# JQuery

what is JQuery
manipulating DOM in JQuery
Events in jquery
animations and examples

# intro

- web API is a set of JavaScript functions that allow you to programmatically access the browser's DOM.
- 3rd party API like bootstrap is used to style the page and make it look nice and responsive.
  - Bootstrap
  - Materialize
  - Semantic UI
- JQuery is a JavaScript library that simplifies DOM manipulation, event handling, Ajax, and animation.

## adding JQuery to DOM

- Free open source js library making JS more easy to use and more powerful
- add JQuery to your HTML
  `<script src="https://code.jquery.com/jquery-3.4.1.min.js"></script>`
- other scripts are loaded after JQuery script
- call it with `$` or `jQuery`
- example of difference click event handling exmaple

```
$("#btn").click(function(){
  $("#div1").hide();
});
//vs Js
document.getElementById("btn").onclick = function(){
  document.getElementById("div1").style.display = "none";
};
```

## jquery statement

- in js we say `document.getElementById("btn").onclick = function(){}` in jquery we say `$("#btn").click(function(){})`
- jquery returns object with jquery wrappers like `animate()` and `css()` (many more methods)
- we can use index to get element wihout jquery wrapper `$("#div1")[0]`
- this wont have access to jquery methods like `animate()`
- example of getting heading and animating it in jquery

```
let heading = $("h1");
heading.animate({
  opacity: 0.5,
  fontSize: "2em"
}, 1000);
```

# jquery selectors

- jquery selectors are like css selectors but with jquery syntax

```
$("#div1")
```

- select an element say h3 and style and animate it

```
$("h2").css("color", "red").animate({fontSize: "2em"}, 1000);
```

- styling and animating usin class selector

```
$(".middle").css({border: "3px solid red"}).animate({fontSize:
"1em"}, 1000);
```

- styling and animating id selector

```
$("#div1").css({border: "3px solid red"}).animate({fontSize:
"1em"}, 1000);
```

# JQuery Filters

- jquery filters are used to refine the selector to get exactly what we want
- just want first ans last child of the list
  `$("header nav ul:first).css("color", "red");` and
  `$("header nav li:last").css("color", "red");`
- `:first-child` and `:last-child` are different filters compared to `:first` and `:last`
- we can get `even` and `odd` children of the list
  `$("header nav li:even").css("color", "red");`
- `:not` is used to get all children except the one we want
  `$('header nav li:not(.about)').css("color", "red");`
- `lt` and `gt` are used to get children with index less than or greater than the index of the element we want
  `$('header nav li:lt(2)').css("color", "red");`

# attribute Filters

- attribute filters are used to get elements with specific attributes
- get all divs with an id give it border pink

```
$("div[id]").css("border", "3px solid pink");
```

- get all p tags with a class of `about`

```
$("p[class=about]").css("border", "3px solid pink");
```

- more filters at jquery selector docs

```
https://api.jquery.com/category/selectors/
```

## traversing the DOM in JQuery

- Dom is the relationship between the html elements on the page
- `next()` and `prev()` are used to get the next and previous element in the dom
- `parent()` and `children()` are used to get the parent and children of the element
- `find()` is used to get the children of the element
- `closest()` is used to get the closest parent of the element

```
let heading = $("h1");
heading.next();
heading.prev();
heading.parent();
heading.children();
heading.find("li");
```

# chaining in jquery

- jquery methods like `animate()`, `css()`, `hide()`, `show()` and `toggle()` can be chained together eg:

```
$("h1").css("color", "red").animate({fontSize: "2em"},
1000).hide(1000).show(1000);
```

- speeds up the way we code
- don't over do it user next line when possible

```
$("h1").css("color", "red").animate({fontSize: "2em"}, 1000).hide(1000).show(1000).next().css("color", "blue").closest("div").css("color", "green");
```

# adding elements to the DOM

- jquer can add elements like `p`, `h1`, `h2`, `h3`, `h4`, `h5`, `h6`, `ul`, `ol`, `li`, `a`, `img`, `div` to the DOM
- `$("body").append("<p>This is a new paragraph</p>");`
- `$("body").append("<h1>This is a new heading</h1>");`
- `$("body").append("<ul><li>This is a new list item</li></ul>");`
- `$("body").append("<a href='https://www.google.com'>This is a new link</a>");`
- `$("body").append("<img src='https://www.google.com/images/branding/googlelogo/2x/googlelogo_`

