# Unit 12: JQuery

**CONTENTS**

## Objectives

After studying this unit, you will be able to:

- Understand JQuery
- Design structure of the elements on a web page
- Custom jQuery extensions to the standard set of CSS selectors
- The DOM traversal methods, which provide greater flexibility for accessing elements on the page

## Introduction

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 is a small, light-weight and fast JavaScript library. It is cross-platform and supports different types of browsers. It is also referred as ?write less do more? because it takes a lot of common tasks that requires many lines of JavaScript code to accomplish, and binds them into methods that can be called with a single line of code whenever needed. It is also very useful to simplify a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.

## 12.1 JQuery

jQuery is a fast, small, and feature-rich JavaScript library. It 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. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

### What jQuery does

The jQuery library provides a general-purpose abstraction layer for common webscripting, and is therefore useful in almost every scripting situation. Its extensiblenature means that we could never cover all the possible uses and functions in asingle book, as plugins are constantly being developed to add new abilities. Thecore features, though, assist us in accomplishing the following tasks:

• Access elements in a document: Without a JavaScript library, web developersoften need to write many lines of code to traverse the Document Object Model(DOM) tree and locate specific portions of an HTML document's structure.With jQuery, developers have a robust and efficient selector mechanism attheir disposal, making it easy to retrieve the exact piece of the document thatneeds to be inspected or manipulated.

```
$('div.content').find('p');
```

• Modify the appearance of a web page: CSS offers a powerful method ofinfluencing the way a document is rendered, but it falls short when not allweb browsers support the same standards. With jQuery, developers canbridge this gap, relying on the same standards support across all browsersIn addition, jQuery can change the classes or individual style propertiesapplied to a portion of the document even after the page has been rendered.

```
$('ul>li:first').addClass('active');
```

• Alter the content of a document: Not limited to mere cosmetic changes,jQuery can modify the content of a document itself with a few keystrokes.Text can be changed, images can be inserted or swapped, lists can bereordered, or the entire structure of the HTML can be rewritten andextended—all with a single easy-to-use Application ProgrammingInterface (API).

```
$('#container').append('<a href="more.html">more</a>');
```

• Respond to a user's interaction: Even the most elaborate and powerfulbehaviors are not useful if we can't control when they take place. The jQuerylibrary offers an elegant way to intercept a wide variety of events, such as auser clicking on a link, without the need to clutter the HTML code itself withevent handlers. At the same time, its event-handling API removes browserinconsistencies that often plague web developers.

```
$('button.show-details').click(function() {
$('div.details').show();
});
```

Animate changes being made to a document: To effectively implementsuch interactive behaviors, a designer must also provide visual feedbackto the user. The jQuery library facilitates this by providing an array ofeffects such as fades and wipes, as well as a toolkit for crafting new ones.

```
$('div.details').slideDown();
```

• Retrieve information from a server without refreshing a page: This codepattern is known as Ajax, which originally stood for Asynchronous JavaScriptand XML, but has since come to represent a much greater set of technologiesfor communicating between the client and the server. The jQuery libraryremoves the browser-specific complexity from this responsive, feature-richprocess, allowing developers to focus on the server-end functionality.

```
$('div.details').load('more.html #content');
```

• Simplify common JavaScript tasks: In addition to all of the documentspecific features of jQuery, the library provides enhancements to basicJavaScript constructs such as iteration and array manipulation.

```
$.each(obj, function(key, value) {
total += value;
});
```

## Why jQuery works well

With the resurgence of interest in dynamic HTML comes a proliferation of JavaScripframeworks. Some are specialized, focusing on just one or two of the tasks previouslymentioned. Others attempt to catalog every possible behavior and animation and servethese all up prepackaged. To maintain the wide range of features outlined earlier whileremaining relatively compact, jQuery employs several strategies:

• Leverage knowledge of CSS: By basing the mechanism for locatingpage elements on CSS selectors, jQuery inherits a terse yet legible way ofexpressing a document's structure. The jQuery library becomes an entry point for designers who want to add behaviors to their pages, because aprerequisite for doing professional web development is knowledge ofCSS syntax.

• Support extensions: In order to avoid "feature creep", jQuery relegatesspecial-case uses to plugins. The method for creating new plugins is simpleand well-documented, which has spurred the development of a wide varietyof inventive and useful modules. Even most of the features in the basicjQuery download are internally realized through the plugin architecture,and can be removed if desired, yielding an even smaller library.

