

JQuery

A better way to do JavaScript



COS216

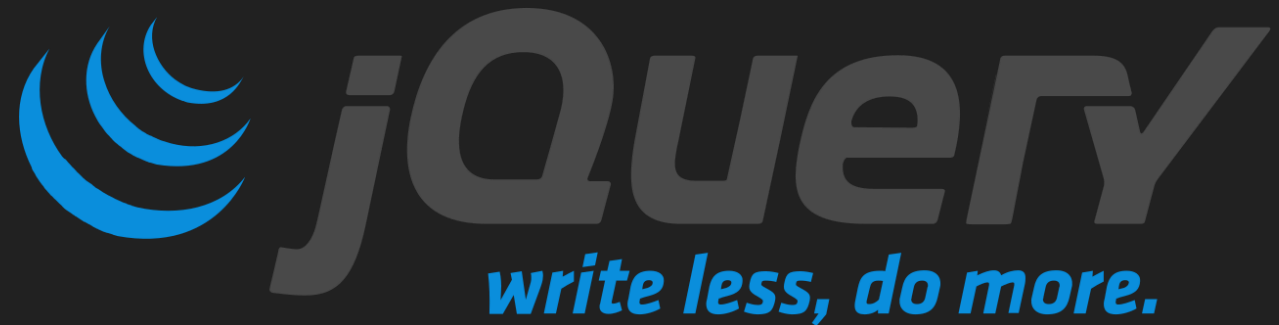
AVINASH SINGH

DEPARTMENT OF COMPUTER SCIENCE

UNIVERSITY OF PRETORIA

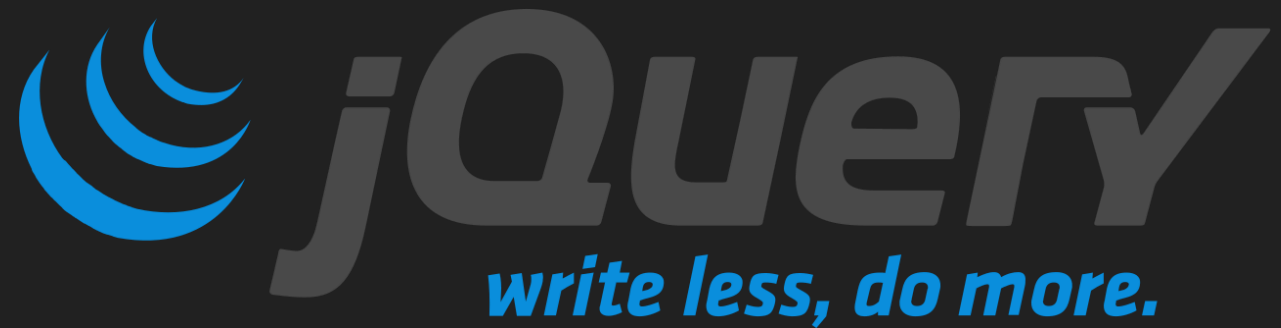
JQuery – Overview

- Collection of JavaScript libraries
- Simplify client-side scripting
- Add improved functions and syntax to JavaScript
- Makes your life as web developer a lot easier



JQuery – Overview

- Initial release in 2006
- Developed by John Resig (BarCamp)
- Used by 80% of the world's top 10,000 most visited websites
- Under the MIT license (free and open-source)



JQuery – Features

- HTML element selection and manipulation
- HTML event handling
- HTML DOM traversal and manipulation
- CSS manipulation
- Effects and animations
- Improved AJAX support
- Improved JSON parsing
- Extendible through plug-ins

