
Session 27

Mohamed Emary

July 17, 2024

1 Revision

When you make a class using sugar syntax, you can't define variables with the `let`, `const`, `var` keywords outside of the `constructor` method. You can define them without the keywords, but they will be treated as properties of the class. You can't also use `function` keyword to define methods.

2 JQuery

2.1 Library vs Framework

- **Library:** A library is a small piece of code that you can call from your own code, to help you do things more quickly/easily. For example, a library might include code related to animation for example. Libraries are usually small and focus on doing one thing and doing it well. Ex: JQuery, React.
- **Framework:** A framework is a collection of code that, together, provides a solution to a problem. Frameworks are usually large and complex. They can include many libraries. Ex: Angular.

2.2 Why JQuery?

- **Simplicity:** It makes things like HTML document traversal and manipulation, event handling, and animation much simpler.
- **Write less, do more:** It allows you to achieve more things using less code.

JQuery has a lot of methods that allow you to manipulate the DOM, create animations, and handle events.

2.3 How to use JQuery?

You can use JQuery by including the library in your HTML file. You can download the library from the JQuery website or use a CDN.

At the end of the body tag, you can include the library before your own script.

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

You can also use CDN or download the library and add it to your project files then include it in your HTML file.

3 How to use JQuery?

You can use JQuery by selecting elements and applying methods on them.

3.1 Selecting elements

You can select elements using CSS selectors, on the form `$('#selector')`.

- **Tag name:** `$('#p')`
- **Class:** `$('.class-name')`
- **ID:** `$('#id-name')`
- **Attribute:** `$('#input[type="text"]')`

Side Note:

You can also use `jQuery('selector')` instead of `$('#selector')` and it will work the same way.

This is useful if you are using another library that uses the `$` symbol.

3.2 Effects

You can use JQuery to create animations and effects on elements.

Examples:

Consider this HTML code:

```
1 <div id="my-div" class="bg-info" style="width: 300px; height: 300px"></div>
2 <button class="btn btn-danger">Click</button>
```

Ex 1:

- To make the `div` element fade out when you click the button you can use the `fadeOut()` method.

```
1 $('#button').click(function() {
2     $('#my-div').fadeOut();
3 });
```

You can also pass a duration to the `fadeOut()` method ex: `fadeOut(1000)`.

To hide the element:

```
1 $('#button').click(function() {
2     $('#my-div').hide();
3 });
```

- If the element has a `display: none` style, the `show()` method will show it.

```
1 $('button').click(function() {
2   $('#my-div').show();
3 });
```

- The `toggle()` method will show the element if it's hidden and hide it if it's shown.

More Functions:

- `slideUp()`, `slideDown()`, `slideToggle()`: Slide an element up, down, or toggle it. These methods work with the element **height**.
 - Toggle means that if the element is hidden, it will show it, and if it's shown, it will hide it.
- `fadeIn()`, `fadeOut()`, `fadeToggle()`, `fadeTo()`: Fade an element in, out, toggle it, or change its **opacity** to a specific value ex. `fadeTo(1000, 0.5)`.

3.3 Effects With Callbacks

Consider this HTML code:

```
1 <div id="demo1" class="bg-info" style="width: 200px; height: 200px"></div>
2 <div id="demo2" class="bg-dark" style="width: 200px; height: 200px"></div>
3 <div id="demo3" class="bg-warning" style="width: 200px; height:
  ↪ 200px"></div>
4
5 <button class="btn btn-danger">Hide All</button>
```

And this JS code:

```
1 $('button').click(function() {
2   $('#demo1').hide(1000);
3   $('#demo2').hide(1000);
4   $('#demo3').hide(1000);
5 });
```

Since the `hide()` method is asynchronous and non-blocking, the three elements will hide at the same time.

How to hide them one after the other?

You can use a callback function by passing a function to the `hide()` method and that function will be called when the element is hidden.

```
1 $('#demo1').hide(1000, function() {
2   $('#demo2').hide(1000, function() {
3     $('#demo3').hide(1000);
4   });
5 });
```

This will hide the elements one after the other.

Notice that if these effects are applied to the same element, they will be executed one after the other without the need for a callback function.

If you apply multiple effects to the same element, you can use the chaining method.

```
1 | $('#demo1').hide(1000).show(1000).fadeOut(1000)
2 |
3 | // Equivalent to
4 | $('#demo1').hide(1000);
5 | $('#demo1').show(1000);
6 | $('#demo1').fadeOut(1000);
```

3.4 Custom Effects With animate()

You can create custom animations using the `animate()` method.

