JQUERY

JQUERY

jQuery is a lightweight, "write less, do more", JavaScript library.

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

jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.

jQuery also simplifies a lot of the complicated things from JavaScript, like DOM manipulation and AJAX calls.

JQUERY

The jQuery library contains the following features:

- HTML/DOM manipulation
- CSS manipulation
- HTML event methods
- Effects and animations
- AJAX

JQUERY vs Java DOM Manipulation

```
$("#myButton").click(function() {
alert("Button clicked!");
});
```

Vs

```
const button = document.getElementById("myButton");
button.onclick= function() {  alert("Button clicked!");});
```

ASPECTS OF THE DOM AND JQUERY

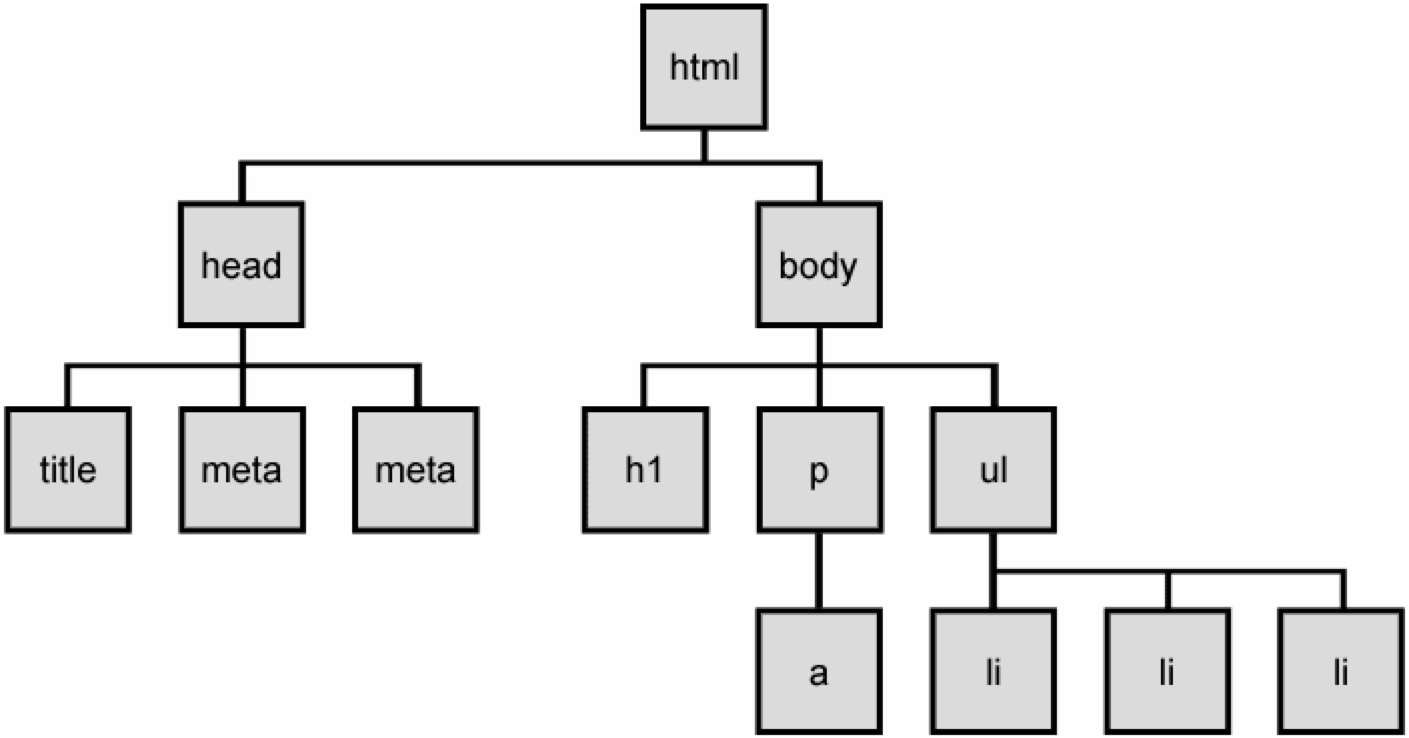
Identification: how do I obtain a reference to the node that I want.

Traversal: how do I move around the DOM tree.

Node Manipulation: how do I get or set aspects of a DOM node.

Tree Manipulation: how do I change the structure of the page.

THE DOM TREE



INCLUDING JQUERY

Download the jQuery library from jQuery.com

Can include using a third-party **content delivery network (CDN)**

- Google CDN

```
<script
src=“https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js” />
```

- Microsoft CDN
- CDNJS CDN
- jsDelivr CDN

JQUERY SYNTAX

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)

JQUERY SELECTORS

jQuery selectors allow you to select and manipulate HTML element(s).

jQuery selectors are used to "find" (or select) HTML elements based on their name, id, classes, types, attributes, values of attributes and much more.

All selectors in jQuery start with the dollar sign and parentheses: $().

BASIC SELECTORS

The four basic selectors include the universal selector, class selectors, id selectors, and elements selectors.

- $("*") **Universal selector matches all elements (and is slow).**
- $("tag") **Element selector matches all elements with the given element name.**
- $(".class") **Class selector matches all elements with the given CSS class.**
- $("#id") **Id selector matches all elements with a given HTML id attribute.**

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".

SELECTORS

```
var singleElement = $("#grab");

var allAs = $("a");

var artistImages = $("img[src^='/artist/']");
```

We can select by attribute with

- square brackets ([attribute]),
- specify a value with an equals sign ([attribute=value]) and
- search for a particular value in the beginning, end, or anywhere inside a string with ^, $, and * symbols respectively

```
([attribute^=value], [attribute$=value], [attribute*=value])
```

THE DOCUMENT READY EVENT

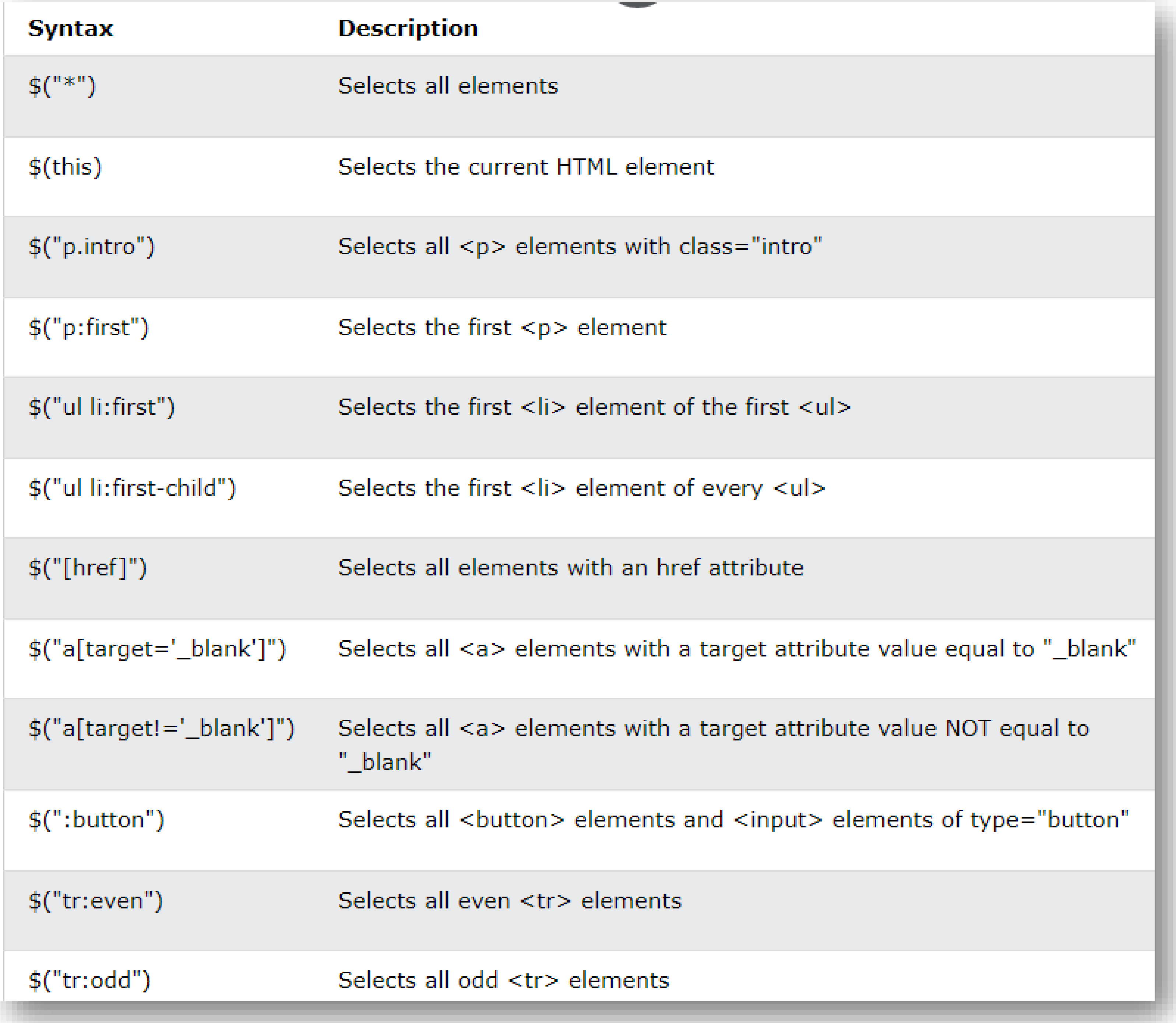
Prevent any jQuery code from running before the document is finished loading

```
$(document).ready(function(){

  // jQuery methods go here...

});

-------------OR---------------

$(function(){

  // jQuery methods go here...

});
```

