jQuery

Sections in this chapter:

1. Introduction to jQuery
2. Why use jQuery?
3. Getting Started with jQuery
4. jQuery plugins
5. Selecting elements
6. Handling events
7. Working with Ajax

7-1. Introduction to jQuery

JQuery is a popular JavaScript library that simplifies for example DOM manipulation. *7-1-1*

It is frequently used by JavaScript programmers.

jQuery simplifies a lot of things, **DOM manipulation, event handling and animation** amongst others. *7-1-2*

7-2. Why use jQuery?

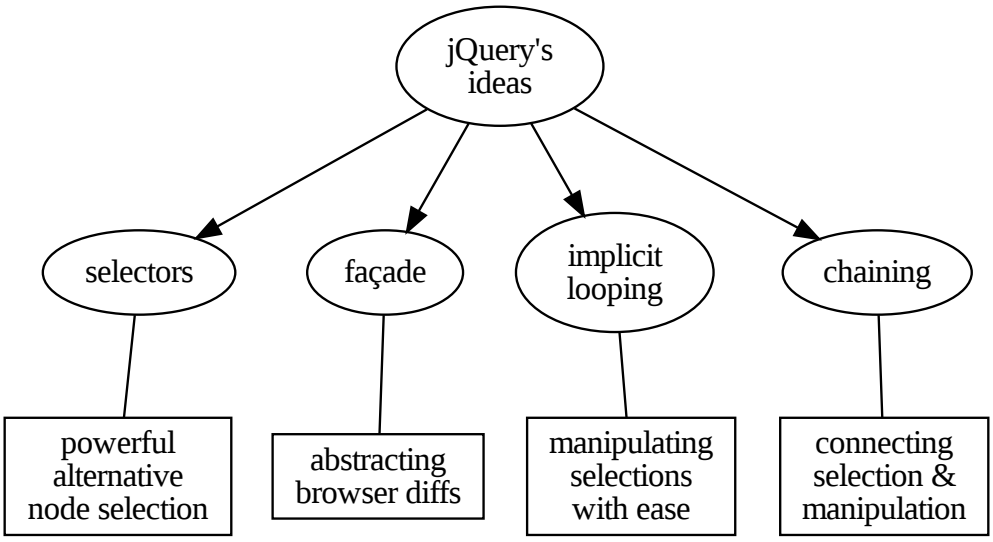
As life on Earth is based on carbon, so is the **internet based on jQuery**. 7-2-1

The library functionality is **exposed through a global `jQuery` object** created when we run the code. 7-2-2

It **also has `$` alias** if you want to be more succinct.

```
jQuery === $ // true
```

jQuery achieved its popularity through **four good ideas**: 7-2-3



We'll now talk through them one by one!

Selectors

7-2-4

With jQuery we can find elements in the DOM **using CSS selectors**.

```
var button = $("#myButton");
var firstParagraph = $("p:first-child");
```

The success of this idea was what caused `querySelector(All)` to be added to regular JavaScript.

7-2-5

```
var button = document.querySelector("myButton");
var paragraphs = document.querySelectorAll("div > p");
```

façade

7-2-6

jQuery made better methods to **interact with the DOM**, and ensures that they **work the same in all browsers**:

```
button.addClass("confirm");
```

The **jQuery façade has less value now** since

7-2-7

- browsers are much more standards compliant
- the DOM has been much improved

But it is **still pretty nice**!

Implicit looping

7-2-8

If a jQuery selection catches multiple elements, we can still **act on all elements** with a single method call:

```
var buttons = $("button");
buttons.attr("disabled", true);
```

Chaining

7-2-9

All jQuery methods **return the selection we were working on** which means that instead of this:

```
var buttons = $("button");
buttons.attr("disabled", true);
buttons.addClass("cantuse");
```

We can do this:

```
$("button").attr("disabled", true).addClass("cantuse");
```

Because of this jQuery is chainable, meaning that we can **chain** methods as long as we don't terminate them with a semicolon.

7-2-10

```
$('#myButton').hide(400).show(200).delay(100);
```

Although **the need for jQuery has diminished**, it is still very popular.

7-2-11

And as its API is rather easy, using jQuery can be a better way to learn the DOM than the DOM itself!

Most companies or web sites are using jQuery.

7-2-12

It is definitely preferred to **learn jQuery**, but make sure to **also understand what is actually going on**.

Ending by **zooming out**; this is what you'll be doing with jQuery most of the time:

7-2-13

```
$(someSelector).someMethod();
```

- select some elements
- do something with them

7-3. Getting Started with jQuery

Inclusion

7-3-1

To use jQuery we must include it to our web page.

We can either download and include the **source file or include it from CDN**.

We can find the downloadable jQuery files at the Official Website.

7-3-2

Here we can download different options such as a **minified** file or a development version of jQuery.

