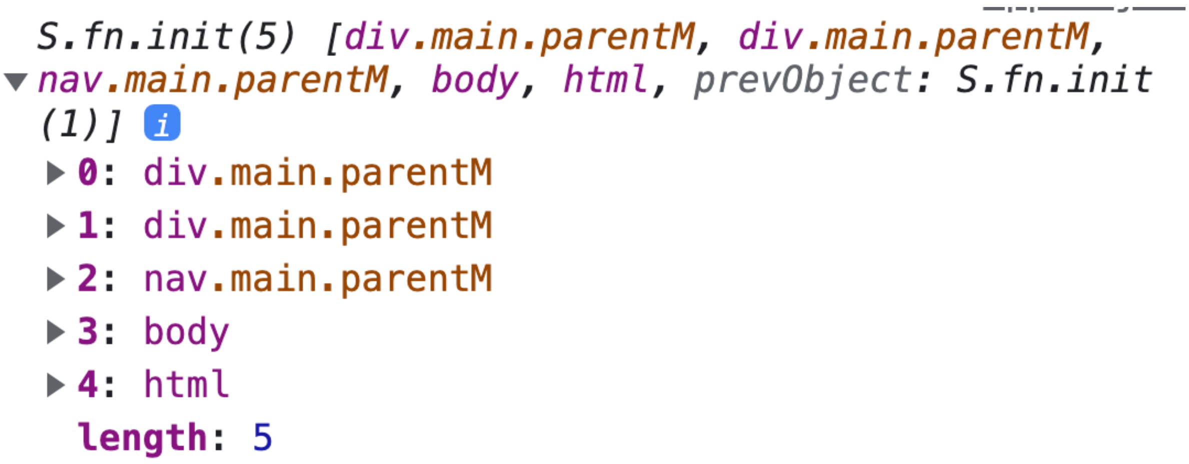
Exercise : Make a selection of an element and then navigate down to the descendants of the selected element.

1. Select an element and output the children of the element into the console.
2. Add an event listener to the first but, when clicked, get the input value and search the main element for that selector element.
3. When the second button is clicked select all the descendants of an element using the wildcard * and find, then apply css styling to the elements.

```
<!DOCTYPE html>
<html>
   <head><title>jQuery Course</title>
       <style>
           .box{
               width:200px;
               border:5px solid #ddd;
               padding:20px;
               margin:10px;
           }
       </style>
   </head>
   <body>
       <div class="main">
           <div xclass="div1">Hello World 1</div>
           <div class="div2"><span>Hello World 2</span></div>
           <div class="div3" id="div3">Hello World 3</div>
           <div class="div4">Laurence Svekis</div>
           <input >
           <div>
           <button id="btn1">1</button>
           <button id="btn2">2</button>
           </div>
           <button id="btn1">3</button>
       </div>
       <button id="btn1">4</button>
       <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
       <script src="app13.js"></script>
   </body>
</html>

//console.log($('.main').children());
```

```
$('#btn1').click(()=>{
   const sel = $('input').val();
   if(sel){
       console.log(sel);
       //$(sel).css('background-color','red');
       $('.main').find(sel).css('color','blue');

   }
})
$('#btn2').click(()=>{
   $('.main div').find('*').css('background-color','purple');
})
```

Traversing Ancestors of Page Elements parent methods with jQuery



```
  S.fn.init(5) [div.main.parentM, div.main.parentM,
▼ nav.main.parentM, body, html, prevObject: S.fn.init
  (1)] ℹ
   ▶ 0: div.main.parentM
   ▶ 1: div.main.parentM
   ▶ 2: nav.main.parentM
   ▶ 3: body
   ▶ 4: html
     length: 5
```

Exercise : Traverse to the selected elements ancestors moving up towards the parent HTML element.

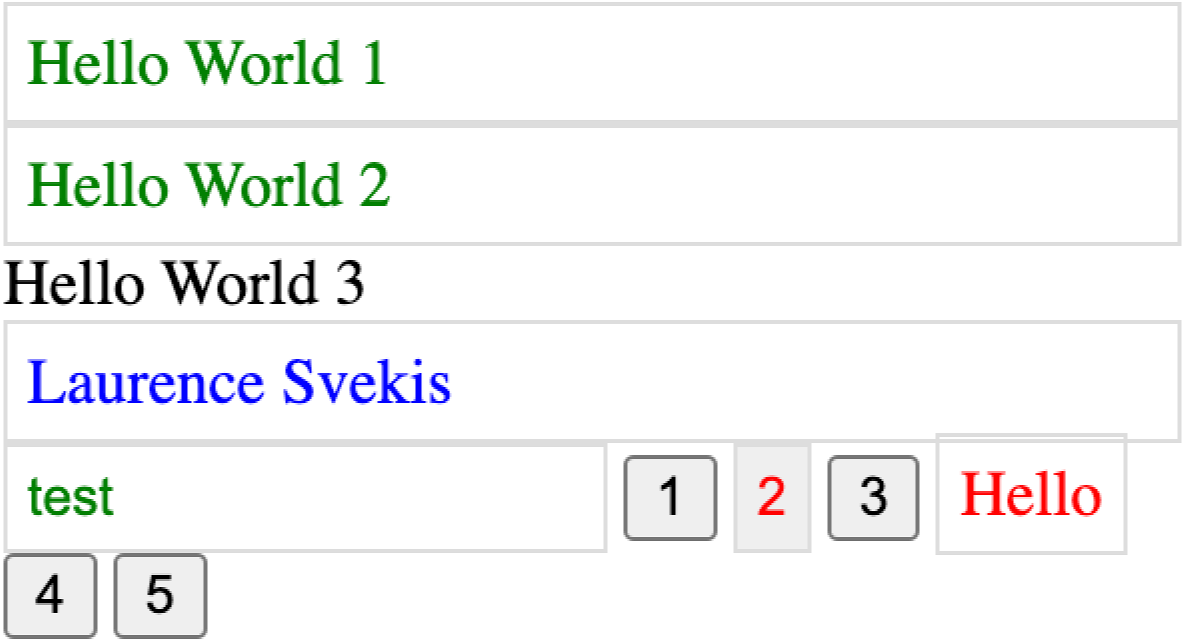
1. Select the button element, when clicked output the length of the parent() and parents().

2. Select the elements that are ancestors of the element using parentsUntil() select a element to stop at.
3. Use the each() to loop through the array of elements. Add a class to all the elements that are ancestors of the second button and have a class of main.

```
<!DOCTYPE html>
<html>
   <head><title>jQuery Course</title>
       <style>
           .box{
               width:200px;
               border:5px solid #ddd;
               padding:20px;
               margin:10px;
           }
       </style>
   </head>
   <body><nav class="main">
       <div class="main">
           <div xclass="div1">Hello World 1</div>
           <div class="div2"><span>Hello World 2</span></div>
           <div class="div3" id="div3">Hello World 3</div>
           <div class="div4">Laurence Svekis</div>
           <input >
           <div class="main">
           <button id="btn1">1</button>
           <button id="btn2">2</button>
           </div>
           <button id="btn1">3</button>
       </div></nav>
       <button id="btn1">4</button>
       <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.m
in.js"></script>
       <script src="app14.js"></script>
   </body>
</html>
```

```
$('#btn1').click(function(){
   console.log($(this).parent().length);
   console.log($(this).parents().length);
   console.log($(this).parents());
   console.log($(this).parents('.main'));
   console.log($(this).parentsUntil('nav'));
})

$('#btn2').click(function(){
   $(this).parents('.main').each(function(){
       $(this).addClass('parentM');
   })
})
```

Traversing Siblings Page Elements and other selections with jQuery



Hello World 1

Hello World 2

Hello World 3

Laurence Svekis

test

Hello

Exercise : Add click events to all the buttons, then apply various sibling methods to traverse to the same level as siblings of the selection element.

1. Apply a class to all siblings of selected element
2. Add to a button the option to remove all element with the class of box
3. On click of a button add a class of box to siblings of an element that have a tag name of div.
4. On a button click use next() nextAll() and nextUntil() methods on page elements apply classes to the elements
5. On a button click use prev() prevAll() and prevUntil() methods on page elements apply classes to the matching elements.

```
<!DOCTYPE html>
<html>

<head>
```

