vanilla-js-samples (mrwhitedeveloper)

jQuery: click event

```
$(document).ready(function() {
  $(".demo").click(function() {
    $(this).hide(200);
  });
});
```

Vanilla JavaScript: click event

```
document.addEventListener('DOMContentLoaded', function() {
  var demoElements = document.querySelectorAll('.demo');
  demoElements.forEach(function(element) {
    element.addEventListener('click', function() {
      this.style.display = 'none';
    });
  });
});
```

jQuery: each

```
$(".demo").each(function() {
  document.write($(this).text() + "\n");
});
```

Vanilla JavaScript: each

```
document.addEventListener('DOMContentLoaded', function() {
  var demoElements = document.querySelectorAll('.demo');
  demoElements.forEach(function(element) {
    document.write(element.textContent + "\n");
  });
});
```

jQuery: Trigger

```
$("a#mylink").trigger("click");
```

Vanilla JavaScript: Trigger

```
document.addEventListener('DOMContentLoaded', function() {
  var myLink = document.querySelector("a#mylink");
  myLink.dispatchEvent(new Event('click'));
});
```

jQuery: noConflict

```
var jq = $.noConflict();
jq(document).ready(function() {
  jq("#demo").text("Hello World!");
});
```

Vanilla JavaScript: noConflict

```
document.addEventListener('DOMContentLoaded', function() {
  var jq = function(selector) {
    return document.querySelector(selector);
  };

  jq(document).addEventListener('DOMContentLoaded', function() {
    jq("#demo").textContent = "Hello World!";
  });
});
```

jQuery: Selectors

```
$("*")                     // all elements
$("p.demo")                // <p> elements with class="demo"
$("p:first")               // the first <p> element
$("p span")                // span, descendant of p
$("p > span")              // span, direct child of p
$("p + span")              // span immediately preceded by a p
$("p ~ span")              // span immediately preceded by p
$("ul li:first")           // the first <li> element of the first <ul>
$("ul li:first-child")     // the first <li> element of every <ul>
$("ul li:nth-child(3)")    // third child
$("[href]")                // any element with an href attribute
$("a[target='_blank']")    // <a> elements with a target="_blank" attribute
$("a[target!='_blank']")   // <a> elements with a target attribute value oth
er than "_blank"
$(":input")                // all form elements
$(":button")               // <button> and <input> elements of type="button"
$("tr:even")               // even <tr> elements
$("tr:odd")                // odd <tr> elements
$("span:parent")           // elements which have child elements
$("span:contains('demo')") // elements containing the specified text
```

Vanilla JavaScript: Selectors

```
document.querySelectorAll("*")                                   // all elemen
ts
document.querySelectorAll("p.demo")                              // <p> elemen
ts with class="demo"
document.querySelector("p")                                      // the first
<p> element
document.querySelectorAll("p span")                              // all <span>
elements inside <p>
document.querySelectorAll("p > span")                            // all <span>
elements that are direct children of <p>
document.querySelectorAll("p + span")                            // all <span>
elements immediately preceded by <p>
document.querySelectorAll("p ~ span")                            // all <span>
elements preceded by <p>
document.querySelectorAll("ul:first-of-type li:first-child")    // the first
```

```
<li> element of the first <ul>
document.querySelectorAll("ul li:first-child")                  // the first
<li> element of every <ul>
document.querySelectorAll("ul li:nth-child(3)")                 // the third
child of every <ul> (index starts from 1)
document.querySelectorAll("[href]")                             // any elemen
t with an href attribute
document.querySelectorAll("a[target='_blank']")                 // <a> elemen
ts with a target="_blank" attribute
document.querySelectorAll("a:not([target='_blank'])")           // <a> elemen
ts with a target attribute value other than "_blank"
document.querySelectorAll(":input")                             // all form e
lements
document.querySelectorAll("button, input[type='button']")       // <button> a
nd <input> elements of type="button"
document.querySelectorAll("tr:nth-child(even)")                 // all even <
tr> elements (index starts from 1)
document.querySelectorAll("tr:nth-child(odd)")                  // all odd <t
r> elements (index starts from 1)
document.querySelectorAll("span:only-child")                    // <span> ele
ments that are the only child of their parent
document.querySelectorAll("span:not(:empty)")                   // <span> ele
ments that are not empty
document.querySelectorAll("span:contains('demo')")             // elements co
ntaining the specified text (custom function required)
```

jQuery: Actions

```
$(this).hide();        // Hide the current element
$("div").hide();       // Hide all <div> elements
$(".demo").hide();     // Hide all elements with class="demo"
$("#demo").hide();     // Hide the element with id="demo"
```

