

GCS 3205 – Mobile Web Application

Department of Computer Science
Gulu University

Daniel Ogenrwot

d.ogenrwot@gu.ac.ug

<https://www.gu.ac.ug>

October 20, 2020



Lecture-3 JavaScript & JQuery

Outline

- ☐ JavaScript
- ☐ JQuery



JavaScript - Overview

- ❑ Originally called LiveScript - in 1995
- ❑ Originally made by NetScape to provide interaction and more functionality in the browser
- ❑ Renamed to JavaScript December 1995
- ❑ JavaScript compared to Java - only in name similar.

Versions ECMAScript

ECMAScript is the standardized format of JavaScript. You will see JavaScript referred to as different version of ecma script or ES.

ES Version Official name

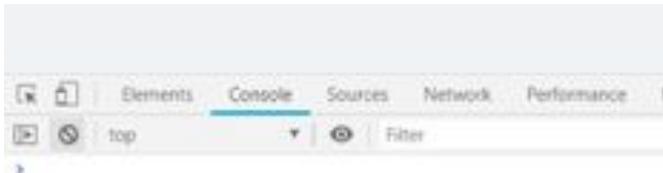
ECMAScript 1 (1997), ECMAScript 2 (1998), ECMAScript 3 (1999),
ECMAScript 4 Never released. ECMAScript 5 (2009), ECMAScript 5.1 (2011),
ECMAScript 2015 *** this is where it gets confusing, ECMAScript 2016, ECMAScript 2017,
ECMAScript 2018, ECMAScript 2019, ECMAScript 2020, ES.NEXT



JavaScript - Chrome Browser Console

The developer console shows you information about the currently loaded Web page, and also includes a command line that you can use to execute JavaScript expressions in the current page. Open try it.

- Will be used throughout the course.
- Test debug - output content from our code
- Most browsers - you can write and execute javascript from your browser



JavaScript - Chrome Browser Console

Outputs a message to the Web Console. Try it:

```
console.log("Hello");
```

console.log prints the element in an HTML-like tree

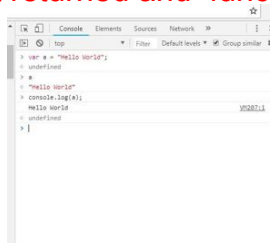
TIP : When you reload it goes away

Try it: `console.dir(document);`

*more about the document Object (DOM) later

`console.dir` prints the element in a JSON-like tree

TIP : Undefined means nothing got returned and function expect a return within the console.



JavaScript - Chrome Browser Console

Inside html file or linked to script file

```
<script src="code.js"></script>
<script>
///
</script>
```

TIP : Alert stops the code execution, if in top does not output the content until the button is clicked. Place JavaScript at the bottom so the rest of the code can render.

```
<!-- HTML4 -->
<script type="text/javascript" src="javascript.js"></script>

<!-- HTML5 -->
<script src="javascript.js"></script>
```



JavaScript - Basics

- ❑ Data Types
- ❑ Declare variable & scope - var, let, const
- ❑ Comments
- ❑ Operators, spread operators, ternary operator
- ❑ Conditionals
- ❑ Functions
- ❑ Loops
- ❑ Alert, Prompt
- ❑ Template literal
- ❑ Immediately Invoked Function Expression (IIFE)
- ❑ Arrow format
- ❑ Object destructuring
- ❑ etc...



JavaScript - Document Object

The Document interface represents any web page loaded in the browser and serves as an entry point into the web page's content, which is the DOM tree. The DOM tree includes elements such as <body> and <table>, among many others. It provides functionality globally to the document

<https://developer.mozilla.org/en-US/docs/Web/API/Document>

1. Select the document body - update page contents
2. View the document and body in the console using dir and log.

```
<!doctype html>
<html>
<head>
  <title>JavaScript Page</title>
</head>
<body>
  <script>
    document.body.innerHTML = "New webPage
    content"; console.log(document.body);
    console.dir(document.body);
    console.dir(document);
  </script>
</body>
</html>
```



JavaScript - Selecting by TAG

The `Element.getElementsByTagName()` method returns a **live HTMLCollection of elements with the given tag name**. All descendants of the specified element are searched, but not the element itself. The returned list is **live**, which means it updates itself with the DOM tree automatically. Therefore, there is no need to call `Element.getElementsByTagName()` with the same element and arguments repeatedly if the DOM changes in between calls.

