

# jQuery

1

A JavaScript library for writing  
shorter code

2

## Prereq: Import the Library

```
<!-- add the following in the header of your HTML -->  
<script src=" " ></script>
```

Several Options (**min**ified production versions)

- <https://code.jquery.com/jquery-3.2.1.min.js>
- <https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js>
- <http://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.2.1.min.js>
- <https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js>

3

jQuery = select (query) elements  
& perform actions

4

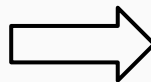
# Why jQuery?

1. Query/select elements from the DOM tree
2. Perform actions on the selected elements
  - `hide()` / `show()`
  - Apply events, attach event listeners
  - Update styles
  - Update attributes
  - Manipulate DOM
  - ....

5

## First use of jQuery: Select Elements

- `document.getElementById()`
- `document.getElementsByTagName()`
- `document.getElementsByClassName()`
- `document.querySelector()`
- `document.querySelectorAll()`



`$("____")`

6

## Pure JavaScript to jQuery

Pure JavaScript	jQuery
<code>var el = document.getElementById("mainTable")</code>	<code>var el = \$("#mainTable");</code>
<code>var els = document.getElementsByTagName("h2");</code>	<code>var els = \$("h2");</code>
<code>var els = document.getElementsByClassName("warn");</code>	<code>var els = \$(".warn");</code>
<code>var els = document.querySelectorAll("tr:even");</code>	<code>var els = \$("tr:even")</code>
<code>var el = document.querySelector("tr");</code>	<code>var el = \$("tr").first();</code>

Any "CSS selectors" you learn from the CSS lecture can be used inside `$("_____")`

7

## jQuery Actions

`$("_____").action_function()`

8

## Event Handling

```
// pure JavaScript
var saveBtn = document.getElementById("save");
saveBtn.addEventListener('click',
    function (event) {
        /* event handling logic goes here */
    }
);
```

```
// jQuery
$("#save").click (
    function (event) {
        /* event handling logic goes here */
    }
);
```

### Other event-related functions

- dblclick()
- mouseenter()
- mouseleave()
- mouseup()
- mousedown()
- hover()
- focus()
- blur()

9

## Manipulate Element Visual Effects

```
// pure JavaScript
var saveBtn = document.getElementById("save");
var loadBtn = document.getElementById("load");
saveBtn.hidden = true;
loadBtn.hidden = false;
```

```
// jQuery
$("#save").hide();
$("#load").show();
```

***So mch shrtr!!***

### Other related functions

- fadeIn() / fadeOut(): changes transparency/opacity
- slideDown()/slideUp(): changes height
- show() / hide(): changes both width & height (scale)
- animate(): *you have more control on which properties to animate*

10

## jQuery animate()

*Desired properties when the animation ends.  
A more intuitive name: animateTo()*

```
$(this).animate({ visual-properties }, 'slow', end_anim_callback);
```

```
$(“_____”).mouseenter( function () {  
    $(this).animate({ fontSize: '120%' },  
        'slow');  
});  
  
$(“_____”).mouseleave( function () {  
    $(this).animate({ fontSize: '100%' },  
        'fast'); /* or time in milliseconds */  
});
```

[List of animatable properties](#)

11

jQuery actions are applied to  
**every one of the**  
selected elements

12

## What does it do?

```
$(“input[type=text]:invalid”)  
  .css({‘background’: ‘red’});
```

13

## jQuery implied loop

```
// pure JavaScript  
var pars = document.getElementsByTagName(“p”);  
for (var k = 0; k < pars.length; k++) {  
  pars[k].style.fontSize = ‘24pt’;  
  pars[k].style.background = ‘green’;  
}
```

```
// jQuery equivalent  
$(“p”).css({‘fontSize’: ‘24pt’,  
            ‘background’: ‘green’});
```

14

# jQuery CSS Class Manipulation Functions

```
var el = document.getElementById("sample");
```

Pure JavaScript	jQuery
<code>el.className += "warn";</code>	<code>\$("#sample").addClass("warn");</code>
<code>var temp = el.className.replace("warn", ""); el.className = temp;</code>	<code>\$("#sample").removeClass("warn");</code>
<code>/* if-else required in JavaScript */</code>	<code>\$("#sample").toggleClass("warn");</code>

15

## jQuery CSS Manipulation Function

```
// pure JavaScript  
var el = document.getElementById('sample');  
el.style.background = 'blue';
```

```
// jQuery  
$('#sample').css('background', 'blue');
```

16



# AJAX in jQuery

17

## AJAX: no post processing

```
<script>
// pure JavaScript
var xhr = new XMLHttpRequest();
var target = document.getElementById("destination");
xhr.onload = function(e) {
    target.innerHTML = xhr.responseText;
};
xhr.open ("GET", url_of_the_datasource, true);
xhr.send();
</script>
```

```
<script>
// Using jQuery
$("#destination").load(url_of_the_datasource);
</script>
```

18

## AJAX: with post processing

```
// pure JavaScript
var xhr = new XMLHttpRequest();
xhr.onload = function(e) {
  if (xhr.status == 200) {
    var json = JSON.parse(xhr.responseText);
    // your code here
  }
};
xhr.open ("GET", url_to_a_JSON_resource, true);
xhr.send();
```

```
// Using jQuery
$.get(url_to_a_JSON_resource, function(data, status) {
  if (status == "success") {
    // data is already parsed (as JSON)
    // your code here
  }
});
```