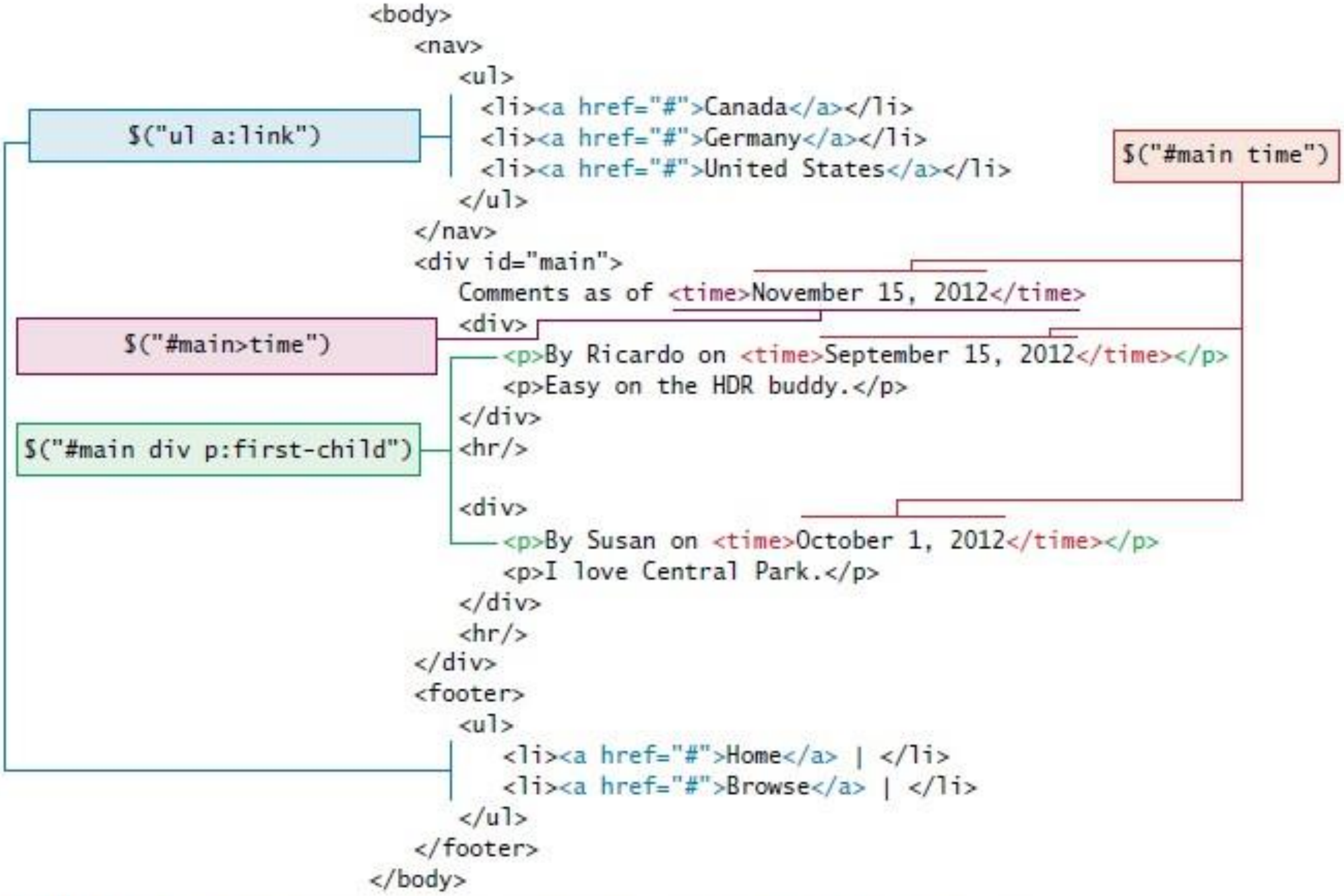
```
Example

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

ATTRIBUTE SELECTOR

An attribute selector provides a way to select elements by either the presence of an element attribute or by the value of an attribute



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



```
<body>
   <nav>
      <ul>
        <li><a href="#">Canada</a></li>
        <li><a href="#">Germany</a></li>
        <li><a href="#">United States</a></li>
      </ul>
   </nav>
   <div id="main">
      Comments as of <time>November 15, 2012</time>
      <div>
         <p>By Ricardo on <time>September 15, 2012</time></p>
         <p>Easy on the HDR buddy.</p>
      </div>
      <hr/>

      <div>
         <p>By Susan on <time>October 1, 2012</time></p>
         <p>I love Central Park.</p>
      </div>
      <hr/>
   </div>
   <footer>
      <ul>
         <li><a href="#">Home</a> | </li>
         <li><a href="#">Browse</a> | </li>
      </ul>
   </footer>
</body>
```

FIGURE 15.4 Illustration of some jQuery selectors and the HTML being selected

PSEUDO-CLASS SELECTOR

Pseudo-classes are used to define a special state of an element. They apply styles based on conditions such as user interaction, the position of an element, or whether an element is checked.

:link, :visited, :focus, :hover, :active, :checked, :first-child.

Example: var visitedLinks = $("a:visited");

PSEUDO-ELEMENT SELECTOR

Pseudo-elements are used to style specific parts of an element's content. They are created to represent a particular part of an element, such as the first letter or line of text.

:first- line, and :first-letter, :before, :after

Example: var firstLine = $("p:first-line");

CONTEXTUAL SELECTOR

These selectors allowed you to specify elements with certain relationships to one another in your CSS.

These relationships included

- descendant (space),
- child (>),
- adjacent sibling (+), //immediate next sibling
- and general sibling (~).

Example:

To select all <p> elements inside of <div> elements you would write

- var para = $("div p");

CONTEXTUAL SELECTOR

Example:

$("main time")

- This selects all <time> elements that are **descendants** of the <main> element, regardless of how deeply nested they are.

```
<main>
  <div>
    <time>2023-09-08</time> <!-- This will be selected -->
  </div>
  <section>
    <article>
      <time>2024-01-01</time> <!-- This will also be selected -->
    </article>
  </section>
</main>
```

CONTEXTUAL SELECTOR

Example:

$("main >time")

- **Meaning**: This selects only the <time> elements that are **direct children** of the <main> element.

```
<main>
<time>2023-09-08</time> <!-- This will be selected -->
<div>
<time>2024-01-01</time> <!-- This will NOT be selected -->
</div>
</main>
```

CONTEXTUAL SELECTOR

Example:

$("main + time")

•**Meaning**: This selects only the <time> elements that are **immediately next** of the <main> element.

```
<main>
      <h1>Event Dates</h1>
      <time>2023-09-08</time>
</main>
<time>2025-01-01</time> <!-- This will be selected -->
<time>2026-01-01</time> <!-- This will NOT be selected -->
```

CONTEXTUAL SELECTOR

Example:

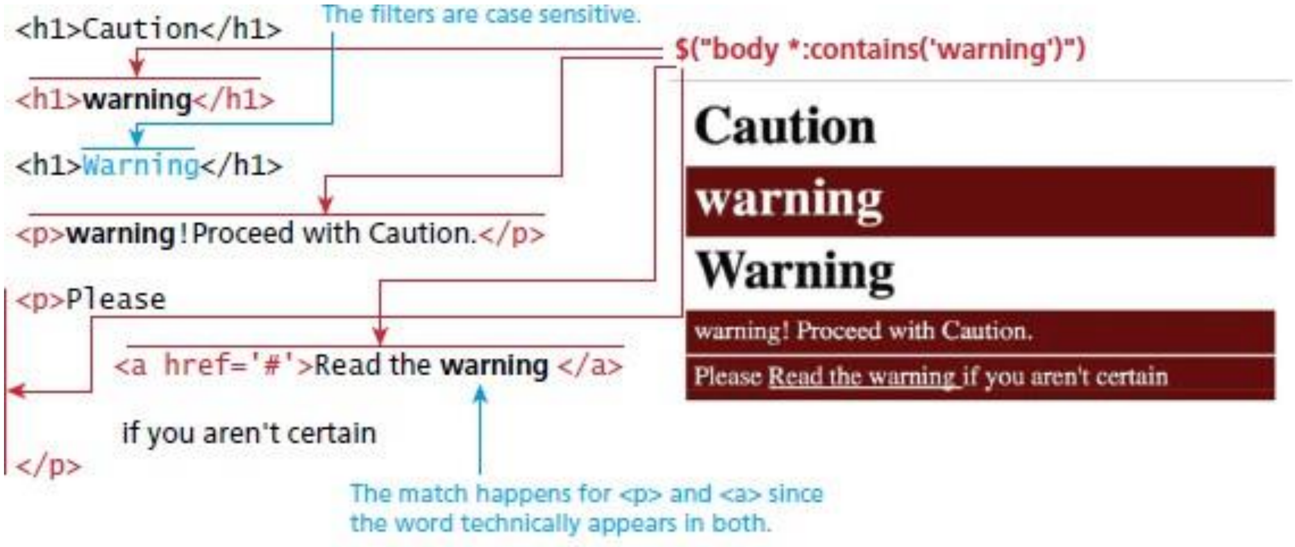
$("main ~ time")

- **Meaning**: This selects only the <time> elements that are **siblings** of the <main> element.

