Web Technologies II

Introducing JQuery – Part A

jQuery is a JavaScript library that lets you jump-start your programming by handling many of the messy details of JavaScript programming for you. jQuery—whose motto is "write less, do more" makes programming fun, fast, and rewarding. With jQuery, you can achieve in a single line of code what could take dozens (if not more) lines of pure JavaScript programming.

About JavaScript Libraries

Many JavaScript programs have to deal with the same set of web page tasks again and again: selecting an element, adding new content, hiding and showing content, modifying a tag's attributes, determining the value of form fields, and making programs react to different user interactions. The details of these basic actions can be quite complicated—especially if you want the program to work in all major browsers. Fortunately, JavaScript libraries offer a faster way to do these time consuming tasks.

Think of it (JQuery) as a collection of prewritten JavaScript functions that you add to your web page. These functions make it easy to complete common tasks. Often, you can replace many lines of your own JavaScript programming (and the hours required to test them) with a single function from a JavaScript library. There are lots of JavaScript libraries out there, and many of them help create major websites like Yahoo!, Amazon, CNN, Apple, and Twitter

In this course we uses the jQuery (a popular JavaScript library) - www.jquery.com But there are other JavaScript libraries and frameworks. Some of them are ,

Libraries	Frameworks
jQuery	Bootstrap
Underscore and Lodash	Angular and AngularJS
D3.js	Ember.js
React	Aurelia
Glimmer	Vue.js

jQuery has many advantages:

Relatively small file size.

A compressed version of the library is only around 96k for version 1.11 and 83k for version 2.1.

Friendly to web designers.

jQuery doesn't assume you're a computer scientist. It takes advantage of CSS knowledge that most web designers already have.

It's tried and true.

jQuery is used on millions of sites, including many popular, highly trafficked websites like Pinterest, MSN.com, Amazon, Microsoft.com,Craigslist, and ESPN

It's free.

Large developer community.

Plug-ins, plug-ins, plug-ins.

jQuery lets other programmers create plug-ins —add-on JavaScript programs that work in conjunction with jQuery to make certain tasks, effects, or features incredibly easy to add to a web page

Getting jQuery

jQuery is simply a bunch of JavaScript code in an external JavaScript file. Like any external JavaScript file , you need to link it to your web page. However, because jQuery is so popular, you havea few choices when it comes to adding it to a web page: *You can either use a version hosted at Google, Microsofit, or jQuery.com or you can download the jQuery fiile to your own computer and add it to your website.*

CDN (Content Distribution Network)

CDN is, another website hosts the jQuery file and sends it out to anyone who requests it. There are a couple of benefits to this approach: First, you can save your web server a few milliseconds by letting Google, Microsoft, or jQuery handle distributing the file to your site's visitors. In addition, CDNs have the added benefit of having servers located around the globe

Why is it an advantage ? Discuss

Linking to the jQuery File on a CDN Server

Microsoft, jQuery, and Google all let you include the jQuery file on one of your web pages using their servers. For example, to link to version 1.11.0 of jQuery using Microsoft's CDN, you would add this line of code in the <head> of your web page (just before the closing </head> tag), like this:

```
<head>
<script src="https://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.3.1.min.js"></script>
</head>
```

Google CDN

```
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
</head>
```

You only need to use one of these lines on your page, based on the CDN you prefer to use. The Google CDN seems to be the most popular, so if you're unsure of which to use, use the Google Servers

Task : Research on how to use the downloaded jQuery file

Modifiying Web Pages: An Overview

JavaScript gives you the power to change a web page before your very eyes. Using JavaScript, you can add pictures and text, remove content, or change the appearance of an element on a page instantly. In fact, dynamically changing a web page is the hallmark of the all JavaScript-powered websites. For example, Google Maps (http://maps.google.com) provides access to a map of the world; when you zoom into the map or scroll across it, the page gets updated without the need to load a new web page. Similarly, when you mouse over a movie title at Netflix (www.netflix.com), an information bubble appears on top of the page providing more detail about the movie. In both of these examples, JavaScript is changing the HTML that the web browser originally downloaded.

Basically we should follow two steps to achieve this

1. ***Select an element on a page.***

An element is any existing tag, and before you can do anything with that element, you need to select it using JavaScript.

2. Do something with the element.

OK, “do something” isn’t a very specific instruction. That’s because there’s nearly an endless number of things you can do with an element to alter the way your web page looks or acts .

Here are some examples

Change a property of the element.

When animating a across a page, for example, you change that element’s position on the page.

Add new content.

If, while filling out a web form, a visitor incorrectly fills out a field, it’s common to make an error message appear—“Please supply an email address,” for example. In this case, you’re adding content some where in relation to that form field.

Remove the element.

For example when you are in Amazon.ca the pop-up box will appear when your mouse on certain link. In this case, JavaScript is adding or removing that box from the page.

Extract information from the element. Other times, you'll want to know something about the tag you've selected. For example, to validate a text field, you need to select that text field, then find out what text was typed into that field—in other words, you need to get the value of that field.

Add/remove a class attribute. Sometimes you'll want an element on a page to change appearance: the text in a paragraph to turn blue, or the background color of a text field to turn red to indicate an error. While JavaScript can make these visual changes, often the easiest way is to simply apply a class and let a web browser make those visual changes based on a CSS style from a style sheet. To change the text of a paragraph to blue, for example, you can simply create a class style with blue text color, and use JavaScript to apply the class to the paragraph dynamically.

