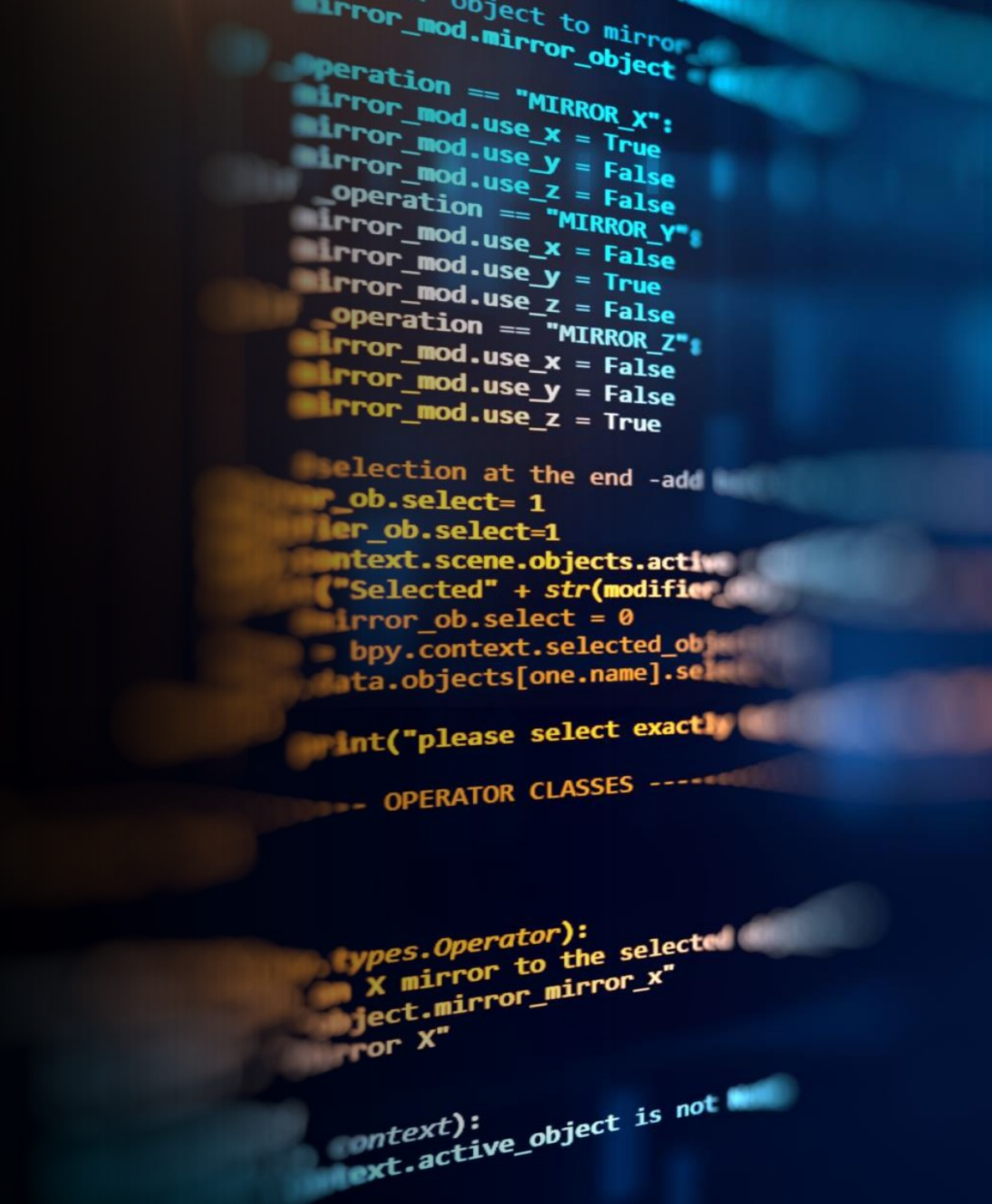
JQUERY

Prepared by
Mr. Renato L. Adriano II



WHAT IS JQUERY?

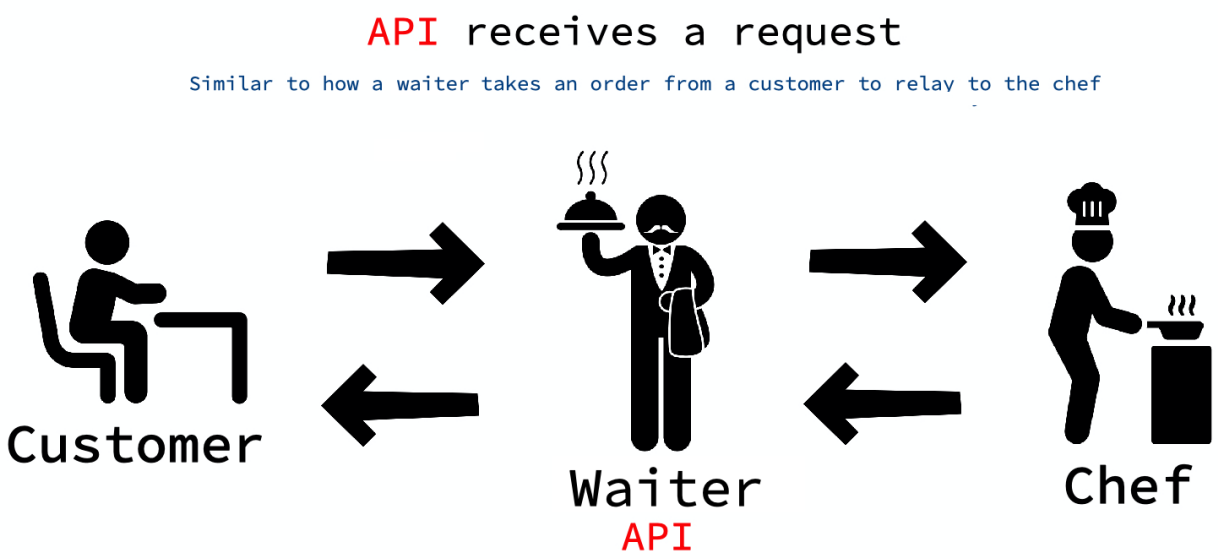
- "The purpose of jQuery is to make it much easier to use JavaScript on your website."

- "makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers."


JQUERY API

WHAT IS JQUERY API?

jQuery API (Application Programming Interface) is a JavaScript file that is used to allow programmers to use the rich contents of jQuery.



HOW TO USE JQUERY API?

Using Downloaded jQuery file:

```
<script src="jquery-3.3.1.min.js"></script>
```

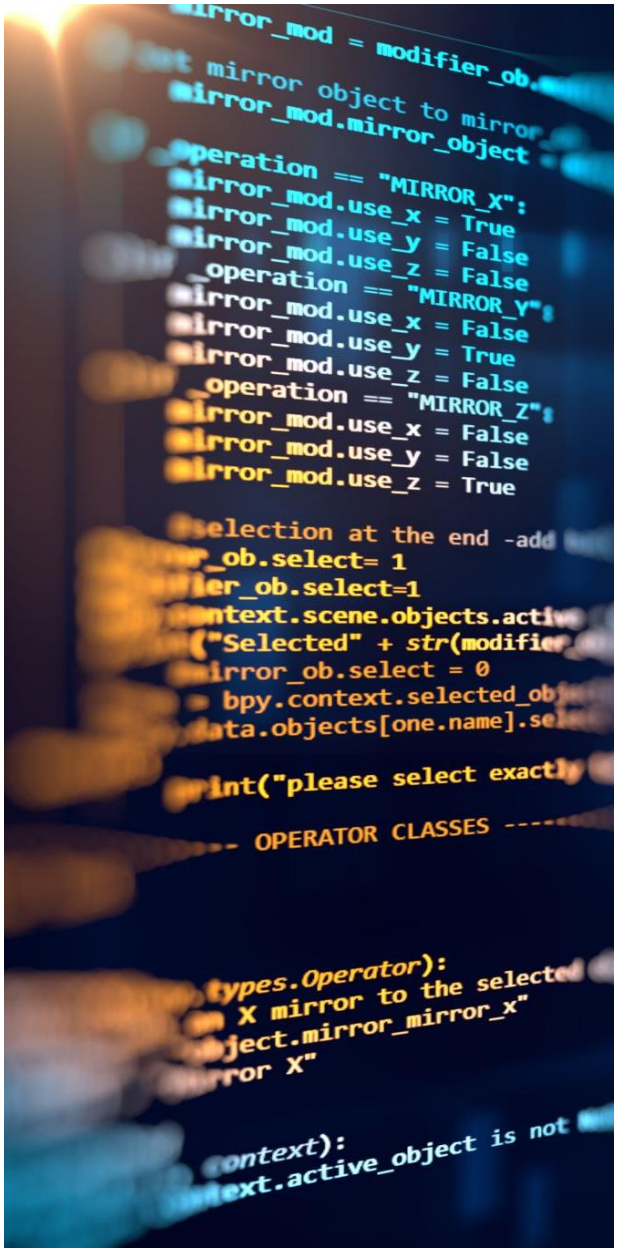
NOTE: If the file accessing jQuery and the jQuery file are not on the same folder, you must provide the correct file path.