JQuery – Features



DIFFERENT JQUERY
LIBRARIES/PROJECTS
EXIST



JQUERY : STANDARD
LIBRARY WITH JQUERY
SYNTAX AND
FUNCTIONS



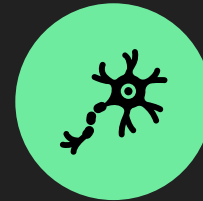
JQUERY UI: ADDS
INTERFACES,
IMPROVED WIDGETS,
ANIMATIONS AND
EFFECTS



JQUERY MOBILE:
IMPROVED JQUERY
FOR MOBILE DEVICES



SIZZLE: A JAVASCRIPT
CSS SELECTOR ENGINE



QUNIT: A JAVASCRIPT
UNIT TESTING
FRAMEWORK

JQuery – Minified

- Most JavaScript libraries have 2 versions
- Full
 - Contains all spacing and indentation
 - Easy to read/understand by human
 - Example: mylibrary.js
- Minified
 - All spacing/indentation removed
 - Library mostly on one single line
 - Small size and bandwidth efficient
 - Example : mylibrary-min.js

JQuery – Documentation

- Directly from JQuery (recommended)

<http://api.jquery.com>

- Or using W3School

<http://www.w3schools.com/JQuery>

JQuery – Include

- Link directly to the online library by using an absolute path:

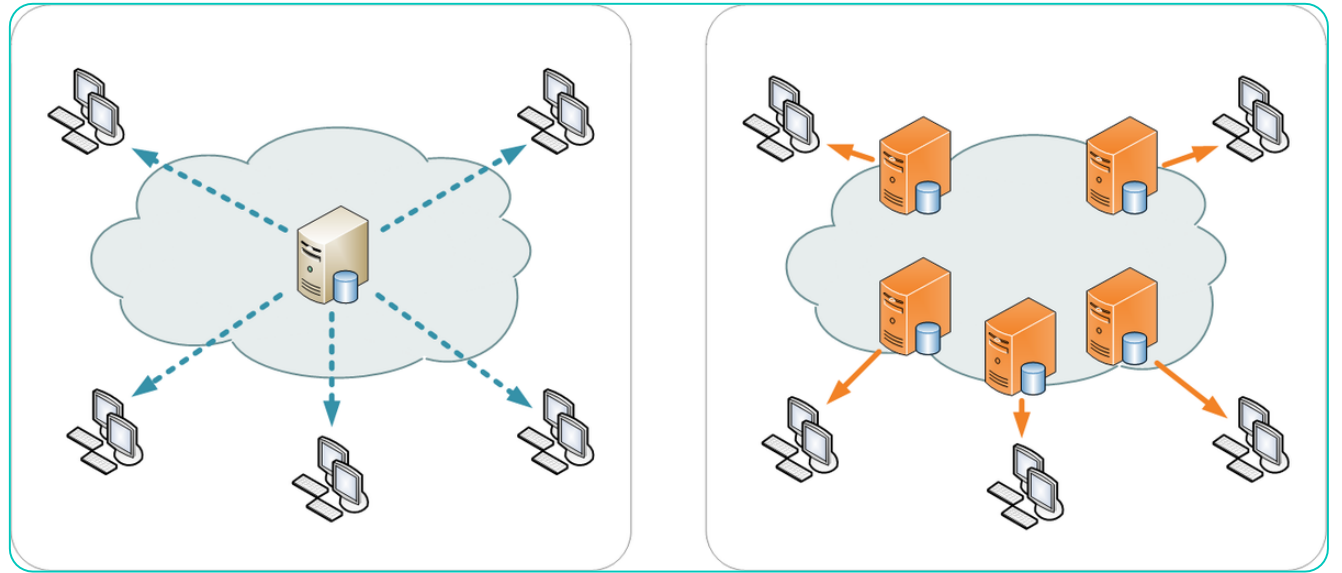
```
<script src="https://code.jquery.com/jquery-3.4.1.min.js"></script>
```

- Or download it (free) and link to the local JS file by using a relative path:

```
<script src="jquery-3.4.1.min.js"></script>
```


Content Delivery Network

- CDN are geographically distributed network of proxy servers
- Provide high availability and performance
- Late 1990s