Selecting a page element is the first step. To understand how to identify and modify a part of a page using JavaScript, you first need to get to know the **Document Object Model.**

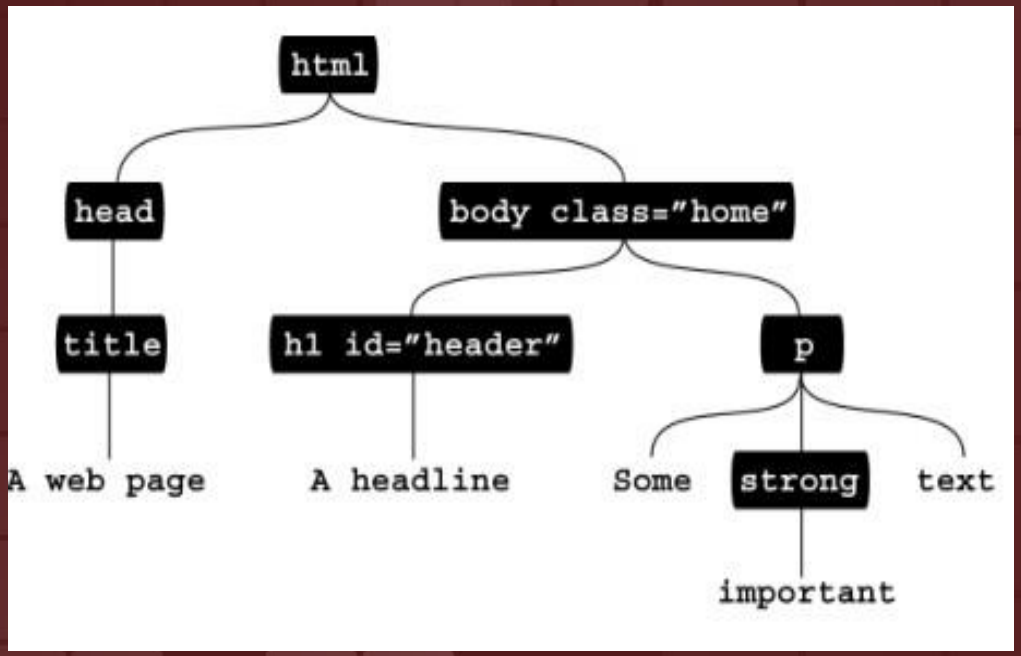
Understanding the Document Object Model

When a web browser loads an HTML file, it displays the contents of that file on the screen (appropriately styled with CSS, of course). But that's not all the web browser does with the tags, attributes, and contents of the file: It also creates and memorizes a "model" of that page's HTML. In other words, the web browser remembers the HTML tags, their attributes, and the order in which they appear in the file—this representation of the page is called the Document Object Model , or DOM for short.

To see how the DOM works, take a look at this very simple web page:

```
<!DOCTYPE HTML>
<html>
<head>
  <meta charset="UTF-8">
  <title>A web page</title>
</head>
<body class="home">
  <h1 id="header">A headline</h1>
  <p>Some <strong>important</strong> text</p>
</body>
</html>
```

On this and all other websites, some tags wrap around other tags—like the <html> tag, which surrounds all other tags, or the <body> tag, which wraps around the tags and contents that appear in the browser window. You can represent the relationship between tags with a kind of family tree. The <html> tag is the “root” of the tree—like the great-great-great grand daddy of all of the other tags on the page—while other tags represent different “branches” of the family tree; for example, the <head> and <body> tags, which each contain their own set of tags.



JavaScript provides several ways to select elements on a page so you can do something to them—like make them fade out of view or move across the page. The document.getElementById() method lets you select an element with a particular ID applied to its HTML. So if you have a tag with the ID banner applied to it— —you could select that div like this:

document.getElementById('banner').

Likewise, the document.getElementsByTagName() method selects every instance of a particular tag—
document.getElementsByTagName('a')

More recent browser versions offer a way to select DOM elements based on CSS selectors. For example, the
document.getElementsByClassName( )
retrieves all elements that share a particular class name. To select all elements with a class of author, you could use this code:
document.getElementsByClassName('author');

Basic Selectors

ID SELECTORS

You can select any page element that has an ID applied to it using jQuery and a CSS ID selector. For example, say you have the following HTML in a web page:

```
<p id="message">Special message</p>
```

To select that element using jQuery looks like this:

```
var messagePara = $('#message');
```

ELEMENT SELECTORS

jQuery also has its own replacement for the getElementsByTagname () method.

```
var linksList = document.getElementsByTagName('a');
```

Just pass the tag's name to jQuery.

```
var linksList = $('a');
```

CLASS SELECTORS

jQuery provides an easy method to select all elements with thesame class name. Just use a CSS class selector like this:

```
$('.submenu')
```

Again, notice that you write the CSS class selector just like, well, a CSS class selector, with the period before the class name. Once you select those tags, you can manipulate them using jQuery. For example, to hide all tags with the class name of .submenu, you'd write this:

```
$('.submenu').hide();
```

Questions ?