jQuery

Introduction:

(a) This guide compiles the information from the following video link:
https://www.youtube.com/watch?v=BWXggB-T1jQ
If you are using this guide to learn jQuery, then I would advise you follow along this video for more details.

(b) jQuery is an extremely popular library of JavaScript. With JavaScript knowledge, everything in jQuery can be replicated, but it may be much more difficult (longer code). I advise learning JavaScript (as well as HTML and maybe CSS too) before learning jQuery.

(c) To include the jQuery library, the following HTML snippet must be included in your HTML document:
<script src=“https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js”></script>
This is the jQuery version 3.31, and it is the current version as of the date this document was created. Find the most up-to-date version at:
https://developers.google.com/speed/libraries/

(d) If you are creating your own ReactJS webserver, then jQuery can be included in the following way, instead of the way provided in part (c):

i. In your terminal, run the command: “npm install jquery −−save”.

ii. At the top of your javascript files include: “import $ from ‘jquery”’ when you want to code with jQuery in that file.

iii. For further information on properly implementing jQuery with React, check out the page: https://reactjs.org/docs/integrating-with-other-libraries.html

Targeting:

(a) Javascript begins between two script tags. This is where jQuery code would also begin. One jQuery code formatting would be:

```
$("document").ready(function(){
//The webpage is targeted (document is targeted)
//The document is targeted when it has just loaded (ready)
//When the criteria is met, the code in the function occurs
});
```

(b) Here are different examples of ways we may target html tags:

```
$("document") //The whole document
```

```
$("#wrapper1") //An element with the id wrapper1
$(".class1") //An element with the class class1
$("h1") //All h1 elements
$("#someId p") //All paragraph tags within the element of id someId
$("p + a") //A link that follows a paragraph (only targets link)
$("div > span") //Every span that is a child of any div
$("#anOrderedList li:nth-child(3)") //The 3rd list item of the list with id anOrderedList
$("li[name]") //All list elements that have the name attribute
$("input[type=‘text’]#id3") //An input element that is of type text that has the id id3
$("a[href*=‘google’]") //A link whose href attribute contains the string ‘google’
$("img[alt^=‘NTT’]") //The image with alt attribute that begins with the string ‘NTT’
$("a[href$=‘pdf’]") //A link with the href attribute that ends with the string ‘pdf’
$("#table1 tr:even") //Targets the even rows of the table with id table1
$("#table1 tr:odd") //Targets the odd rows instead
$("#table1 tr:first") //First row
$("#table1 tr:last") //Last row
$("#orderedList1 li:not(:contains(and))") //Ordered list rows of the OL with id orderedList1 that do not contain the string ‘and’
$("a:contains(gravy)") //Links that contain the string ‘gravy’
$("p:has(i)") //All paragraphs that contain italics tags
$(this) //When within a function, the keyword ‘this’ will target the element that the function is targeting
```

Functions:

(a) Here are some functions that can be used on targeted elements:

```
.ready() //When this element has loaded
.css(String) //For editing CSS
.css({String}) //For editing a CSS property with mutliple inputs
.html(String) //Edits the inner html (needs tags) to what is inside
.val(String) //Edits the value of inputs, similar to the .html() for inputs
.hide() //Hides the element
.append(String) //Adds text to the end of targeted text
.prepend(String) //Adds text to the beginning of targeted text
.before(String) //Adds html tag(s) before another targeted element
.after(String) //Adds html tag(s) after another targeted element
.click(function(){}) //Triggers when clicked
```

```
.click(function(e){}) //Triggers when clicked, but also passes the event as e. The events attributes can be accessed.
.remove() //Removes the element
.replaceWith(String) //Replaces with some other element
.each(function(index)) //Similar to a for-each loop, and will call a function on each index of what is targeted
.addClass(String) //Add a class to an element(s)
.toggleClass(String) //Toggles a class on and off
.attr(String) //Returns the value of the given attribute
.attr(String1, String2) //Change the attribute in String1 to the value of String2
.mouseover(function(){}) //Occurs on mouseing over an element
.mouseout(function(){}) //Occurs on mousing off of an element
.hover(function(){}, function(){}) //The first function handles mouseover and the second function handles mouseout
.dblclick(function(){}) //Occurs on double click of an element
.mousemove(function(){}) //Occurs when the pointer changes position
.mousemove(function(e){}) //Occurs when the pointer moves, but the event of moving is passed as well
.keypress(function(){}) //Occurs when a key is pressed
.keypress(function(e){}) //Occurs when a key is pressed, but also passes the key pressing event
.blur(function(){}) //Occurs when an element is clicked off of after being clicked on
.change(function(){}) //Occurs when an input's values change
.focus(function(){}) //When an input box has been clicked
.select(function(){}) //When text in an input is highlighted
.on(String, var, function) //A triggering event detailed in String such as click, mouseover, etc., will trigger a function which takes as inputs the variable var.
.fadeOut(int) //Fades an element into deletion in int milliseconds
.fadeOut(String) //Fades an element into deletion in String speed (‘slow’, ‘fast’, etc.)
.fadeToggle(int) //Fades an element into deletion or creates an element and fades it in in int milliseconds (also has the String option like fadeOut)
.fadeTo(int, double) //Fades to double percentage (.5 would be 50 percent opacity) in int milliseconds (also has the String option like fadeOut)
```

(b) Things that are targeted have an output; if we target a paragraph then the output would be that paragraph. We can redirect these outputs into any variable or function. This means that the first example syntax from can be used as a get and a set.

(c) There are many more animation types of similar form to *fadeOut*, *fadeToggle*, etc. I would advise inputing your animations in your CSS code as the syntax is simpler.