```
<main>
      <h1>Event Dates</h1>
      <time>2023-09-08</time>
</main>
<time>2025-01-01</time> <!-- This will be selected -->
<time>2026-01-01</time> <!-- This will also be selected -->
```

CONTENT FILTERS

Allows you to append filters to all of the selectors you've used this far and match a particular pattern



```
<h1>Caution</h1>

<h1>warning</h1>

<h1>Warning</h1>

<p>warning!Proceed with Caution.</p>

<p>Please
    <a href='#'>Read the warning </a>
    if you aren't certain
</p>
```

The filters are case sensitive.

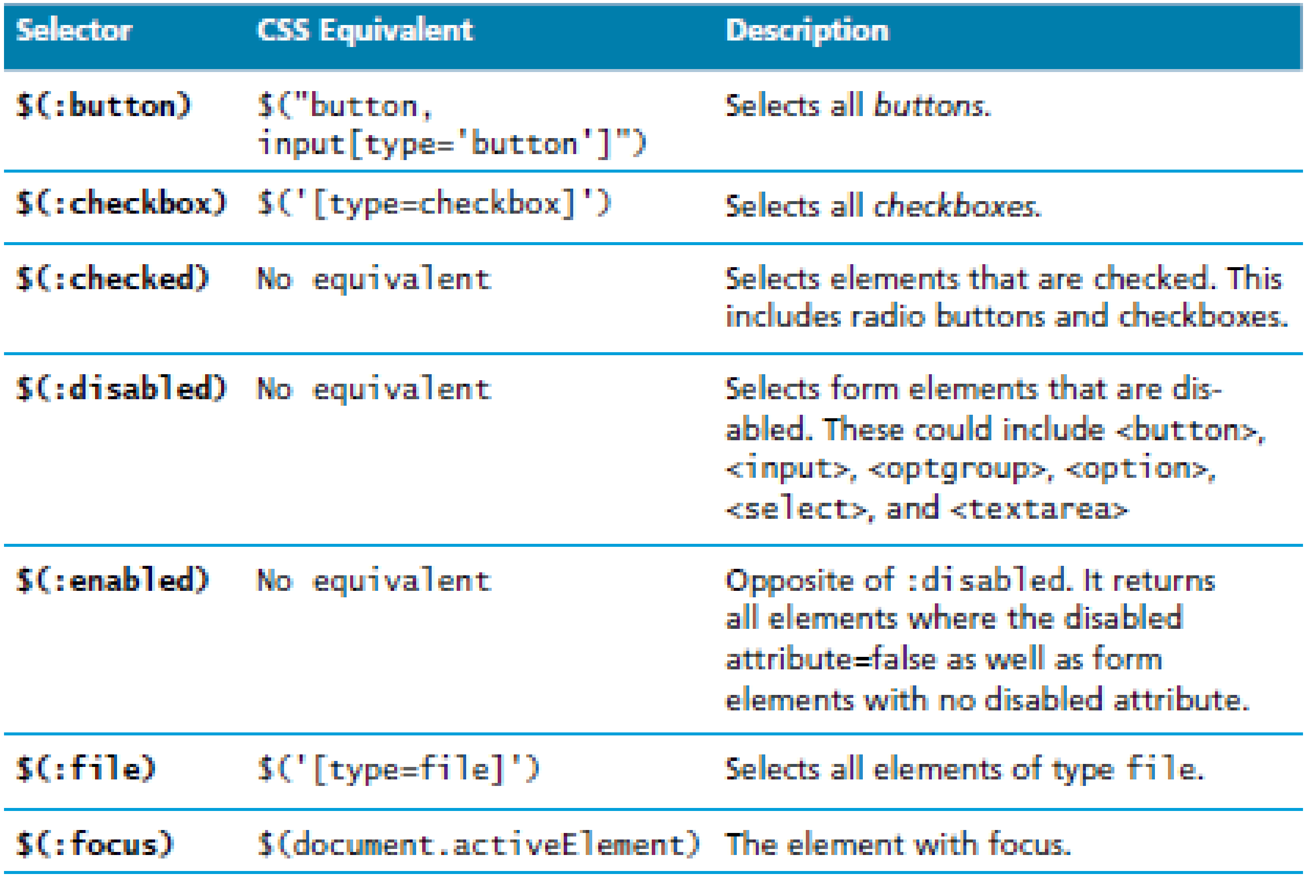
$("body *:contains('warning')")

The match happens for <p> and <a> since the word technically appears in both.

Caution
warning
Warning
warning! Proceed with Caution.
Please Read the warning if you aren't certain

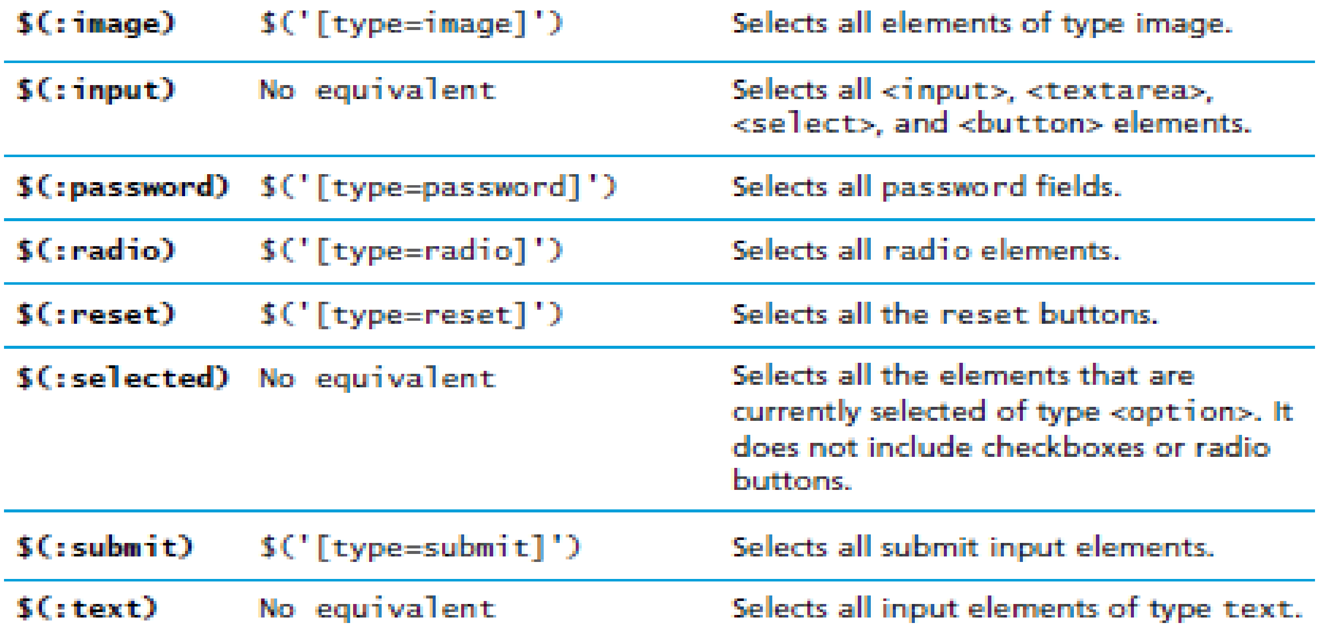
FIGURE 15.5 An illustration of jQuery's content filter selector

FORM SELECTORS



Selector	CSS Equivalent	Description
`$(:button)`	`$("button, input[type='button']")`	Selects all *buttons*.
`$(:checkbox)`	`$('[type=checkbox]')`	Selects all *checkboxes*.
`$(:checked)`	No equivalent	Selects elements that are checked. This includes radio buttons and checkboxes.
`$(:disabled)`	No equivalent	Selects form elements that are disabled. These could include `<button>`, `<input>`, `<optgroup>`, `<option>`, `<select>`, and `<textarea>`
`$(:enabled)`	No equivalent	Opposite of `:disabled`. It returns all elements where the disabled attribute=false as well as form elements with no disabled attribute.
`$(:file)`	`$('[type=file]')`	Selects all elements of type `file`.
`$(:focus)`	`$(document.activeElement)`	The element with focus.

FORM SELECTORS



$(:image)	$('[type=image]')	Selects all elements of type image.
$(:input)	No equivalent	Selects all <input>, <textarea>, <select>, and <button> elements.
$(:password)	$('[type=password]')	Selects all password fields.
$(:radio)	$('[type=radio]')	Selects all radio elements.
$(:reset)	$('[type=reset]')	Selects all the reset buttons.
$(:selected)	No equivalent	Selects all the elements that are currently selected of type <option>. It does not include checkboxes or radio buttons.
$(:submit)	$('[type=submit]')	Selects all submit input elements.
$(:text)	No equivalent	Selects all input elements of type text.

FORM SELECTORS

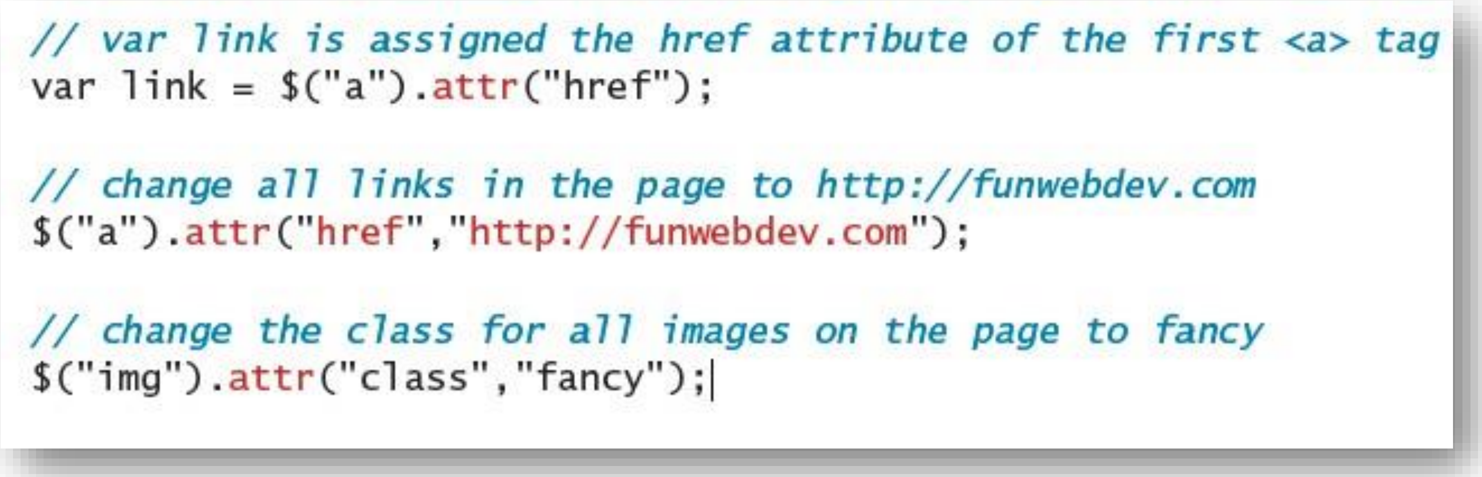
```
<input type="text" name="username">
<input type="password" name="password">

