



มหาวิทยาลัยขอนแก่น

วิทยา ชริยา มัญญา



KHON KAEN UNIVERSITY

jQuery

Assoc. Prof. Dr. Kanda Runapongsa Saikaew
(krunapon@kku.ac.th)

Department of Computer Engineering
Khon Kaen University



คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น
FACULTY OF ENGINEERING KHON KAEN UNIVERSITY

Agenda

- What and Why jQuery?
- jQuery Syntax
- jQuery Effects
- jQuery HTML
- jQuery AJAX
- jQuery Examples
- jQuery References



What is jQuery?

- jQuery is a lightweight, "write less, do more", JavaScript library.
- The purpose of jQuery is to make it much easier to use JavaScript on your website.
- jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code



jQuery Library

- The jQuery library contains the following features:
 - HTML/DOM manipulation
 - CSS manipulation
- HTML event methods
- Effects and animations
- AJAX
- Utilities



jQuery Install

- Adding jQuery to Your Web Pages
 - There are several ways to start using jQuery on your web site. You can:
 - Download the jQuery library from [jQuery.com](http://jquery.com)
 - Include jQuery from a CDN, like Google
- Downloading jQuery
 - There are two versions of jQuery available for downloading from <http://jquery.com/download/>
 - Production version - this is for your live website because it has been minified and compressed
 - Development version - this is for testing and development (uncompressed and readable code)



jQuery Library

- The jQuery library is a single JavaScript file, and you reference it with the HTML <script> tag (notice that the <script> tag should be inside the <head> section):

```
<head>  
<script src=" jquery-3.3.1.min.js "></script>  
</head>
```



Alternatives to Downloading

- If you don't want to download and host jQuery yourself, you can include it from a CDN (Content Delivery Network).
- Both Google and Microsoft host jQuery.
- To use jQuery from Google or Microsoft, use one of the following:

– Google CDN:

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



Get the Latest Available

- If you look at the Google URL above - the version of jQuery is specified in the URL (3.3.1)
- If you would like to use the latest version of jQuery
 - You can either remove a number from the end of the version string (for example 3.3), then Google will return the latest version available in the 3.3 series (3.3.0, 3.3.1, etc.)
 - Or you can take it up to the whole number (1), and Google will return the latest version available in the 3 series (from 3.1.0 to 3.3.1)



Advantages of Using the hosted jQuery from Google or Microsoft

- Many users already have downloaded jQuery from Google or Microsoft when visiting another site
- As a result, it will be loaded from cache when they visit your site, which leads to faster loading time
- Also, most CDN's will make sure that once a user requests a file from it, it will be served from the server closest to them, which also leads to faster loading time.



jQuery Syntax

- The jQuery syntax is tailor made for **selecting** HTML elements and performing some **action** on the element(s).
- 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)



jQuery Syntax Examples

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



The Document Ready Event

- You might have noticed that all jQuery methods in our examples, are inside a 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).



# Why Do We Prefer Document Ready

- It is good practice to wait for the document to be fully loaded and ready before working with it
- This also allows you to have your JavaScript code before the body of your document, in the head section.
- Here are some examples of actions that can fail if methods are run before the document is fully loaded:
  - Trying to hide an element that is not created yet
  - Trying to get the size of an image that is not loaded yet



# Shorter Syntax

- **Tip:** The jQuery team has also created an even shorter method for the document ready event:
- `$(function(){`

*//jQuery methods go here...*

`});`

- Use the syntax you prefer. We think that the document ready event is easier to understand when reading the code.



# jQuery Selectors

- jQuery selectors allow you to select and manipulate HTML element(s)
- jQuery selectors are used to "find" (or select) HTML 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 selectors in jQuery start with the dollar sign and parentheses: `$( )`.



# The Element Selector

- The jQuery element selector selects elements based on the element name.
- You can select all `<p>` elements on a page like this: `$( "p" )`
- **Example**

When a user clicks on a button, all `<p>` elements will be hidden:

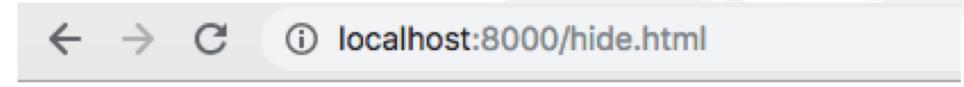
- Code Example

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



# Hiding Elements Based on Element Names

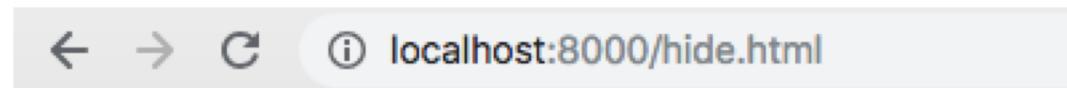
```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/j
query.min.js"></script>
<script>
$(document).ready(function(){
 $("button").click(function(){
 $("p").hide(1000);
 });
});
</script>
</head>
<body>
<button>Hide</button>
<p>This is a paragraph with little content.</p>
<p>This is another small paragraph.</p>
</body>
</html>
```



Hide

This is a paragraph with little content.

This is another small paragraph.



Hide



# Hiding Elements Based on Element IDs and Element Class

```
<script>
$(document).ready(function(){
 $("button").click(function(){
 $("#test").hide();
 });
});
</script>
</head>
<body>
<h2>This is a heading</h2>
<p>This is a paragraph.</p>
<p id="test">This is another paragraph with
id = test</p>
<button>Click me to hide the paragraph
with id = test</button>
```

```
<script>
$(document).ready(function(){
 $("button").click(function(){
 $(".test").hide();
 });
});
</script>
</head>
<body>
<h2 class="test">This is a heading with class =
test</h2>
<p class="test">This is a paragraph with class
= test</p>
<p>This is another paragraph.</p>
<button>Click me</button>
```



# What are Events?

- All the different visitor's actions that a web page can respond to are called events
- An event represents the precise moment when something happens
- Example: clicking on an element
- The term “fires” is often used with events
  - The keypress event fires the moment you press a key



# Some Common DOM Events

- Mouse Events
  - click, dblclick, mouseenter, mouseleave
- Keyboard Events
  - keypress, keydown, keyup
- Form Events
  - submit, change, focus, blur
- Document/Window Events
  - load, resize, scroll, unload

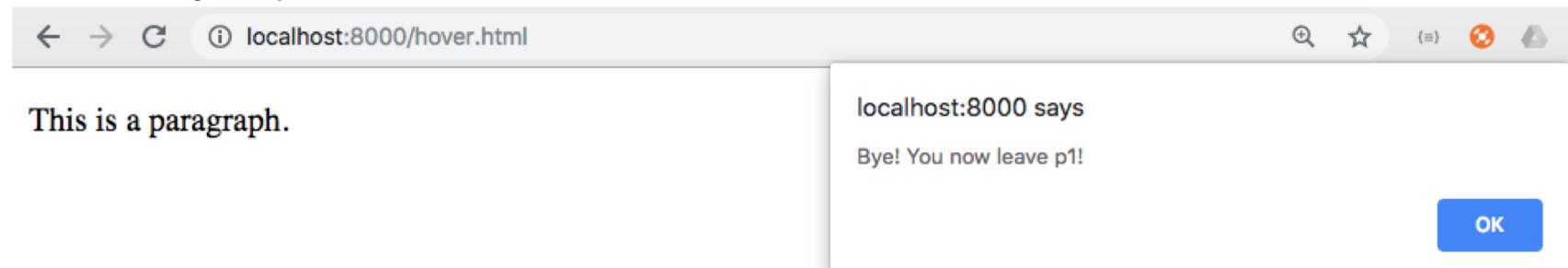
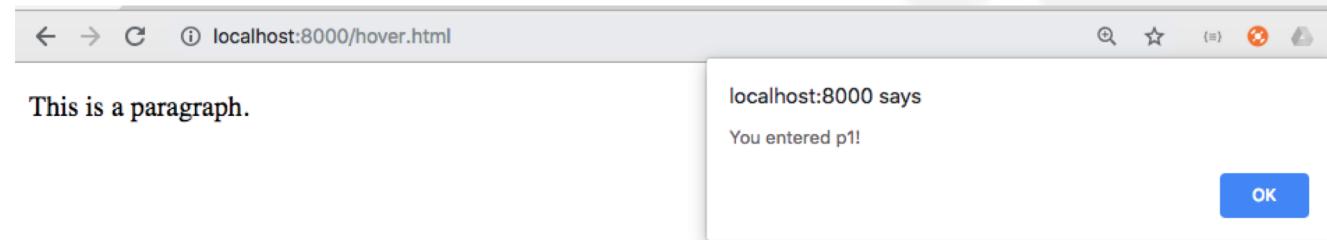


# Hover

The hover() method takes two functions and is a combination of the mouseenter() and mouseleave() methods.