HOW TO USE JQUERY API?

Using Content Delivery Network (CDN):

```
<script src="https://code.jquery.com/jquery-3.6.0.js"
integrity="sha256-
H+K7U5CnXl1h5ywQfKtSj8PCmoN9aaq30gDh27Xc0jk="
crossorigin="anonymous"></script>
```

SOURCE: https://code.jquery.com/



CONTENTS


JQUERY SYNTAX

JQUERY SYNTAX

jQuery syntax always starts with the keyword **jQuery** or **$.**

Example:

```
jQuery(selector).action();
$(selector).action();
```

NOTE: $ is an alias for **jQuery** and it's often employed because it's shorter and faster to write.


JQUERY SELECTORS

JQUERY SELECTORS

jQuery selectors are the same as CSS selectors.

1. Element Selector
2. Class Selector
3. Id Selector
4. Attribute Selector
5. Pseudo-class Selector
6. Group Selector
7. Descendant Selector
8. **this** Selector

ELEMENT SELECTOR

Element selectors are selected using the element's tag name.

Examples:

$(“a”)

$(“div”)

$(“span”)

$(“form”)

$(“input”)

$(“button”)

CLASS SELECTOR

Class selectors are selected using the element's class name preceded by a dot.

Examples:

$(".btn")

$(".class1")

$(".col1")

ID SELECTOR

Id selectors are selected using the element's id preceded by a hash (#).

Examples:

$(“#error”)

$(“#btnSave”)

$(“#txtName”)

ATTRIBUTE SELECTOR

Attribute selectors are selected using attributes of an element. Attribute selector uses brackets ([ ]).

Examples:

$(“[id]”)

$(“[type=‘text’]”)

$(“[class=‘first’]”)

PSEUDO CLASS SELECTOR

Pseudo-classes are reserved keywords used by CSS to provide additional style to an element. Pseudo-class selectors are selected using the pseudo-class name preceded by a colon (:).

Examples:

$(“tr:even”)

$(“p:first”)

$(“a:active”)

GROUP SELECTOR

Group selectors are just combination of other selectors meant to target multiple elements selected in a single statement. Group selection is separated by a comma (,).

Examples:

$(“#div1, span”)

$(“.col1, .btn”)

$(“a, p”)

DESCENDANT SELECTOR

Descendant selector selects the element in accordance with the hierarchy provided. Descendant selection is separated by a space.

Examples:

$(“div p”)

$(“form .btn”)

$(“table tr td img”)

THE “THIS” SELECTOR

The “**this**” selection targets the outer selector to be affected by the action.

Example:

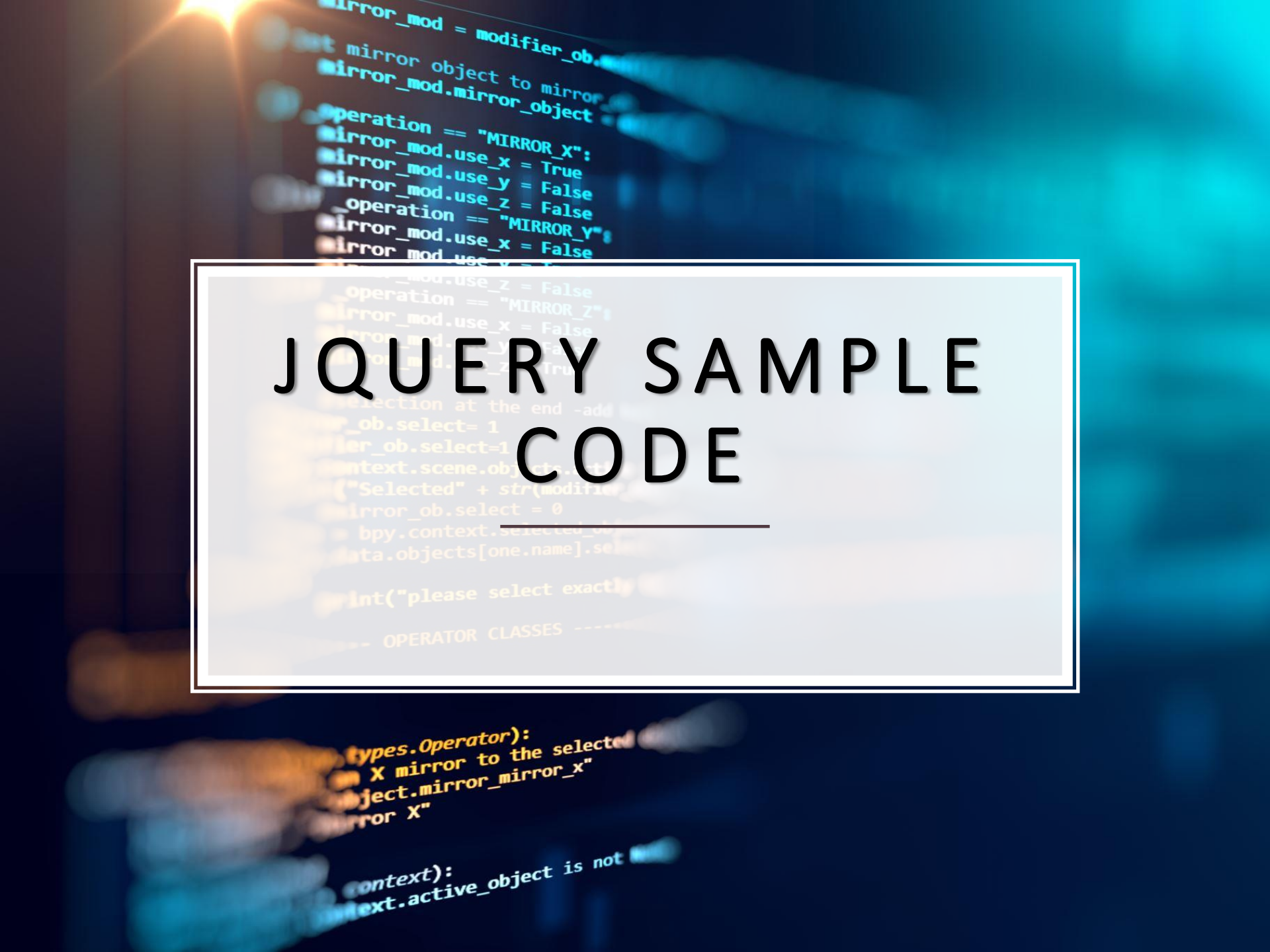
```
$( “#div1” ).click(function() {
    $(this).hide(2000);
});

//hides the element with id “div1” for 2 seconds (2000 milliseconds).
```


JQUERY EVENTS

JQUERY EVENT METHODS

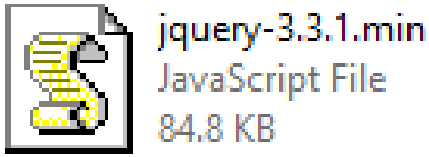
jQuery event methods allow elements to do something from triggering the element. Some of the commonly used events are:

FORM EVENTS	KEYBOARD EVENTS	MOUSE EVENTS
submit	keyup	click
change	keydown	dblclick
focus	keypress	hover
blur		mouseenter
		mouseleave


JQUERY SAMPLE CODE

SAMPLE CODE (JQUERY API)

NOTE: Before coding jQuery, do not forget to add jQuery API in your code. jQuery codes will not work without the API.



Download the JQUERY API on the link below:

https://drive.google.com/file/d/1tVfmvn282JtGCciOk-XAwvseOrXhFZmM/view?usp=sharing

SAMPLE CODE (HTML FILE)

NOTE: The first <script> tag adds the jQuery API on the code. The second <script> tag adds the external JavaScript.

Step 1: Create the HTML file **sample.html**

```
<!doctype html>
<html>
    <head>
        <script src="jquery-3.3.1.min.js"></script>
        <script src="sample.js"></script>
    </head>
    <body>
        <button id="btnHide">Hide div</button>
        <div id="display">Make me disappear! Click the button.</div>
    </body>
</html>
```

SAMPLE CODE (JAVASCRIPT FILE)

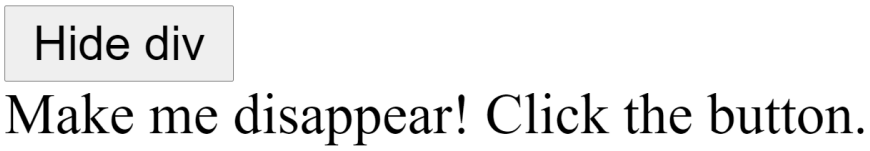
NOTE: This code makes the button clickable. If the button is clicked, the element with id “display” will be hidden after two-second animation.

