Lecture 3.1
Introduction to JQuery

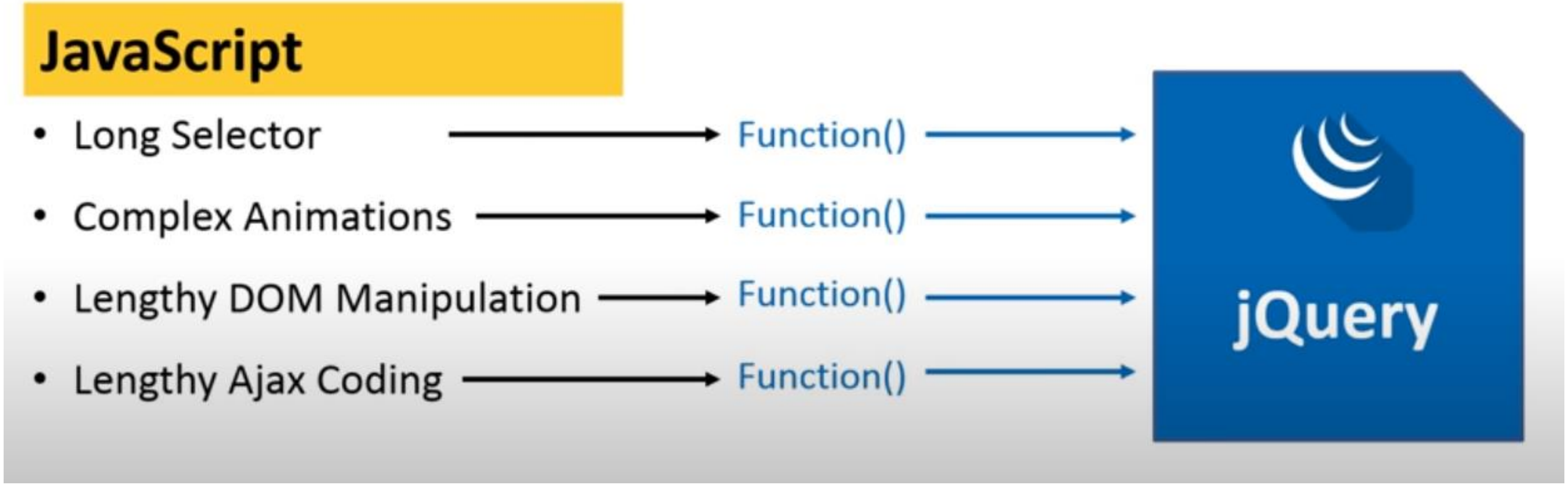
Course Instructor

Engr. Madeha Mushtaq

JQuery

- jQuery is a JavaScript library of helper functions that are cross browser-compatible.
- It is an open source add-in.
- Make your JavaScript code shorter, faster and cross browser.
- Manipulate your HTML, like showing or hiding something from the page.

JQuery



JavaScript

- Long Selector
- Complex Animations
- Lengthy DOM Manipulation
- Lengthy Ajax Coding

JQuery

- The jQuery library contains the following features:
  - HTML/DOM manipulation
  - CSS manipulation
  - HTML event handling
  - Effects and animations
  - AJAX (Asynchronous JavaScript and XML)

Adding jQuery to Your Web Pages

- We can download the library locally and include the file
  - <script src = "JS\jquery-3.6.4.js"> </script>
- Where Jquery-3.6.4.js is the local copy of the library.
- Alternatively, we can use the CDN on the Web:
  - <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"> </script>

JQuery Initialization

- JQuery depends on the DOM being ready for access
  - We do not want to use it until the page has been completely loaded.
  - Once this has occurred we can use JQuery to access parts of our document (in various ways) and to manipulate them (also in various ways).

JQuery Initialization

- A good way to make sure the DOM is ready before using JQuery is to put our JQuery access statements into a callback function:
- <script type="text/javascript">
- $(document).ready(function(){

  //Rest of our JQuery code here will execute when
  //ready function runs
- });
- </script>
- Note the syntax:
- Most JQuery commands are going to be prefixed by $.

