



JQUERY BASICS

TechRacine Academy

<https://techracine.com>

<https://academy.techracine.com>

Frontend Developer Intermediate Path

Table of Contents

1. Introduction to jQuery.....	3
2. Why jQuery Became Popular	3
3. Installing jQuery.....	3
4. jQuery Syntax	4
5. Selectors	4
6. DOM Manipulation.....	4
7. Event Handling.....	5
8. Effects and Animations.....	5
9. Traversing Elements	5
10. Working with CSS	6
11. Forms and Validation.....	6
12. AJAX in jQuery	7
13. JSON Handling	7
14. jQuery Utility Functions.....	8
15. jQuery Chaining	8
16. Dynamic Content Loading	8
17. Common Plugins.....	9
18. Performance Optimization.....	9
19. Best Practices	9
20. Common Mistakes.....	10
21. Modern Relevance of jQuery	10
22. Real World Usage	10
23. Summary.....	10
24. Exercises	11

1. Introduction to jQuery

jQuery is a fast and lightweight JavaScript library.

It simplifies DOM manipulation, event handling, AJAX, and animations.

Before modern JavaScript frameworks became popular, jQuery was heavily used in almost every web application.

Many enterprise applications and legacy systems still depend on jQuery.

Understanding jQuery is valuable because many existing projects continue to use it.

```
<script src="https://code.jquery.com/jquery-3.7.1.min.js"></script>
```

2. Why jQuery Became Popular

jQuery solved browser compatibility issues during the early web development era.

It simplified JavaScript syntax significantly.

Developers could perform complex tasks with very little code.

jQuery accelerated frontend development and improved productivity.

3. Installing jQuery

jQuery can be added using CDN or downloaded locally.

CDN is the easiest way for development and testing.

```
<script src="https://code.jquery.com/jquery-3.7.1.min.js"></script>
```

4. jQuery Syntax

jQuery syntax is built around selectors and actions.

The dollar symbol represents jQuery.

```
$(document).ready(function() {  
  
    console.log("Page Loaded");  
  
});
```

5. Selectors

Selectors identify HTML elements for manipulation.

jQuery supports element, class, ID, and advanced selectors.

Selectors are one of the most important jQuery concepts.

```
$("#title").hide();  
  
$(".card").fadeIn();  
  
$("p").css("color", "blue");
```

6. DOM Manipulation

jQuery simplifies modifying webpage content dynamically.

Developers can easily change text, HTML, and attributes.

```
$("#title").text("Welcome");  
  
$("#box").html("<b>Hello</b>");
```

7. Event Handling

Event handling responds to user interactions.

Common events include click, hover, keypress, and submit.

```
$("#btnSave").click(function() {  
  
    alert("Saved");  
  
});
```

8. Effects and Animations

jQuery provides built-in effects and animations.

Animations improve user interaction and visual feedback.

```
$("#panel").slideDown();  
  
$("#box").fadeOut();
```

9. Traversing Elements

Traversing allows movement through DOM elements.

Methods such as parent, children, find, and closest are commonly used.

```
$("#menu").children().hide();
```

10. Working with CSS

jQuery can dynamically modify CSS properties.

This helps create interactive UI behaviors.

```
$("#card").css({  
  
    background: "blue",  
    color: "white"  
  
});
```

11. Forms and Validation

jQuery is widely used for client-side form validation.

Validation improves user experience and data quality.

```
$("#frmLogin").submit(function() {  
  
    if($("#txtName").val() == "") {  
  
        alert("Name Required");  
  
        return false;  
    }  
  
});
```

12. AJAX in jQuery

AJAX allows webpages to communicate with servers without reloading.

jQuery simplified AJAX compared to traditional JavaScript.

```
$.ajax({  
  
    url: "data.php",  
    method: "GET",  
  
    success: function(data) {  
  
        console.log(data);  
  
    }  
  
});
```

13. JSON Handling

JSON is commonly used in APIs and AJAX responses.

jQuery works easily with JSON data.

```
$.getJSON("users.json", function(data) {  
  
    console.log(data);  
  
});
```

14. jQuery Utility Functions

jQuery provides utility functions for arrays, objects, and iteration.

These functions simplify development.

```
$.each(users, function(index, user) {  
  
    console.log(user);  
  
});
```

15. jQuery Chaining

jQuery supports chaining multiple actions together.

This improves readability and reduces code duplication.

```
$("#box")  
  
    .addClass("active")  
  
    .fadeIn()  
  
    .css("color", "red");
```

16. Dynamic Content Loading

jQuery can dynamically load external content into webpages.

This technique is common in dashboards and admin portals.

```
$("#content").load("menu.html");
```

17. Common Plugins

jQuery supports thousands of plugins.

Popular plugins include DataTables, Select2, and jQuery UI.

Plugins significantly extend functionality.

18. Performance Optimization

Efficient selectors improve performance.

DOM manipulation should be minimized when possible.

Large applications should avoid excessive jQuery nesting.

19. Best Practices

Use meaningful selectors.

Keep jQuery organized and modular.

Minimize direct DOM manipulation.

Prefer reusable functions.

Validate forms properly.

20. Common Mistakes

Avoid excessive nested selectors.

Avoid duplicate event bindings.

Do not overuse animations.

Keep AJAX calls optimized.

Test interactions across browsers.

21. Modern Relevance of jQuery

Modern frameworks reduced the dominance of jQuery.

However, many enterprise applications still use jQuery.

Maintaining older systems often requires strong jQuery knowledge.

22. Real World Usage

jQuery is still widely used in ERP systems, admin dashboards, enterprise portals, and legacy web applications.

Many plugins and CMS platforms continue to depend on jQuery.

23. Summary

jQuery simplified frontend development and became one of the most important JavaScript libraries.

Understanding selectors, DOM manipulation, AJAX, animations, and events remains valuable for intermediate developers.

jQuery knowledge is especially useful in enterprise maintenance projects.

24. Exercises

1. Create a dynamic form validation system.
2. Build a hide/show content panel.
3. Practice AJAX requests.
4. Create a dynamic table search.
5. Build interactive buttons using animations.
6. Practice jQuery chaining.
7. Load external content dynamically.
8. Build a responsive image gallery.
9. Practice DOM traversal methods.
10. Create a mini admin dashboard interaction.