Step 2: Create the JavaScript file **sample.js**

```
$(document).ready(function(){
    $("#btnHide").click(function(){
        $("#display").hide(2000);
    });
});
```

SAMPLE CODE (OUTPUT)

WHEN THE PAGE LOAD:



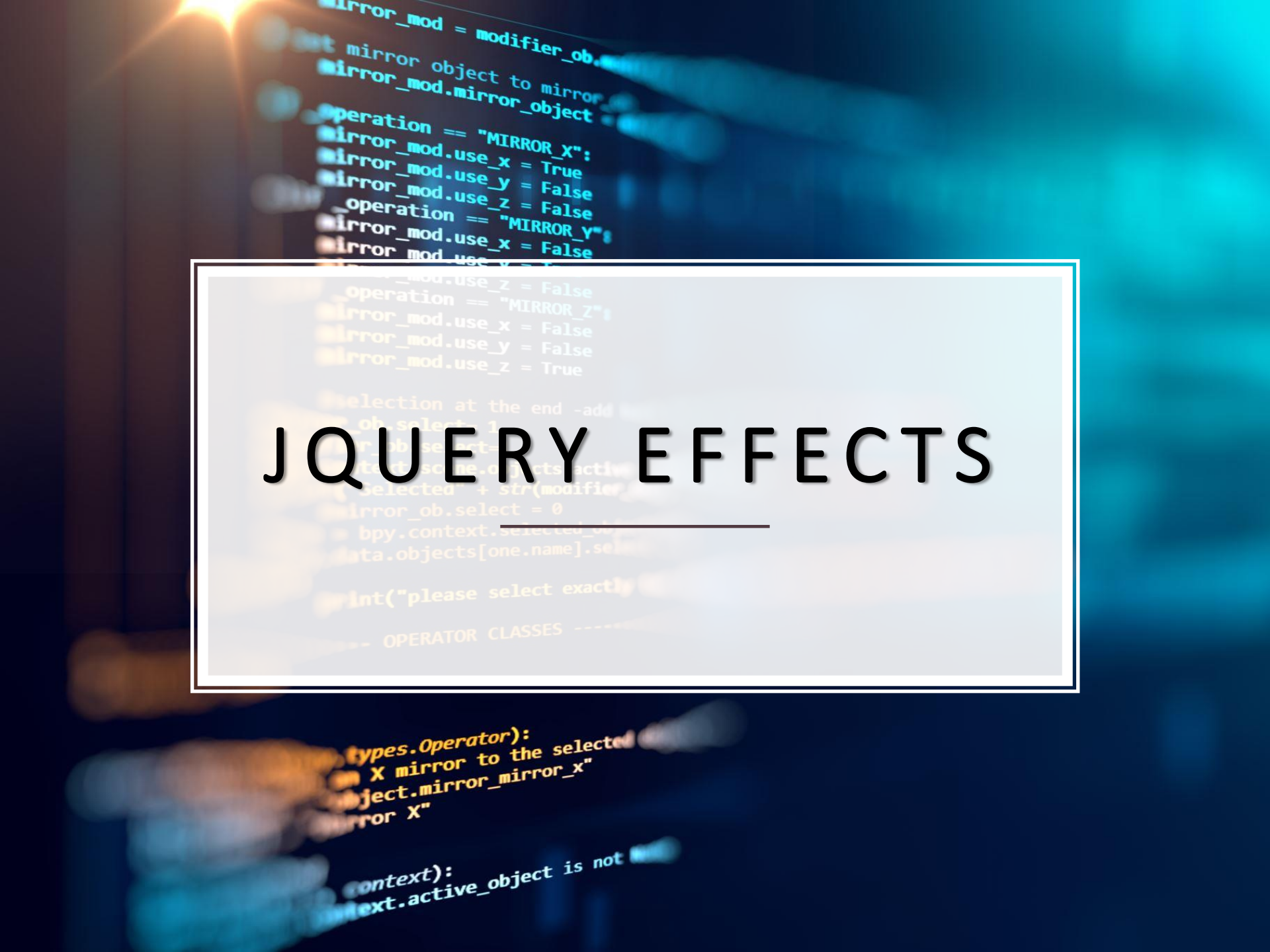
AFTER THE BUTTON WAS CLICKED:

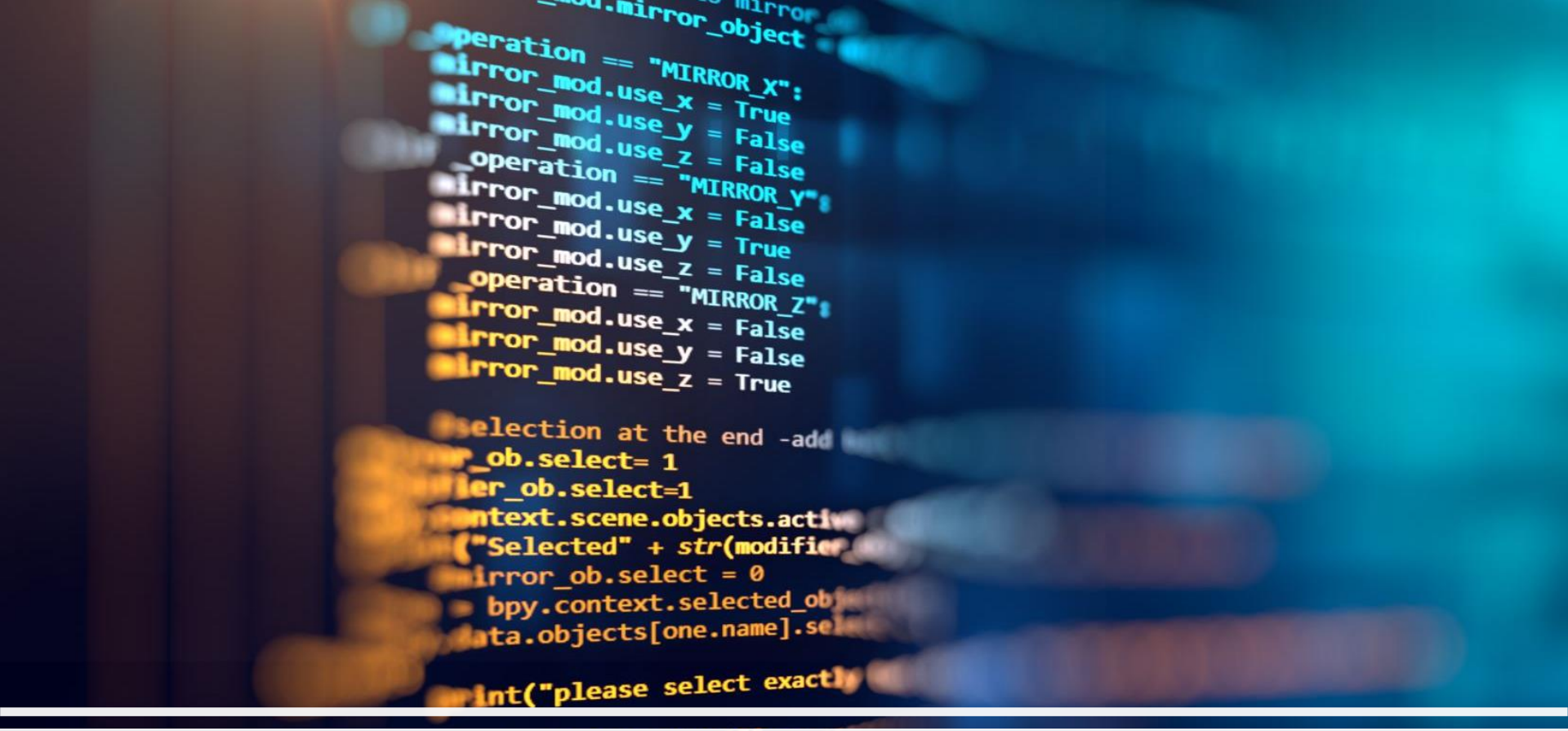
THE $(DOCUMENT).READY()

On the given example, the code for making the button clickable is inside the $(document).ready() statement.

The question is, WHY???

Answer: $(document).ready() allows loading of events after the page has been fully loaded. This avoids any unnecessary user interactions on the website, e.g., clicking a button while the page still loads.


JQUERY EFFECTS



HIDE & SHOW EFFECTS

JQUERY HIDE EFFECT

The **hide()** effect hides the selected element with the given speed.

Syntax:

hide([speed , object callback_function])

slow, fast or milliseconds like 100, 1000, 2000 etc.

NOTE: Make sure that the element you are trying to hide is visible on the page.

JQUERY SHOW EFFECT

The **show()** effect shows the selected element with the given speed.

Syntax:

show([speed , object callback_function])

NOTE: Make sure that the element you are trying to show is hidden on the page.