JQuery Initialization

- The ready event fires when the DOM is ready, and the function is called.
- Here are some examples of actions that can fail if methods are run before the document is fully loaded:
  - Trying to hide an element that is not created yet
  - Trying to get the size of an image that is not loaded yet.

JQuery Selection

- The first step is, we must be able to select elements/ items in our document.
- The jQuery selectors allow you to search or select HTML elements and then perform intended action on those.
- $(selector).action()
- There are MANY ways of selection in Jquery.

JQuery Selection

- The jQuery syntax is tailor-made for selecting HTML elements and performing some action on the element(s).
- Basic syntax is: **$(selector).action()**

- A $ sign to define/access jQuery
- A (selector) to "query (or find)" HTML elements
- A jQuery action() to be performed on the element(s)
- Examples:
  - $(this).hide() - hides the current element.
  - $("p").hide() - hides all <p> elements.
  - $(".test").hide() - hides all elements with class="test".
  - $("#test").hide() - hides the element with id="test".

JQuery Selection

- Selecting by TAG name: $(“tagname”)
  - Returns an array of tags that match tag name
- Selecting by ID: $(“#theid”)
  - Returns element with id equal to the id
- Selection by CSS class: $(“.className”)
  - Returns an array of elements with class .className
- Selection by odd/even: $(“element:odd”)
  - Returns array of items matching element with odd index values (with indices starting at 0)

JQuery Selection

- Selection by index:
- $(“element:eq(2)”) $(“element:lt(4)”) $(“element:gt(1)”)
  - Returns elements specified by index (eq = equal, lt = less than, gt = greater than)
- $("p:first"), selects the first <p> element.
- $("p:last"), selects the last <p> element.