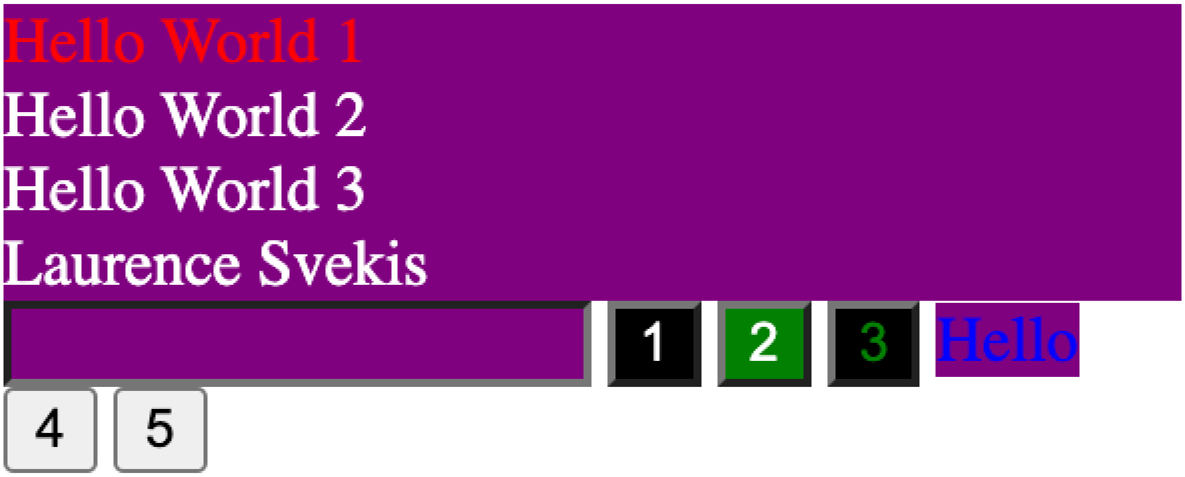
```
    <title>jQuery Course</title>
    <style>
        .box {
            border: 1px solid #ddd;
            padding: 5px;
            color: red;
        }

        .box1 {
            border: 1px solid #ddd;
            padding: 5px;
            color: blue;
        }

        .box2 {
            border: 1px solid #ddd;
            padding: 5px;
            color: green;
        }
    </style>
</head>

<body>
    <nav>
        <div class="main">
            <div xclass="div1">Hello World 1</div>
            <div class="div2"><span>Hello World 2</span></div>
            <div class="div3" id="div3">Hello World 3</div>
```

```
           <div class="div4">Laurence Svekis</div>
           <input>

           <button id="btn1">1</button>
           <button id="btn2">2</button>

           <button id="btn3">3</button>
           <span>Hello</span>
       </div>
   </nav>
   <button id="btn4">4</button>
   <button id="btn5">5</button>
   <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
   <script src="app15.js"></script>
</body>

</html>
```

```
$('#btn1').click(()=>{
   console.log($('#btn1').siblings());
   $sel = $('#btn1').siblings();
   $sel.addClass('box');
})
$('#btn2').click(()=>{
   $('.box').removeClass('box');
})
```

```
$('#btn3').click(()=>{
   $('#btn1').siblings('div').addClass('box');
})
$('#btn4').click(()=>{
   $('#btn1').next().addClass('box');
   $('#btn1').nextAll('span').addClass('box');
   $('#div3').nextUntil('button').addClass('box1');
})
$('#btn5').click(()=>{
   $('#btn1').prev().addClass('box2');
   $('#btn1').prevAll('span').addClass('box2');
   $('#div3').prevUntil('button').addClass('box2');
})
```

Traversing Filtering Page Elements to select them using jQuery Methods



Hello World 1
Hello World 2
Hello World 3
Laurence Svekis

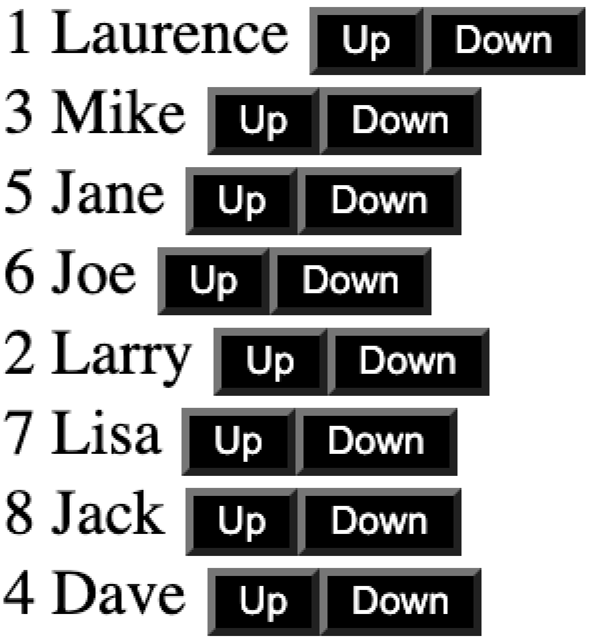
Exercise : Use of filtering to make selection of specific elements and traverse to the result

1. Add click events to all 5 page buttons
2. On click of the first button select an element collection and then traverse to the first() last() and eq() for selecting a specific element using the position in the collection apply styling to the resulting elements.
3. On the click of the second button using filter() to select matching element results apply styling to the resulting elements.
4. On the third button click, select the elements from the collection that do not match the request and apply styling to the resulting elements.
5. On the fourth and fifth buttons, try out variations of filter() and not() to make element selections traversing from the selected element, apply styling to the matching results.

```
<!DOCTYPE html>
<html>

<head>
   <title>jQuery Course</title>
   <style>
```

```
        .box {
            border: 1px solid #ddd;
            padding: 5px;
            color: red;
        }

        .box1 {
            border: 1px solid #ddd;
            padding: 5px;
            color: blue;
        }

        .box2 {
            border: 1px solid #ddd;
            padding: 5px;
            color: green;
        }
    </style>
</head>

<body>
    <nav>
        <div class="main">
            <div class="div1">Hello World 1</div>
            <div class="div2"><span>Hello World 2</span></div>
            <div class="div3" id="div3">Hello World 3</div>
            <div class="div4">Laurence Svekis</div>
            <input>
            <button id="btn1">1</button>
            <button id="btn2">2</button>

            <button id="btn3">3</button>
            <span>Hello</span>
        </div>
    </nav>
    <button id="btn4">4</button>
    <button id="btn5">5</button>
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
    <script src="app16.js"></script>
</body>

</html>

$('#btn1').click(()=>{
    console.log($('.main').children());
    $main = $('.main').children();
    $main.first().css('color','red');
    $main.last().css('color','blue');
    $main.eq(7).css('color','green');
    $main.eq(6).css('background-color','green');
})

$('#btn2').click(()=>{
    $main = $('.main').children();
    $btns = $main.filter('button');
    $main.filter('button').css('background-color','black');
    $btns.css('color','white');
})

$('#btn3').click(()=>{
    $main = $('.main').children();
    $btns = $main.not('button');
    $main.not('button').css('background-color','purple');
    $btns.css('color','white');
})

$('#btn4').click(()=>{
    $('.main div').not('.div1').css('color','red');
})

$('#btn5').click(()=>{
    $('.main div').filter('div').css('color','green');
})
```

How to create a Dynamic List with jQuery project



1 Laurence Up Down
3 Mike Up Down
5 Jane Up Down
6 Joe Up Down
2 Larry Up Down
7 Lisa Up Down
8 Jack Up Down
4 Dave Up Down

Exercise : Dynamic page elements with jQuery

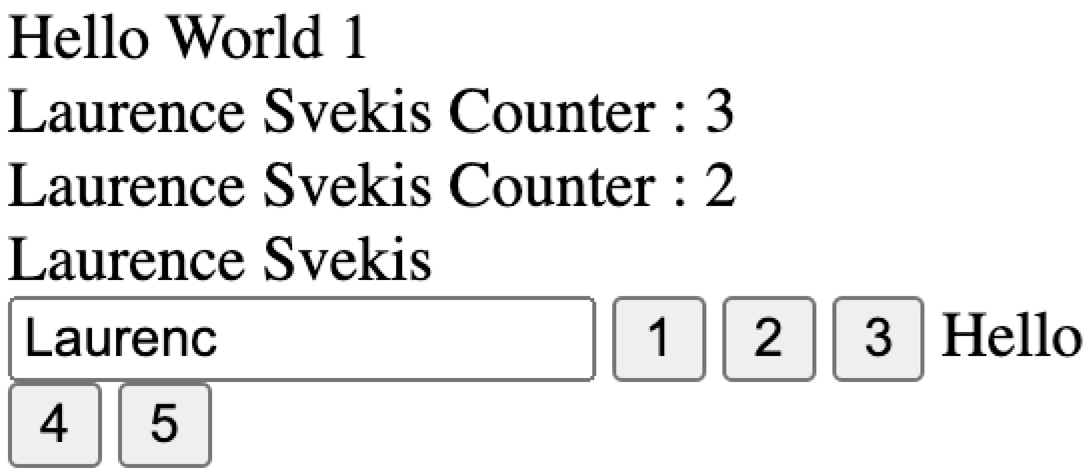
1. Create an array of string values
2. Loop through the array items using $.each() In the callback function add the value as divs to the page, add a class of box.
3. Select all elements with a class of ‘box’. Using each loop through the items
4. Select the text() value of the div element, and store it as a variable value. Then use empty() to clear contents of the element.
5. Create a span with the text content from the original element, append the span into the element.
6. Add 2 buttons into the element, one that has text of Up and the other Down.
7. Add a click event to the first button, selec the prev() element to the main parent of the button.
8. add a condition if the length of the element is greater than zero, then insert the main element before the prev() with insertBefore()

9. Add a click event to the second button to move the main element down. Do the same except use the next() sibling to the parent, and use insertAfter() to add the selected parent.
10. Apply some animations and styling to the element for better visuals of the movement.

```
<!DOCTYPE html>
<html>
   <head><title>jQuery Course</title>
       <style>
           .btn{

               background-color:black;
               color:white;
               font-size:0.6em;
           }
           .output span{
               margin : 5px;
           }
           .active{
               color : red;
           }
       </style>
  </head>
   <body>
       <div class="output"></div>
       <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.m
in.js"></script>
       <script src="exe2.js"></script>
   </body>
</html>

const arr =
['Laurence','Larry','Mike','Dave','Jane','Joe','Lisa','Jack'];
$.each(arr,(ind,val)=>{
   console.log(ind,val);
   let html = `${ind+1} ${val}`;
   $('<div>').html(html).addClass('box').appendTo('.output');
})
$('.box').each(function(){
   $ele = $(this);
   $par = $ele.parent();
   console.log($par);
   const temp = $ele.text();
   $ele.empty();
   $('<span>').text(temp).appendTo($ele);
   $btn1 =
$('<button>').text('Up').addClass('btn').appendTo($ele);
   $btn2 =
$('<button>').text('Down').addClass('btn').appendTo($ele);
   $btn1.click(function(){
       $sel = $(this).parent().prev();

       if($sel.length > 0){
           console.log($sel);
           $(this).parent().addClass('active');
           $(this).parent().slideUp('slow',function(){
               $(this).removeClass('active');
               $(this).insertBefore($sel);
               $(this).slideDown('fast');
           })

       }
   })
   $btn2.click(function(){
       $sel = $(this).parent().next();
       if($sel.length > 0){
           console.log($sel);
           console.log($sel);
           $(this).parent().addClass('active');
           $(this).parent().slideUp('slow',function(){
               $(this).removeClass('active');
               $(this).insertAfter($sel);
               $(this).slideDown('fast');
           })
       }
   })
})
```