Vanilla JavaScript: Actions

```
// Assuming you have a reference to the current element as 'currentElement'
currentElement.style.display = 'none';

// Hide all <div> elements
var divElements = document.querySelectorAll('div');
divElements.forEach(function(element) {
  element.style.display = 'none';
});

// Hide all elements with class="demo"
var demoElements = document.querySelectorAll('.demo');
demoElements.forEach(function(element) {
  element.style.display = 'none';
});
```

```
// Hide the element with id="demo"
var demoElement = document.getElementById('demo');
if (demoElement) {
  demoElement.style.display = 'none';
}
```

jQuery: Traversing

```
$("#demo").parent();                // accessing direct parent
$("span").parent().hide();          // changing parent color
$("#demo").parents();               // all ancestors of the element
$("#demo").parentsUntil("#demo2");  // all ancestors between two - demo is in
side demo2
$("#demo").children();              // all direct children
$("#demo").children(".first");      // all direct children having a specified
class
$("#demo").find("span");            // all span elements inside #demo
$("#demo").find("*");               // all descendants
$("#demo").siblings("span");        // span siblings of #demo
$("#demo").next();                  // the next sibling
$("p").nextAll();                   // all next siblings
$("#demo").nextUntil("#demo2");     // siblings between two arguments
$("#demo").prev();                  // the previous sibling
$("p").prevAll();                   // all previous siblings
$("#demo").prevUntil("#demo2");     // previous siblings between two argument
s
```

Vanilla JavaScript: Traversing

```
document.getElementById("demo").parentNode;                            // acc
essing direct parent

// Changing parent color (assuming "span" elements have their own CSS class)
var spans = document.querySelectorAll("span");
spans.forEach(function(span) {
  var parent = span.parentNode;
  if (parent) {
    parent.style.color = "red";
  }
});

// All ancestors of the element
var allAncestors = [];
var currentElement = document.getElementById("demo");
while (currentElement.parentNode) {
  allAncestors.push(currentElement.parentNode);
  currentElement = currentElement.parentNode;
}
```

```
// All ancestors between two elements (demo is inside demo2)
var ancestorsBetween = [];
var startElement = document.getElementById("demo");
var endElement = document.getElementById("demo2");
currentElement = startElement;
while (currentElement && currentElement !== endElement) {
  ancestorsBetween.push(currentElement.parentNode);
  currentElement = currentElement.parentNode;
}

// All direct children
var directChildren = document.getElementById("demo").children;

// All direct children having a specified class (e.g., "first")
var childrenWithClass = document.querySelectorAll("#demo .first");

// All span elements inside #demo
var spansInsideDemo = document.getElementById("demo").querySelectorAll("span"
);

// All descendants
var allDescendants = document.getElementById("demo").querySelectorAll("*");

// All siblings of span elements
var spanSiblings = document.querySelectorAll("span ~ *");

// The next sibling
var nextSibling = document.getElementById("demo").nextElementSibling;

// All next siblings
var allNextSiblings = [];
currentElement = document.getElementById("demo");
while (currentElement.nextElementSibling) {
  allNextSiblings.push(currentElement.nextElementSibling);
  currentElement = currentElement.nextElementSibling;
}

// Next siblings between two elements (demo is inside demo2)
var nextSiblingsBetween = [];
currentElement = document.getElementById("demo");
while (currentElement && currentElement !== endElement) {
  nextSiblingsBetween.push(currentElement.nextElementSibling);
  currentElement = currentElement.nextElementSibling;
}
```

```
// The previous sibling
var prevSibling = document.getElementById("demo").previousElementSibling;



// All previous siblings
var allPrevSiblings = [];
currentElement = document.getElementById("demo");
while (currentElement.previousElementSibling) {
  allPrevSiblings.push(currentElement.previousElementSibling);
  currentElement = currentElement.previousElementSibling;
}

// Previous siblings between two elements (demo is inside demo2)
var prevSiblingsBetween = [];
currentElement = document.getElementById("demo");
while (currentElement && currentElement !== endElement) {
  prevSiblingsBetween.push(currentElement.previousElementSibling);
  currentElement = currentElement.previousElementSibling;
}
```

jQuery: Filtering

```
$("span strong").first();   // first strong in first span
$("span strong").last();    // last strong in last span
$("div").eq(9);             // element with a specific index
$("div").filter(".big");    // all div elements with .big class
$("div").not(".big");       // opposite of filter
```

Vanilla JavaScript: Filtering

```
// First strong in the first span
var firstStrongInFirstSpan = document.querySelector("span strong");

// Last strong in the last span
var spans = document.querySelectorAll("span");
var lastStrongInLastSpan = spans[spans.length - 1].querySelector("strong");

// Element with a specific index (e.g., 9)
var divs = document.querySelectorAll("div");
var elementAtIndex9 = divs[9];

// All div elements with the .big class
var divsWithBigClass = document.querySelectorAll("div.big");

// Opposite of filter (All div elements without the .big class)
var divsWithoutBigClass = document.querySelectorAll("div:not(.big)");
```

jQuery: Ajax

```
$("#demo").load("file.txt h1.main"); // returns the h1 tag in the text file

$("#demo").load("file.txt", function(responseTxt, statusTxt, xhr) {  // callback function
  if (statusTxt == "success") {
    document.write("Content loaded successfully!");
  } else {
    document.write("Error: " + xhr.status + ": " + xhr.statusText);
  }
});
```