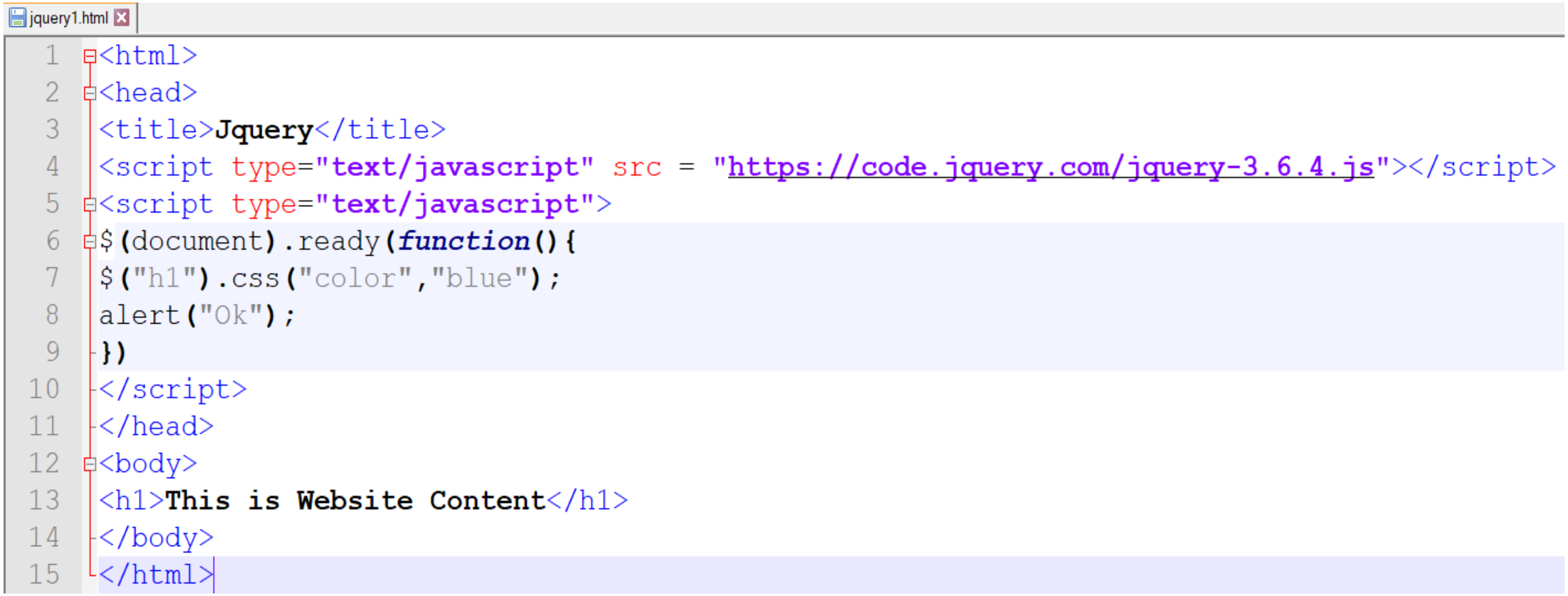
jQuery - css() Method

- The css() method sets or returns one or more style properties for the selected elements.
- The following example will return the background-color value of the FIRST matched element:
- $("p").css("background-color");
- The following example will set the background-color value for ALL matched elements:
- $("p").css("background-color", "yellow");

Set Multiple CSS Properties

- The following example will set a background-color and a font-size for ALL matched elements:
- Example:
- $("p").css({"background-color": "yellow", "font-size": "200%“});

JQuery Selectors



```
<html>
<head>
<title>Jquery</title>
<script type="text/javascript" src = "https://code.jquery.com/jquery-3.6.4.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("h1").css("color","blue");
alert("Ok");
})
</script>
</head>
<body>
<h1>This is Website Content</h1>
</body>
</html>
```

The Element Selector

- The jQuery element selector selects elements based on the element name.
- You can select all <p> elements on a page like this:

```
$("p")
$(document).ready(function(){
$("button").click(function(){
$("p").hide();
});
});
```

The # Id Selector

- The jQuery #id selector uses the id attribute of an HTML tag to find the specific element.
- To find an element with a specific id, write a hash character, followed by the id of the HTML element:

-
  ```
  $(document).ready(function(){
        $("button").click(function(){
       $("#test").hide();
       });
       });
  ```

The .Class Selector

- The jQuery .class selector finds elements with a specific class.

- To find elements with a specific class, write a period character, followed by the name of the class:

-
  ```
  $(document).ready(function(){
    $("button").click(function(){
      $(".test").hide();
    });
  });
  ```

JQuery Selectors

Syntax	Description
$("*")	Selects all elements
$(this)	Selects the current HTML element
$(“p.intro")	Selects all <p> elements with class="intro"
$("p:first")	Selects the first <p> element
$("ul li:first")	Selects the first <li> element of the first <ul>
$("ul li:first-child")	Selects the first <li> element of every <ul>
$("[href]")	Selects all elements with an href attribute
$(“a[target='_blank']")	Selects all <a> elements with a target attribute value equal to "_blank"
$("a[target!='_blank']")	Selects all <a> elements with a target attribute value NOT equal to "_blank"
$(“:button")	Selects all <button> elements and <input> elements of type="button"
$("tr:even")	Selects all even <tr> elements
$("tr:odd")	Selects all odd <tr> elements

Modifying Elements

- Once an element has been selected we can do whatever we want to it.
- Some examples:
- Update the CSS of the selected element(s)
  - $(selector).css()
- Many DOM methods to update properties of the element
  - $(selector).append()
  - $(selector).addClass()
  - $(selector).attr()

Modifying Elements

- Methods to change appearance of elements:
  - $(selector).hide()
  - $(selector).show()
- Methods to deal with events and event handling
  - $(selector).bind()
  - $(selector).click()
  - $(selector).focus()
  - $(selector).mouseover()

JQuery Events

Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

Commonly Used jQuery Event Methods

- **$(document).ready()**

The $(document).ready() method allows us to execute a function when the document is fully loaded.