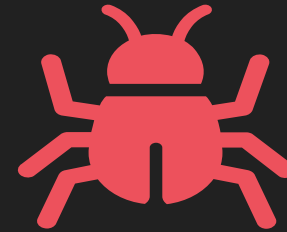
<https://www.cloudflare.com/learning/cdn/what-is-a-cdn/>

Content Delivery Network



Advantages

- Faster Loading
- Less Resources/Space
- Better Uptime
- Web Application Firewall (WAF)
- Load Balancing
- DDOS Prevention



Disadvantages

- Malware
- Versioning
- Customizability

JQuery – Object

- JQuery has a global object
- All JQuery functions and variables have to be accessed through this object
- Either use the full name: JQuery
- Or the more commonly used short-hand version: \$

JQuery – Selectors

- Select HTML elements using CSS selectors

```
$("p") // by tag name
$("#theid") // by id
$(".myclass") // by class name
$("p#paragraph1") // by tag and id
$("p.center") // by tag and class
$("*") // all elements
$("[src]") // with specific attribute
$("p:first") // first element by tag
```

JQuery – Selectors

- Note that JQuery selectors return JQuery objects, not pure JS objects
- Hence, you cannot call JS functions on JQuery objects
- Use the JQuery `get()` function or the subscript operator `[]` to get the pure JS object
- Should be avoided, since JQuery has its own version for all pure JS functions

JQuery – Selectors

```
// Valid
console.log(document.getElementById('main').innerHTML);
console.log($('#main').html());

// Invalid
console.log(document.getElementById('main').html());
console.log($('#main').innerHTML);

// Valid
console.log($('#main').get(0).innerHTML);
console.log($('#main')[0].innerHTML);
```

JQuery – Getters

- Get inner HTML (including HTML tags)

```
var html = $("#par").html();
```

- Get raw text (same as HTML with all HTML tags removed)

```
var text = $("#par").text();
```

JQuery – Getters

- Get an input value

```
var user = $("#user").val();
```

- Get an attribute

```
var href = $("#link").attr("href");
```


JQuery – Setters

- Set inner HTML

```
$("#par").html("<p>New HTML</p>");
```

- Set raw text

```
$("#par").text("New text");
```

JQuery – Setters

```
$("#user").val("Default user");
```

- Set an input value

- Set an attribute

```
$("#link").attr("href", "http://cs.up.ac.za");
```

JQuery – Inside Add

- Insert content at the end (still inside) the element

```
$("#par").append("Appended text.");
```

- Insert content at the beginning (still inside) the element

```
$("#par").prepend("Prepended text.");
```

JQuery – Outside Add

- Insert content after (outside) of the selected element

```
$("#par").after("Appended text.");
```

- Insert content before (outside) of the selected element

```
$("#par").before("Prepended text.");
```

JQuery – Delete

- Remove selected element and all its children

```
$("#par").remove();
```

- Clear an element. Removes all children and inner text

```
$("#par").empty();
```

JQuery – CSS

- Get a CSS attribute

```
$("#par").css("background-color");
```

- Set a CSS attribute

```
$("#par").css("background-color", "red");
```

JQuery – CSS

- Set a number of CSS attributes at the same time

```
$("#par").css({  
    "background" : "red",  
    "font-size" : "200%"  
});
```