Vanilla JavaScript: Ajax

```
// Equivalent to $("#demo").load("file.txt h1.main")
fetch("file.txt")
  .then(response => response.text())
  .then(html => {
    var tempDiv = document.createElement("div");
    tempDiv.innerHTML = html;
    var h1Element = tempDiv.querySelector("h1.main");
    document.getElementById("demo").innerHTML = h1Element ? h1Element.outerHTML : "No h1.main found in the file.";
  })
  .catch(error => console.error("Error:", error));

// Equivalent to the callback function in $("#demo").load("file.txt", function(responseTxt, statusTxt, xhr) {...})
fetch("file.txt")
  .then(response => {
    if (response.ok) {
      return response.text();
    } else {
      throw new Error("Network response was not ok");
    }
  })
  .then(responseTxt => {
    document.write("Content loaded successfully!");
  })
  .catch(error => {
    console.error("Error:", error);
  });
```

jQuery: get()

```
$.get("demo.asp", function(data, status) {
  document.write("Data: " + data + "\nStatus: " + status);
});
```

Vanilla JavaScript: get()

```
fetch("demo.asp")
  .then(response => {
    if (response.ok) {
      return response.text();
    } else {
      throw new Error("Network response was not ok");
    }
  })
  .then(data => {
    document.write("Data: " + data + "\nStatus: success");
  })
  .catch(error => {
    console.error("Error:", error);
    document.write("Status: error");
  });
```

jQuery: post()

```
$.post("demo.asp", { name: "John", age: 30 }, function(data, status) {
  console.log("Data: " + data + "\nStatus: " + status);
});
```

Vanilla JavaScript: post()

```
fetch("demo.asp", {
  method: "POST",
  headers: {
    "Content-Type": "application/json",
  },
  body: JSON.stringify({ name: "John", age: 30 }),
})
  .then(response => {
    if (response.ok) {
      return response.text();
    } else {
      throw new Error("Network response was not ok");
    }
  })
  .then(data => {
    console.log("Data: " + data + "\nStatus: success");
  })
  .catch(error => {
    console.error("Error:", error);
    console.log("Status: error");
  });
```

jQuery: click and hide

```
$(".demo").click(function() {
  $(this).hide(200);
});
```

Vanilla JavaScript: click and hide

```
// Assuming you have elements with the class "demo"
var demoElements = document.querySelectorAll(".demo");

demoElements.forEach(function(element) {
  element.addEventListener("click", function() {
    this.style.transition = "opacity 200ms"; // Adding a smooth transition
    this.style.opacity = "0"; // Hiding the element by setting opacity to 0
    // You can also use 'display' property: this.style.display = 'none';
  });
});
```

jQuery: Mouse Events

```
scroll, click, dblclick, mousedown, mouseup, mousemove, mouseover, mouseout,
mouseenter, mouseleave, load, resize, scroll, unload, error
```

Vanilla JavaScript: Mouse Events

```
// Scroll event
window.addEventListener('scroll', function(event) {
  // Your scroll event handling code here
});

// Click event
document.addEventListener('click', function(event) {
  // Your click event handling code here
});

// Double-click event
document.addEventListener('dblclick', function(event) {
  // Your double-click event handling code here
});

// Mousedown event
document.addEventListener('mousedown', function(event) {
  // Your mousedown event handling code here
});

// Mouseup event
document.addEventListener('mouseup', function(event) {
  // Your mouseup event handling code here
});

// Mousemove event
document.addEventListener('mousemove', function(event) {
  // Your mousemove event handling code here
});
```

```javascript
// Mouseover event
document.addEventListener('mouseover', function(event) {
  // Your mouseover event handling code here
});

// Mouseout event
document.addEventListener('mouseout', function(event) {
  // Your mouseout event handling code here
});

// Mouseenter event
document.addEventListener('mouseenter', function(event) {
  // Your mouseenter event handling code here
});

// Mouseleave event
document.addEventListener('mouseleave', function(event) {
  // Your mouseleave event handling code here
});

// Load event
window.addEventListener('load', function(event) {
  // Your load event handling code here
});

// Resize event
window.addEventListener('resize', function(event) {
  // Your resize event handling code here
});

// Unload event
window.addEventListener('unload', function(event) {
  // Your unload event handling code here
});

// Error event
window.addEventListener('error', function(event) {
  // Your error event handling code here
});
```