<script>
  $(document).ready(function(){
    $(":password").css("background-color", "lightblue");
  });
</script>
```

JQUERY ATTRIBUTES

JQUERY ATTRIBUTE

In jQuery we can both set and get an attribute value by using the `attr()` method on any element from a selector

- This function takes a parameter to specify which attribute, and the optional second parameter is the value to set it to.
- If no second parameter is passed, then the return value of the call is the current value of the attribute.

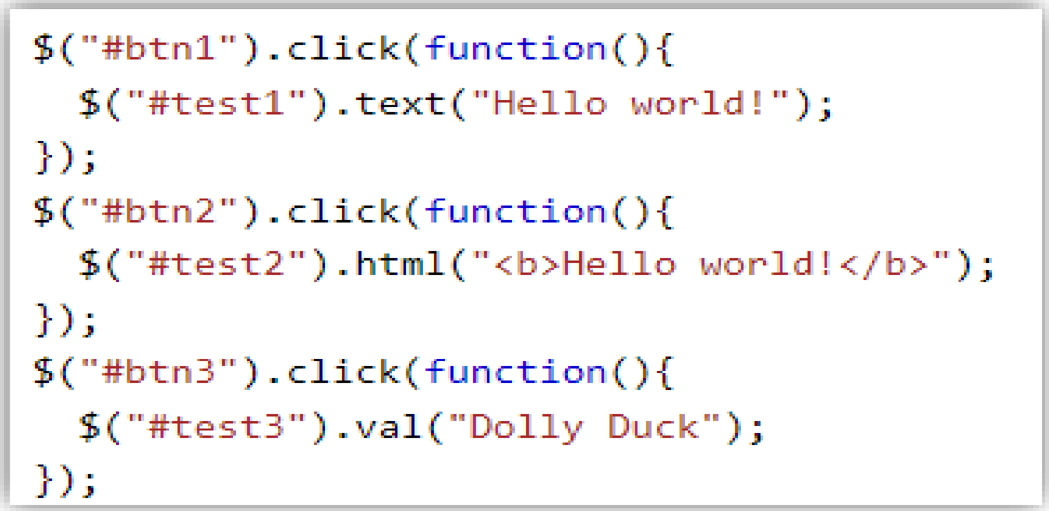


```
// var link is assigned the href attribute of the first <a> tag
var link = $("a").attr("href");

// change all links in the page to http://funwebdev.com
$("a").attr("href","http://funwebdev.com");

// change the class for all images on the page to fancy
$("img").attr("class","fancy");
```

jQuery-Get Content and Attributes

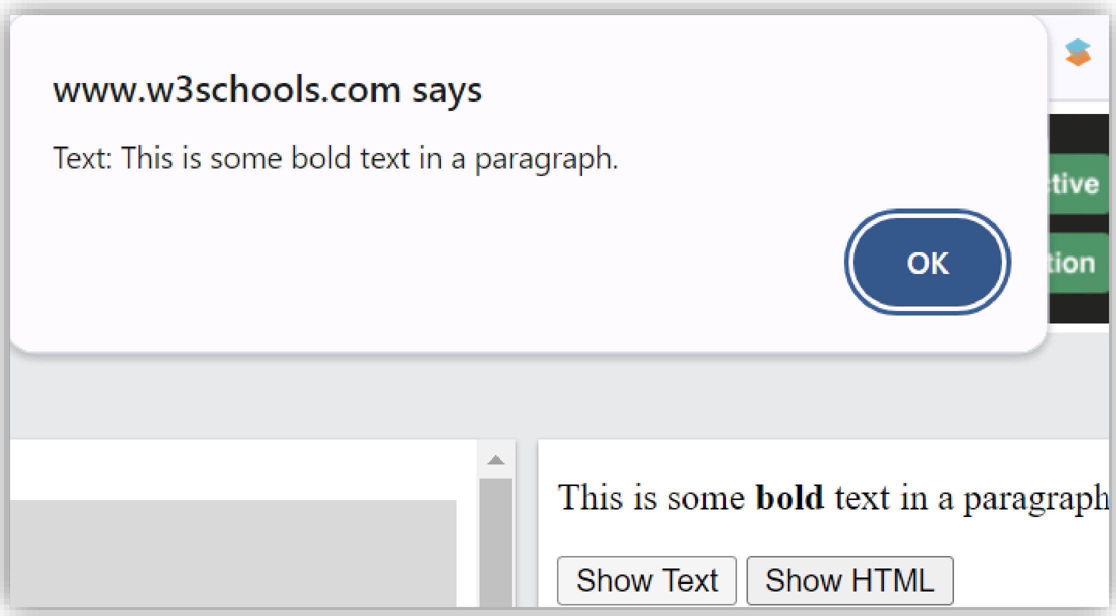
Three simple, but useful, jQuery methods for DOM manipulation are:

- text() - Sets or returns the text content of selected elements
- html() - Sets or returns the content of selected elements (including HTML markup)
- val() - Sets or returns the value of form fields