- Sample script

```
<script>
$(document).ready(function(){
 $("#p1").hover(function(){
 alert("You entered p1!");
 },
 function(){
 alert("Bye! You now leave p1!");
 });
});
...
<p id="p1">This is a paragraph.</p>
```



# Agenda

- What and Why jQuery?
- jQuery Syntax
- **jQuery Effects**
- jQuery HTML
- jQuery AJAX
- jQuery Examples
- jQuery References



# jQuery Fade

- With jQuery you can fade an element in and out of visibility.
- jQuery has the following fade methods:
  - fadeIn() : Fade in a hidden element
  - fadeOut() : Fade out a visible element
  - fadeToggle() : Toggles between the fadeIn() and fadeOut() methods
  - fadeTo() : Allows fading to a given opacity (value between 0 and 1)



# jQuery Fade In

```
<script>
$(document).ready(function(){
 $("button").click(function(){
 $("#div1").fadeIn();
 $("#div2").fadeIn("slow");
 $("#div3").fadeIn(3000);
 });
});
</script>
```

...

```
<div id="div1"
style="width:80px;height:80px;display:none;background-color:red;"></div>

<div id="div2"
style="width:80px;height:80px;display:none;background-color:green;"></div>

<div id="div3"
style="width:80px;height:80px;display:none;background-color:blue;"></div>
```

← → ⌂ ⓘ localhost:8000/fadein.html

Demonstrate fadeIn() with different parameters.

Click to fade in boxes

← → ⌂ ⓘ localhost:8000/fadein.html

Demonstrate fadeIn() with different parameters.

Click to fade in boxes



# jQuery fadeIn() Method

- The jQuery fadeIn() method is used to fade in a hidden element
- Syntax
  - `$(selector).fadeIn(speed,callback);`
- Speed: “slow”, “fast”, or milliseconds
- Callback: A function to be executed after the fading completes



# jQuery Slide

- The jQuery slide methods slide elements up and down
- Syntax: `$(selector).slide<A>(speed,callback);`  
A: Down, Up, Toggle
  - slideDown: used to slide down an element
  - slideUp: used to slide up an element
  - slideToggle: toggle between the slideDown() and slideUp() methods



# jQuery Slide

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3
.2.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
 $("#flip").click(function(){
 $("#panel").slideDown("slow");
 });
});
</script>

<style>
#panel, #flip {
 padding: 5px;
 text-align: center;
 background-color: #e5eecc;
 border: solid 1px #c3c3c3;
 #panel {
 padding: 30px;
 background-color: #d9d9d9;
 display: none;
 }
}
</style>
</head>
<body>

<div id="flip">Click to slide down panel</div>
<div id="panel">Hello world!</div>

</body>
</html>
```



# jQuery Slide output



# jQuery Animate

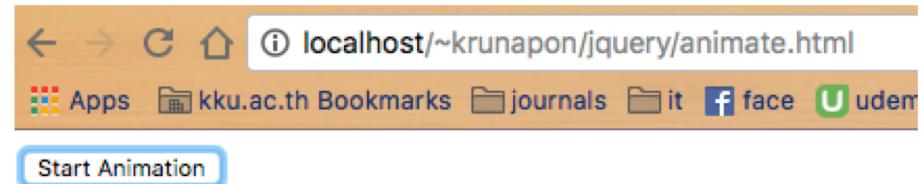
- The jQuery animate() method lets you create custom animations
- Syntax: `$(selector).animate({params}, speed, callback);`
  - The required params parameter defines the CSS properties to be animated



# Animate with Single Property

```
$(document).ready(function(){
 $("button").click(function(){
 $("div").animate({left:'250px'});
 });
});
</script>