Syntax:

```
1 | $('#selector').animate({styles}, duration, easing, callback);
```

Example:

```
1 | $('#demo1').animate({width: '300px'}, 1000);
```

This will animate the width of the element to 300px in 1 second.

You can have multiple styles in the object passed to the `animate()` method.

```
1 | $('#demo1').animate({width: '300px', height: '300px', borderRadius: '50%'},
   | ↪ 1000);
```

This will animate the width, height, and border-radius of the element at the same time to the specified values in 1 second.

To make them animate one after the other, you can just chain the `animate()` methods.

```
1 | $('#demo1').animate({width: '300px'}, 1000).animate({height: '300px'},
   | ↪ 1000).animate({borderRadius: '50%'}, 1000);
2 |
3 |
4 | // or
5 | $('#demo1').animate({width: '300px'}, 1000);
6 | $('#demo1').animate({height: '300px'}, 1000);
7 | $('#demo1').animate({borderRadius: '50%'}, 1000);
```

If you have two elements and you want to animate them one after the other, you can use the callback function.

```
1 | $('#demo1').animate({width: '300px'}, 1000).animate({height: '300px'},
   | ↪ 1000, function() {
2 |   $('#demo2').animate({width: '300px'}, 1000).animate({height: '300px'},
   | ↪ 1000);
3 | });
```

This will animate the first element by changing its width then height, then animate the second element by changing its width then height.

Above example Steps:

First element width $\xrightarrow{\text{Then}}$ Its height \rightarrow Second element width $\xrightarrow{\text{Then}}$ Its height

If you have an element you can even make the element increase its width by 50px from its current width with each click.

```
1 | $('button').click(function() {
2 |     $('#demo1').animate({width: '+=50px'}, 1000);
3 | });
```

Important Note

`animate()` method works with CSS properties that have numeric value only.

To stop animation with a button click, you can use the `stop()` method.

```
1 | $('button').click(function() {
2 |     $('#demo1').stop();
3 | });
```

This will stop the animation of the element but other animations will continue.

If you pass `true` to the `stop()` method, it will clear the animation queue and stop all animations.

If you stop an animation (even without `true` as parameter) and this animation calls back another animation, the callback function will not be executed.

4 DOM Manipulation With JQuery

JQuery allows you to manipulate the DOM easily.

4.1 Get and Set

You can get and set the content, HTML, value, attributes, and CSS properties of elements using JQuery.

4.1.1 Text Content

To get the content of an element, you can use the `text()` method.

```
1 | $('selector').text();
```

You can also pass a string to the `text()` method to set the content of the element.

```
1 | $('selector').text('New text');
```

4.1.2 HTML Content

To get the HTML content of an element, you can use the `html()` method.

```
1 | $('selector').html();
```

You can also pass a string to the `html()` method to set the HTML content of the element.

```
1 | $('selector').html('<h1>New content</h1>');
```

4.1.3 Input Value

To get the value of an input element, you can use the `val()` method.

```
1 | $('input').val();
```

You can also pass a string to the `val()` method to set the value of the input element.

```
1 | $('input').val('New value');
```

4.1.4 Attribute

To get the attribute of an element, you can use the `attr()` method.

```
1 | $('selector').attr('attribute-name');
```

You can also pass a string to the `attr()` method to set the attribute of the element.

```
1 | $('selector').attr('attribute-name', 'value');
```

Example:

```
1 | <input type="text" id="my-input">
1 | $('input').attr('type'); // 'text'
```

4.1.5 CSS

You can get the CSS value of a property of an element using the `css()` method.

```
1 | $('selector').css('property-name');
```

You can also pass a string to the `css()` method to set the CSS property of the element.

```
1 | $('selector').css('property-name', 'value');
```

Example:

```
1 | $('#demo1').css('background-color'); // 'rgb(255, 0, 0)'  
2 | $('#demo1').css('color', 'red');  
3 |  
4 |  
5 | // To set multiple CSS properties  
6 | $("#demo1").css({  
7 |   width: "100px",  
8 |   height: "100px",  
9 |   borderRadius: "50%",  
10 |   transitionDuration: "1s",  
11 | });
```

4.1.6 Width, Height

You can get and set the width and height of an element using the `width()` and `height()` methods.

```
1 | $('selector').width();  
2 | $('selector').height();
```

You can also pass a number to the `width()` and `height()` methods to set the width and height of the element.

```
1 | $('selector').width(100);
2 | $('selector').height(100);
```

You can get the inner width and height of an element using the `innerWidth()` and `innerHeight()` methods.