Abstract away browser quirks: An unfortunate reality of web development isthat each browser has its own set of deviations from published standards. Asignificant portion of any web application can be relegated to handling featuresdifferently on each platform. While the ever-evolving browser landscapemakes a perfectly browser-neutral code base impossible for some advancedfeatures, jQuery adds an abstraction layer that normalizes the common tasks,reducing the size of code while tremendously simplifying it.

• Always work with sets: When we instruct jQuery to find all elements withthe class collapsible and hide them, there is no need to loop througheach returned element. Instead, methods such as .hide() are designedto automatically work on sets of objects instead of individual ones. Thistechnique, called implicit iteration, means that many looping constructsbecome unnecessary, shortening code considerably.

• Allow multiple actions in one line: To avoid overuse of temporaryvariables or wasteful repetition, jQuery employs a programming patterncalled chaining for the majority of its methods. This means that the resultof most operations on an object is the object itself, ready for the next actionto be applied to it.These strategies keep the file size of the jQuery package small, while at the sametime providing techniques for keeping our custom code that uses the librarycompact as well.

The elegance of the library comes about partly by design and partly due to theevolutionary process spurred by the vibrant community that has sprung up aroundthe project. Users of jQuery gather to discuss not only the development of plugins,but also enhancements to the core library. The users and developers also assist incontinually improving the official project documentation, which can be found athttp://api.jquery.com.

Despite all the efforts required to engineer such a flexible and robust system, theend product is free for all to use. This open source project is licensed under theMIT License to permit free use of jQuery on any site and facilitate its use withinproprietary software. If a project requires it, developers can relicense jQuery underthe GNU Public License for inclusion in other GNU-licensed open source projects.

## 12.2 jQuerySelectors

Understanding the DOM

One of the most powerful aspects of jQuery is its ability to make selecting elements in the DOM easy. The DOM serves as the interface between JavaScript and a web page; it provides a representation of the source HTML as a network of objects rather than as plain text.

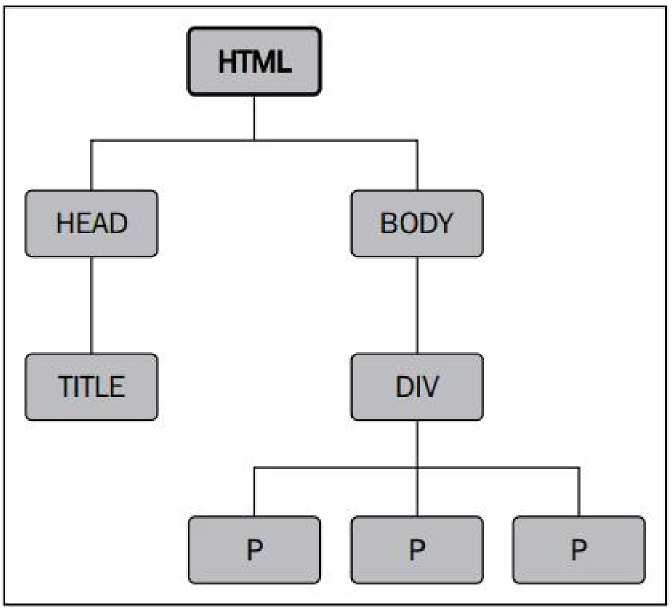
This network takes the form of a family tree of elements on the page. When we refer to the relationships that elements have with one another, we use the same terminology that we use when referring to family relationships: parents, children, and so on. A simple example can help us understand how the family tree metaphor

applies to a document:

```
<html>
<head>
<title>the title</title>
</head>
<body>

<div>

<p>This is a paragraph.</p>

<p>This is another paragraph.</p>

<p>This is yet another paragraph.</p>
```

```
</div>
</body>
</html>
```

Here, <html> is the ancestor of all the other elements; in other words, all the other elements are descendants of <html>. The <head> and <body> elements are not only descendants but children of <html> as well. Likewise, in addition to being the ancestor of <head> and <body>, <html> is also their parent. The <p> elements are children (and descendants) of <div>, descendants of <body> and <html>, and siblings of each other.



To help visualize the family tree structure of the DOM, we can use a numberof software tools, such as the Firebug plugin for Firefox or the Web Inspector inSafari or Chrome.

With this tree of elements at our disposal, we'll be able to use jQuery to efficientlylocate any set of elements on the page. Our tools to achieve this are jQuery selectorsand traversal methods.