More jQuery events and helpful methods

jQuery Data Method save values into the element object



```
Hello World 1
Laurence Svekis Counter : 3
Laurence Svekis Counter : 2
Laurence Svekis
Laurenc 1 2 3 Hello
4 5
```

Exercise : Store a value of the text() from the clicked element which can then be restored from the data() of the element

1. Add a click event to all the divs
2. When the div is clicked get the current text() value and store it to the data()
3. Update the text() of the element
4. Add events to the buttons to use the data value of the element and output the data() value as the new text() of the element

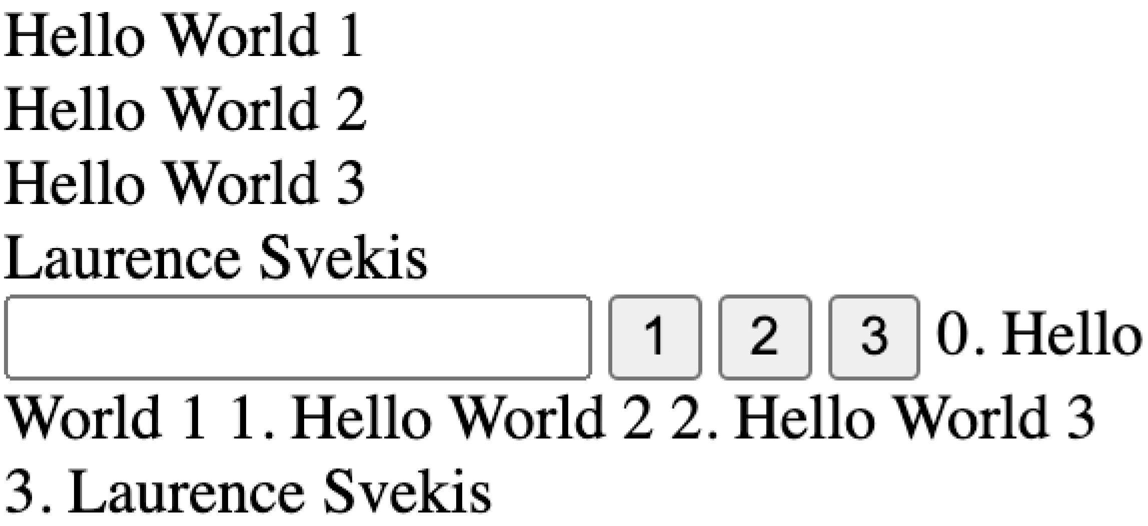
```
<!DOCTYPE html>
<html>
```

```
<head>
    <title>jQuery Course</title>
</head>

<body>
    <nav>
        <div class="main">
            <div class="div1">Hello World 1</div>
            <div class="div2"><span>Hello World 2</span></div>
            <div class="div3" id="div3">Hello World 3</div>
            <div class="div4">Laurence Svekis</div>
            <input>

            <button id="btn1">1</button>
            <button id="btn2">2</button>

            <button id="btn3">3</button>
            <span>Hello</span>
        </div>
    </nav>
    <button id="btn4">4</button>
    <button id="btn5">5</button>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
    <script src="app17.js"></script>
</body>
```

```
</html>
let counter = 0;
let temp = '';
$('input').val('Laurence Svekis');
$('.main div').click(function(){
   $(this).data('past',$(this).text());
   counter++;
   const temp = $('input').val();
   const html = `${temp} Counter : ${counter}`;
   $(this).html(html);
})
$('#btn1').click(()=>{
   const val = $('.div1').data('past');
   $('.div1').text(val);
   console.log(val);
})
$('#btn2').click(()=>{
   const val = $('.div2').data('past');
   $('.div2').text(val);
   console.log(val);
})
$('#btn3').click(()=>{
   const val = $('.div3').data('past');
   $('.div3').text(val);
   console.log(val);
})
$('#btn4').click(()=>{
   const val = $('.div4').data('past');
```

```
    $('.div4').text(val);
    console.log(val);
})
```

jQuery get Method to get the index value of a page element



Hello World 1
Hello World 2
Hello World 3
Laurence Svekis

0. Hello World 1 1. Hello World 2 2. Hello World 3 3. Laurence Svekis

Exercise : Use the get() method to return the DOM element object

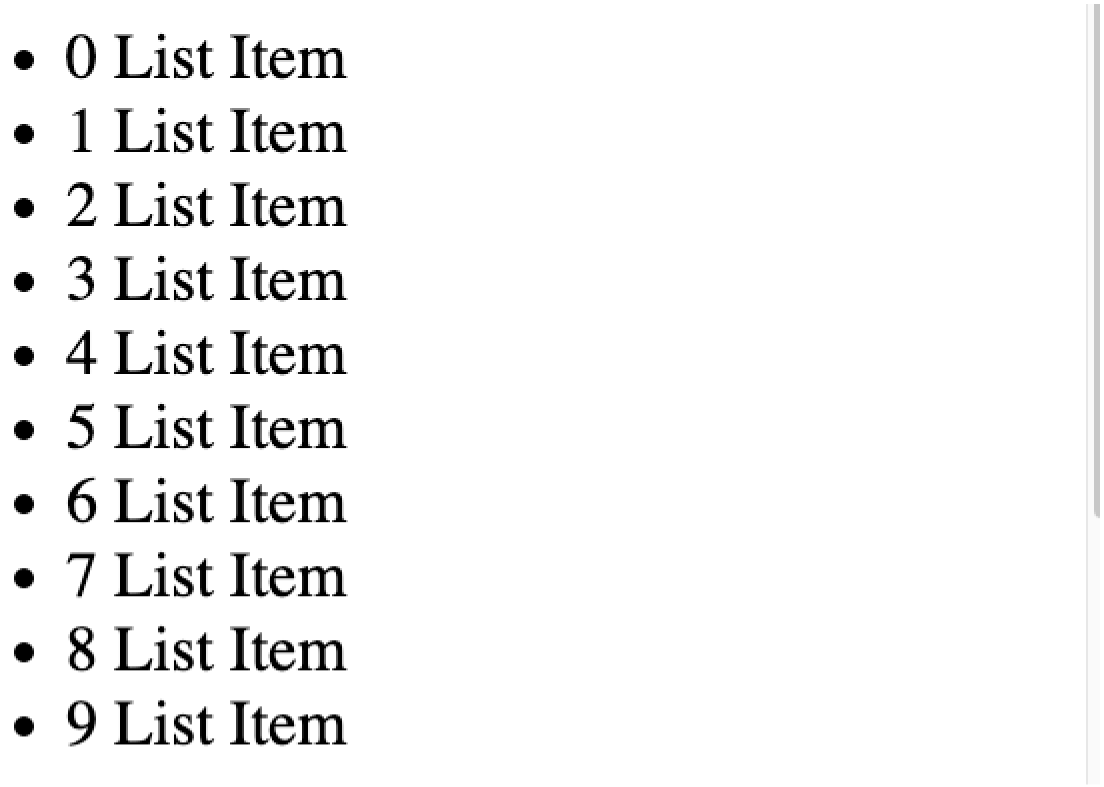
1. Select a jQuery set of elements with the selector
2. Use the get() and an index value of the element you want to interact with. Then use JavaScript DOM method and properties to get values from the element
3. Create a click on the button that will get the length of items in the jQuery selection, then loop through the items and use get the iterator value as the index returns the element. Get the textContent value of the element and add it to the page with jQuery.

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

```html
<head>
   <title>jQuery Course</title>
</head>

<body>
   <nav>
       <div class="main">
           <div class="div1">Hello World 1</div>
           <div class="div2"><span>Hello World 2</span></div>
           <div class="div3" id="div3">Hello World 3</div>
           <div class="div4">Laurence Svekis</div>
           <input>

           <button id="btn1">1</button>
           <button id="btn2">2</button>

           <button id="btn3">3</button>
           <span class="span1">Hello</span>
       </div>
   </nav>
   <button id="btn4">4</button>
   <button id="btn5">5</button>
   <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.m
in.js"></script>
   <script src="app18.js"></script>
</body>
```

```
</html>

$('#btn1').click(()=>{
   const temp = $('.main div').get(1);
   console.log(temp.innerHTML);
   console.log(temp.textContent);
})

$('#btn2').click(()=>{
   const len = $('.main div').length;
   console.log(len);
   let html = '';
   for(let i=0;i<len;i++){
       const temp = $('.main div').get(i);
       console.log(temp.textContent);
       html += `${i}. ${temp.textContent} `;
   }
   $('.main .span1').text(html);
})
```

jQuery index of page element index Method



- 0 List Item
- 1 List Item
- 2 List Item
- 3 List Item
- 4 List Item
- 5 List Item
- 6 List Item
- 7 List Item
- 8 List Item
- 9 List Item

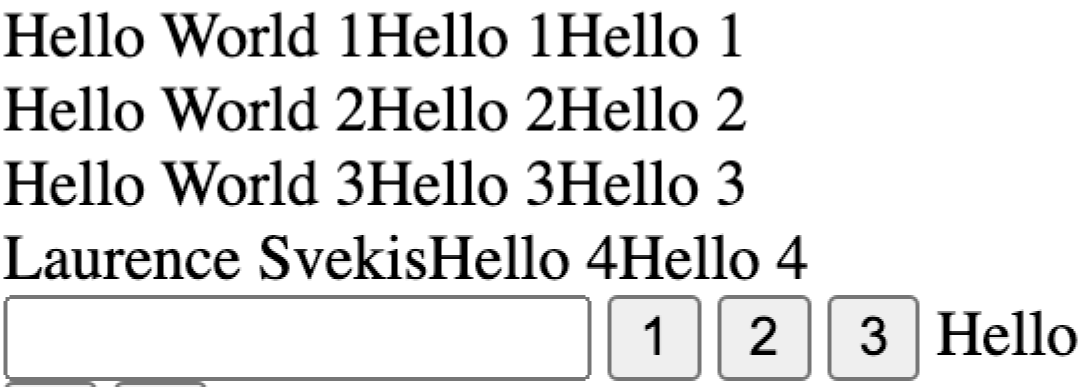
Exercise : Make a selection of page elements with jQuery , return the index() value of the selected element.