HIDE AND SHOW (HTML FILE)

Step 1: Create the HTML file **hideshow.html**

```
<!doctype html>
<html>
    <head>
        <script src="jquery-3.3.1.min.js"></script>
        <script src="hideshow.js"></script>
    </head>
    <body>
        <button id="btnHide">Hide div</button>
        <button id="btnShow">Show div</button>
        <div id="display">Make me disappear! Click Hide button.</div>
        <div id="display2">Make it appear, Click Show button.</div>
    </body>
</html>
```

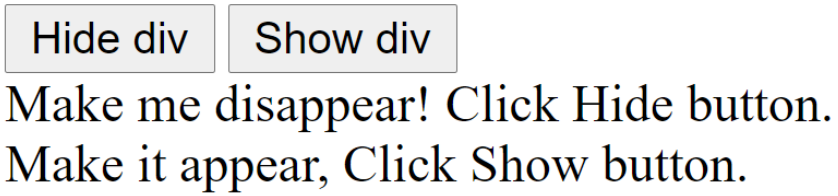
HIDE AND SHOW (JS FILE)

Step 2: Create the JavaScript file **hideshow.js**

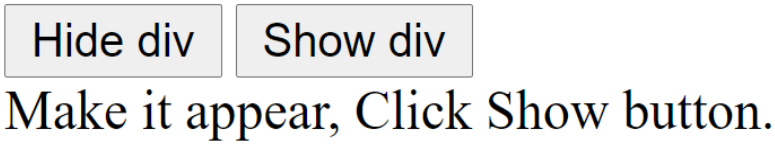
```
$(document).ready(function(){
    $("#btnHide").click(function(){
        $("#display").hide(2000);
    });
    $("#btnShow").click(function(){
        $("#display").show(2000);
    });
});
```

HIDE AND SHOW (OUTPUT)

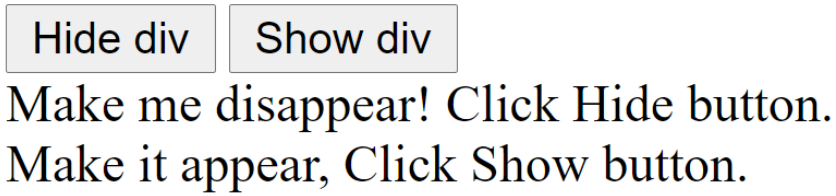
WHEN THE PAGE LOAD:



AFTER THE HIDE BUTTON WAS CLICKED:



AFTER THE SHOW BUTTON WAS CLICKED:



We want to hide the for two seconds. Then right after, we want it to be displayed again for two seconds.

WHAT SHOULD WE DO?

Answer: jQuery effects can use either:

a.) callback function

b.) jQuery chaining.

EXPLAINING CALLBACK

THE CALLBACK FUNCTION

The **Callback function** is a function executed right after the effect was finished on a given speed. Most of the time, callback functions are the last parameter of an effect.

For instance,

hide(speed, **callback**);

show(speed, **callback**);

CALLBACK (HTML FILE)

Step 1: Create the HTML file **callback.html**

```
<!doctype html>
<html>
    <head>
        <script src="jquery-3.3.1.min.js"></script>
        <script src="callback.js"></script>
    </head>
    <body>
        <button id="btnHideShow">Hide then Show div</button>
        <div id="display">Make me disappear and appear again!
        Click the button.</div>
    </body>
</html>
```

CALLBACK (JS FILE)

Step 2: Create the JavaScript file **callback.js**

```
$(document).ready(function(){
    $("#btnHideShow").click(function(){
        $("#display").hide(2000,function(){$(this).show(2000);});
    });
});
```

Callback Function

CALLBACK (OUTPUT)

WHEN THE PAGE LOAD:

AFTER THE "HIDE THEN SHOW" BUTTON WAS CLICKED:

AFTER THE DIV HAS BEEN HIDDEN FOR 2 SECONDS:

EXPLAINING CHAINING

THE JQUERY CHAINING

The **jQuery Chaining** allows us to run multiple jQuery methods (on the same element) within a single statement.

For instance,

```
.hide(2000).show(2000)
```

CHAINING (HTML FILE)

Step 1: Create the HTML file **chaining.html**

```
<!doctype html>
<html>
    <head>
        <script src="jquery-3.3.1.min.js"></script>
        <script src="chaining.js"></script>
    </head>
    <body>
        <button id="btnHideShow">Hide then Show div</button>
        <div id="display">Make me disappear and appear again!
        Click the button.</div>
    </body>
</html>
```

CHAINING (JS FILE)

Step 2: Create the JavaScript file **chaining.js**

```
$(document).ready(function(){
    $("#btnHideShow").click(function(){
        $("#display").hide(2000).show(2000);
    });
});
```

jQuery Chaining

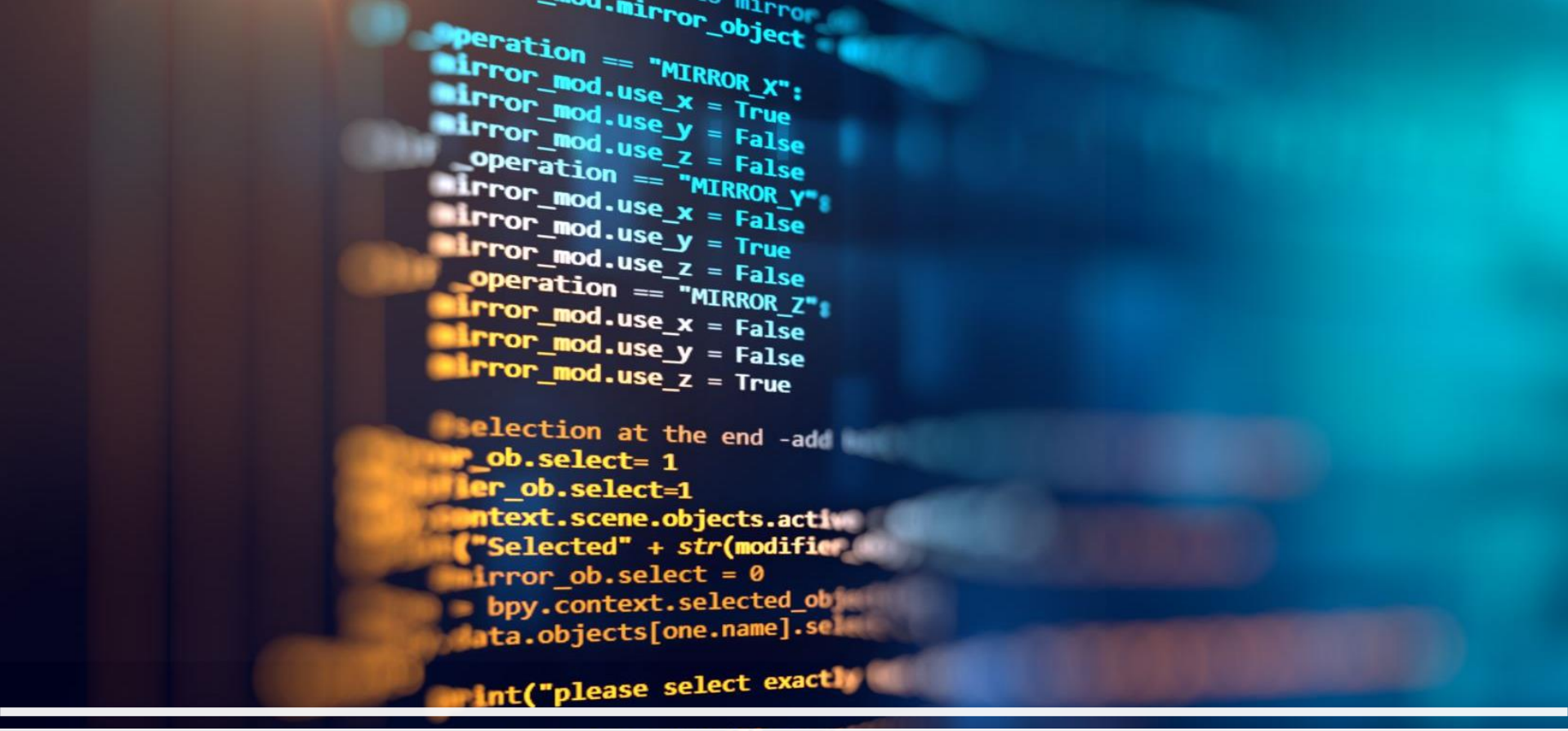
NOTE: We could also have added more method calls if needed.

CHAINING (OUTPUT)

WHEN THE PAGE LOAD:

AFTER THE "HIDE THEN SHOW" BUTTON WAS CLICKED:

AFTER THE DIV HAS BEEN HIDDEN FOR 2 SECONDS:



TOGGLE EFFECT

JQUERY TOGGLE EFFECT

The **toggle()** effect hides/shows the selected element with the given speed.

Syntax:

toggle([speed , object callback_function])

NOTE: toggle() effect transitions from hide to show and vice versa.