```
$("#btn1").click(function(){
  $("#test1").text("Hello world!");
});
$("#btn2").click(function(){
  $("#test2").html("<b>Hello world!</b>");
});
$("#btn3").click(function(){
  $("#test3").val("Dolly Duck");
});
```

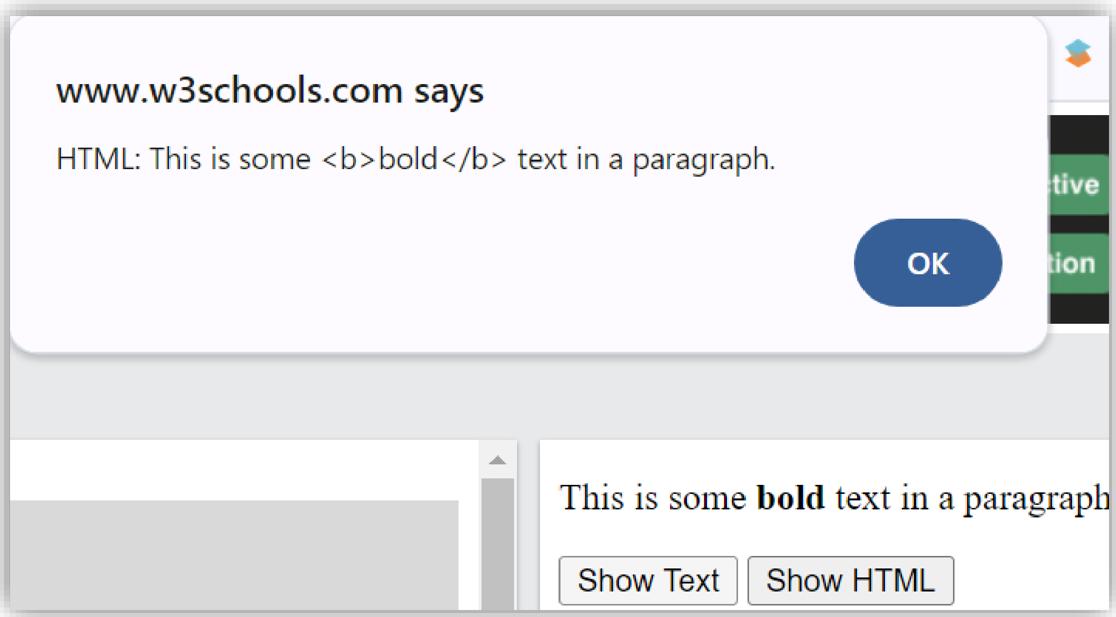
```
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    alert("Text: " + $("#test").text());
  });
  $("#btn2").click(function(){
    alert("HTML: " + $("#test").html());
  });
});
</script>
</head>
<body>
<p id="test">This is some <b>bold</b> text in a paragraph.</p>
<button id="btn1">Show Text</button>
<button id="btn2">Show HTML</button>
</body>
```



www.w3schools.com says

Text: This is some bold text in a paragraph.

This is some **bold** text in a paragraph

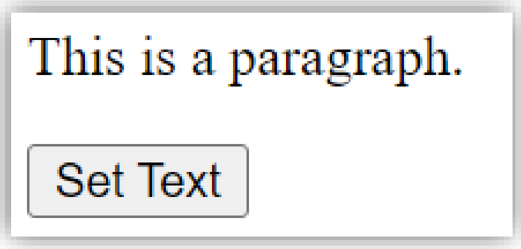


www.w3schools.com says

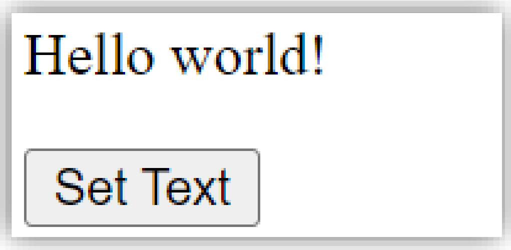
HTML: This is some <b>bold</b> text in a paragraph.

This is some **bold** text in a paragraph

```
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    $("#test1").text("Hello world!");
  });
});
</script>
</head>
<body>
<p id="test1">This is a paragraph.</p>
<button id="btn1">Set Text</button>
</body>
</html>
```

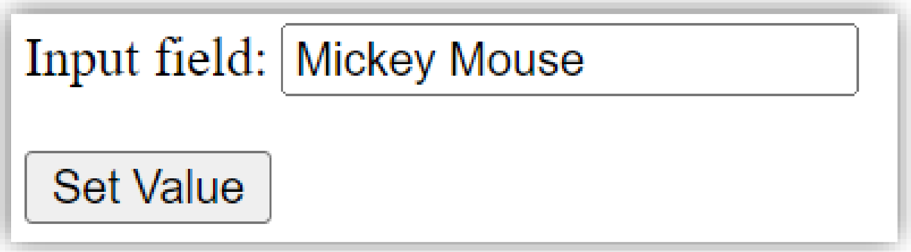


This is a paragraph.

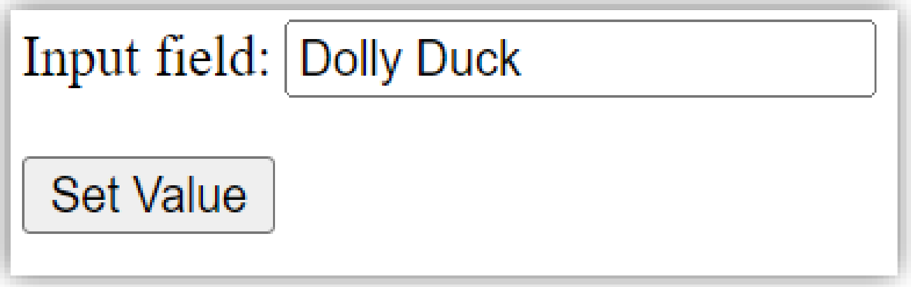


Hello world!

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#btn3").click(function(){
    $("#test3").val("Dolly Duck");
  });
});
</script>
</head>
<body>
<p>Input field: <input type="text" id="test3" value="Mickey Mouse"></p>
<button id="btn3">Set Value</button>
</body>
</html>
```



Input field: Mickey Mouse

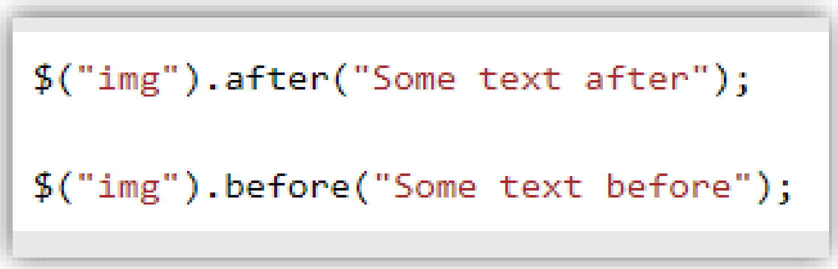


Input field: Dolly Duck

jQuery - Get Content and Attributes

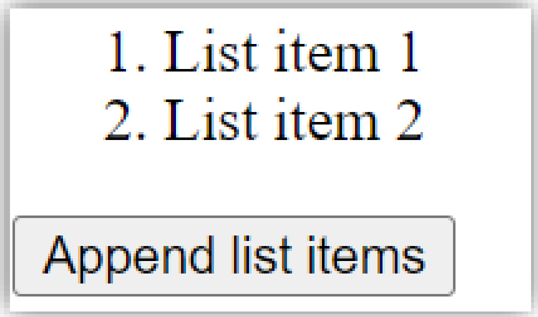
Four jQuery methods that are used to add new content

- 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

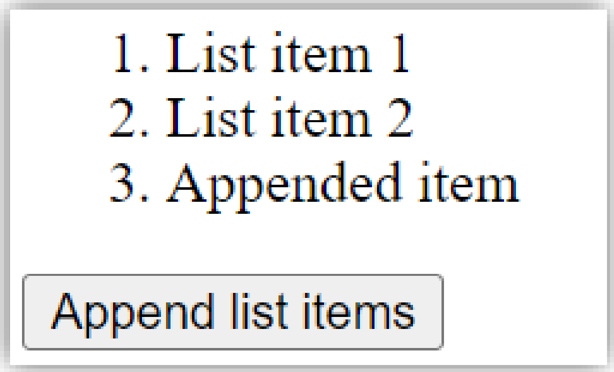


```
$("img").after("Some text after");

$("img").before("Some text before");
```

```
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    $("ol").append("<li>Appended item</li>");
  });
});
</script>
</head>
<body>
<ol>
  <li>List item 1</li>
  <li>List item 2</li>
</ol>
<button id="btn1">Append list items</button>
</body>
</html>
```

