$("h1")*/\*The return is always a list regardless of the amount of h1, like querySelectorAll()\*/*

$("li a")

$("h1").css("color", "yellow")

$ ("a").css("font-size", "40px")

$("div").css("background", "purple")

$("div.highlight").css("width", "200px");

$("#third").css("border", "4px solid orange");

$("div:first-of-type").css("color", "pink");

**var** styles = {

color: "red",

background: "pink",

border: "2px solid purple"

}

$("h1").css(styles)

Note: jQuery can select a list and style it. However, in DOM:

**var** lis = document.querySelectorAll("li");

**for**(**var** i = 0; i < lis.length; i++){

lis[i].addEventListener("click", **function**(){

**this**.style.color = "pink" *//this denotes the lis[i]*

});

}

$("li").css({

fontSize: "10px",

border: "3px dashed purple",

background: "rgba(89, 45, 20, 0.5)"

})

Method

$("ul").text()*/\*textContent in DOM\*/*

"

Skit

Skaboe

Twix

"

$("li").text("Rusty")

[<li>Rusty</li>, <li>Rusty</li>, <li>Rusty</li>]

###############################################

$("ul").html() */\*like innerHTML()\*/*

"

<li>Rusty</li>

<li>Rusty</li>

<li>Rusty</li>

"

$("ul").html("<li>Hacked your UL</li><li>Cute</li>")

$("li").html("<a href='Googel.com'>Google</a>")

"

Google

Google

"

######################################################

$("img").attr("src")

<https://www.w3schools.com/w3images/fjords.jpg>

$("img").attr("src", "https://www.w3schools.com/w3images/lights.jpg")

$("input").attr("type")

"text"

$("input").attr("type", "color")

$("img:first-of-type").attr("src", "https://www.w3schools.com/w3images/lights.jpg")

$("img").last().attr("src", "https://www.w3schools.com/w3images/lights.jpg")

#########################################################

$("input").val()

""

//After typing Quan Luu in textbox

$("input").val()

"Quan Luu"

$("input").val("Rusty")*//update the input to Rusty*

####################################################

CSS File or style tag:

.correct{

color: green;

}

.wrong{

color: red;

background: pink;

}

$("h1").addClass("correct")

$("h1").removeClass("wrong")

$("li").first().toggleClass("correct")

########################################################

$("h1").click(**function**(){

alert("h1 clicked!");

});

$("button").click(**function**(){

$(**this**).css("background", "pink");

});

$("h1").on("click", **function**(){

$(**this**).css("color", "purple");

});

$("button").on("mouseenter", **function**(){

$(**this**).css("font-weight", "bold");

});

$("button").on("mouseleave", **function**(){

$(**this**).css("font-weight", "normal");

});

$("input[type='text']").keypress(**function**(){

console.log("You pressed a key!")

});

$("input[type='text']").keypress(**function**(event){

console.log(event)

});

$("input[type='text']").keypress(**function**(event){

**if**(event.which === 13){

alert("You Hit Enter");

}

});

Note: The console will wait fadeout, after that implement function()

$("button").on("click", **function**(){

$("div").fadeOut(1000, **function**(){

$(**this**).remove();

});

});

$("button").on("click", **function**(){

$("div").fadeToggle(1000, **function**(){

});

});

$("button").on("click", **function**(){

$("div").slideDown();

});

$("button").on("click", **function**(){

$("div").slideUp();

});

$("button").on("click", **function**(){

$("div").slideToggle(1000, **function**(){

console.log("Slide is done");

$(**this**).remove();

});

});

Practice: 63ToDoList

/\* jquery only add click/ listerner only when the code runs, so if you write like code below, jquery only add code in exist li at the time the code runs. When you add more li in UI, jquery will not add click/ listerner for that li\*/

$("li").on("click", **function**(){

$(**this**).toggleClass("completed");*//toggle in li which parent is ul*

})

/\*To avoid:\*/

$("ul").on("click", "li", **function**(){

$(**this**).toggleClass("completed");*//toggle in li which parent is ul*

})

/\*this shows li and toggleClass will be in effect in ul which includes all li\*/

#################################################

/\* When clicking on Span and the li is parent of span which means you click li, ul is parent of li which means you click ul\*/

$("span").click(**function**(){

alert("Clicked on the span");

})

/\*To avoid, we use event.stopPropagation\*/

$("ul").on("click", "span", **function**(event){

$(**this**).parent().fadeOut(500, **function**(){

$(**this**).remove();

});

event.stopPropagation();

});

###################################################

$("input[type='text']").keypress(**function**(event){

**if**(event.which === 13){

**var** todoText = $(**this**).val();

$(**this**).val("");

$("ul").append("<li><span><i class='fa fa-trash'></i></span> " + todoText + "</li>");

}

})

$(".fa-plus").click(**function**(){

$("input[type='text']").fadeToggle();

})

Practice: 64Patatap