- **click()**
- The click() method attaches an event handler function to an HTML element.
- The function is executed when the user clicks on the HTML element.
- The following example says: When a click event fires on a <p> element; hide the current <p> element:
- $("p").click(function() { $(this).hide();

});

Commonly Used jQuery Event Methods

- **dblclick()**
- The dblclick() method attaches an event handler function to an HTML element.
- The function is executed when the user double-clicks on the HTML element:
- $("p").dblclick(function(){ $(this).hide(); });
- **mouseenter()**
- The mouseenter() method attaches an event handler function to an HTML element.
- The function is executed when the mouse pointer enters the HTML element:
- $("#p1").mouseenter(function(){ alert("You entered p1!");

});

Commonly Used jQuery Event Methods

- **mouseleave()**
- The mouseleave() method attaches an event handler function to an HTML element.
- The function is executed when the mouse pointer leaves the HTML element:
-
  ```
  $("#p1").mouseleave(function(){
    alert("Bye! You left p1!"); });
  ```
- **hover()**
- The hover() method takes two functions and is a combination of the mouseenter() and mouseleave() methods
-
  ```
  $("#p1").hover(function(){
    alert("You entered p1!"); },
  function(){ alert("Bye! You left p1!"); });
  ```

Commonly Used jQuery Event Methods

- **focus()**
- The focus() method attaches an event handler function to an HTML form field.

-
  ```
  $("input").focus(function(){
    $(this).css("background-color", "#cccccc");
  });
  ```
- **blur()**
- The blur() method attaches an event handler function to an HTML form field.

-
  ```
  $("input").blur(function(){
    $(this).css("background-color", "#ffffff");
  });
  ```

Commonly Used jQuery Event Methods

- **The on() Method**
- The on() method attaches one or more event handlers for the selected elements.

-
  ```
  $("p").on("click", function(){
    $(this).hide();
  });
  ```

```
$("p").on({
mouseenter: function(){
    $(this).css("background-color", "lightgray");
  },
  mouseleave: function(){
    $(this).css("background-color", "lightblue");
  },
  click: function(){
    $(this).css("background-color", "yellow");
  }
});
```