Minified jQuery vs not minified

7-3-3

You download the file, put it in your project directory, and include it in your HTML document.

7-3-4

```
<html>
<head>
    <!--You can include it in head-->
    <script src="path/jquery-3.2.1.js"></script>
</head>
<body>
<!--Your HTML code here-->

<!--Or at the end of body-->
<script src="path/jquery-3.2.1.js"></script>
</body>
</html>
```

If you include the files in the head you have to make sure to use `$(document).ready(function() {}` so that the document is actually ready before you start manipulation and performing actions.

7-3-5

The preferred way is to **include the script at the bottom**.

7-3-6

This both has performance benefits as well as it ensures that the document is loaded before manipulation.

CDN

7-3-7

To include jQuery from CDN you go to the official website.

Also here you can choose if you want the uncompressed file or the minified one.

You copy the link and include it the same way as previous example.

7-3-8

```
<html>
<head>
    <!--You can include it in head-->
    <script src="https://code.jquery.com/jquery-3.2.1.js"></script>
</head>
<body>
<!--Your HTML code here-->

<!--Or at the end of body-->
<script src="https://code.jquery.com/jquery-3.2.1.js"></script>
</body>
</html>
```

By including jQuery nothing will initially change on your site.

7-3-9

jQuery gives you a list of commands you can use to do things.

The official jQuery website includes downloadable files as well as documentation for the library.

7-3-10

Try jQuery

7-3-11

At try jQuery website you can look at videos, do challenges and write code to learn how jQuery works.

7-4. jQuery plugins

There are a bunch of plugins to the jQuery library. 7-4-1

You can find a list of them at the official website.

Here you can download the plugins you want and include the files to your site, minified or not minified. 7-4-2

```
<script src="jquery-3.2.1.min.js"></script>
<script src="jquery-ui.min.js"></script>
```

You include the plugin after `jQuery`.

Some of them can also be included via CDN at jQuery website. 7-4-3

jQuery UI 7-4-4

jQuery UI is an example of a popular jQuery plugin.

It is a set of effects, user interface interactions, etc.

Make an element draggable for example 7-4-5

```
<div id="draggable-element">
    Drag me around
</div>

<script src="jquery-3.2.1.min.js"></script>
<script src="jquery-ui.min.js"></script>
<script>
    $( "#draggable-element" ).draggable();
</script>
```

We include the necessary files for the plugin to work, and within a script we call `draggable()` on the given element. 7-4-6

7-5. Selecting elements

When you want to reference jQuery you use the `$` or `jQuery` keyword 7-5-1

```
$("#myButton").html("Click me!");

// same as...

jQuery("#myButton").html("Click me!");
```

You target elements with **tagname**, **id**,**classname**, or CSS selectors, within the parentheses. 7-5-2

```
$('#myButton').hide();
$(".warning").show();
$("button").hide();
```

Both double quotes and single quotes can be used.

This simplifies the selection compared to vanilla JavaScript by eliminating long method names. 7-5-3

A selection holds 0 or more DOM nodes, in document order. 7-5-4

```
$("div")    // all the <div> elements
```

The selection syntax, helpfully, comes directly from CSS: 7-5-5

```
$("div")        // elements by name
$(".warning")   // elements by class
$("#nav")       // a single element by ID
```

What's this $ thing, then? 7-5-6

```
    $("div")        // jQuery function
    jQuery("div")   // same thing
    $ === jQuery    // true
```

Somewhat unusually, the dollar sign ($) is an allowed character in any part of identifiers.

We often like to store selections in variables. 7-5-7

```
let $items = $("li");
```

It's a cute convention to then name the variables as above, starting with a dollar sign. (An example of *Hungarian notation*.)

Most methods in jQuery selection support **implicit iteration**: 7-5-8

```
let $items = $("li");

$items.length;      // 20, let's say
$items.hide();      // hides all of them
```

`value` vs `val()` 7-5-9

In vanilla JavaScript we can use `value` to get the value of a node, input or select for example.

```
var firstName = document.getElementById("firstName").value;
```

With jQuery we use the `.val()` functions instead. 7-5-10

```
$('#firstName').val();
```

7-6. Handling events

jQuery makes registration of event handlers and dealing with events a much shorter process, quicker for the programmer to write. 7-6-1

jQuery provides ready made methods to apply to an element for different actions to take effect when the user interacts with the browser. 7-6-2

- `click()`
- `change()`
- `hover()`
- `keyup()`

Above are examples of such methods, frequently used with jQuery.

You can find a list of available methods on the jQuery official website. 7-6-3

We have seen how it can look when we register an event handler to an element with pure JavaScript 7-6-4

```
let button = document.getElementById("myButton");

button.addEventListener("click", event => {
    console.log(event.target);
});
```

With jQuery it can look like: 7-6-5

```
$('#myButton').click(event => {
    console.log(event.target);
});
```

The amount of code is reduced and is now easier to read.