## various ways to add elements to the DOM

- `append()` adds elements to the end of the list `$("body").append("<p>This is a new paragraph</p>");`
- `prepend()` adds elements to the beginning of the list `$("body").prepend("<p>This is a new paragraph</p>");`
- `before()` adds elements before the element `$("body").before("<p>This is a new paragraph</p>");`
- `after()` adds elements after the element `$("body").after("<p>This is a new paragraph</p>");`
- `html()` adds elements to the DOM `$("body").html("<p>This is a new paragraph</p>");`
- `text()` adds text to the DOM `$("body").text("This is a new paragraph");`

## Wrap and unwrap elements

- Wrapping is used to wrap elements in a new element
  `$("#section").wrap("<div class='wrapper'></div>");`
  *will take individual elements and wrap them in a new element
- Un-wrapping is used to unwrap elements `$("#section").unwrap();`
  - get all the tags with id section and unwrap them, takes out the parent element
- `wrapAll` will take all elements and wrap them in a new element
  `$("#section").wrapAll("<div class='wrapper'></div>");`

```
wrapper = $("<ul>");
var actionButton = $("#action-button");
var wrapped = true;
actionButton.click(function(){
  if(wrapped){
    $('.listItem').unwrap();
    wrapped = false;
    actionButton.text("Wrap");
  } else {
    $('.listItem').wrap(wrapper);
    wrapped = true;
    actionButton.text("Unwrap");
  }
```

# removing elements from the DOM

- `remove()` removes the element from the DOM
  `$("#section").remove();`
- `empty()` removes all the children of the element
  `$("#section").empty();`

```
let actionButton = $("#action-button");
let list = $("ul");
for(let i = 0; i < 10; i++){
  list.append("<li class='listItem'>" + i + "</li>");
}
actionButton.click(function(){
  list.empty();
});
```

# Changing text and attribute

- `text()` changes the text of the element
  `$("#section").text("This is a new paragraph");`
- `.attr()` changes the attribute of the element
  `$("#section").attr("id", "new-id");`
- `.removeAttr()` removes the attribute of the element
  `$("#section").removeAttr("id");`

```
let attrValue = $("#section").attr("id");
$("#section").attr("id", "new-id");
$("#section").removeAttr("id");
```

## control css in Jquery

- `.css()` changes the css of the element `$("#section").css("color", "red");`
- `.addClass()` adds a class to the element `$("#section").addClass("new-class");`
- get css property `$("#section").css("color");` //red
- add multiple css property
  `$("#section").css({"color": "red", "background-color": "blue"});` // all element with id have text color is red and background color is blue
- `.removeClass()` removes a class from the element
  `$("#section").removeClass("new-class");`
- `.toggleClass()` toggles (add and remove)a class on the element
  `$("#section").toggleClass("new-class");`

```
let cssObj = {"color": "red", "background-color": "blue"};
$("ul").css(cssObj);
$("ul").addClass("new-class");
$("ul").removeClass("new-class");
$("ul").toggleClass("new-class");
```

# Event binding and unbinding

- `on` binds an event to an element
  `$("#section").on("click", function(){});`
- `off` unbinds an event to an element `$("#section").off("click");`
- once the function fires we can get the object in context with `this` and wrap it in jQuery to get Jquery wrapper `$(this)`

# event helpers

- `.one()` binds an event to an element and fires only once
  `$("#section").one("click", function(){});`
- `click` is a built in event for single click
  `$("#section").click(function(){});`
- `dblclick` is a built in event for double click
  `$("#section").dblclick(function(){});`

# event object

- get the type of event that occurred `event.type`
- get event target `event.target`
- stope event propogation `event.stopPropagation()`
- get x coordinate of mouse `event.pageX`

# Animation in Jquery

- `.animate()` animates an element

```
$("#section").animate({width: "200px", height: "200px"}, 1000);
```

- anything with numeric value can be animated, you can add the animate time
- second parameted is the time in milliseconds
- third is the easing function eg: `linear` `easeIn` `easeOut` `easeInOut`
- fourth is the callback function

```
$("#section").animate({width: "200px", height: "200px"}, 1000,
'linear', function(){});
```

# Fading elements

- `.fadeIn()` fades in an element `$("#section").fadeIn(1000);`
- `.fadeOut()` fades out an element `$("#section").fadeOut(1000);`
- `.fadeToggle()` toggles the fade in and out of an element
  `$("#section").fadeToggle(1000);`
- `.fadeTo()` fades an element to a specific opacity
  `$("#section").fadeTo(1000, .5);`
- argument is the speed in milliseconds, and the second argument is the opacity

# Hide show and toggle

- `.hide()` hides an element `$("#section").hide();`
- `.show()` shows an element `$("#section").show();`
- `.toggle()` toggles the visibility of an element
  `$("#section").toggle();`

```
let actionButton = $("#action-button");
actionButton.click(function(){
  $("#section").toggle();
});
```