`innerWidth = width + padding`

```
1 | $('selector').innerWidth();
2 | $('selector').innerHeight();
```

You can get the outer width and height of an element using the `outerWidth()` and `outerHeight()` methods.

`outerWidth = width + padding + border`

```
1 | $('selector').outerWidth();
2 | $('selector').outerHeight();
```

You can also pass `true` to the `outerWidth()` and `outerHeight()` methods to include the margin.

```
1 | $('selector').outerWidth(true);
2 | $('selector').outerHeight(true);
```

4.2 Add and Remove

You can add and remove elements, classes, and attributes using JQuery.

4.2.1 Elements

To add elements, you can use the `append()`, `prepend()`, `after()`, and `before()` methods.

- `append()`: Add elements inside the selected element at the end.
- `prepend()`: Add elements inside the selected element at the beginning.
- `after()`: Add elements after the selected element.
- `before()`: Add elements before the selected element.

Example:

Consider this HTML:

```
1 | <div id="my-div"></div>
```

And in JS:

```
1 | $('#my-div').append('<p>New paragraph</p>'); // Child at the end
2 | // <div id="my-div"><p>New paragraph</p></div>
3 |
4 | $('#my-div').prepend('<p>New paragraph</p>'); // Child at the beginning
5 | // <div id="my-div"><p>New paragraph</p></div>
6 |
7 | $('#my-div').after('<p>New paragraph</p>'); // Sibling after
8 | // <div id="my-div"></div><p>New paragraph</p>
9 |
```

```
10 | $('#my-div').before('<p>New paragraph</p>'); // Sibling before
11 | // <p>New paragraph</p><div id="my-div"></div>
```

You can even move existing element by passing it to the `append()`, `prepend()`, `after()`, and `before()` methods.

```
1 | $('#my-div').append($('p')); // Moves the p element inside the div element
```

To remove elements, you can use the `remove()` method.

```
1 | $('selector').remove();
```

This will remove the selected element.

To empty an element, you can use the `empty()` method.

```
1 | $('selector').empty();
```

This will remove all the children of the selected element.

4.2.2 Classes

You can add and remove classes using JQuery.

To add a class to an element, you can use the `addClass()` method.

```
1 | $('selector').addClass('class-name');
2 |
3 | // To add multiple classes
4 | $('selector').addClass('class1 class2 class3');
```

To remove a class from an element, you can use the `removeClass()` method.

```
1 | $('selector').removeClass('class-name');
```

To toggle a class on an element, you can use the `toggleClass()` method.

```
1 | $('selector').toggleClass('class-name');
```

This will add the class if it's not there and remove it if it's there.

4.2.3 Attributes

You can add and remove attributes using JQuery.

To add an attribute to an element, you can use the `attr()` method.

```
1 | $('selector').attr('attribute-name', 'value');
```

To remove an attribute from an element, you can use the `removeAttr()` method.

```
1 | $('selector').removeAttr('attribute-name');
```

5 Traversing

JQuery allows you to traverse the DOM easily.

- `.parent()`: The direct parent of the selected element.
- `.parents()`: All the ancestors of the selected element.

- `.parentsUntil('selector')`: All the ancestors of the selected element until a specific element, and that element is not included.

You can use the methods mentioned before to apply styles, add classes, and remove elements from the elements you are traversing.

- `.children()`: The direct children of the selected element.
 - `.children('selector')`: The direct children of the selected element that match the selector.
- `.find('selector')`: The descendants of the selected element that match the selector.
 - `.find('*')`: All the descendants of the selected element.
- `.siblings()`: The siblings of the selected element.
 - `.siblings('selector')`: The siblings of the selected element that match the selector.
- `.next()`, `.prev()`: The next or previous sibling of the selected element.
- `.nextAll()`, `.prevAll()`: All the next or previous siblings of the selected element.
- `.nextUntil('selector')`, `.prevUntil('selector')`: All the next or previous siblings of the selected element until a specific element, and that element is not included.
- `$('span').not('selector')`: Select all span elements that do not match the selector.
- `$('span').eq(index)`: Select the span element at the specified index.
- `$('span').first()`: Select the first span element.
- `$('span').last()`: Select the last span element.
- `$('span').filter('selector')`: Select the span elements that match the selector.

For a nice JQuery cheat sheet, you can visit [this link](#).