jQuery: Keyboard Events

```
keydown, keypress, keyup
```

Vanilla JavaScript: Keyboard Events

```
// Keydown event
document.addEventListener('keydown', function(event) {
  // Your keydown event handling code here
  console.log('Keydown event:', event.key);
});

// Keypress event
document.addEventListener('keypress', function(event) {
  // Your keypress event handling code here
  console.log('Keypress event:', event.key);
});

// Keyup event
document.addEventListener('keyup', function(event) {
  // Your keyup event handling code here
  console.log('Keyup event:', event.key);
});
```

jQuery: Form Events

```
submit, change, focus, blur
```

Vanilla JavaScript: Form Events

```
// Assuming you have a form with ID "myForm"
var myForm = document.getElementById('myForm');

myForm.addEventListener('submit', function(event) {
  // Your submit event handling code here
  event.preventDefault(); // Prevent form submission to avoid page reload
  console.log('Form submitted');
});
```

```
// Assuming you have an input element with ID "myInput"
var myInput = document.getElementById('myInput');

myInput.addEventListener('change', function(event) {
  // Your change event handling code here
  console.log('Input value changed:', event.target.value);
});

// Assuming you have an input element with ID "myInput"
var myInput = document.getElementById('myInput');

myInput.addEventListener('focus', function(event) {
  // Your focus event handling code here
  console.log('Input focused');
});

// Assuming you have an input element with ID "myInput"
var myInput = document.getElementById('myInput');

myInput.addEventListener('blur', function(event) {
  // Your blur event handling code here
  console.log('Input lost focus');
});
```

jQuery: Dom Elements

Vanilla JavaScript: Dom Elements

```
// Assuming you have an input element with ID "myInput"
var myInput = document.getElementById('myInput');

myInput.addEventListener('blur', function(event) {
  // Your blur event handling code here
  console.log('Input lost focus');
});
// Assuming you have an input element with ID "myInput"
var myInput = document.getElementById('myInput');

myInput.addEventListener('focus', function(event) {
  // Your focus event handling code here
  console.log('Input focused');
});
// Assuming you have an input element with ID "myInput"
var myInput = document.getElementById('myInput');

myInput.addEventListener('focusin', function(event) {
  // Your focusin event handling code here
  console.log('Input received focus');
});
```

```
// Assuming you have an input element with ID "myInput"
var myInput = document.getElementById('myInput');

myInput.addEventListener('focusout', function(event) {
  // Your focusout event handling code here
  console.log('Input lost focus');
});
// Assuming you have an input element with ID "myInput"
var myInput = document.getElementById('myInput');

myInput.addEventListener('change', function(event) {
  // Your change event handling code here
  console.log('Input value changed:', event.target.value);
});
// Assuming you have a textarea element with ID "myTextarea"
var myTextarea = document.getElementById('myTextarea');

myTextarea.addEventListener('select', function(event) {
  // Your select event handling code here
  console.log('Text selected:', event.target.value.substring(event.target.selectionStart, event.target.selectionEnd));
});
// Assuming you have a form with ID "myForm"
var myForm = document.getElementById('myForm');

myForm.addEventListener('submit', function(event) {
  // Your submit event handling code here
  event.preventDefault(); // Prevent form submission to avoid page reload
  console.log('Form submitted');
});
```

jQuery: Browser

```
// Load event on the window
$(window).load(function () {
    console.log('Page is fully loaded');
});
// Load event on the window
$(window).resize(function () {
     console.log('Window resized');
});
// Scroll event on the window
$(window).scroll(function () {
     console.log('Page scrolled');
});
// unload event on the window
$(window).unload(function () {
       console.log('Page is being unloaded');
});
// error event on the window
```

```
$("#myImage").bind("error",function(){
      console.log('Image failed to load');
});
```

Vanilla JavaScript: Browser

```
// Load event on the window
window.addEventListener('load', function(event) {
  // Your load event handling code here
  console.log('Page is fully loaded');
});

// Load event on individual elements (e.g., an image with ID "myImage")
var myImage = document.getElementById('myImage');
myImage.addEventListener('load', function(event) {
  // Your load event handling code here
  console.log('Image is fully loaded');
});

window.addEventListener('resize', function(event) {
  // Your resize event handling code here
  console.log('Window resized');
});

// Scroll event on the window
window.addEventListener('scroll', function(event) {
  // Your scroll event handling code here
  console.log('Page scrolled');
});

// Scroll event on individual elements (e.g., a div with ID "myDiv")
var myDiv = document.getElementById('myDiv');
myDiv.addEventListener('scroll', function(event) {
  // Your scroll event handling code here
  console.log('Div scrolled');
});

window.addEventListener('unload', function(event) {
  // Your unload event handling code here
  console.log('Page is being unloaded');
});

var myImage = document.getElementById('myImage');
myImage.addEventListener('error', function(event) {
  // Your error event handling code here
  console.log('Image failed to load:', event.target.src);
});
```

jQuery: bind()

```
$(document).ready(function() {
  $("#demo").bind('blur', function(e) {
    // DOM event fired
    // Your blur event handling code here
  });
});
```