### *Using the $() function*

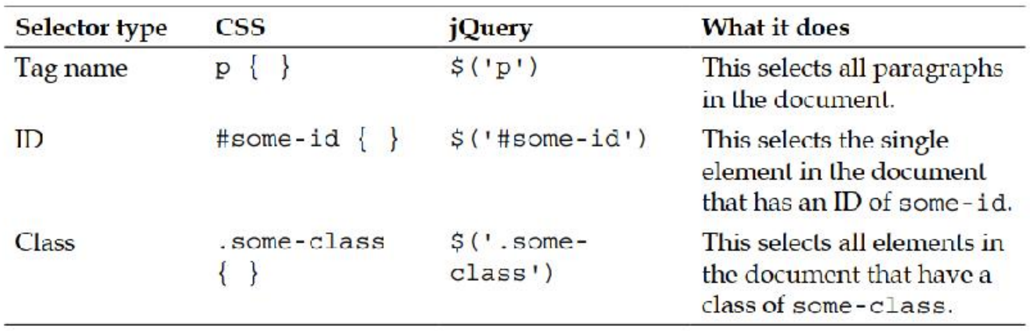
The resulting set of elements from jQuery's selectors and methods is alwaysrepresented by a jQuery object. Such a jQuery object is very easy to work withwhen we want to actually do something with the things that we find on a page.
We can easily bind events to these objects and add slick effects to them, as wellas chain multiple modifications or effects together.

**Notes**

Note that jQuery objects are different from regular DOM elements ornode lists, and as such do not necessarily provide the same methodsand properties for some tasks. In the final part of this chapter, we willlook at ways to directly access the DOM elements that are collectedwithin a jQuery object.

In order to create a new jQuery object, we use the $() function. This function typicallyaccepts a CSS selector as its sole parameter and serves as a factory returning a newjQuery object pointing to the corresponding elements on the page. Just about anythingthat can be used in a stylesheet can also be passed as a string to this function, allowingus to apply jQuery methods to the matched set of elements.

The three primary building blocks of selectors are tag name, ID, and class. They canbe used either on their own or in combination with others. The following simpleexamples illustrate how these three selectors appear in code:



| Selector type | CSS | jQuery | What it does |
|---|---|---|---|
| Tag name | `p { }` | `$('p')` | This selects all paragraphs in the document. |
| ID | `#some-id { }` | `$('#some-id')` | This selects the single element in the document that has an ID of `some-id`. |
| Class | `.some-class { }` | `$('.some-class')` | This selects all elements in the document that have a class of `some-class`. |

### *CSS selectors*

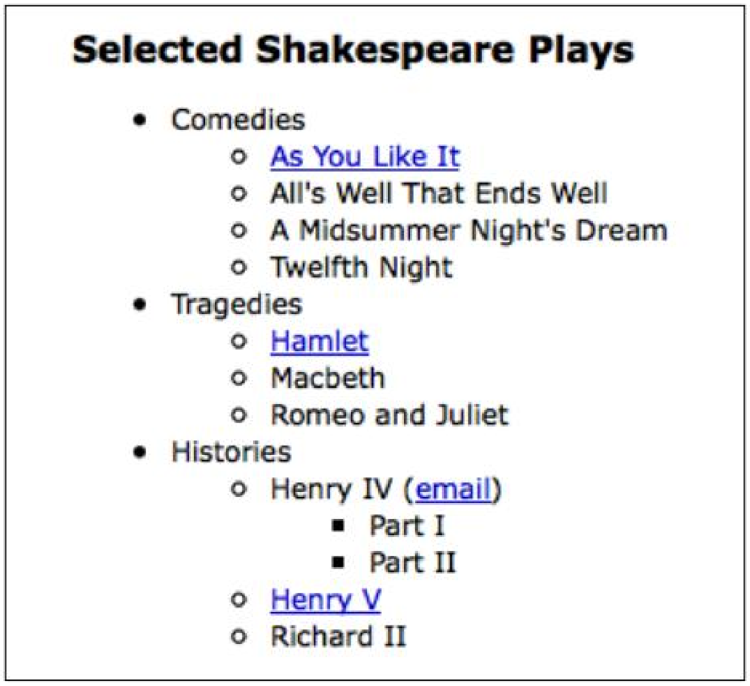
The jQuery library supports nearly all the selectors included in CSS specifications1 through 3, as outlined on the World Wide Web Consortium's site: http://www.w3.org/Style/CSS/specs. This support allows developers to enhance their websiteswithout worrying about which browsers might not understand more advancedselectors, as long as the browsers have JavaScript enabled.