JQuery – Classes

- Dynamically add a new CSS class:

```
$("div").addClass("highlight");
```

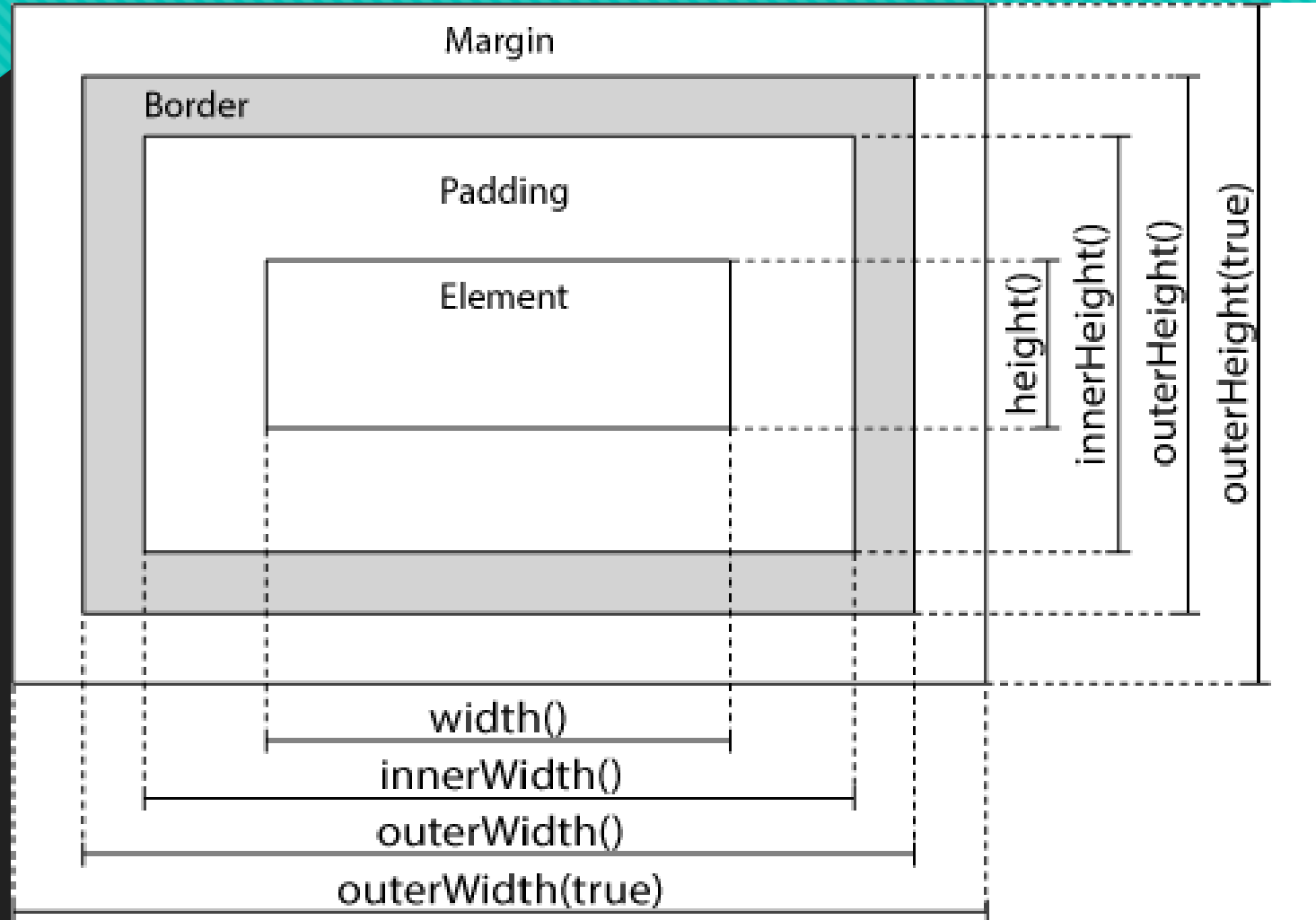
- Dynamically remove a new CSS class:

```
$("div").removeClass("highlight");
```

- Toggle between add/remove CSS classes:

```
$("div").toggleClass("highlight");
```


JQuery – Dimensions



JQuery – Delete

- Directly get/set the dimensions of a HTML element

```
$("#div").width(); // get  
$("#div").height(500); // set  
$("#div").innerWidth(); // get  
$("#div").innerHeight("200px"); // set  
$("#div").outerWidth(); // get  
$("#div").outerHeight("150%"); // set
```