Vanilla JavaScript: bind()

```
document.addEventListener('DOMContentLoaded', function() {
  var demoElement = document.getElementById('demo');
  demoElement.addEventListener('blur', function(e) {
    // DOM event fired
    // Your blur event handling code here
  });
});
```

jQuery: show,hide

```
$("#demo").hide(); // sets to display: none

$("#demo").show(200); // shows hidden element with animation (speed)

$("#demo").toggle(); // toggle between show and hide

$( "#element" ).hide( "slow", function() {
  console.log( "Animation complete." );
});
```

Vanilla JavaScript: show,hide

```
// Assuming you have an element with ID "demo"
var demoElement = document.getElementById("demo");

// Hide element (sets display: none)
demoElement.style.display = "none";

// Show hidden element with animation (speed)
```

```
function showElementWithAnimation(element, speed) {
  element.style.opacity = "0"; // Start with opacity 0 (hidden)
  element.style.display = "block"; // Make the element visible (display: block)

  var opacity = 0;
  var interval = 20; // Animation interval in milliseconds
  var duration = speed; // Animation duration in milliseconds

  function fadeIn() {
    opacity += interval / duration;
    element.style.opacity = String(opacity);

    if (opacity < 1) {
      setTimeout(fadeIn, interval);
    }
  }

  fadeIn();
}

showElementWithAnimation(demoElement, 200);

// Toggle between show and hide
function toggleElement(element) {
  if (element.style.display === "none") {
    showElementWithAnimation(element, 200);
  } else {
    element.style.display = "none";
  }
}

toggleElement(demoElement);

// Hide with callback function
```

```
function hideElementWithCallback(element, speed, callback) {
  element.style.opacity = "1"; // Start with opacity 1 (visible)

  var opacity = 1;
  var interval = 20; // Animation interval in milliseconds
  var duration = speed; // Animation duration in milliseconds

  function fadeOut() {
    opacity -= interval / duration;
    element.style.opacity = String(opacity);

    if (opacity > 0) {
      setTimeout(fadeOut, interval);
    } else {
      element.style.display = "none"; // Hide the element after animation
      if (typeof callback === "function") {
        callback();
      }
    }
  }

  fadeOut();
}

hideElementWithCallback(document.getElementById("element"), "slow", function(
) {
  console.log("Animation complete.");
});
```

jQuery: fade

```
$("#demo").fadeIn(); // fade in a hidden element

$("#demo").fadeOut(300); // fade out with a duration of 300 milliseconds

$("#demo").fadeToggle("slow"); // toggle between fadeIn and fadeOut with a "s
low" speed

$("#demo").fadeTo("slow", 0.25); // fades to 0.25 opacity with a "slow" speed
```

Vanilla JavaScript: fade

```
// Assuming you have an element with ID "demo"
var demoElement = document.getElementById("demo");

// Fade in a hidden element
```

```
function fadeInElement(element, duration) {
  element.style.opacity = "0"; // Start with opacity 0 (hidden)
  element.style.display = "block"; // Make the element visible (display: block)

  var opacity = 0;
  var interval = 20; // Animation interval in milliseconds

  function fadeIn() {
    opacity += interval / duration;
    element.style.opacity = String(Math.min(opacity, 1));

    if (opacity < 1) {
      setTimeout(fadeIn, interval);
    }
  }

  fadeIn();
}

fadeInElement(demoElement, 300);

// Fade out with a duration of 300 milliseconds
function fadeOutElement(element, duration) {
  var opacity = 1;
  var interval = 20; // Animation interval in milliseconds

  function fadeOut() {
    opacity -= interval / duration;
    element.style.opacity = String(Math.max(opacity, 0));

    if (opacity > 0) {
      setTimeout(fadeOut, interval);
    } else {
      element.style.display = "none"; // Hide the element after animation
    }
  }

  fadeOut();
}

fadeOutElement(demoElement, 300);

// Toggle between fadeIn and fadeOut with a "slow" speed
```

```
function fadeToggleElement(element, speed) {
  if (element.style.opacity === "0" || element.style.display === "none") {
    fadeInElement(element, speed);
  } else {
    fadeOutElement(element, speed);
  }
}

fadeToggleElement(demoElement, 600);

// Fades to 0.25 opacity with a "slow" speed
function fadeToElement(element, duration, targetOpacity) {
  var opacity = parseFloat(element.style.opacity);
  var interval = 20; // Animation interval in milliseconds

  function fadeTo() {
    if (opacity < targetOpacity) {
      opacity += interval / duration;
      element.style.opacity = String(Math.min(opacity, targetOpacity));
    } else if (opacity > targetOpacity) {
      opacity -= interval / duration;
      element.style.opacity = String(Math.max(opacity, targetOpacity));
    }

    if (opacity !== targetOpacity) {
      setTimeout(fadeTo, interval);
    }
  }

  fadeTo();
}

fadeToElement(demoElement, 600, 0.25);
```

jQuery: slide

```
$("#demo").slideDown(); // slide down to reveal the element

$("#demo").slideUp("slow"); // slide up to hide the element with a "slow" spe
ed

$("#demo").slideToggle(); // toggle between slideDown and slideUp
```