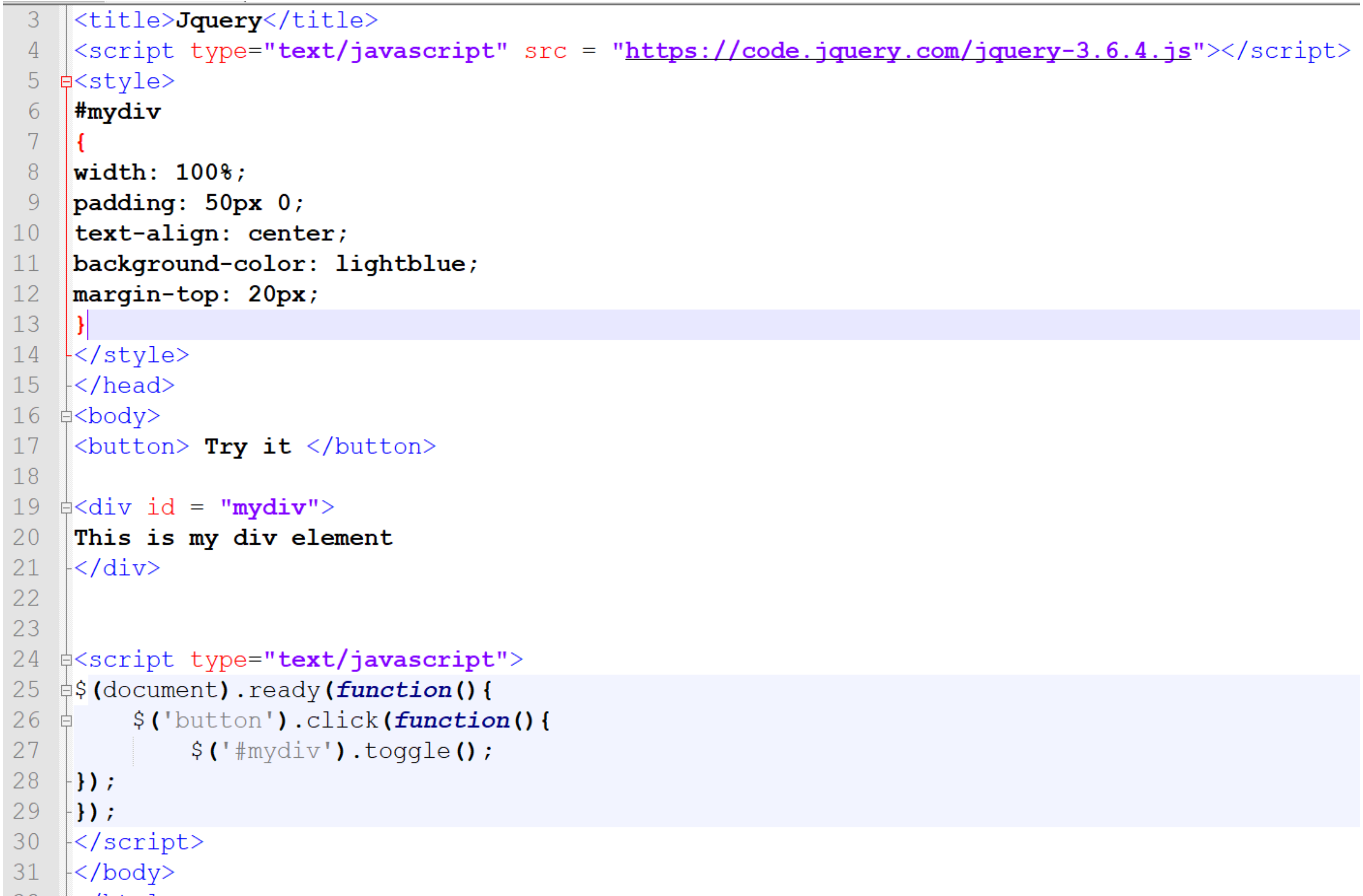
jQuery hide() and show()

```
<script>
$(document).ready(function(){
  $("#hide").click(function(){
    $("p").hide();
  });
 $("#show").click(function(){
    $("p").show();
  });
});
</script>
<p>If you click on the "Hide" button, I will disappear.</p>
<button id="hide">Hide</button>
<button id="show">Show</button>
```

jQuery hide() and show()

- The optional speed parameter specifies the speed of the hiding/showing, and can take the following values: "slow", "fast", or in milliseconds
-
  ```
  $("button").click(function(){
    $("p").hide(3000);
  });
  ```

jQuery toggle()

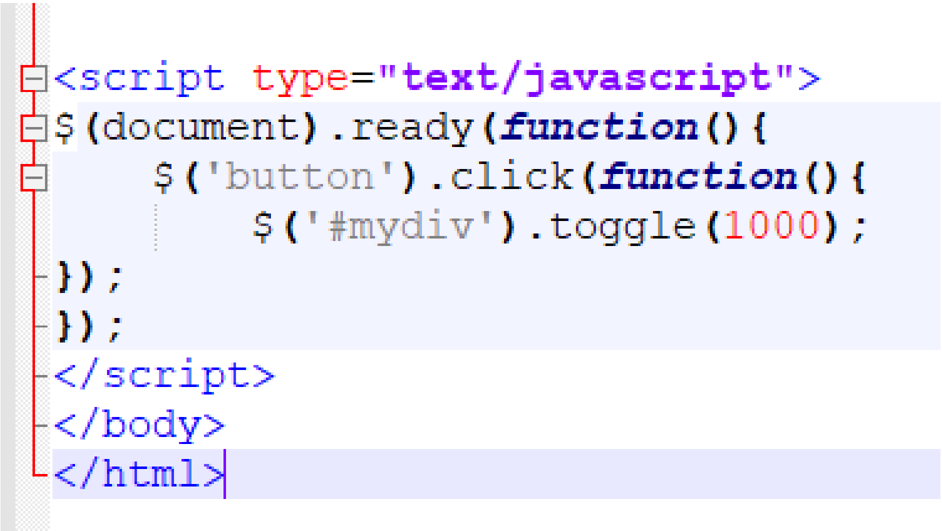


```
<title>Jquery</title>
<script type="text/javascript" src = "https://code.jquery.com/jquery-3.6.4.js"></script>
<style>
#mydiv
{
width: 100%;
padding: 50px 0;
text-align: center;
background-color: lightblue;
margin-top: 20px;
}
</style>
</head>
<body>
<button> Try it </button>

<div id = "mydiv">
This is my div element
</div>


<script type="text/javascript">
$(document).ready(function(){
    $('button').click(function(){
        $('#mydiv').toggle();
});
});
</script>
</body>
```