1. On the click of the button output the elements index value into the console.
2. Add click events to all the page divs within the main class. Get the index value and add it to the element text()
3. Create an UL adding LI items to the UL. Add Click events to the LI and output their index() value into the console.

```
<!DOCTYPE html>
<html>

<head>
   <title>jQuery Course</title>
</head>
```

```
<body>
   <nav>
       <div class="main">
           <div class="div1">Hello World 1</div>
           <div class="div2"><span>Hello World 2</span></div>
           <div class="div3" id="div3">Hello World 3</div>
           <div class="div4">Laurence Svekis</div>
           <input>

           <button id="btn1">1</button>
           <button id="btn2">2</button>

           <button id="btn3">3</button>
           <span class="span1">Hello</span>
       </div>
   </nav>
   <button id="btn4">4</button>
   <button id="btn5">5</button>
   <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.m
in.js"></script>
   <script src="app19.js"></script>
</body>

</html>

$('#btn1').click(function(){
   console.log($(this).index());
})
$('.main div').click(function(){
   const val = $(this).index();
   const temp = $(this).text();
   $(this).text(`${val} ${temp}`);
})
const $ul = $('<ul>').prependTo('nav');
for(let i=0;i<10;i++){
   $('<li>').text(`${i} List Item`).appendTo($ul).click(function(){
       console.log($(this).index());
   })
}
```

jQuery toArray Method to get the jQuery elements as DOM elements



Hello World 1Hello 1Hello 1
Hello World 2Hello 2Hello 2
Hello World 3Hello 3Hello 3
Laurence SvekisHello 4Hello 4

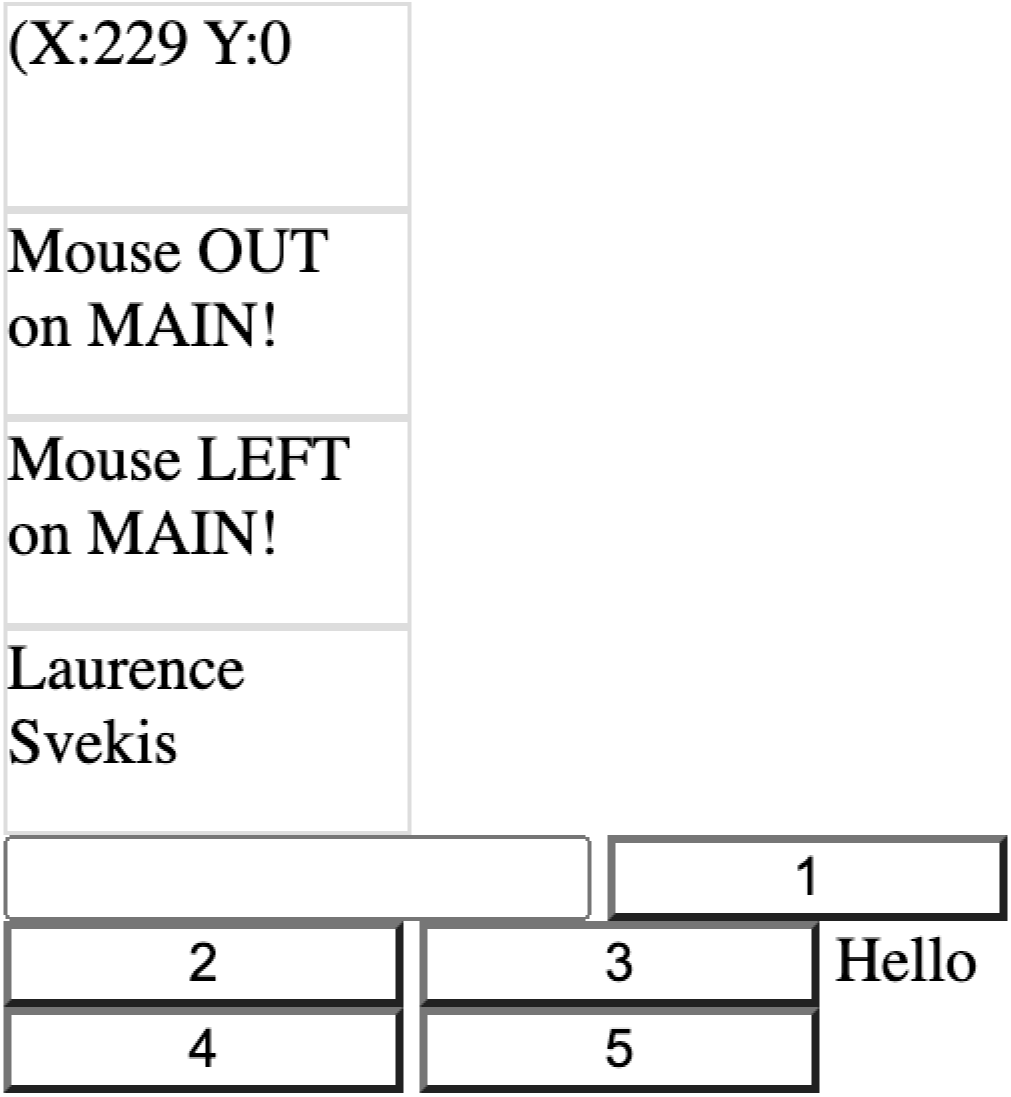
Exercise : Select the jQuery elements as DOM elements with toArray()

1. Use the jQuery selector to make a selection of elements
2. Convert them to DOM elements with toArray()
3. Using JavaScript methods get the DOM elements and make updates to them

```
<!DOCTYPE html>
<html>


<head>
   <title>jQuery Course</title>
</head>


<body>
   <nav>
       <div class="main">
           <div class="div1">Hello World 1</div>
           <div class="div2"><span>Hello World 2</span></div>
           <div class="div3" id="div3">Hello World 3</div>
           <div class="div4">Laurence Svekis</div>
           <input>

           <button id="btn1">1</button>
           <button id="btn2">2</button>

           <button id="btn3">3</button>
           <span class="span1">Hello</span>
       </div>
   </nav>
   <button id="btn4">4</button>
   <button id="btn5">5</button>
   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
   <script src="app20.js"></script>
</body>

</html>
```

```
$('#btn1').click(()=>{
   const eles = $('.main div').toArray();
   console.log(eles);
   eles.forEach((ele,ind)=>{
       console.log(ele.textContent);
       const span = document.createElement('span');
       span.textContent = `Hello ${ind+1}`;
       ele.append(span);
   })
})
```

jQuery Mouse Move Events Listeners and hover Events



(X:229 Y:0

Mouse OUT on MAIN!

Mouse LEFT on MAIN!

Laurence Svekis

Exercise : attach various mouse movement events to page elements.

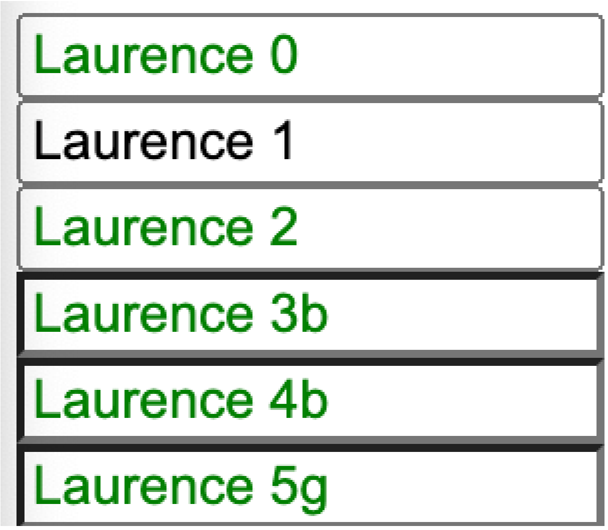
1. Add the hover() event to the divs within the main class. On enter update the background color to red and then on leave set it back to white.
2. Add mouseup() and mousedown() to the divs updating the text and the color of the background on the different mouse press states.
3. Add mouseout() and mouseleave() to the main element, check the difference
4. Add mouseenter() and mouseover() to the main element, check the difference

5. Add mouseenter() and mouseleave() to the buttons, to update the background color just like the hover() event.
6. On the main document track the mousemove() from the event object, get the values of pageX and pageY property values and then output them into a page element.

```
<!DOCTYPE html>
<html>

<head>
    <title>jQuery Course</title>
    <style>
        .box{
            border:1px solid #ddd;
            height:50px;
            width:100px;
        }
    </style>
</head>

<body>
    <nav>
        <div class="main">
            <div class="div1">Hello World 1</div>
            <div class="div2"><span>Hello World 2</span></div>
            <div class="div3" id="div3">Hello World 3</div>
            <div class="div4">Laurence Svekis</div>
            <input>

            <button id="btn1">1</button>
            <button id="btn2">2</button>

            <button id="btn3">3</button>
            <span class="span1">Hello</span>
        </div>
    </nav>
    <button id="btn4">4</button>
    <button id="btn5">5</button>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
    <script src="app21.js"></script>
</body>

</html>
```

```
$('.main div').addClass('box').hover(function(){
    $(this).css('background-color','red')},function(){
    $(this).css('background-color','white')
}).mouseenter(function(e){
    //$(this).text(`(X:${e.pageX} Y:${e.pageY}`);
})
$('.main div').mouseup(function(){
    $(this).text('MOUSE UP');
    $(this).css('background-color','blue');
}).mousedown(function(){
    $(this).text('MOUSE DOWN');
    $(this).css('background-color','green');
})
$('.main').mouseout(()=>{
   $('.div2').text('Mouse OUT on MAIN!');
})
$('.main').mouseover(()=>{
   $('.div2').text('Mouse OVER on MAIN!');
})
$('.main').mouseleave(()=>{
   $('.div3').text('Mouse LEFT on MAIN!');
})
$('.main').mouseenter(()=>{
   $('.div3').text('Mouse ENTER on MAIN!');
})
$('button').css('width','100px');
$('button').mouseenter(function(e){
   $(this).css('background-color','red');

})
$('button').mouseleave(function(){
   $(this).css('background-color','white');
})
$(document).mousemove(function(e){
   $('.div1').text(`(X:${e.pageX} Y:${e.pageY}`);
})
```

jQuery to listen for keyboard Events and get values from the event object



Laurence 0

Laurence 1

Laurence 2

Laurence 3b

Laurence 4b

Laurence 5g

Exercise :

1. Using a loop create several input fields add them all to main
2. On the input field add keydown() method update the CSS background color and output the event method into the console including the element index() value.
3. On the input field add keypress() method update the CSS background color and output the keypress into the console including the element index() value.
4. On the input field add focus() and blur() methods update the CSS background color and output the event methods into the console including the element index() value.
5. On the input field add keyup() method update the CSS background color and output the event method into the console including the element index() value.
6. On the input field add focusin() and focusout() methods which will include the children, then update the CSS background color and output the event method into the console including the element index() value.