Vanilla JavaScript: slide

```
// Assuming you have an element with ID "demo"
var demoElement = document.getElementById("demo");

// Slide down to reveal the element
```

```
function slideDownElement(element, duration) {
  element.style.display = "block"; // Make the element visible (display: bloc
k)
  var startHeight = element.clientHeight;
  element.style.height = "0"; // Start with height 0 (hidden)

  var height = 0;
  var interval = 10; // Animation interval in milliseconds

  function slideDown() {
    height += interval / duration * startHeight;
    element.style.height = String(Math.min(height, startHeight)) + "px";

    if (height < startHeight) {
      setTimeout(slideDown, interval);
    }
  }

  slideDown();
}

slideDownElement(demoElement, 600);

// Slide up to hide the element with a "slow" speed
function slideUpElement(element, duration) {
  var startHeight = element.clientHeight;
  var height = startHeight;
  var interval = 10; // Animation interval in milliseconds

  function slideUp() {
    height -= interval / duration * startHeight;
    element.style.height = String(Math.max(height, 0)) + "px";

    if (height > 0) {
      setTimeout(slideUp, interval);
    } else {
      element.style.display = "none"; // Hide the element after animation
    }
  }

  slideUp();
}

slideUpElement(demoElement, 600);

// Toggle between slideDown and slideUp
```

```
function slideToggleElement(element, speed) {
  if (element.style.display === "none") {
    slideDownElement(element, speed);
  } else {
    slideUpElement(element, speed);
  }
}

slideToggleElement(demoElement, 600);
```

jQuery: get()

```
$("div").animate({
  opacity: '0.5',
  left: '200px',
  height: '200px'
});
```

Vanilla JavaScript: get()

```
// Assuming you have div elements that you want to animate
var divElements = document.querySelectorAll("div");

// Function to animate properties
function animateElement(element, properties, duration) {
  var startValues = {};
  var changeValues = {};
  var startTime = null;

  for (var property in properties) {
    if (properties.hasOwnProperty(property)) {
      startValues[property] = parseFloat(element.style[property]) || 0;
      changeValues[property] = parseFloat(properties[property]) - startValues
[property];
    }
  }
function animate(timestamp) {
    if (!startTime) startTime = timestamp;
    var progress = timestamp - startTime;

    if (progress >= duration) progress = duration;

    for (var property in properties) {
      if (properties.hasOwnProperty(property)) {
        var newValue = startValues[property] + (changeValues[property] * prog
ress) / duration;
        element.style[property] = newValue + (property === "opacity" ? "" : "
px");
      }
    }

    if (progress < duration) {
      requestAnimationFrame(animate);
    }
  }

  requestAnimationFrame(animate);
}

// Call the animateElement function for each div element
divElements.forEach(function(divElement) {
  animateElement(divElement, {
    opacity: '0.5',
    left: '200',
    height: '200'
  }, 1000); // Duration in milliseconds (e.g., 1000 ms = 1 second)
});
```

jQuery: stop

```
$("#demo").stop();

$('#demo').mouseleave(function(event) {
  $('.tab').stop().animate({
    opacity: '0.5',
    marginTop: '10px'
  }, 500, function() {
    $('#demo').removeClass('hovered');
  });
});
```

```
$('#demo').mouseover(function(event) {
  $('.tab').stop().animate({
    opacity: '1',
    marginTop: '0px'
  }, 300, function() {
    $('#demo').addClass('hovered');
  });
});
```