JQuery – Traversing

- Get the parent element

```
$("span").parent();
```

- Get all ancestors of the element

```
$("span").parents();
```

JQuery – Traversing

- Get direct children

```
$("#div").children();  
$("#div").children("p.1");
```

- Get descendants

```
$("#div").find("*");  
$("#div").find("span");
```

JQuery – Traversing

- Get siblings of an element

```
$("#h2").siblings();  
$("#h2").siblings("p");
```

- Get the next/previous sibling of an element

```
$("#h2").next();  
$("#h2").prev();
```

JQuery – Visibility

- Hide/show HTML elements

```
$("#p").hide();  
$("#p").show();
```

- Hide/show elements with timeout (animation)

```
$("#p").hide(1000);
```

- Toggle visibility (hide if shown, show if hidden)

```
$("#p").toggle();
```

JQuery – Chained Calls

- References are returned by JQuery functions
- Functions can be chained

```
$("#div").width(400).height(500);
```

JQuery – Events

- Assign handlers to JavaScript events

```
$("#p").click(function(){  
    // action goes here!  
});
```

// or

```
$("#p").on('click', function(){  
    // action goes here!  
});
```


JQuery – Events

- Not all events have their own JQuery functions, if not, use the on() function and pass the event as a string parameter
- All normal JavaScript events can be used
 - click, dblclick
 - mouseenter, mouseleave
 - keypress, keydown, keyup
 - submit, change
 - resize, scroll
 - focus, blur
 - load, unload
 - Many more ...

```
$( "p" ).on( "click", function() {  
    alert( $( this ).text() );  
});
```

JQuery – Events

- JQuery has an additional ready event
- The ready event is fired the moment the DOM becomes safe to manipulate
- In contrast, the JS onload function only fires if everything was loaded, including non-DOM resources (images, videos, external files, etc)
- Always consider using the ready event in JQuery to execute code if the page is ready

```
$(document).ready(function(){  
    // action to execute if the page is ready  
});
```

JQuery – AJAX

- Simple AJAX using the load function

```
$(selector).load(URL, data, callback);
```

- Load data from a URL into a div

```
$("#div1").load("http://cs.up.ac.za/data.txt");
```

JQuery – AJAX

- Do manual AJAX request
- If no type is specified, GET is assumed
- If the returned Content-Type is JSON, JQuery will automatically parse the results for you, no need to call `JSON.parse(...)`
- Various events can be used (fired before/after request, on error, on success, etc)
- Separate GET and POST functions exists
 - Same as AJAX, just with some preset parameters

JQuery – AJAX

```
$.ajax({  
  url: "https://api.coinmarketcap.com/v1/ticker/",  
  type: "POST",  
  data: null, // Optional POST data/body  
  success: function(result){  
    console.log("Success");  
    console.log(result);  
  },  
  error: function(xhr, status, error){  
    console.log("Error: " + error);  
  },  
});
```

JQuery – AJAX GET

- Short-hand version of GET

```
$.get(  
    "https://api.coinmarketcap.com/v1/ticker/",  
    function(data, status){  
        alert("Status: " + status);  
        alert("Data: " + data);  
    }  
);
```

JQuery – AJAX POST

- Short-hand version of POST

```
$.post(  
    "https://api.coinmarketcap.com/v1/ticker/",  
    { type:"BTC", "amount" : 5},  
    function(data, status){  
        alert("Status: " + status);  
        alert("Data: " + data);  
    }  
);
```

JQuery – UI

- The JQuery UI library provides many advanced widgets and animations
 - Movement, fading, change animations, etc
 - Progress bars, tabs, drag-and-drop, spinners, popup dialogs, etc
- Most of the features are now available in HTML5 and CSS3
- Bootstrap (discussed later in the semester) also provides “better” widgets
- JQuery UI will therefore not be discussed