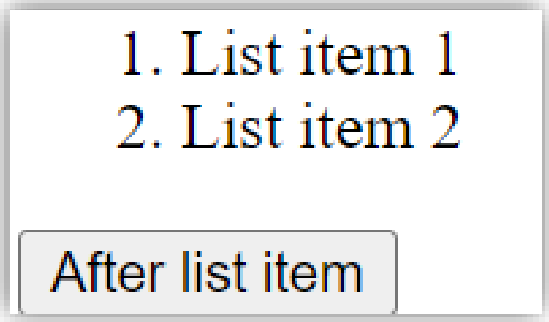


1. List item 1
2. List item 2

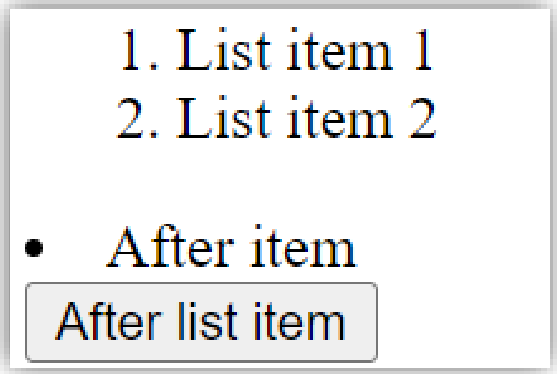


1. List item 1
2. List item 2
3. Appended item

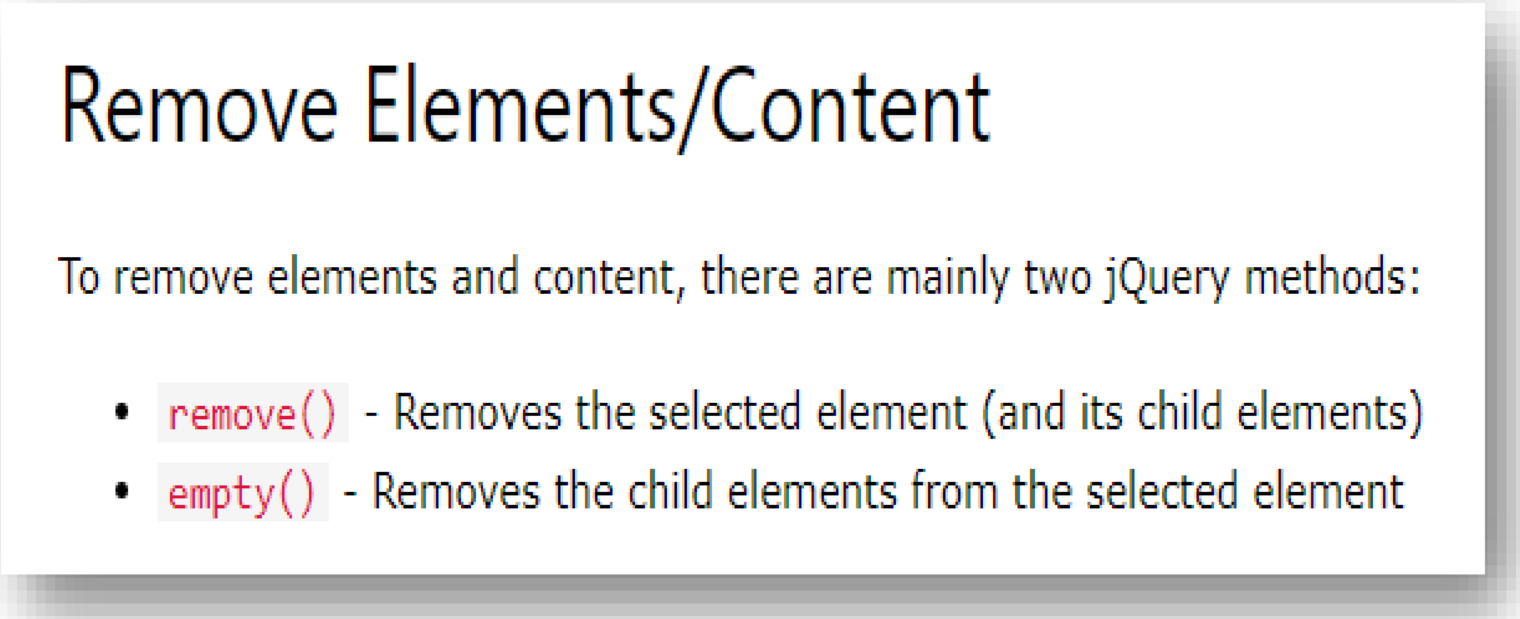
```
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    $("ol").after("<li>After item</li>");
  });
});
</script>
</head>
<body>
<ol>
  <li>List item 1</li>
  <li>List item 2</li>
</ol>
<button id="btn1">After list item</button>
</body>
</html>
```



```
1. List item 1
2. List item 2
```



```
1. List item 1
2. List item 2

• After item
```

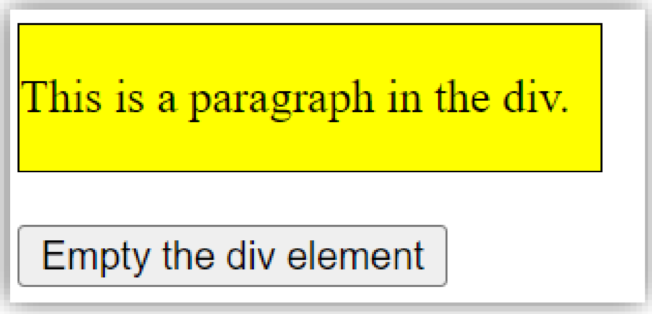


Remove Elements/Content

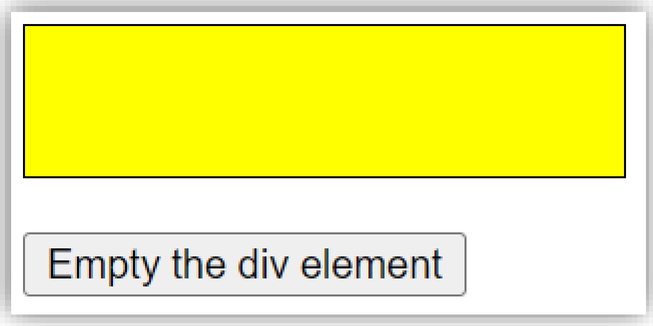
To remove elements and content, there are mainly two jQuery methods:

- `remove()` - Removes the selected element (and its child elements)
- `empty()` - Removes the child elements from the selected element

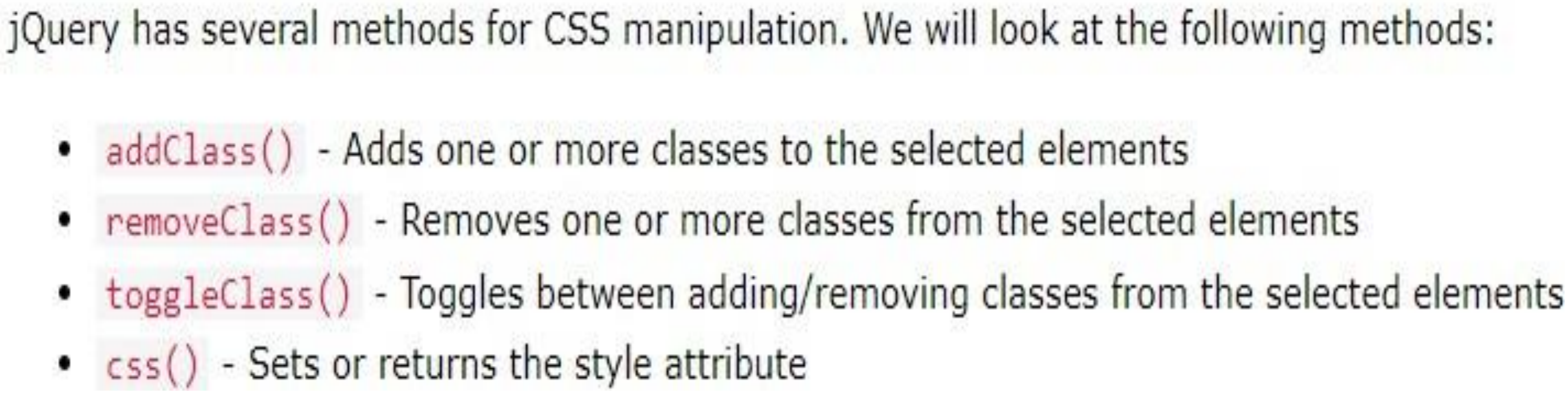
```
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").empty()   ;});    });
</script>
</head>
<body>
<div id="div1" style="height:50px; width:200px; border:1px solid black;
background-color:yellow;">
<p>This is a paragraph in the div.</p>
</div>
<br>
<button>Empty the div element</button>
</body>
</html>
```



This is a paragraph in the div.