Vanilla JavaScript: stop

```
// Assuming you have an element with ID "demo" and elements with the class "tab"
var demoElement = document.getElementById("demo");
var tabElements = document.querySelectorAll(".tab");

// Function to animate properties
function animateElement(element, properties, duration, callback) {
  var startValues = {};
  var changeValues = {};
  var startTime = null;

  for (var property in properties) {
    if (properties.hasOwnProperty(property)) {
      startValues[property] = parseFloat(element.style[property]) || 0;
      changeValues[property] = parseFloat(properties[property]) - startValues[property];
    }
  }
```

```
function animate(timestamp) {
    if (!startTime) startTime = timestamp;
    var progress = timestamp - startTime;

    if (progress >= duration) progress = duration;

    for (var property in properties) {
      if (properties.hasOwnProperty(property)) {
        var newValue = startValues[property] + (changeValues[property] * progress) / duration;
        element.style[property] = newValue + (property === "opacity" ? "" : "px");
      }
    }

    if (progress < duration) {
      requestAnimationFrame(animate);
    } else {
      if (typeof callback === "function") {
        callback();
      }
    }
  }

  requestAnimationFrame(animate);
}

// Function to stop the ongoing animation of an element
function stopAnimation(element) {
  var computedStyle = getComputedStyle(element);
  var currentOpacity = computedStyle.opacity;
  var currentMarginTop = computedStyle.marginTop;

  // Set the current computed values as the final values for the animation
  element.style.opacity = currentOpacity;
  element.style.marginTop = currentMarginTop;
}

// Hover end
demoElement.addEventListener('mouseleave', function(event) {
  tabElements.forEach(function(tabElement) {
    stopAnimation(tabElement);
    animateElement(tabElement, {
      opacity: '0.5',
      marginTop: '10'
    }, 500, function() {
      demoElement.classList.remove('hovered');
    });
  });
});

// Hover begin
demoElement.addEventListener('mouseover', function(event) {
  tabElements.forEach(function(tabElement) {
    stopAnimation(tabElement);
    animateElement(tabElement, {
      opacity: '1',
      marginTop: '0'
    }, 300, function() {
      demoElement.classList.add('hovered');
    });
  });
});
```

jQuery: chaining

```
$("#demo").css("backgroundColor", "green").slideUp(500).slideDown(500);
```

Vanilla JavaScript: chaining

```
// Assuming you have an element with ID "demo"
var demoElement = document.getElementById("demo");

// Change background color to green
demoElement.style.backgroundColor = "green";

// Slide up animation
function slideUpElement(element, duration) {
  var startHeight = element.clientHeight;
  var height = startHeight;
  var interval = 10; // Animation interval in milliseconds

  function slideUp() {
    height -= interval / duration * startHeight;
    element.style.height = String(Math.max(height, 0)) + "px";

    if (height > 0) {
      setTimeout(slideUp, interval);
    } else {
      element.style.display = "none"; // Hide the element after animation
    }
  }

  slideUp();
}
```

```
// Slide down animation
function slideDownElement(element, duration) {
  element.style.display = "block"; // Make the element visible (display: block)
  var startHeight = element.clientHeight;
  element.style.height = "0"; // Start with height 0 (hidden)

  var height = 0;
  var interval = 10; // Animation interval in milliseconds

  function slideDown() {
    height += interval / duration * startHeight;
    element.style.height = String(Math.min(height, startHeight)) + "px";

    if (height < startHeight) {
      setTimeout(slideDown, interval);
    }
  }

  slideDown();
}

// Chained animations (change background color, slideUp, slideDown)
demoElement.style.transition = "background-color 500ms";
setTimeout(function() {
  demoElement.style.backgroundColor = ""; // Reset background color (remove inline style)
  slideUpElement(demoElement, 500);
  setTimeout(function() {
    slideDownElement(demoElement, 500);
  }, 500); // Wait for slideUp animation to finish before starting slideDown
}, 500); // Wait for background color transition to finish before starting slideUp
```

jQuery: Dom Manipulation Content

```
$("#demo").text(); // returns text content

$("#demo").html(); // returns content, including HTML markup

$("#demo").val(); // returns field value

$("#demo").html('Hey <em>yo</em>'); // sets HTML content
```

Vanilla JavaScript: Dom Manipulation Content

```
// Assuming you have an element with ID "demo"
var demoElement = document.getElementById("demo");

// Get text content
var textContent = demoElement.textContent;

// Get HTML content
var htmlContent = demoElement.innerHTML;

// Get field value (for input, textarea, and select elements)
var fieldValue = demoElement.value;

// Set HTML content
demoElement.innerHTML = 'Hey <em>yo</em>';
```

jQuery: Dom Manipulation Attributes

```
$("#link").attr("href"); // get an attribute

$("#link").attr("href", "https://htmlg.com"); // set attribute

$("#link").attr({
  "href": "https://htmlg.com", // setting multiple attributes
  "title": "HTML Editor"
});

$("#link").attr("href", function(i, origValue) {
  return origValue + "/help"; // callback function gets and changes the attribute
});
```

Vanilla JavaScript: Dom Manipulation Attributes

```
// Assuming you have an element with ID "link"
var linkElement = document.getElementById("link");

// Get an attribute
var hrefAttribute = linkElement.getAttribute("href");

// Set attribute
linkElement.setAttribute("href", "https://htmlg.com");

// Setting multiple attributes
linkElement.setAttribute("href", "https://htmlg.com");
linkElement.setAttribute("title", "HTML Editor");
```

```
// Callback function to get and change the attribute
linkElement.setAttribute("href", function(i, origValue) {
  return origValue + "/help";
});
```

jQuery: Dom Manipulation Add

```
$(".demo").prepend("Yo!");

$(".demo").append("<em>Hey!</em>");

$(".demo").before("Cheers");

$(".demo").after("<em>Peace</em>");
```