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

```
    <title>jQuery Course</title>
    <style>
        .box {
            border: 1px solid #ddd;
            height: 50px;
            width: 100px;
        }
        input{
            display:block;
        }
    </style>
</head>
<body>
    <div class="main"></div>
    <button id="btn1">1</button>
    <button id="btn2">2</button>
    <button id="btn3">3</button>
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
    <script src="app22.js"></script>
</body>

</html>

for(let i=0;i<10;i++){
    const html = `Laurence ${i}`;
    adder(html);
```

```
}

$('input').keydown(function(e){
   console.log(e.key,e.code);
   $(this).css('background-color','yellow');
   console.log('keydown '+$(this).index());
})
$('input').keypress(function(e){
   console.log(e.key,e.code);
   $(this).css('background-color','red');
   console.log('keypress '+$(this).index());
})
$('input').keyup(function(e){
   console.log(e.key,e.code);
   $(this).css('background-color','white');
   console.log('keyup '+$(this).index());
})
$('input').focus(function(e){
   $(this).css('color','red');
   console.log('focus '+$(this).index());
})
$('input').blur(function(e){
   $(this).css('color','green');
   console.log('blur '+$(this).index());
})
$('input').focusin(function(e){
   $(this).css('color','blue');
   console.log('focusIn '+$(this).index());
})
$('input').focusout(function(e){
    $(this).css('color','pink');
    console.log('focusOut '+$(this).index());
})
function adder(val){
    $('<input>').attr({
        type:'text',
        value:val
    }).appendTo('.main');
}
```

jQuery form Events on submit and more

Exercise : Attach events to form elements and form submission

1. Add a select element with options to the HTML page
2. On change of the select output the value to the element with the class of output.
3. Select the input text type element, and on change event output the value of the input field to the output element
4. On input field select of text add a select() method event to the element, output the active element text into the output element.
5. On the form submission, add an event which says in the console that the form submitted, as well add a preventDefault() to prevent the default action of form submission.
6. On a button add when clicked add within the callback function that the main form is submitted.

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

```
    <title>jQuery Course</title>
    <style>
        .box {
            border: 1px solid #ddd;
            height: 50px;
            width: 100px;
        }
        input{
            display:block;
        }
    </style>
</head>
<body>
    <div class="output"></div>
    <form class="main">
        <input type="text" name="name" value="Laurence Svekis">
        <input type="submit" value="Submit">
    </form>
    <button id="btn1">1</button>
    <button id="btn2">2</button>
    <button id="btn3">3</button>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
    <script src="app23.js"></script>
</body>

</html>
```

```
$('<select>').appendTo('.main');
for(let i=0;i<5;i++){
   const val = `${i+1} Value`;
   $('<option>').val(val).text(val).appendTo('select');
}
$('select').change(function(){
   $('.output').text($(this).val());
})

$('input[type="text"]').change(function(){
   $('.output').text($(this).val());
})

$(':input').select(function(){
   $('.output').text($(this).val());
})

$('.main').submit(function(e){
   e.preventDefault();
   console.log('form submitted');
})

$('#btn1').click(()=>{
   $('.main').submit();
})
```

Attach events with the on Method with more powerful events

Exercise : use of on() method for adding events to future elements created after the event code is run.

1. Create an element with a class of box, add it to the start at the top of the page elements.
2. Add multiple events the the element with a class of box, click() mouseenter() and mouseleave()
3. Add to all the button elements, within the .btns class element, the click event with the on() method. Once clicked it should output the element index value to the text area of the element with the class of output.
4. Create an object with some data, then using the on() method, add a click event including the data to be sent to the callback function of adder.
5. In the adder() function output the event object data into the console.
6. Use the short hand click on the button elements, add when clicked that the background changes to red and the element index value is output into the console log.
7. Create page buttons and append them to the btns class element. The buttons created after the click() event was added will not have the click event.

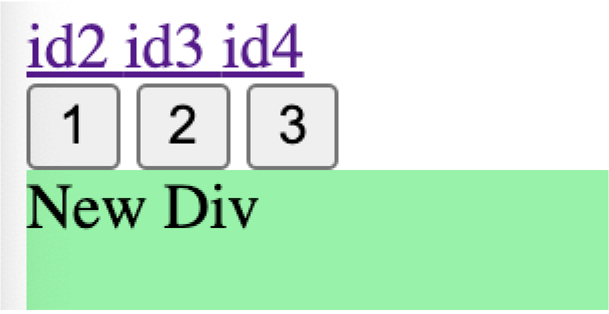
```
<!DOCTYPE html>
<html>
<head>
   <title>jQuery Course</title>
   <style>
       .box {
           border: 1px solid #ddd;
           height: 50px;
           width: 100px;
       }
```

```
        input{
            display:block;
        }
        .active{
            background-color:red;
            color:green;
        }
    </style>
</head>
<body>
    <div class="output"></div>
    <div class="btns">
        <button id="btn1">1</button>
        <button id="btn2">2</button>
        <button id="btn3">3</button>
    </div>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
    <script src="app24.js"></script>
</body>

</html>
```

```
$('<div>').text('LaurenceSvekis').addClass('box').prependTo('body');

$('.box').on({
   click : function() {$(this).toggleClass('active')},
   mouseenter:function(){$(this).css('background-color','blue')}
   ,

mouseleave:function(){$(this).css('background-color','white')}
})

$('.btns').on('click','button',function(){
   const val = `${$(this).index()} index button`;
   $('.output').text(val);
})
const obj = {
   first : 'Laurence',
   last : 'Svekis',
   id : 100
};
$('.output').on('click',obj,adder);

function adder(e){
   console.log(e.data);
}



$('.btns button').click(function(){
   const val = `${$(this).index()} index button`;
   $(this).css('background-color','red');
   console.log(val);
})
```

```
for(let i=0;i<5;i++){
   const temp = `${i+4}`;
   $('<button>').text(temp).appendTo('.btns');
}
```

jQuery scroll event on Browser Events and window events



Exercise : Add elements and connect the anchor tag to the elements with id, add event of scroll and resize to the Window object.

1. When the button is clicked, create a random value to be used for height, and a random color to be used for background color of a new element. Create a new div, set the css() add an attribute of id and append it to the body of the page.
2. Add a anchor tag, with a hyperlink to the # + element id and append this to the element with a class of output.
3. Add a scroll() event to the window and a resize() to the window. Output those events to the console.

