

# jQuery

`\$("element subelement[attribute=value]:filter").doSomething(parameters)`

## Importing

```
<head>  
    <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>  
</head>
```

## Selecting Elements

| Select By         | Example  |
|-------------------|--|
| ID                | <code>\$("#header")</code>   |
| Class             | <code>\$(".updated")</code>  |
| Tag Name          | <code>\$("table")</code>   |
| Combination       | <code>\$("table.user-list")</code> or <code>\$("#footer ul.menu li")</code>  |
| Basic Filters     | <code>:first</code> , <code>:last</code> , <code>:even</code> , <code>:odd</code>  |
| Content Filters   | <code>:empty</code> , <code>:contains(text)</code> , <code>:has(selector)</code>   |
| Attribute Filters | <code>[attribute]</code> , <code>[attribute=value]</code> , <code>[attribute!=value]</code>  |
| Forms             | <code>:input</code> , <code>:text</code> , <code>:submit</code> , <code>:password</code> , <code>:enabled</code> , <code>:checked</code> |

## Actions

| Action           | Example   |
|------------------|---|
| DOM Manipulation | <code>before()</code> , <code>after()</code> , <code>append()</code> , <code>appendTo()</code>                                      |
| Attributes       | <code>addClass()</code> , <code>css()</code> , <code>attr()</code> , <code>html()</code> , <code>val()</code> , <code>text()</code> |
| Events           | <code>click()</code> , <code>on()</code> , <code>bind()</code> , <code>unbind()</code> , <code>live()</code>                        |
| Effects          | <code>hide()</code> , <code>fadeOut()</code> , <code>toggle()</code> , <code>animate()</code>                                       |
| AJAX             | <code>load()</code> , <code>get()</code> , <code>ajax()</code> , <code>post()</code> , <code>getJSON()</code>                       |

# Events

```
$("#span#message").click(function(event){...});
```

OR

```
$("#span#message").on("click", function(event) {...});
```

Without function, it becomes a manual triggering of the event: `$("#span#message").click()` clicks the span with ID message.

- `this` attribute is used in a event handler function to reference the element that's been selected

```
$( "p" ).click(function() {  
    var htmlString = $( this ).html();  
})
```

- `hide()` / `show()` : Hide or show elements.
- `toggle()` : Toggle visibility of elements.
- `fadeIn()` / `fadeOut()` : Fade in or out by changing opacity.
- `slideToggle()` : Toggle visibility with a sliding effect.
- `animate()` : Create custom animations by changing CSS properties over time.

# Promises

## Producing Code (Creates the promise):

```
let fetchData = new Promise(function(resolve, reject) {  
    let data = "data fetched";  
  
    if (data) {  
        resolve(data); // Data is successfully fetched  
    } else {  
        reject("Error fetching data");  
    }  
});
```

## Consuming Code (Waits for the result):

```
fetchData.then(function(result) {  
    console.log(result); // Logs "data fetched"  
}).catch(function(error) {  
    console.error(error); // Logs "Error fetching data" if rejected  
});
```

## 1. `.then()` Method:

- The `.then()` method is primarily used to handle the resolution (success) of a promise.
- It takes two arguments:
  - The first function is called when the promise is resolved.
  - The second function is called when the promise is rejected (although this is less common and not best practice).

## 2. `.catch()` Method:

- The `.catch()` method is specifically designed to handle rejections (failures) of a promise.
- It can be used after a `.then()` to catch any errors that occurred in the promise chain.

## Promise Object

- Two properties -> state and result

| State     | Result    |
|-----------|-----------|
| Pending   | Undefined |
| Fulfilled | Value     |
| Rejected  | Error     |

- Cannot be accessed directly, promise methods need to be used