jQuery toggle()



```
<script type="text/javascript">
$(document).ready(function(){
    $('button').click(function(){
        $('#mydiv').toggle(1000);
});
});
</script>
</body>
</html>
```

Animation Examples

- Hide Show Content on Button Click
- Slide Down & Slide Up Content on Click
- Toggle & Fade Text on Button Click
- Animations on Click

jQuery fadeIn() Method

- $(*selector*).fadeIn(*speed, callback*);
- $(*selector*).fadeOut(*speed,callback*);
- $(*selector*).fadeToggle(*speed,callback*);
- $(*selector*).fadeTo(*speed,opacity,callback*);

-
  ```
  $("button").click(function(){
    $("#div1").fadeTo("slow", 0.15);
    $("#div2").fadeTo("slow", 0.4);
    $("#div3").fadeTo("slow", 0.7);
  });
  ```

jQuery Sliding Methods

- With jQuery you can create a sliding effect on elements.
- jQuery has the following slide methods:
- slideDown()
- slideUp()
- slideToggle()

jQuery Sliding Methods

- $(*selector*).slideDown(*speed,callback*);
- The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.
- The optional callback parameter is a function to be executed after the sliding completes.

jQuery animate()

- **animate()** is a jQuery method that allows you to animate HTML elements using JavaScript code. The basic syntax of **animate()** is:
- $(*selector*).animate({*params*},*speed,callback*);
- The required params parameter defines the CSS properties to be animated.
- The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.
- The optional callback parameter is a function to be executed after the animation completes.

jQuery animate()

-
  ```
  $("button").click(function(){
    $("div").animate({left: '250px'}, “”);
  });
  ```

jQuery animate()

By default, all HTML elements have a static position, and cannot be moved. To manipulate the position, remember to first set the CSS position property of the element to relative, fixed, or absolute!

jQuery - Add Elements

- Add New HTML Content
- We will look at four jQuery methods that are used to add new content:

- append() - Inserts content at the end of the selected elements
- prepend() - Inserts content at the beginning of the selected elements
- after() - Inserts content after the selected elements
- before() - Inserts content before the selected elements

jQuery - Add Elements

```
<head>
<script
src="/jquery/3.4.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
 $("#btn1").click(function(){
  $("img").before("<p>Brfore</p>");
 });
 $("#btn2").click(function(){
  $("img").after("<h1>After</h1>");
 });
});
</script>
</head>
```

```
<body>

<img src="image.gif" alt="jQuery"
width="100" height="140"><br><br>

<button id="btn1">Insert before</button>
<button id="btn2">Insert after</button>

</body>
</html>
```

jQuery - Remove Elements

- Remove Elements/Content
- To remove elements and content, there are mainly two jQuery methods:
- remove() - Removes the selected element (and its child elements)
- empty() - Removes the child elements from the selected element

jQuery - Remove Elements

```
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").remove();
  });
});
</script>
```

jQuery - Remove Elements

- The following example removes all <p> elements with class="test":
- $("p").remove(".test");
- This example removes all <p> elements with class="test" or class="demo":
- $(“p").remove(".test, .demo");

Class Task