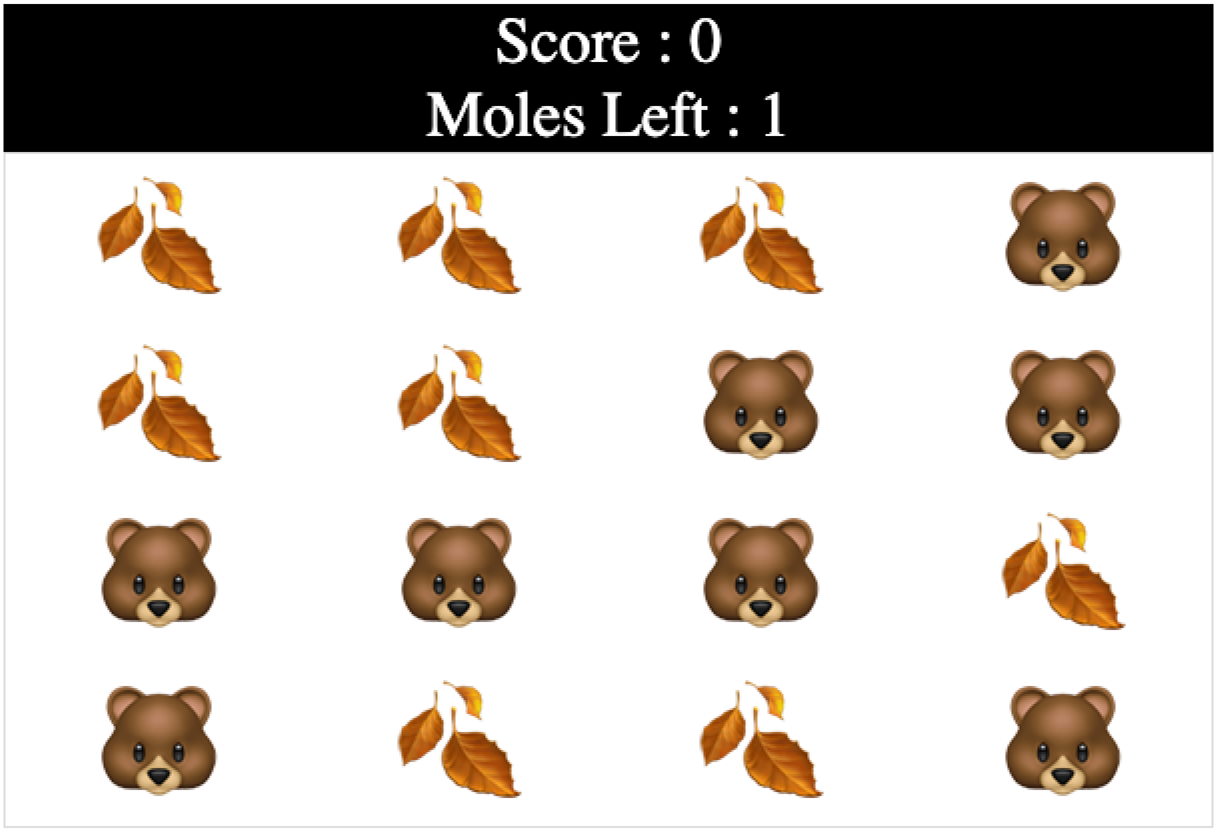
```
<!DOCTYPE html>
<html>
```

```
<head>
    <title>jQuery Course</title>
    <style>
        .box {
            border: 1px solid #ddd;
            height: 50px;
            width: 100px;
        }
        input{
            display:block;
        }
        .active{
            background-color:red;
            color:green;
        }
    </style>
</head>
<body>
    <div class="output"></div>
    <div class="btns">
        <button id="btn1">1</button>
        <button id="btn2">2</button>
        <button id="btn3">3</button>
    </div>
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
    <script src="app25.js"></script>
```

```
</body>

</html>
```

```
let counter = 1;
$('button').click(()=>{
   counter++;
   const r = Math.floor(Math.random()*200)+100;
   const c = '#'+Math.random().toString(16).substring(2,8);
   const id = 'id'+counter;
   $('<div>').text('New
Div').css('height',r+'px').attr('id',id).css('background-color',
c).appendTo('body');
   //output
   $('<a>').attr('href','#'+id).text(id+'
').appendTo('.output');
})
$(window).scroll((e)=>{
   console.log(e);
})
$(window).resize((e)=>{
   console.log($(window).height());
   console.log($(document).height());
})
```

How to create a jQuery Mole hit game



Score : 0

Moles Left : 1

Objective of the game is to hit the moles, they will appear and disappear randomly. Click the mole to score, continue the game until the round is over and the number of moles has reached 0.

Exercise : Build dynamic and interactive game play with jQuery

1. Use the HTML and CSS to setup the basic game play area
2. Setup a global game object of data that tracks the score, gameover boolean, last hole, the number of holes for the game, the end of the moles countdown, and the total of mole to show in the round.
3. Build the game board with holes for the number of elements to show in the game play area, within the hole add the elements with the mole, the leaves, and the hit icons. Give them classes so that they can be selected. Create a unique hole class so that each hole can be randomly selected from the board.
4. Add to the start button to start the game function.
5. In the start game function hide the start button. Set the game score, game end counter, and game over boolean to starting values. This will allow the player to start again once

the countdown of moles is complete. Launch a function that will start the random moles to show.

6. Create a function to output the score and countdown of the moles left. Invoke the function to update the game score.
7. In the starter() function use the selector to select an element, use a function to generate the random hole value to select.
8. In the random hole value generator, get a random number and check to see if the new value is not the same as the last value, if it is to generate a new number.
9. Once you have the random element hole selected, send it into a new function to showMole(). This will be the hole that the mole will show in.
10. In the showMole() function subtract the countdown of moles, update the score. If the countdown is less than or equal to zero then show the start button.
11. If the countdown is greater than zero, then generate a random value for a timer. This can be used to set a random time in milliseconds in the timeout function.
12. From the hole element, hide the dirt element, show the mole in the element.
13. Set a setTimeout() which in the callback shows the dirt element, hides the ‘mole’ and ‘hit’ class elements. Also checks if the game is over, and if it is not then invokes starter() to continue the loop.

```
<!DOCTYPE html>
<html>
<head>
   <title>jQuery Exercise Game </title>
   <style>
       *{
           box-sizing:border-box;
       }
       .game{
           margin:auto;
           padding:0;
           max-width:800px;
           border:1px solid #ddd;
```

```
}
.hole{
    text-align:center;
    width:25%;
    display:inline-block;
    font-size:4em;
}
.mole{

    cursor:crosshair;
}
.hit{
    display:none;
}
.dirt{
    display:none;
    cursor:grab;
}
#start{
    padding:10px;
    font-size:2em;
    margin:auto;
    display:block;
}
.message{
    text-align:center;
    background-color:black;
    color:white;
```

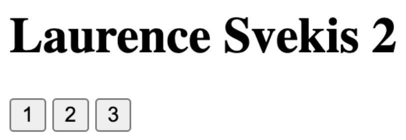
```
           font-size:2em;
       }
   </style>
</head>
<body>
   <div class="message"></div>
   <div class="game"></div>
   <button id="start">Start Game !</button>
   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
   <script src="exe3.js"></script>

</body>
</html>
```

```
const game = {score:0,gameover:true,last:0,holes:16,ender:10,total:10};
makeGameBoard();
$('.game').on('click','.mole',hitMole);
$('#start').click(startGame);

function hitMole(){

$(this).hide().parent().find('.hit').show().fadeOut(200,function(){
       $(this).parent().find('.dirt').show();
   })
```

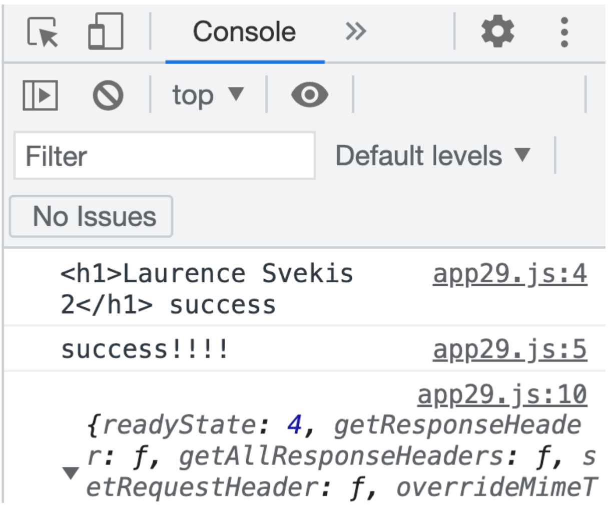
```
    game.score++;
    mes();
}

function mes(){
    let html = `<div>Score : ${game.score}</div><div>Moles Left :
${game.ender}`;
    $('.message').html(html);
}

function starter(){
    const $ele = $('.hole'+ranHole());
    showMole($ele);
}

function showMole($ele){
    game.ender--;
    mes();
    if(game.ender <= 0){
        game.gameover =  true;
        $('#start').show();
    } else{
    const timer = Math.round(Math.random()*1000)+200;
    $ele.find('.dirt').hide(0,()=>{
        $ele.find('.mole').show();
    })
    setTimeout(()=>{
        $ele.find('.dirt').show();
        $ele.find('.mole').hide();
        $ele.find('.hit').hide();
        if(!game.gameover) starter();
    },timer);}
}

function ranHole(){
    const val = Math.floor(Math.random()*game.holes);
    if(val == game.last){
        return ranHole();
    }
    game.last = val;
    return val;
}

function startGame(){
    $('#start').hide();
    game.gameover = false;
    game.score = 0;
    game.ender = game.total;
    starter();
    mes();
}

function makeGameBoard(){
    for(let i=0;i<game.holes;i++){
        $div = $('<div>');
        const temp = `hole${i}`;
        $div.addClass('hole').addClass(temp).appendTo('.game');
        $('<div>').addClass('mole').html('🐻').appendTo($div);
        $('<div>').addClass('dirt').html('🍂').appendTo($div);
        $('<div>').addClass('hit').html('☠️').appendTo($div);
    }
}
```

AJAX requests with jQuery JSON POST GET data from external sources

Load a file with jQuery load method



Laurence Svekis 2



```
<h1>Laurence Svekis          app29.js:4
2</h1> success
success!!!!                  app29.js:5
                            app29.js:10
  {readyState: 4, getResponseHeade
  r: f, getAllResponseHeaders: f, s
▼ etRequestHeader: f, overrideMimeT
```

Exercise : AJAX example connects to an external file and loads the contents directly into a page element.