TOGGLE (HTML FILE)

Step 1: Create the HTML file **toggle.html**

```
<!doctype html>
<html>
    <head>
        <script src="jquery-3.3.1.min.js"></script>
        <script src="toggle.js"></script>
    </head>
    <body>
        <button id="btnHideShow">Hide/Show div</button>
        <div id="display">Make me disappear/appear, Click the button.</div>
    </body>
</html>
```

TOGGLE (JS FILE)

Step 2: Create the JavaScript file **toggle.js**

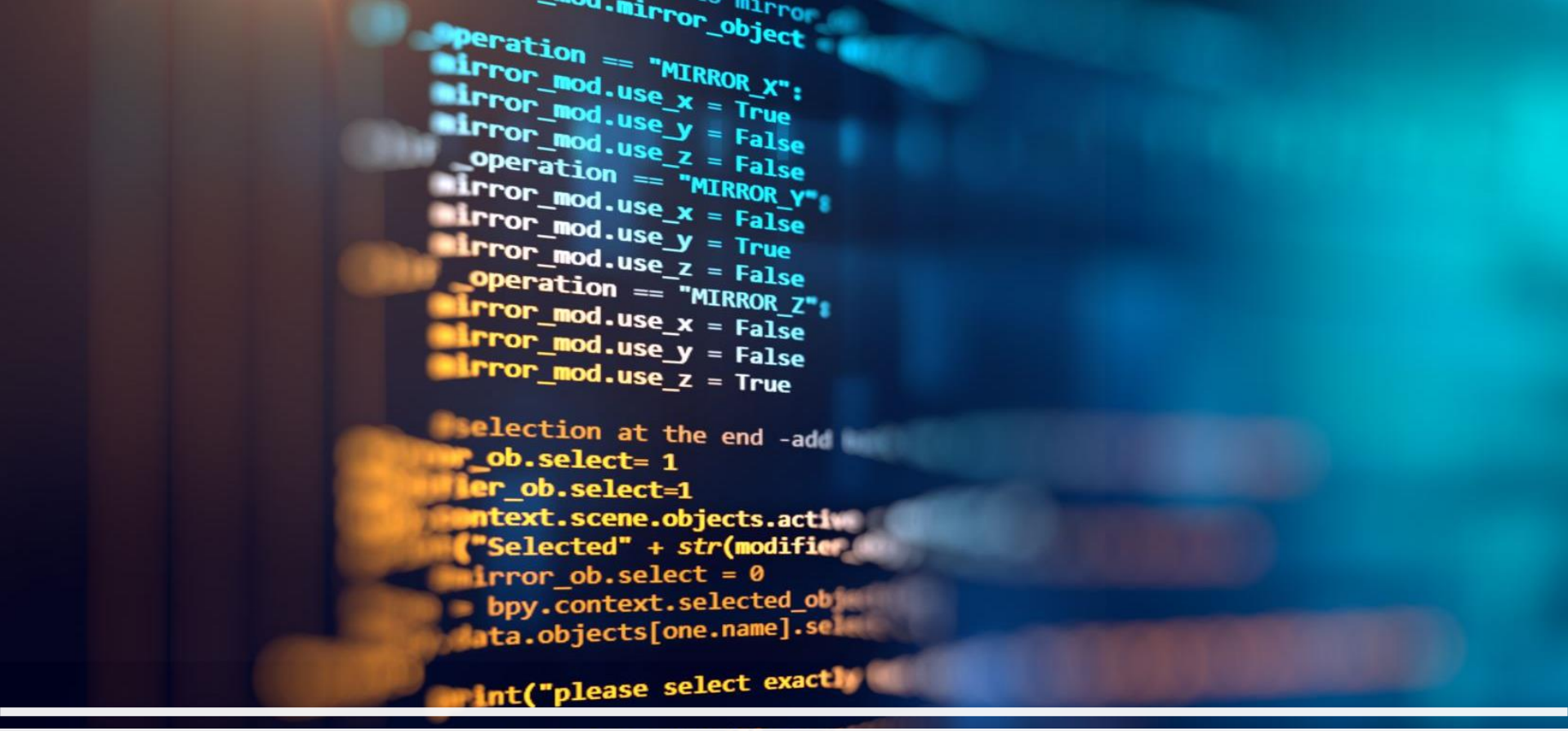
```
$(document).ready(function(){
    $("#btnHideShow").click(function(){
        $("#display").toggle(2000);
    });
});
```

TOGGLE (OUTPUT)

WHEN THE PAGE LOAD:

AFTER THE HIDE/SHOW BUTTON WAS CLICKED FOR THE FIRST TIME:

AFTER THE HIDE/SHOW BUTTON WAS CLICKED FOR THE SECOND TIME :



FADE EFFECTS

JQUERY FADE OUT EFFECT

The **fadeOut()** effect hides the selected element, with a fading effect, with the given speed.

Syntax:

fadeOut([speed , object callback_function])

NOTE: Make sure that the element you are trying to hide is visible on the page.

JQUERY FADE IN EFFECT

The **fadeIn()** effect shows the selected element, with a fading effect, with the given speed.

Syntax:

fadeIn([speed , object callback_function])

NOTE: Make sure that the element you are trying to show is hidden on the page.

JQUERY FADE TO EFFECT

The **fadeTo()** effect hides the selected element, with a fading effect, with the given speed and a given point of transparency.

Syntax:

fadeTo([speed , opacity , object callback_function])

Specifies the opacity to fade to. Must be a number between 0 and 1

JQUERY FADE TOGGLE EFFECT

The **fadeToggle()** effect shows/hides the selected element, with a fading effect, with the given speed.

Syntax:

fadeToggle([speed , object callback_function])

FADE (HTML FILE)

Step 1: Create the HTML file **fade.html**

```
<!doctype html>
<html>
    <head>
        <script src="jquery-3.3.1.min.js"></script>
        <script src="fade.js"></script>
    </head>
    <body>
        <button id="btnFadeOut">Fade Out</button>
        <button id="btnFadeIn">Fade In</button>
        <button id="btnFadeToggle">Fade Toggle</button>
        <br/><br/><img src="Thanos.jpg" id="thanos"
        style="height: 100px; width: 200px"></img>
        <div>Move mouse in/out the image for fadeTo effect.</div>
    </body>
</html>
```

FADE (JS FILE)

Step 2: Create the JavaScript file **fade.js**

```
$(document).ready(function(){
    $("#btnFadeOut").click(function(){
        $("#thanos").fadeOut(1000);
    });
    $("#btnFadeIn").click(function(){
        $("#thanos").fadeIn(1000);
    });
    $("#btnFadeToggle").click(function(){
        $("#thanos").fadeToggle(1000);
    });
    $("#thanos").mouseenter(function(){
        $("#thanos").fadeTo(1000, 0.50);
    });
    $("#thanos").mouseleave(function(){
        $("#thanos").fadeTo(1000, 1);
    });
});
```

FADE (OUTPUT)

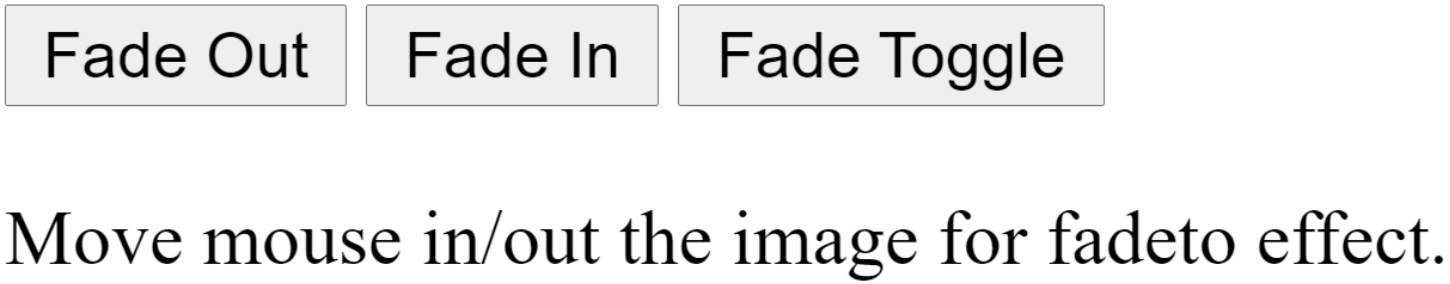
WHEN THE PAGE LOAD:

Fade Out | Fade In | Fade Toggle

Move mouse in/out the image for fadeto effect.

FADE (OUTPUT)

AFTER THE FADE OUT BUTTON WAS CLICKED:



FADE (OUTPUT)

AFTER THE FADE IN BUTTON WAS CLICKED:

Fade Out | Fade In | Fade Toggle

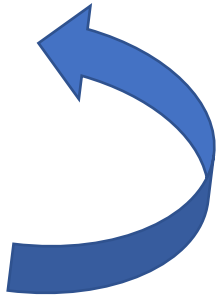
Move mouse in/out the image for fadeto effect.