To begin learning how jQuery works with CSS selectors, we'll use a structure thatappears on many websites, often for navigation – the nested unordered list:

```
<ul id="selected-plays">
<li>Comedies
<ul>
<li><a href="/asyoulikeit/">As You Like It</a></li>
<li>All's Well That Ends Well</li>
<li>A Midsummer Night's Dream</li>
<li>Twelfth Night</li>
</ul>
</li>
<li>Tragedies
<ul>
<li><a href="hamlet.pdf">Hamlet</a></li>
<li>Macbeth</li>
<li>Romeo and Juliet</li>
</ul>
</li>
<li>Histories
<ul>
<li>Henry IV (<a href="mailto:henryiv@king.co.uk">email</a>)
<ul>
<li>Part I</li>
<li>Part II</li>
</ul>
<li><a href="http://www.shakespeare.co.uk/henryv.htm">
Henry V</a></li>
<li>Richard II</li>
```

```
</ul>
</li>
</ul>
```

Notice that the first <ul> has an ID of selecting-plays, but none of the <li> tagshave a class associated with them. Without any styles applied, the list looks like this:



**Selected Shakespeare Plays**

- Comedies
  - As You Like It
  - All's Well That Ends Well
  - A Midsummer Night's Dream
  - Twelfth Night
- Tragedies
  - Hamlet
  - Macbeth
  - Romeo and Juliet
- Histories
  - Henry IV (email)
    - Part I
    - Part II
  - Henry V
  - Richard II

The nested list appears as we would expect it to—a set of bulleted items arrangedvertically and indented according to their level.

### Accessing DOM elements

Every selector expression and most jQuery methods return a jQuery object. Thisis almost always what we want because of the implicit iteration and chainingcapabilities that it affords.

Still, there may be points in our code when we need to access a DOM elementdirectly. For example, we may need to make a resulting set of elements available toanother JavaScript library, or we might need to access an element's tag name, whichis available as a property of the DOM element. For these admittedly rare situations,jQuery provides the .get() method. To access the first DOM element referred to bya jQuery object, for example, we would use .get(0). So, if we want to know the tag name of an element with an ID of my-element, we would write:

varmyTag = $('#my-element').get(0).tagName;

For even greater convenience, jQuery provides a shorthand for .get(). Insteadof writing the previous line, we can use square brackets immediately followingthe selector:

varmyTag = $('#my-element')[0].tagName;

It's no accident that this syntax appears to treat the jQuery object as an array of DOMelements; using the square brackets is like peeling away the jQuery layer to get atthe node list, and including the index (in this case, 0) is like plucking out the DOMelement itself.

## 12.3 Event Handling

All the different visitors' actions that a web page can respond to are called events.

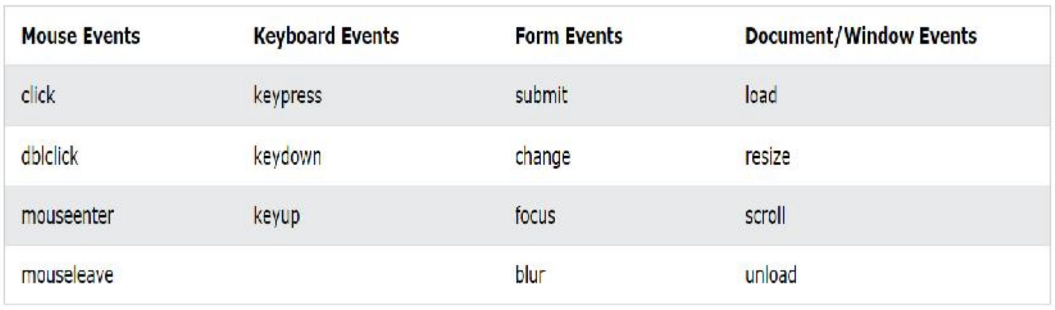
An event represents the precise moment when something happens.

Examples:

- moving a mouse over an element
- selecting a radio button
- clicking on an element

The term "fires/fired" is often used with events. Example: "The keypress event is fired, the moment you press a key".