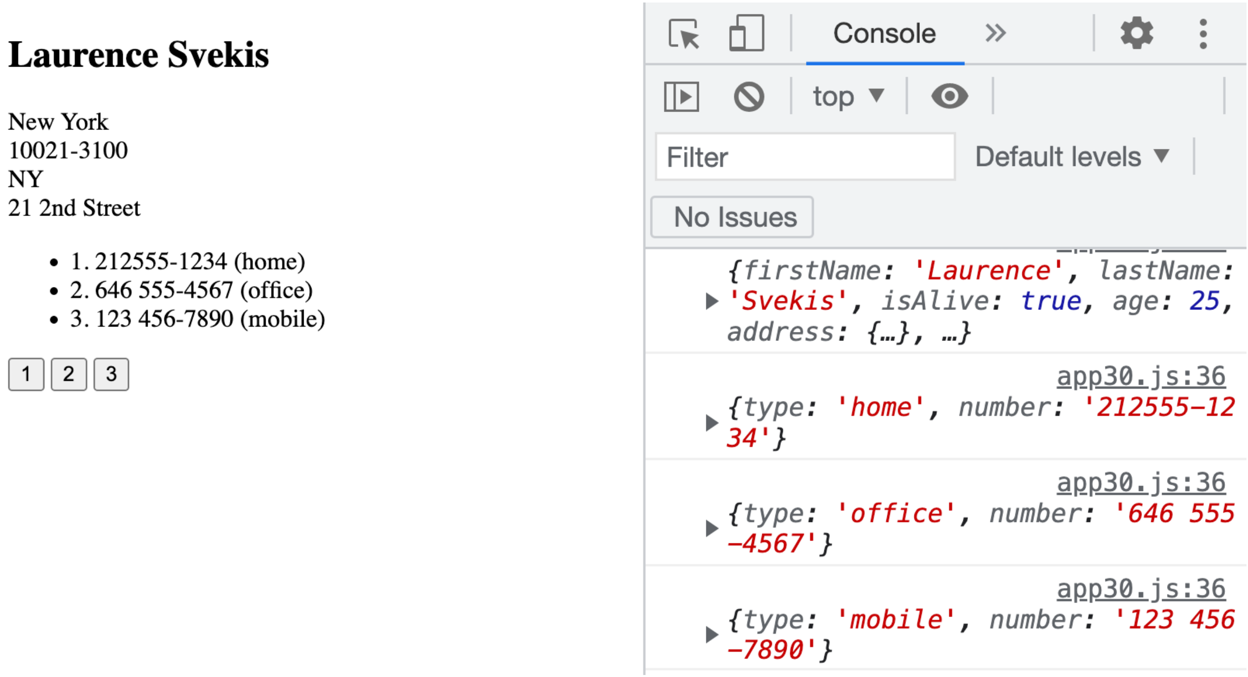
1. Create several text and html files with content that you want to load directly into your page elements.
2. Using a click event on the button, and the load() method select the element and in the load parameters add a string value of the path to the file.
3. Optional callback function, can return the response text, the response status and the entire xHR object which can then be used within the callback function. Check the status, if the status returns ‘success’ then load a message into the console.

```
<!DOCTYPE html>
<html>
<head>
   <title>jQuery Course</title>
```

```
</head>
<body>
   <div class="output"></div>
   <div class="btns">
       <button id="btn1">1</button>
       <button id="btn2">2</button>
       <button id="btn3">3</button>
   </div>
   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
   <script src="app30.js"></script>
</body>

</html>

$('#btn1').click(()=>{
   $('.output').load('file29.txt',(rep,status,xhr)=>{
       if(status == 'success'){
           console.log(rep,status);
           console.log('success!!!!');
           console.log(rep,status);
           console.log('Help!!!!');
       }
       console.log(xhr);
   });
})
$('#btn2').click(()=>{
   $('.output').load('file29.html');
})
$('#btn3').click(()=>{
   $('.output').load('file29b.txt');
})
```

Get JSON data with jQuery get method

Endpoints you can use to make GET method requests with practice data
https://www.discoveryvip.com/shared/



Laurence Svekis

New York
10021-3100
NY
21 2nd Street

- 1. 212555-1234 (home)
- 2. 646 555-4567 (office)
- 3. 123 456-7890 (mobile)

```
{firstName: 'Laurence', lastName:
'Svekis', isAlive: true, age: 25,
address: {…}, …}
                              app30.js:36
{type: 'home', number: '212555-12
34'}
                              app30.js:36
{type: 'office', number: '646 555
-4567'}
                              app30.js:36
{type: 'mobile', number: '123 456
-7890'}
```

Exercise : Make and AJAX request to a JSON file and retrieve the values as an object in the code

1. Create a JSON file locally and use the get() method to get the JSON object data.
2. Use the JSON object data by outputting the values into the web page once the button is clicked.

3. Make several different connections, create elements with jQuery and add the JSON response data values into the page elements
4. Practice connecting with different objects, try the endpoint URL in the browser to check to see if GET will work to retrieve the data.

```
<!DOCTYPE html>
<html>
<head>
   <title>jQuery Course</title>
</head>
<body>
   <div class="output"></div>
   <div class="btns">
       <button id="btn1">1</button>
       <button id="btn2">2</button>
       <button id="btn3">3</button>
   </div>
   <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.m
in.js"></script>
   <script src="app30.js"></script>
</body>

</html>
```

```
$('#btn1').click(adder1);
$('#btn2').click(adder2);
$('#btn3').click(adder3);
```

```
function adder2(){

$.get('https://www.discoveryvip.com/shared/people.json',(data)=>
{
       console.log(data);
       $.each(data.people,(ind,val)=>{
           console.log(val.first);
       });
   })
}
function adder3(){

$.get('https://www.discoveryvip.com/shared/books1.json',(data)=>
{
       console.log(data.books);
       let val1 = JSON.stringify(data.books);
       $.each(data.books,(ind,val)=>{
           val1 += `<div>(${val.title}) `;
           val1 += `(${val.author})</div>`;
       });
       $('<div>').html(val1).appendTo('.output');
   })
}

function adder1(){
   $.get('file30.json',(res)=>{
       console.log(res);
       let html = `${res.firstName} ${res.lastName}`;
```

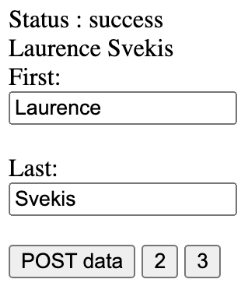
```
        $('<h2>').text(html).appendTo('.output');
        const a = res.address;
        const val1 =
`${a.city}<br>${a.postalCode}<br>${a.state}<br>${a.streetAddress
}`;
        $('<div>').html(val1).appendTo('.output');
        $ul = $('<ul>');
        $ul.appendTo('.output');
        $.each(res.phoneNumbers,(ind,val)=>{
            console.log(val);
            const temp = `${ind+1}. ${val.number} (${val.type})`;
            $('<li>').text(temp).appendTo($ul);
        })

    })
}
```

```
{"firstName":"Laurence","lastName":"Svekis","isAlive":true,"age"
:25,"address":{"streetAddress":"33 Street","city":"New
York","state":"NY","postalCode":"10044"},"phoneNumbers":[{"type"
:"home","number":"1222-1234"},{"type":"office","number":"333
567-1234"},{"type":"mobile","number":"123 444-1234"}]}
```

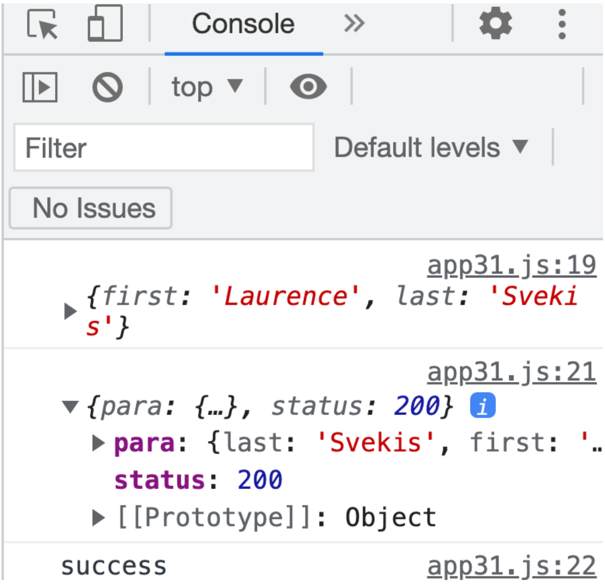
POST data with jQuery using post method

POST requires a server response, there are not many server endpoints to test with. I've set up one using Google Apps Script which can be used, and returns a simple response object back from the data that was sent. You can also set up your own server endpoint using backend coding to provide a response on POST methods.

const url = 'https://script.google.com/macros/s/AKfycbzuGPGSdXCOOD9jQq4S0sx-OQxIOdL70sC_Z4AmWsRC6jnaxnwg8HGWfMBvalb-aD4U/exec';



Status : success
Laurence Svekis
First:
Laurence
Last:
Svekis



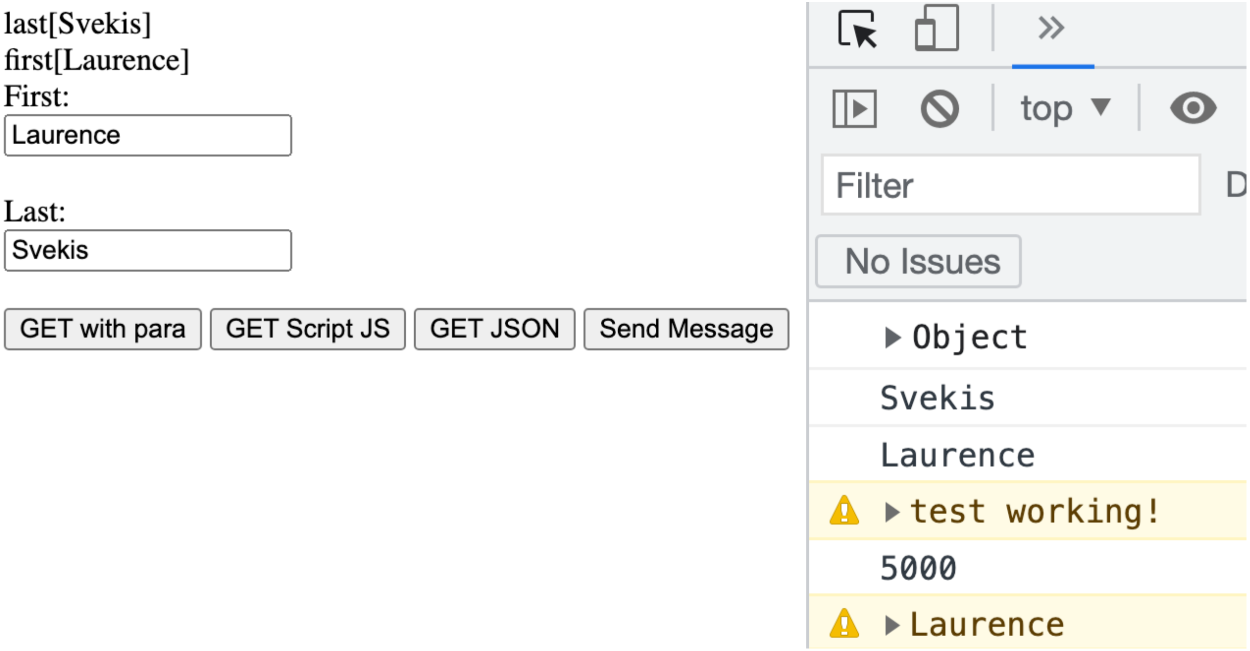
```
app31.js:19
▸ {first: 'Laurence', last: 'Svekis'}
app31.js:21
▾ {para: {…}, status: 200}
  ▸ para: {last: 'Svekis', first: '…
    status: 200
  ▸ [[Prototype]]: Object
success                              app31.js:22
```

Exercise : Send data to a server using the POST method.