1. Select the element in different ways
2. Update the `textContent`

```
const e11 =  
document.querySelector('h1');  
e11.textContent = 'First H1 on  
Page';  
console.log(e11);  
const e12 = document.getElementsByTagName('h1'); //  
return HTMLcollection  e12.textContent = 'First H1 by  
Tag';  
console.log(e12);  
e12[0].textContent = 'FirstH1 by Tag';  
console.log(e12[0]);
```

`getElementById()`



What is JQuery ?

JQuery is a lightweight, "write less, do more", JavaScript library. An open source JavaScript library that simplifies the interaction between HTML and JavaScript. JQuery is easy to learn and master.

What you should already know?

- ☐ HTML
- ☐ CSS
- ☐ JavaScript



JQuery - Installation

- ❑ Steps for adding JQuery to your web pages .
- ❑ Go to <http://docs.jquery.com/> OR
- ❑ Go to <https://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js>
- ❑ Use following code snippet

```
< html >  
    < head >  
        < script src="Jquery.js">  
    </head>  
    < body>  
    </body>  
</html>
```

OR

```
<script scr  
="http://ajax.googleapis.com/ajax/libs/jquery/1.8.0/jqueryy  
.min.js"></script>
```



JQuery - Syntax

- ❑ With jQuery you select (query) HTML elements and perform "actions" on them.
- ❑ Basic syntax is: ***\$(selector).action()***
- ❑ A \$ sign to define/access jQuery
- ❑ A (selector) to "query (or find)" HTML elements
- ❑ A jQuery *action()* to be performed on the element(s)

- ❑ Examples:
- ❑ `$("p").hide()` - hides all <p> elements.
- ❑ `$("#test").hide()` - hides the element with id="test".
- ❑ `$(".test").hide()` - hides all elements with class="test".



JQuery - The Document Ready Event

The Document Ready Event

```
$(document).ready(function(){  
    // jQuery methods go here...  
});
```

This is to prevent any jQuery code from running before the document is finished loading (is ready).

- ❑ Trying to hide an element that is not created yet.
- ❑ Trying to get the size of an image that is not loaded yet

Alternate Syntax

```
$(function(){  
    // jQuery methods go here...  
});
```

Use the syntax you prefer



JQuery - Selectors

- ❑ jQuery selectors allow you to select and manipulate HTML elements
- ❑ With jQuery selectors you can find elements based on their id, classes, types, attributes, values of attributes and much more. It's based on the existing [CSS Selectors](#) and in addition, it has some own custom selectors.
- ❑ All type of selectors in jQuery, start with the dollar sign and parentheses: \$().
- ❑ **Types of jquery selectors:**
 - ❑ Element selector
 - ❑ Id (#) selector
 - ❑ Class (.) selector



JQuery - Selectors

Element Selector

The jQuery element selector selects elements based on their tag names.

```
$(document).ready(function() {  
    $("#button").click(function() {  
        $("p").hide();  
    });  
});
```

Id (#) Selector

The jQuery #id selector uses the id attribute of an HTML tag to find the specific element.

```
$(document).ready(function() {  
    $("#button").click(function() {  
        $("#test").hide();  
    });  
});
```



JQuery - Selectors

Class (.) Selector

The jQuery class selector finds elements with a specific class.

```
$(document).ready(function() {  
    $("#button").click(function() {  
        $(".test").hide();  
    });  
});
```