Here are some common DOM events:



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

jQuery Syntax For Event

To assign a click event to all paragraphs on a page, you can do this:

```
$("p").click();
```

The next step is to define what should happen when the event fires. You must pass a function to the event:

```
$("p").click(function(){
  // action goes here!!
});
```

Commonly Used jQuery Event Methods

$(document).ready()

The $(document).ready() method allows us to execute a function when the document is fully loaded. This event is already explained in the jQuery Syntax chapter.

click()

The click() method attaches an event handler function to an HTML element.

The function is executed when the user clicks on the HTML element.

The following example says: When a click event fires on a <p> element; hide the current <p> element:

**#Code**

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

## Summary

- With the techniques that we have covered in this chapter, we should now be able to locate sets of elements on the page in a variety of ways. In particular, we learned how to style top-level and sub-level items in a nested list by using basic CSS selectors, how to apply different styles to different types of links by using attribute selectors, add rudimentary striping to a table by using either the custom jQuery selectors:odd and:even or the advanced CSS selector:nth-child(), and highlight text within certaintable cells by chaining jQuery methods.
- So far, we have been using the $(document).ready() method to add a class to amatched set of elements. In the next chapter, we'll explore ways in which to add aclass in response to a variety of user-initiated events.

## Keywords

**JQuery :** The jQuery library provides methods to handle DOM events. Most jQuery methods correspond to native DOM events.

**Events :** To handle DOM events using jQuery methods, first get the reference of DOM element(s) using jQuery selector and invoke appropriate jQuery event method.

**DOM:** The Document Object Model is a cross-platform and language-independent interface that treats an XML or HTML document as a tree structure wherein each node is an object representing a part of the document. The DOM represents a document with a logical tree.

## Self Assessment

1. Which of the following is a single global function defined in the jQuery library?

A. jQuery()
B. $()
C. Queryanalysis()
D. global()

2. Which of the following is a factory function?

A. $()
B. jQuery()
C. Queryanalysis()
D. onclick()

3. Which of the following is a heavily overloaded function?

A. jQuery()

B. $()

C. script()

D. Both jQuery() and $()

4. Which jQuery method is used to hide selected elements?

A. hidden()

B. hide()

C. visible(false)

D. display(none)

5. Which built-in method returns the character at the specified index?

A. characterAt()

B. getCharAt()

C. charAt()

D. None of the above

6. Using ________ function, we can hold or release the execution of jQuery's ready event.

A. jQuery.holdReady()

B. jQuery.ready()

C. jQuery.hold()

D. jQuery.holdready()

7. Which of the following jQuery method checks the current selection against an expression?

A. getIs( selector )

B. is( selector )

C. checkIs( selector )

D. None of the above

8. Is jQuery a library for client scripting or server scripting?

A. Server scripting

B. Client scripting

C. Both of these

D. None

9. In which year jQuery developed?

A. 2001

B. 2004

C. 2005

D. 2006

10. Which of the following sign is used as a shortcut for jQuery?
A. the % sign
B. the & sign
C. the $ sign
D. the @ sign

11. Which jQuery method is used to set one or more style properties to the selected element?
A. The html() method
B. The style() method
C. The css() method
D. All of the above

12. Which of the jQuery function prevents the code from running before the loading of the document finishes?
A. $(document).load()
B. $(document).unload()
C. $(document).ready()
D. $(document).trim()

13. JQuery is server side scripting.
A. True
B. False

14. With jQuery, look at the following selector: $("div.intro"). What does it select?
A. The first div element with class="intro"
B. All div elements with class="intro"
C. The first div element with id="intro"correct
D. All div elements with id="intro"

15. Which of the following is correct?
A. jQuery is a JavaScript Library
B. jQuery is a JSON Library
C. All of above
D. None of above

## Answers for SelfAssessment

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| l. | A | 2. | B | 3. | D | 4. | B | 5. | C |
| 6. | A | 7. | B | 8. | B | 9. | A | 10. | C |
| 11. | C | 12. | C | 13. | B | 14. | C | 15. | A |

## Review Questions

1. What is jQuery?
2. What is the difference between JavaScript and jQuery?
3. What are the effects methods used in jQuery?
4. What are the event methods used in jQuery?
5. Explain the followings.
    i. JQuery events
    ii. Applications of JQuery

## Further Readings

Bootstrap by Example, Silvio MoratoLearning Bootstrap 4, Matt Lambart

Bootstrap in 24 Hours, By SamsJump Start Bootstrap,

**Web Links**

https://api.jquery.com/category/events/

https://api.jquery.com