FADE (OUTPUT)

AFTER THE FADE TOGGLE BUTTON WAS CLICKED:

Fade Out | Fade In | Fade Toggle

Move mouse in/out the image for fadeto effect.

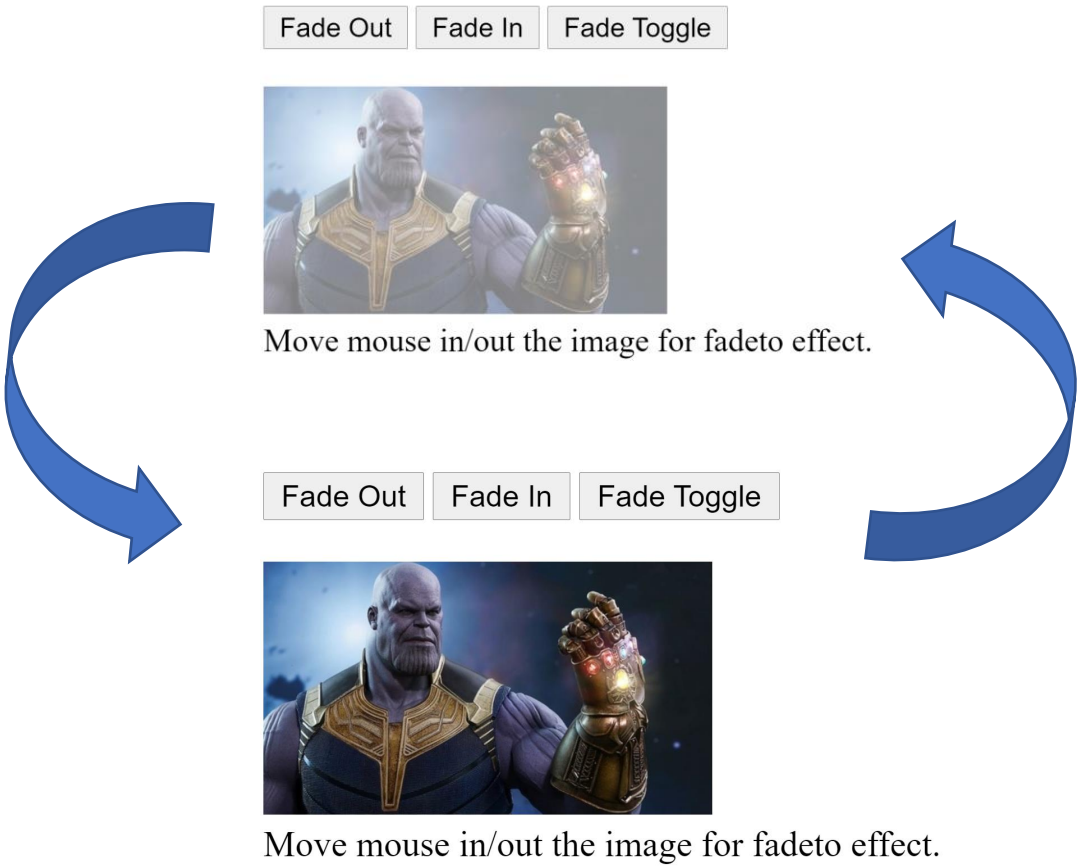


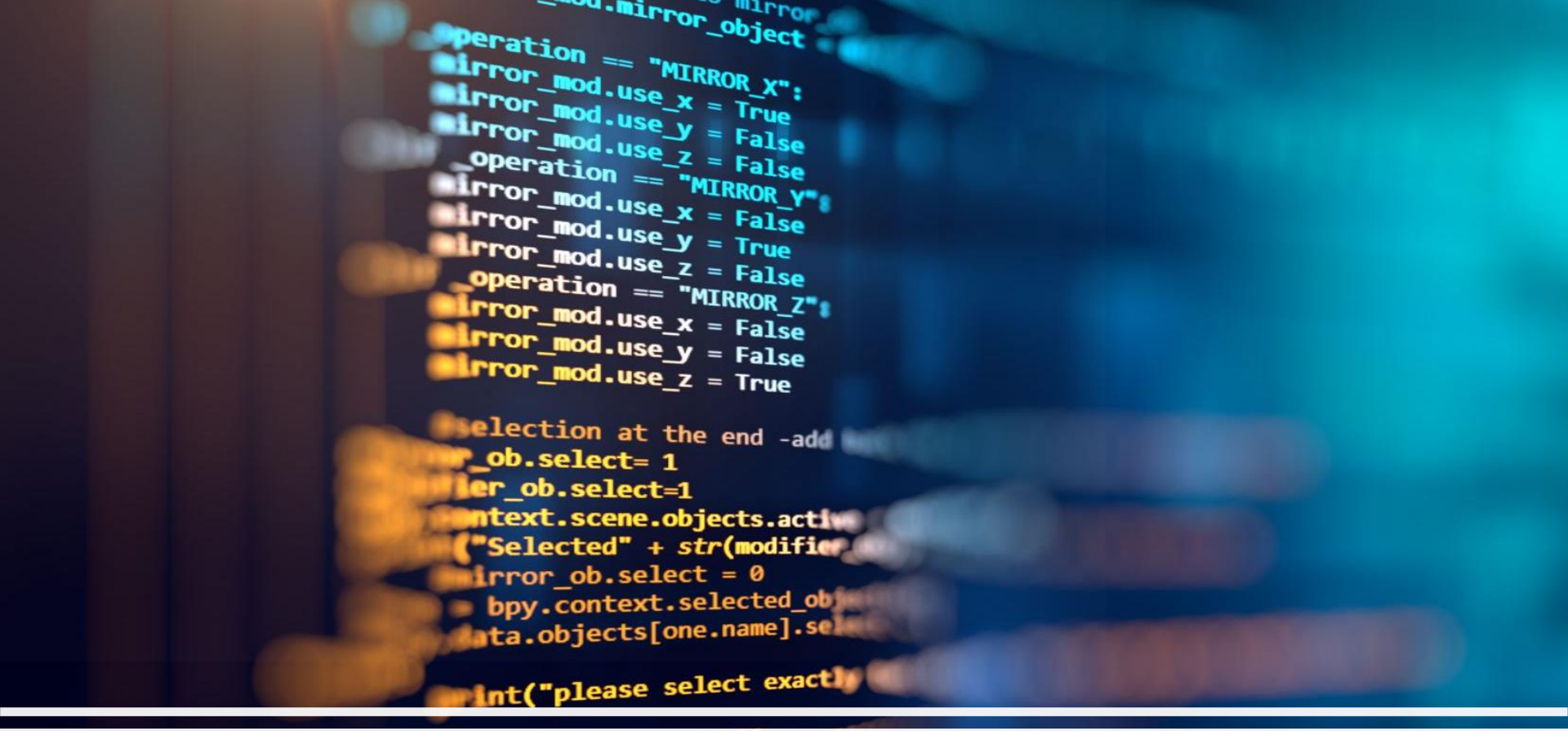
Fade Out | Fade In | Fade Toggle

Move mouse in/out the image for fadeto effect.

FADE (OUTPUT)

AFTER THE MOUSE ENTER/LEAVE THE IMAGE:





SLIDE EFFECTS

JQUERY SLIDE UP EFFECT

The **slideUp()** effect hides the selected element, with a sliding effect, with the given speed.

Syntax:

slideUp([speed , object callback_function])

JQUERY SLIDE DOWN EFFECT

The **slideDown()** effect shows the selected element, with a sliding effect, with the given speed.

Syntax:

slideDown([speed , object callback_function])

JQUERY SLIDE TOGGLE EFFECT

The **slideToggle()** effect shows/hides the selected element, with a sliding effect, with the given speed.

Syntax:

slideToggle([speed , object callback_function])

SLIDE (HTML FILE)

Step 1: Create the HTML file **slide.html**

```
<!doctype html>
<html>
    <head>
        <script src="jquery-3.3.1.min.js"></script>
        <script src="slide.js"></script>
    </head>
    <body>
        <button id="btnSlideUp">Slide Up</button>
        <button id="btnSlideDown">Slide Down</button>
        <button id="btnSlideToggle">Slide Toggle</button>
        <br/><br/><img src="Netflix.jpg" id="netflix"
        style="height: 100px; width: 200px"></img>
    </body>
</html>
```

SLIDE (JS FILE)

Step 2: Create the JavaScript file **slide.js**

```
$(document).ready(function(){
    $("#btnSlideUp").click(function(){
        $("#netflix").slideUp("fast");
    });
    $("#btnSlideDown").click(function(){
        $("#netflix").slideDown("slow");
    });
    $("#btnSlideToggle").click(function(){
        $("#netflix").slideToggle(1000);
    });
});
```

SLIDE (OUTPUT)

WHEN THE PAGE LOAD:

Slide Up | Slide Down | Slide Toggle

SLIDE (OUTPUT)

AFTER THE SLIDE UP BUTTON WAS CLICKED:

SLIDE (OUTPUT)

AFTER THE SLIDE DOWN BUTTON WAS CLICKED:

Slide Up | Slide Down | Slide Toggle

SLIDE (OUTPUT)

AFTER THE SLIDE TOGGLE BUTTON WAS CLICKED:


JQUERY HTML

JQUERY HTML

The **html()** property allows the getting and setting of text in an HTML container tags. html() property is the equivalent of JavaScript's innerHTML.