1. Create two input fields for first and last names
2. On the button click send the data to the server URL.
3. Use the post() method, create the data argument from the values in the input fields.
4. Add a callback on the post request, to return the response object, that will be in JSON from this endpoint. Also include the status
5. Check if the returned response from the server is 'success', and if it is output the response JSON data selecting the properties and adding the values into the web page.

```
$('#btn1').click(sendDataOver);
$('<input>').attr({
   type:'text',
   value:'Laurence',
   name:'first'
}).appendTo('.ins').before('First:').after('<br>');
$('<input>').attr({
   type:'text',
   value:'Svekis',
   name:'last'
}).appendTo('.ins').before('Last:').after('<br>');
$('#btn1').text('POST data');
const url =
'https://script.google.com/macros/s/AKfycbzuGPGSdXCOOD9jQq4S0sx-
OQxlOdL70sC_Z4AmWsRC6jnaxnwg8HGWfMBvalb-aD4U/exec';
function sendDataOver(){
   const data = {
       first:$('input[name="first"]').val(),
       last:$('input[name="last"]').val()
   }
   console.log(data);
   $.post(url,data,(rep,status)=>{
       console.log(rep);
       console.log(status);
       if(status == 'success'){
           const p = rep.para;
           const temp = `Status : ${status}<br>${p.first}
${p.last}`;
           $('<div>').html(temp).appendTo('.output');
       }
   })
}
```

GET shorthand Methods jQuery getScript getJSON methods



last[Svekis]
first[Laurence]
First:
Laurence

Last:
Svekis

```
▶ Object
Svekis
Laurence
⚠ ▶ test working!
5000
⚠ ▶ Laurence
```

Exercise : jQuery offers several GET methods to either send GET data, import a JS file to use in the current file and to return response data as JSON.

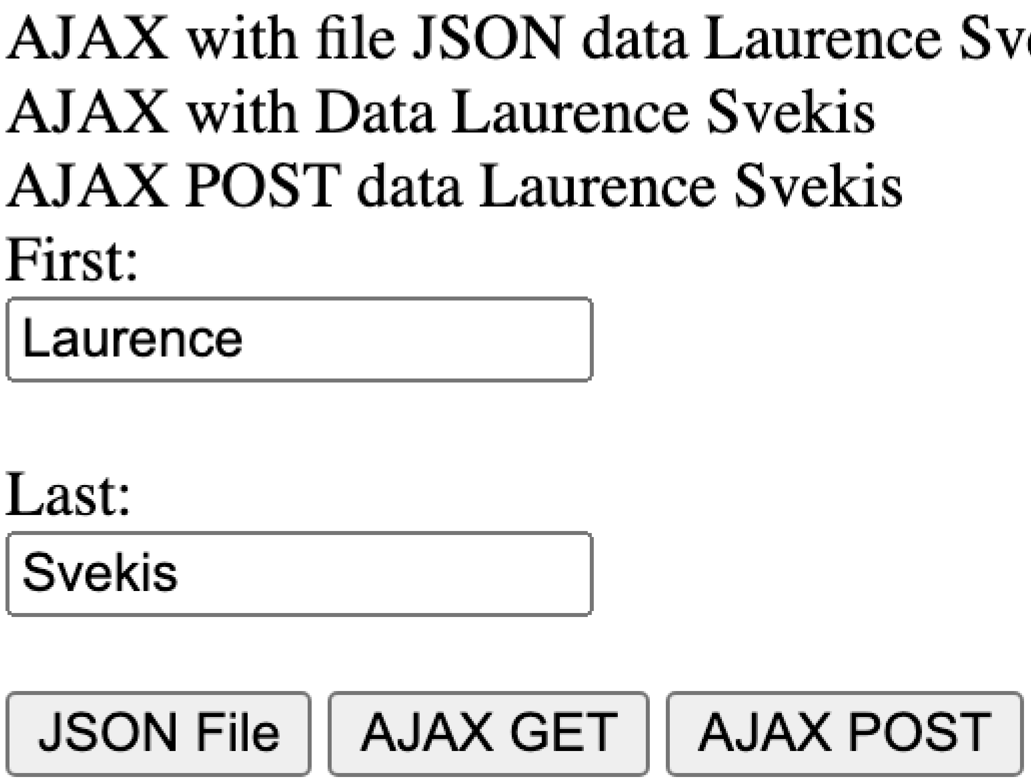
1. Create input elements for the page using jQuery
2. On the first button click, make a get method request to the endpoint and retrieve the JSON data. Output the JSON data object contents to the web page. get() method
3. On the second button click import a JavaScript JS script file. include a function in the new js file, that can then be invoked. getScript() method
4. Add a new button to the page that will invoke the new JS function when clicked, outputting the contents of the input field into the browser console.
5. On the third button click, get the endpoint data as JSON using getJSON() method. output part of the response object data, selecting it by the property value of the object into the console of the browser.

```
const url =
'https://script.google.com/macros/s/AKfycbzuGPGSdXCOOD9jQq4S0sx-
OQxlOdL70sC_Z4AmWsRC6jnaxnwg8HGWfMBvalb-aD4U/exec';
$('<input>').attr({
   type:'text',
   value:'Laurence',
   name:'first'
}).appendTo('.ins').before('First:').after('<br>');
$('<input>').attr({
   type:'text',
   value:'Svekis',
   name:'last'
}).appendTo('.ins').before('Last:').after('<br>');
$('#btn1').click(adder1);
$('#btn2').click(adder2);
$('#btn3').click(adder3);
$('#btn1').text('GET with para');
$('#btn2').text('GET Script JS');
$('#btn3').text('GET JSON');
function adder1(){
   const obj1 = {
       first : $('input').first().val(),
       last : $('input').last().val()
   }
   $.get(url,obj1,call1);
}

function call1(rep){
   console.log(rep.para);
   $.each(rep.para,(prop,val)=>{
       console.log(val);
       const html = `${prop}[${val}]`;
       $('<div>').html(html).appendTo('.output');
   })
}

function adder2(){
   $.getScript('app32a.js',()=>{
       logger('test working!');
       $('<button>').text('Send Message').click(()=>{
           logger($('input').first().val());
       }).appendTo('.btns');
   })
}

function adder3(){
   const url1 = `${url}?id=5000`;
   $.getJSON(url1,(data)=>{
       console.log(data.para.id);
   })
}
```

JAVASCRIPT 2

```
function logger(val){
   console.warn(val);
}
```

jQuery AJAX method and callback options



AJAX with file JSON data Laurence Svekis
AJAX with Data Laurence Svekis
AJAX POST data Laurence Svekis

First: Laurence

Last: Svekis

Exercise : Use the ajax() method to connect to endpoints and use the response data within the web page elements.

1. Use the ajax() method, within the argument object add a property value for the url of the source file. Add a callback function for the success property value. With the successful request output the data values into the web page element.
2. Use the ajax() method and a GET method type, in the ajax argument object, set the url to the endpoint URL, set the type to GET, set the data to an object of the input field values and names, set the response data as dataType json.

3. Add chained methods to ajax() of done() which will fire the callback function once the connection has been successful. Use the data values returned from the server endpoint and add them into your web page elements.
4. Chain to the ajax() method, the fail() and always() methods and output the response data into the console.
5. Use the same code as the GET request, update the type to POST. This will now send the request as a POST with the data from the web page input elements.

```
const url =
'https://script.google.com/macros/s/AKfycbzuGPGSdXCOOD9jQq4S0sx-
OQxlOdL70sC_Z4AmWsRC6jnaxnwg8HGWfMBvalb-aD4U/exec';
$('<input>').attr({
   type:'text',
   value:'Laurence',
   name:'first'
}).appendTo('.ins').before('First:').after('<br>');
$('<input>').attr({
   type:'text',
   value:'Svekis',
   name:'last'
}).appendTo('.ins').before('Last:').after('<br>');

$('#btn1').text('JSON File').click(()=>{
    $.ajax({
        url : 'file30.json',
        success: (result)=>{
           console.log(result);
           const html = `AJAX with file JSON data
${result.firstName} ${result.lastName}`;
           $('<div>').html(html).appendTo('.output');
        }
    });
});


$('#btn2').text('AJAX GET').click(()=>{
   $.ajax({
       url:url,
       type:'GET',
       data:{
           first : $('input').first().val(),
           last : $('input').last().val()
       },
       dataType:'json'
   })
   .done((data)=>{
       console.log(data);
       const html = `AJAX with Data ${data.para.first} ${data.para.last}`;
       $('<div>').html(html).appendTo('.output');
   })
   .fail((err)=>{
       console.log(err);
   })
   .always((data)=>{
       console.log(data);
   })
})
```

```
$('#btn3').text('AJAX POST').click(()=>{
   $.ajax({
       url:url,
       type:'POST',
       data:{
           first : $('input').first().val(),
           last : $('input').last().val()
       },
       dataType:'json'
   })
   .done((data)=>{
       console.log(data);
       const html = `AJAX POST data ${data.para.first}
${data.para.last}`;
       $('<div>').html(html).appendTo('.output');
   })
   .fail((err)=>{
       console.log(err);
   })
   .always((data)=>{
       console.log(data);
   })
})
```