Syntax	Description
\$("#")	Selects all elements
\$(this)	Selects the current HTML element
\$("#p.intro")	Selects all <p> elements with class="intro"
\$("#p:first")	Selects the first <p> element
\$("#ul li:first")	Selects the first element of the first
\$("#ul li:first-child")	Selects the first element of every
\$("#[href]")	Selects all elements with an href attribute
\$("#a[target='_blank']")	Selects all <a> elements with a target attribute value equal to "_blank"
\$("#tr:even")	Selects all even <tr> elements
\$("#tr:odd")	Selects all odd <tr> elements



jQuery - Effects

- ❑ jQuery hide() , show(), toggle(), fadeIn(), fadeOut(), fadeTo(), fadeToggle(), slideDown(), slideUp(), slideToggle(),...

- ❑ Syntax:

```
$(selector).hide(speed, callback);  
$(selector).show(speed, callback);  
$(selector).toggle(speed, callback);  
$(selector).fadeIn(speed, callback);  
$(selector).fadeOut(speed, callback);  
$(selector).fadeTo(speed, callback);  
$(selector).fadeToggle(speed, callback);  
$(selector).slideDown(speed, callback);  
$(selector).slideUp(speed, callback);  
$(selector).slideToggle(speed, callback);
```

- ❑ Speed and callback are optional parameters.



JQuery - Get Content & Attributes

jQuery contains powerful methods for changing and manipulating HTML elements and attributes.

Three simple, but useful, jQuery methods for DOM manipulation is:

`text()` - Sets or returns the text content of selected elements

`html()` - Sets or returns the content of selected elements (including HTML markup)

`val()` - Sets or returns the value of form fields

Syntax

`$(selector).text();` Get the content of selector

`$(selector).html();` Get the content with html markup of selector

`$(selector).val();` Get value of form elements

`$(selector).attr(attribute name);` Get value elements attribute



JQuery - Add & Remove Elements

With jQuery, it is easy to add or remove new elements/content.

Methods:

`append()` - Inserts content at the end of the selected elements

`prepend()` - Inserts content at the beginning of the selected elements

`after()` - Inserts content after the selected elements

`before()` - Inserts content before the selected elements

`remove()` - Removes the selected element (and its child elements)

`empty()` - Removes the child elements from the selected element

Syntax .

`$(selector).remove();`

`$(selector).empty(string);` string optional parameter.



JQuery - Get and Set CSS classes

With jQuery, it is easy to manipulate the CSS of elements.
Methods:

`addClass()` - Adds one or more classes to the selected elements
`removeClass()` - Removes one or more classes from the selected elements
`toggleClass()` - Toggles between adding/removing classes

The **css()** method sets or returns one or more style properties for the selected elements.

To return the value of a specified CSS property, use the following syntax:

```
$(selector).css("propertyname", "value");
```

To set a CSS Property , use the following syntax:

```
$(selector).css("propertyname", "value");
```



JQuery - AJAX

AJAX is the art of exchanging data with a server, and updating parts of a web page - without reloading the whole page.

AJAX = Asynchronous JavaScript and XML.

In short; AJAX is about loading data in the background and display it on the webpage, without reloading the whole page.

Examples of applications using AJAX: Gmail, Google Maps, Youtube, and Facebook tabs, etc...

jQuery provides several methods for AJAX functionality.

With the jQuery AJAX methods, you can request text, HTML, XML, or JSON from a remote server using both HTTP Get and HTTP Post - And you can load the external data directly into the selected HTML elements of your web page!



JQuery - AJAX Methods

load

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

```
$("#button").click(function(){
    $("#div1").load("demo_test.txt", function(responseTxt, statusTxt, xhr){
        if(statusTxt == "success")
            alert("External content loaded successfully!");
        if(statusTxt == "error")
            alert("Error: " + xhr.status + ": " + xhr.statusText);
    });
});
```

\$.get() Method

\$.post() Method

\$.ajax() Method - more flexible



THANK YOU