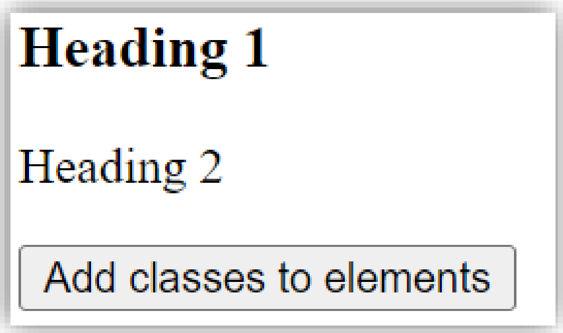
jQuery Manipulating CSS



jQuery has several methods for CSS manipulation. We will look at the following methods:

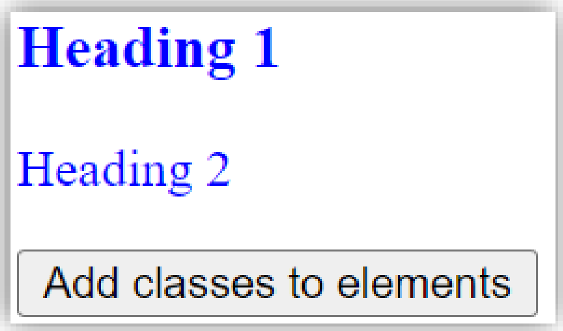
- `addClass()` - Adds one or more classes to the selected elements
- `removeClass()` - Removes one or more classes from the selected elements
- `toggleClass()` - Toggles between adding/removing classes from the selected elements
- `css()` - Sets or returns the style attribute

```
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("h3, p").addClass("blue");
  });
});
</script>
<style>
.blue { color: blue; }
</style>
</head>
<body>
<h3>Heading 1</h3>
<p>Heading 2</p>
<button>Add classes to elements</button>
</body>
```



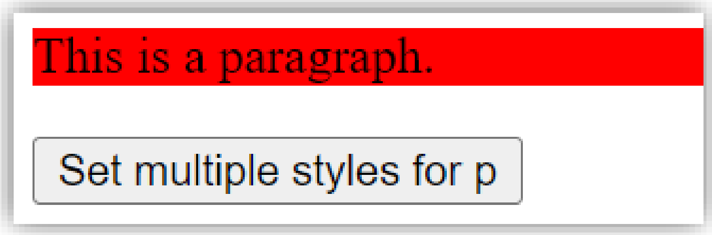
Heading 1

Heading 2



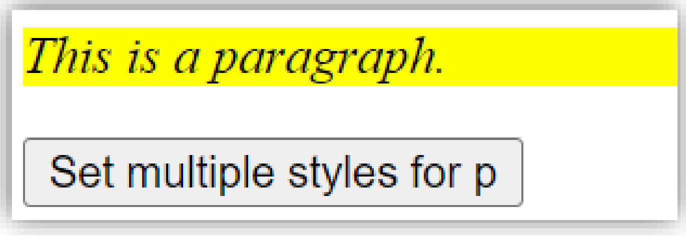
Heading 1

Heading 2



This is a paragraph.

```
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").css({"background-color": "yellow", "font-style": "italic"});
  });
});
</script>
</head>
<body>
<p style="background-color:#ff0000">This is a paragraph.</p>
<button>Set multiple styles for p</button>
</body>
</html>
```

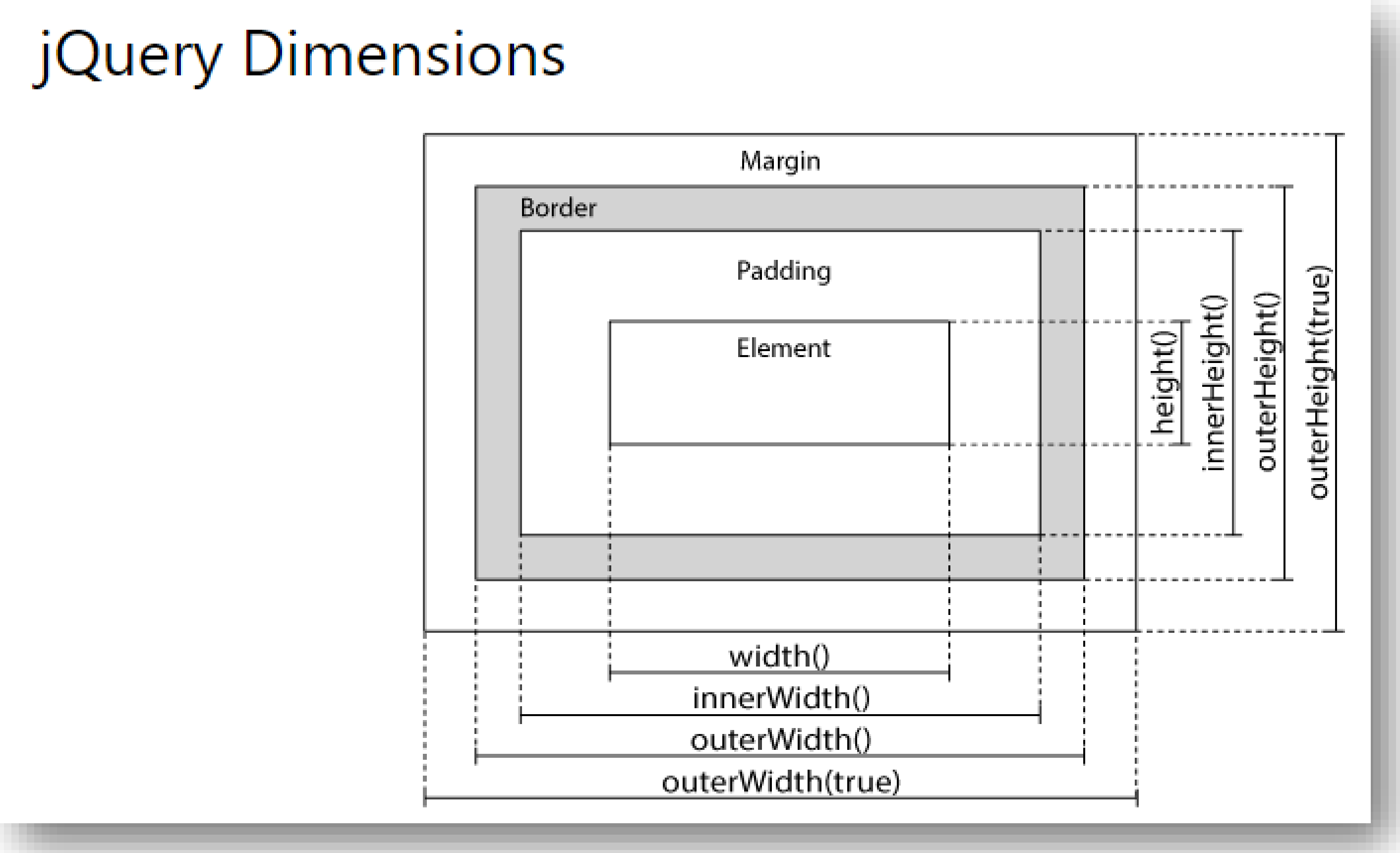


This is a paragraph.

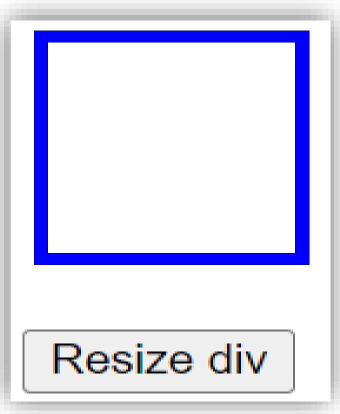
jQuery Dimension Methods

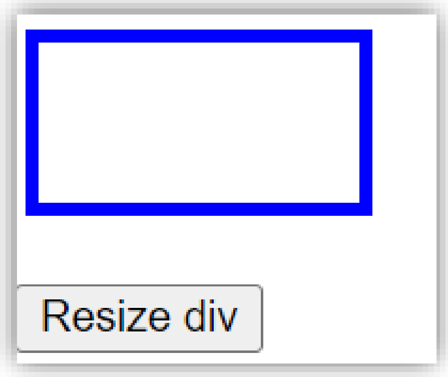
jQuery has several important methods for working with dimensions:

- `width()`
- `height()`
- `innerWidth()`
- `innerHeight()`
- `outerWidth()`
- `outerHeight()`