Syntax:

html([string value])

JQUERY HTML (HTML FILE)

Step 1: Create the HTML file **jhtml.html**

```
<!doctype html>
<html>
    <head>
        <script src="jquery-3.3.1.min.js"></script>
        <script src="jhtml.js"></script>
    </head>
    <body>
        <button id="btnChange">Change Text</button>
        <button id="btnAlert">Alert Text</button>
        <div id="display">Hello!</div>
    </body>
</html>
```

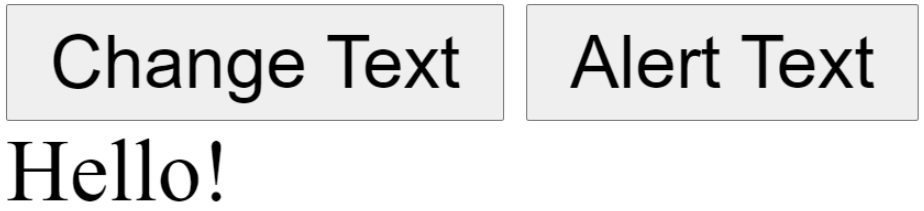
JQUERY HTML (JS FILE)

Step 2: Create the JavaScript file **jhtml.js**

```
$(document).ready(function(){
    $("#btnChange").click(function(){
        $("#display").html("Hi!");
    });
    $("#btnAlert").click(function(){
        str = $("#display").html();
        alert(str);
    });
});
```

JQUERY HTML (OUTPUT)

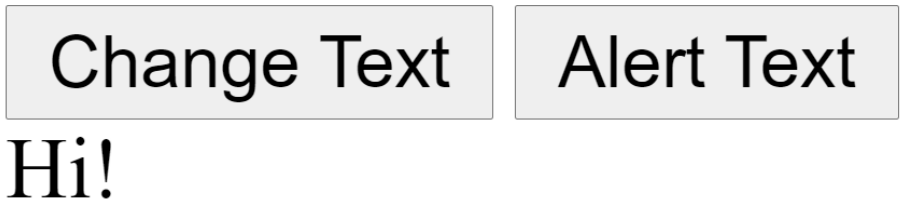
WHEN THE PAGE LOAD:



AFTER THE ALERT TEXT BUTTON WAS CLICKED:

JQUERY HTML (OUTPUT)

AFTER THE CHANGE TEXT BUTTON WAS CLICKED:



AFTER THE ALERT TEXT BUTTON WAS CLICKED:

SIZE SELECTION

JQUERY HTML (HTML FILE)

Step 1: Create the HTML file **size.html**

```
<!doctype html>
<html>
    <head>
        <script src="jquery-3.3.1.min.js"></script>
        <script src="size.js"></script>
    </head>
    <body>
        <select id="sizes">
            <option selected disabled>Select Size</option>
            <option value="100">Small</option>
            <option value="200">Medium</option>
            <option value="300">Large</option>
        </select>
        <label id="display">Price appears here.</label>
    </body>
</html>
```

JQUERY HTML (JS FILE)

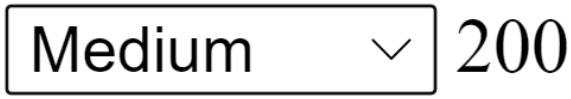
Step 2: Create the JavaScript file **size.js**

```
$(document).ready(function(){
    $("#sizes").change(function(){
        price = $("option:selected").val();
        $("#display").html(price);
    });
});
```

JQUERY HTML (OUTPUT)

WHEN THE PAGE LOAD:

AFTER A SIZE WAS SELECTED:



COURSE SELECTION

JQUERY HTML (HTML FILE)

Step 1: Create the HTML file **course.html**

```
<!doctype html>
<html>
    <head>
        <script src="jquery-3.3.1.min.js"></script>
        <script src="course.js"></script>
    </head>
    <body>
        Select course:
        <input type="radio" name="course" id="bsit" value="BSIT">
        <label for="bsit">BSIT</label>
        <input type="radio" name="course" id="blis" value="BLIS">
        <label for="blis">BLIS</label>
        <div id="display"></div>
    </body>
</html>
```

JQUERY HTML (JS FILE)

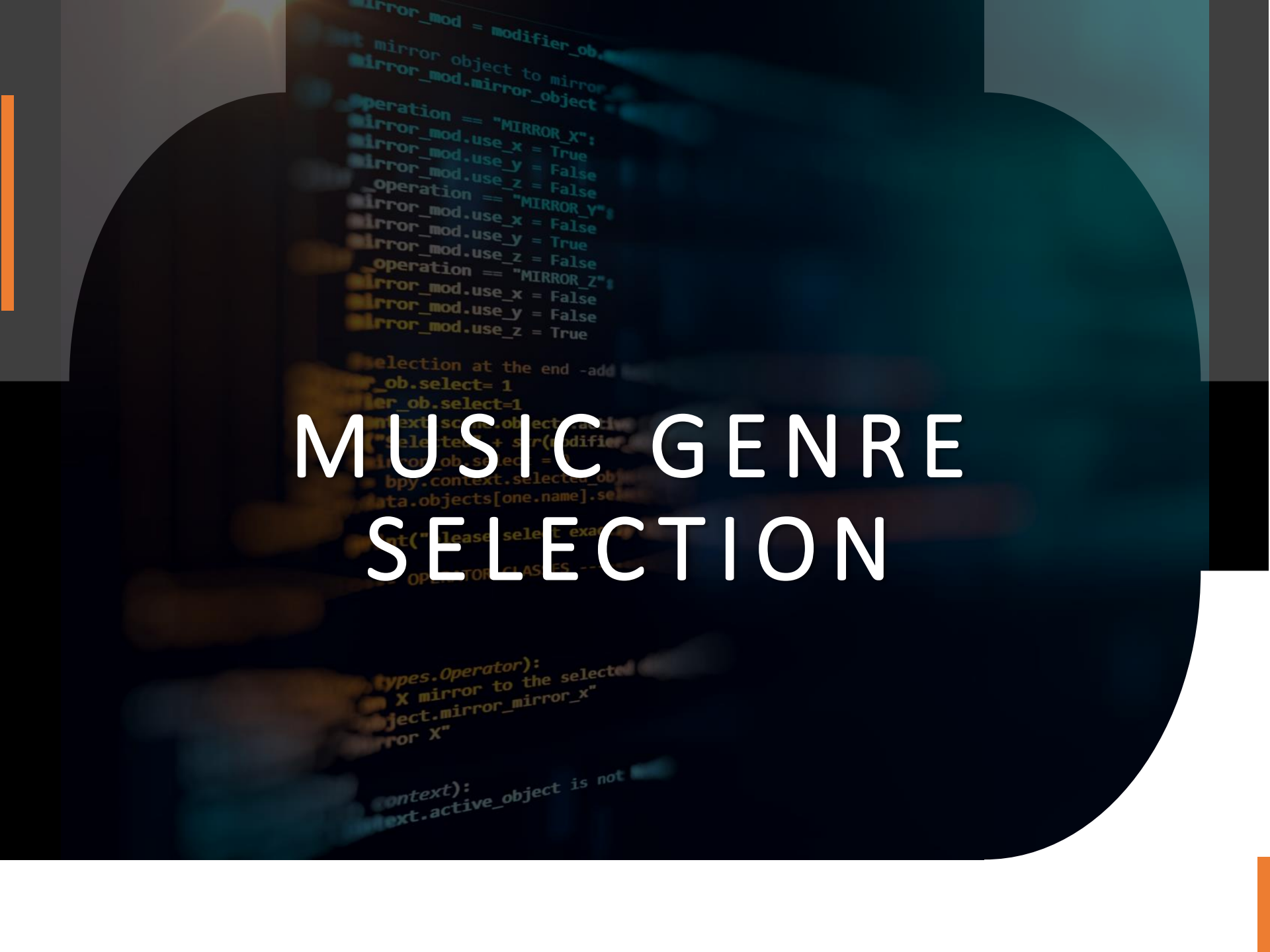
Step 2: Create the JavaScript file **course.js**

```
$(document).ready(function(){
    $("[name='course']").change(function(){
        course = $("[name='course']:checked").val();
        $("#display").html("The course selected is " + course);
    });
});
```

JQUERY HTML (OUTPUT)

WHEN THE PAGE LOAD:

AFTER A COURSE WAS SELECTED:



MUSIC GENRE SELECTION

JQUERY HTML (HTML FILE)

Step 1: Create the HTML file **genre.html**

```
<!doctype html>
<html>
    <head>
        <script src="jquery-3.3.1.min.js"></script>
        <script src="genre.js"></script>
    </head>
    <body>
        <input type="checkbox" value="Pop">Pop<br>
        <input type="checkbox" value="Rock">Rock<br>
        <input type="checkbox" value="R&B">R&amp;B<br>
        <input type="checkbox" value="Jazz">Jazz<br>
        <div id="display">You selected:</div>
    </body>
</html>
```

JQUERY HTML (JS FILE)

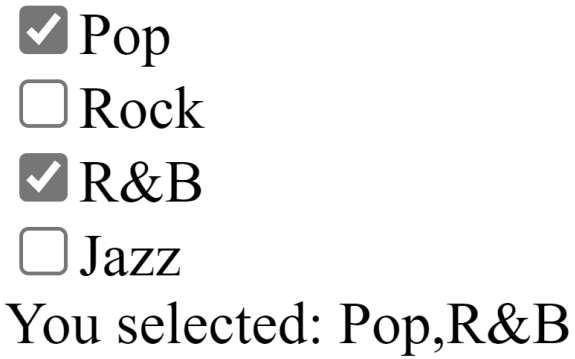
Step 2: Create the JavaScript file **genre.js**

```
$(document).ready(function(){
    $("[type='checkbox']").change(function(){
        var choice = [];
        $("[type='checkbox']:checked").each(function(){
            choice.push(this.value);
        });
        $("#display").html("You selected: " + choice);
    });
});
```

JQUERY HTML (OUTPUT)

WHEN THE PAGE LOAD:

AFTER POP AND R&B CHECKBOXES WERE CLICKED:




JQUERY CSS

JQUERY CSS

The **css()** property allows the altering of CSS.

Syntax:

css([string property , string value])

NOTE: All hyphenated words in CSS properties shall be converted to camel case, with no spaces provided.

For instance,

background-color → backgroundColor

text-align → textAlign

JQUERY CSS (HTML FILE)

Step 1: Create the HTML file **jcss.html**

```
<!doctype html>
<html>
    <head>
        <script src="jquery-3.3.1.min.js"></script>
        <script src="jcss.js"></script>
    </head>
    <body>
        <div id="display">Hello World!</div>
    </body>
</html>
```

JQUERY CSS (JS FILE)

Step 2A: Create the JavaScript file **jcss.js**

```
$(document).ready(function(){
    $("#display").css("backgroundColor","royalblue");
    $("#display").css("border","1px solid lightblue");
    $("#display").css("textAlign","center");
    $("#display").css("color","white");
    $("#display").css("height","auto");
    $("#display").css("width","auto");
});
```

JQUERY CSS (JS FILE)

Step 2B: Create the JavaScript file **jcss2.js**

```
$(document).ready(function(){
    $("#display").css({
        backgroundColor: "royalblue",
        border: "1px solid lightblue",
        textAlign: "center",
        color: "white",
        height: "auto",
        width: "auto"
    });
});
```

NOTE: This JavaScript file contains the same CSS styles from the previous slide but uses different format in writing the CSS properties and values.

JQUERY CSS (OUTPUT)

WHEN THE PAGE LOAD:

LIGHT MODE AND DARK MODE

JQUERY CSS (HTML FILE)

Step 1: Create the HTML file **darkmode.html**

```
<!doctype html>
<html>
    <head>
        <script src="jquery-3.3.1.min.js"></script>
        <script src="darkmode.js"></script>
    </head>
    <body>
        <button id="btnToggle">Dark Mode</button>
        <div id="display">
            <!--INSERT LOREM IPSUM HERE-->
        </div>
    </body>
</html>
```

JQUERY CSS (JS FILE)

Step 2A: Create the JavaScript file **darkmode.js**

```
$(document).ready(function(){
    $("#btnToggle").click(function(){
        mode = $("#btnToggle").html();
        if(mode=="Dark Mode"){
            $("#display").css({
                backgroundColor: "black",
                color: "white"
            });
            $("#btnToggle").html("Light Mode");
        }
        else if(mode=="Light Mode"){
            $("#display").css({
                backgroundColor: "white",
                color: "black"
            });
            $("#btnToggle").html("Dark Mode");
        }
    });
});
```

JQUERY CSS (OUTPUT)

WHEN THE PAGE LOAD:

Dark Mode

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took agalley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with therelease of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

AFTER THE BUTTON DARK MODE WAS CLICKED:

Light Mode

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took agalley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with therelease of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.


JQUERY ANIMATE

JQUERY ANIMATE

The **animate()** method allows the altering of CSS and shows effect in a given speed.

Syntax:

animate([object css , speed , easing , callback_function])

Linear or Swing

JQUERY ANIMATE (HTML FILE)

Step 1: Create the HTML file **animate.html**

```
<!doctype html>
<html>
    <head>
        <script src="jquery-3.3.1.min.js"></script>
        <script src="animate.js"></script>
        <link rel="stylesheet" href="animate.css">
    </head>
    <body>
        <div id="display"><img src="Home.png" id="img1"/>
        <img src="Users.png" id="img2"/>
        <img src="Files.png" id="img3"/>
        <img src="Exit.png" id="img4"/></div>
    </body>
</html>
```

JQUERY ANIMATE (JS FILE)

Step 2: Create the JavaScript file **animate.js**

```
$(document).ready(function(){
    $("img").mouseenter(function(){
        $(this).animate({height: "30px", width: "30px"},100);
    });
    $("img").mouseleave(function(){
        $(this).animate({height: "25px", width: "25px"},100);
    });
});
```

JQUERY ANIMATE (CSS FILE)

Step 3: Create the CSS file **animate.css**

```
img{height: 25px; width: 25px;}

#img1{position:absolute; top:15px; left: 60px;}
#img2{position:absolute; top:15px; left:120px;}
#img3{position:absolute; top:15px; left:180px;}
#img4{position:absolute; top:15px; left:240px;}

#display{background-color: gold;
height:40px; width:100%};
```

JQUERY ANIMATE (OUTPUT)

WHEN THE PAGE LOAD:

AFTER THE MOUSE ENTERED AN ICON:

ZOOM IN AND ZOOM OUT

JQUERY ANIMATE (HTML FILE)

Step 1: Create the HTML file **zoom.html**

```
<!doctype html>
<html>
    <head>
        <script src="jquery-3.3.1.min.js"></script>
        <script src="zoom.js"></script>
        <link rel="stylesheet" href="zoom.css">
    </head>
    <body>
        <button id="btnMinus">Zoom Out</button>
        <button id="btnPlus">Zoom In</button><br/><br/>
        <img src="Denjiro.jpg" id="denjiro"></div>
    </body>
</html>
```

JQUERY ANIMATE (JS FILE)

Step 2: Create the JavaScript file **zoom.js**

```
$(document).ready(function(){
    $("#btnPlus").click(function(){
        $("#denjiro").animate({height:"+=100px",width:"+=100px"},800);
    });
    $("#btnMinus").click(function(){
        $("#denjiro").animate({height:"-=100px",width:"-=100px"},800);
    });
});
```

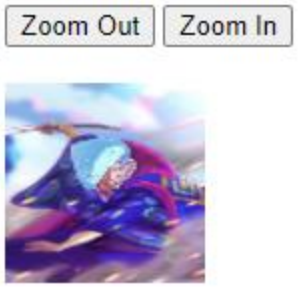
JQUERY ANIMATE (CSS FILE)

Step 3: Create the CSS file **zoom.css**

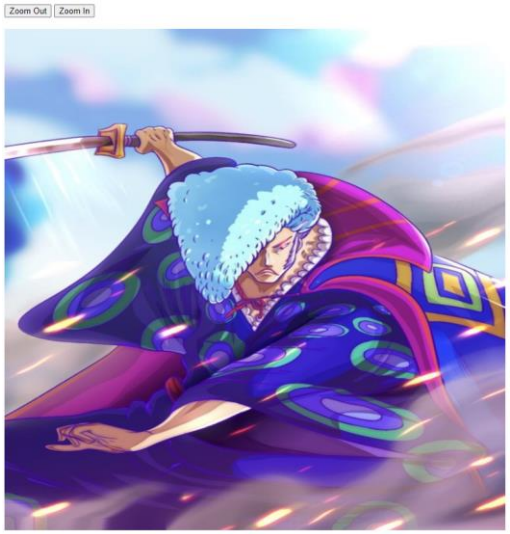
```
#denjiro{
    max-height: 800px;
    min-height: 100px;
    height: 100px;
    max-width: 800px;
    min-width: 100px;
    width: 100px;
}
```

JQUERY ANIMATE (OUTPUT)

WHEN THE PAGE LOAD:



AFTER THE ZOOM IN BUTTON WAS CLICKED SEVEN (7) TIMES:




END OF LESSON