Vanilla JavaScript: Dom Manipulation Add

```
// Assuming you have elements with the class "demo"
var demoElements = document.querySelectorAll(".demo");

// Add content at the beginning in the selected elements
demoElements.forEach(function(demoElement) {
  demoElement.insertAdjacentText("afterbegin", "Yo!");
});

// Add content at the end in the selected elements
demoElements.forEach(function(demoElement) {
  demoElement.insertAdjacentHTML("beforeend", "<em>Hey!</em>");
});

// Add content before the selected elements
demoElements.forEach(function(demoElement) {
  var newElement = document.createElement("span");
  newElement.textContent = "Cheers";
  demoElement.parentNode.insertBefore(newElement, demoElement);
});

// Add content after the selected elements
demoElements.forEach(function(demoElement) {
  var newElement = document.createElement("span");
  newElement.innerHTML = "<em>Peace</em>";
  demoElement.parentNode.insertBefore(newElement, demoElement.nextSibling);
});
```

jQuery: Dom Manipulation Remove

```
$("#demo").remove(); // removes the selected element

$("#demo").empty(); // removes children

$("div").remove(".cl1, .cl2"); // removes divs with the listed classes
```

Vanilla JavaScript: Dom Manipulation Remove

```javascript
// Assuming you have an element with ID "demo" and div elements with classes "cl1" and "cl2"
var demoElement = document.getElementById("demo");
var divElements = document.querySelectorAll("div.cl1, div.cl2");

// Remove the selected element
if (demoElement.parentNode) {
  demoElement.parentNode.removeChild(demoElement);
}

// Remove children of the selected element
demoElement.innerHTML = "";

// Remove div elements with classes "cl1" and "cl2"
divElements.forEach(function(divElement) {
  if (divElement.parentNode) {
    divElement.parentNode.removeChild(divElement);
  }
});
```

jQuery: Dom Manipulation Classes

```javascript
$("#demo").addClass("big red"); // add class

$("h1, p").removeClass("red"); // remove class

$("#demo").toggleClass("big"); // toggle between adding and removing
```

Vanilla JavaScript: Dom Manipulation Classes

```javascript
// Assuming you have an element with ID "demo" and h1, p elements in the document
var demoElement = document.getElementById("demo");
var h1Elements = document.querySelectorAll("h1");
var pElements = document.querySelectorAll("p");

// Add class "big" and "red" to the selected element
demoElement.classList.add("big", "red");

// Remove class "red" from h1 and p elements
h1Elements.forEach(function(h1Element) {
  h1Element.classList.remove("red");
});
pElements.forEach(function(pElement) {
  pElement.classList.remove("red");
});

// Toggle class "big" on the selected element
demoElement.classList.toggle("big");
```

jQuery: Dom Manipulation Css

```
$("#demo").css("background-color"); // returns CSS value

$("#demo").css("color", "blue"); // sets CSS rule

$("#demo").css({ "color": "blue", "font-size": "20px" }); // sets multiple CSS properties
```

Vanilla JavaScript: Dom Manipulation Css

```
// Assuming you have an element with ID "demo"
var demoElement = document.getElementById("demo");

// Get CSS value for background-color
var backgroundColorValue = window.getComputedStyle(demoElement).backgroundColor;

// Set CSS rule for color
demoElement.style.color = "blue";

// Set multiple CSS properties using an object
demoElement.style.color = "blue";
demoElement.style.fontSize = "20px";
```

jQuery: Dom Manipulation Dimensions

```
width, height, innerWidth, innerHeight, outerWidth, outerHeight
inner - includes padding
outer - includes padding and border
```

Vanilla JavaScript: Dom Manipulation Dimensions

```
// Assuming you have an element with ID "demo"
var demoElement = document.getElementById("demo");

var width = demoElement.offsetWidth;
var height = demoElement.offsetHeight;

// Assuming you have an element with ID "demo"
var demoElement = document.getElementById("demo");

var innerWidth = demoElement.clientWidth;
var innerHeight = demoElement.clientHeight;

// Assuming you have an element with ID "demo"
var demoElement = document.getElementById("demo");

// Get the computed styles of the element, including padding and border
var computedStyles = window.getComputedStyle(demoElement);
```

```
// Calculate outer width and height
var outerWidth = demoElement.offsetWidth + parseInt(computedStyles.marginLeft) + parseInt(computedStyles.marginRight);
var outerHeight = demoElement.offsetHeight + parseInt(computedStyles.marginTop) + parseInt(computedStyles.marginBottom);
```