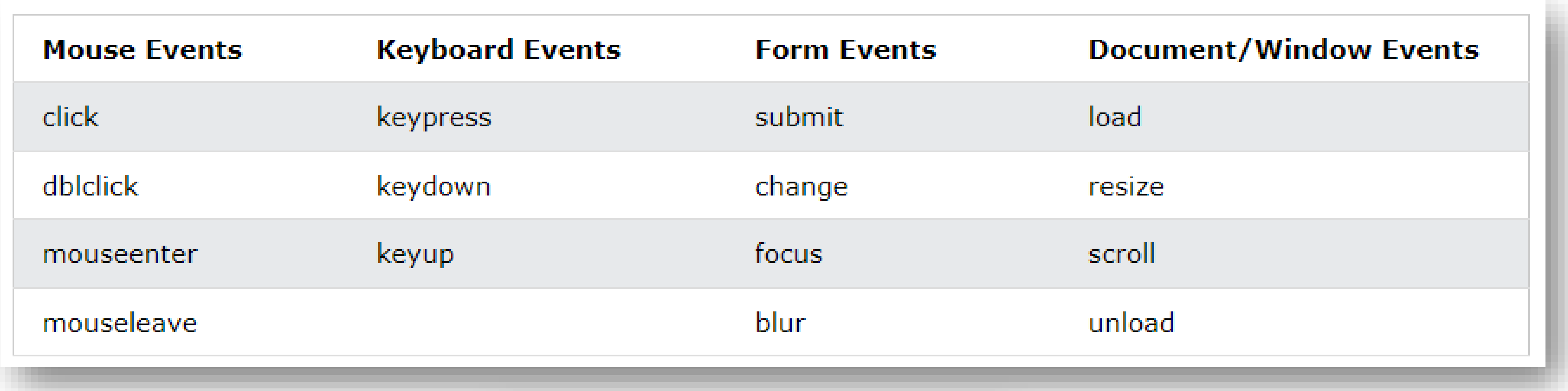
```
<head><script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").width(80).height(30);
  });
});</script>
<style>
#div1 {
  height: 50px;
  width: 50px;
  padding: 10px;
  margin: 3px;
  border: 4px solid blue;
}</style>
</head>
<body>
<div id="div1"></div>
<br>
<button>Resize div</button>
</body>
```



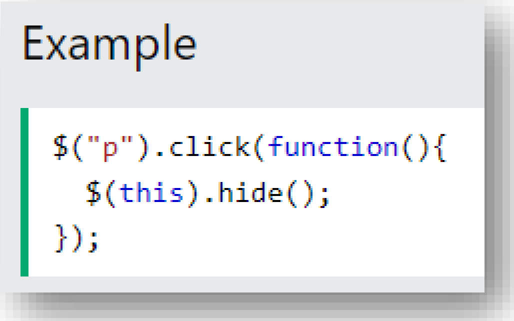


JQUERY LISTENERS

JQUERY LISTENERS

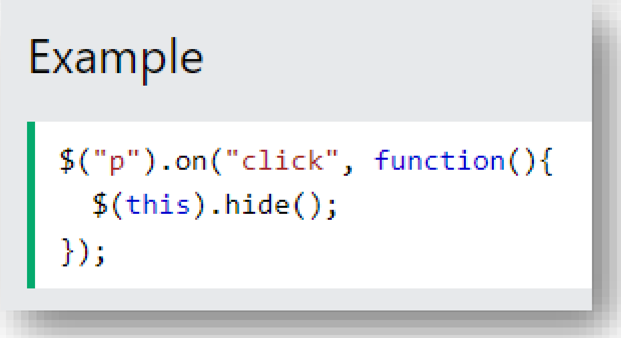


Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload



Example

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

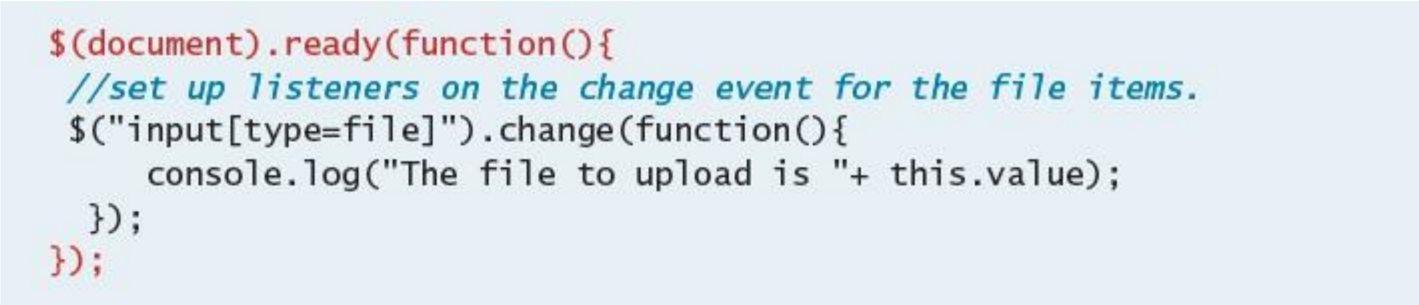


Example

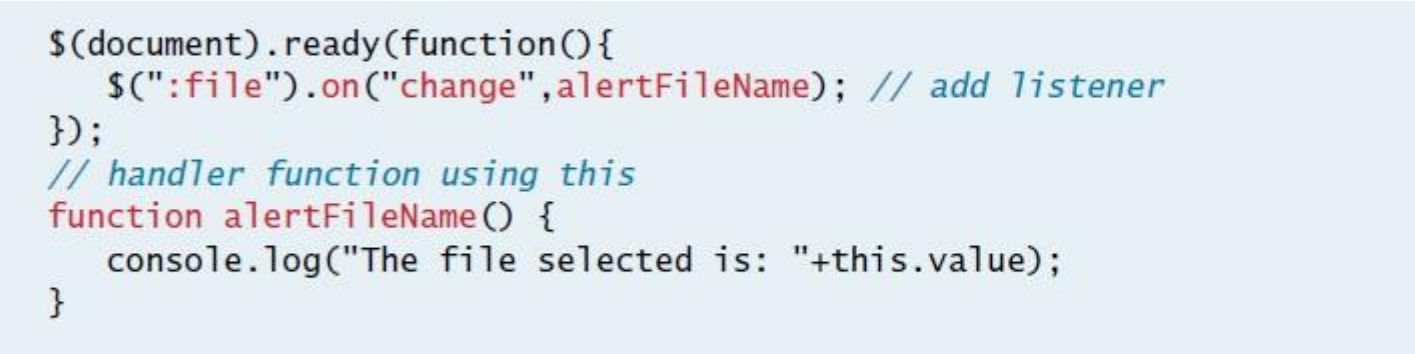
```
$("p").on("click", function(){
  $(this).hide();
});
```

JQUERY LISTENERS

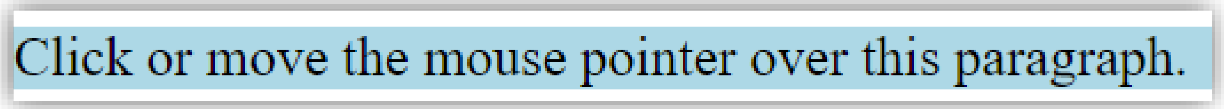
JavaScript, jQuery supports creation and management of listeners/handlers for JavaScript events.



```
$(document).ready(function(){
 //set up listeners on the change event for the file items.
 $("input[type=file]").change(function(){
     console.log("The file to upload is "+ this.value);
  });
});
```

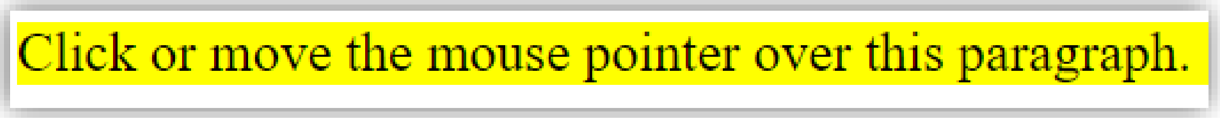


```
$(document).ready(function(){
   $(":file").on("change",alertFileName); // add listener
});
// handler function using this
function alertFileName() {
   console.log("The file selected is: "+this.value);
}
```



Click or move the mouse pointer over this paragraph.

```
<head>
<script src=“…"></script>
<script>
$(document).ready(function(){
  $("p").on({
    mouseenter: function(){
      $(this).css("background-color", "lightgray"); },
    mouseleave: function(){
      $(this).css("background-color", "lightblue"); },
    click: function(){
      $(this).css("background-color", "yellow");  }
  });
});
</script>
</head>
<body>
<p>Click or move the mouse pointer over this paragraph.</p>
</body>
```



Click or move the mouse pointer over this paragraph.

EXERCISE (1/3)

1. Use the correct selector to hide all <p> elements.
2. Use the correct selector to hide an element with id="test".
3. Use the correct selector to hide all elements with class="test".
4. Use the correct selector to hide all elements in the document.
5. Use the correct selector to hide all elements with an href attribute.
6. Use the correct selector to hide all odd table rows in a table.

```
$("     ").hide();
```

EXERCISE (2/3)

1. Use the correct event to hide all <p> elements with a "click".
2. Use the correct event to hide all <p> elements with a "double-click".
3. When the mouse pointer enters a <p> element, it should be hidden. Use the correct event to do so.
4. If you press a keyboard key inside an <input> element, it should be hidden. Use the correct event to do so.

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

EXERCISE (3/3)

1. Use the `on()` method to attach a click event handler to all <p> elements.

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

REFERENCES

jQuery

https://jquery.com/

https://api.jquery.com/

jQuery at W3Schools

https://www.w3schools.com/jquery/