...
<button>Start Animation</button>
<div
style="background:#98bf21;height:10
0px;width:100px;position:absolute;">
</div>
```



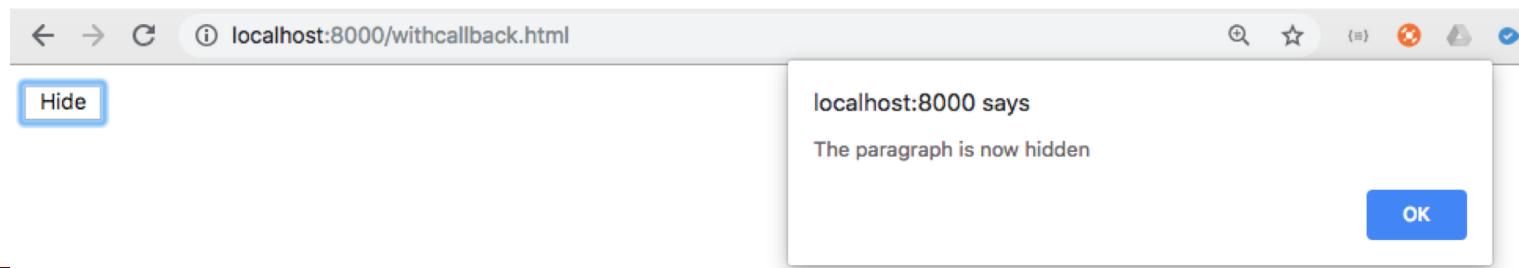
# jQuery Callback

- A callback function is executed after the current effect is 100% finished
- JavaScript statements are executed line by line. However, with effects, the next line of code can be run even though the effect is not finished. This can create errors.
- To prevent this, you can create a callback function.
- A callback function is executed after the current effect is finished.
- Typical syntax: **`$(selector).hide(speed,callback);`**



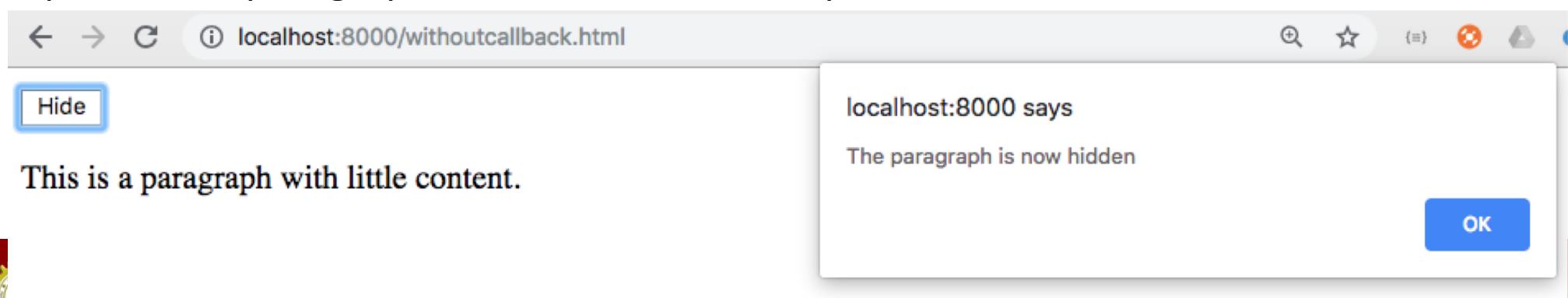
# Example with Callback

```
<script>
$(document).ready(function(){
 $("button").click(function(){
 $("p").hide("slow", function(){
 alert("The paragraph is now hidden");
 });
 });
});
</script>
</head>
<body>
<button>Hide</button>
<p>This is a paragraph with little content.</p>
```



# Example Without Callback

```
<script>
$(document).ready(function(){
 $("button").click(function(){
 $("p").hide(1000);
 alert("The paragraph is now hidden");
 });
});
</script>
</head>
<body>
<button>Hide</button>
<p>This is a paragraph with little content.</p>
```



# Agenda

- What and Why jQuery?
- jQuery Syntax
- jQuery Effects
- **jQuery HTML**
- jQuery AJAX
- jQuery Examples
- jQuery References



# jQuery Get Content and Attributes

- jQuery contains powerful methods for changing and manipulating HTML elements and attributes
- One very important part of jQuery is the possibility to manipulate the DOM
- jQuery comes with a bunch of DOM related methods that make it easy to access and manipulate elements and attributes



# Get Content – text(), html(), and val()

- Three simple, but useful, jQuery methods for DOM manipulation are:
  - `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 form fields



# Sample Code to Get Content

```
<script>
$(document).ready(function(){
 $("#btn1").click(function(){
 alert("Text: " + $("#test").text());
 });
 $("#btn2").click(function(){
 alert("HTML: " + $("#test").html());
 });
});
</script>
</head>
<body>
<p id="test">This is some bold text in a paragraph.</p>
<button id="btn1">Show Text</button>
<button id="btn2">Show HTML</button>
```

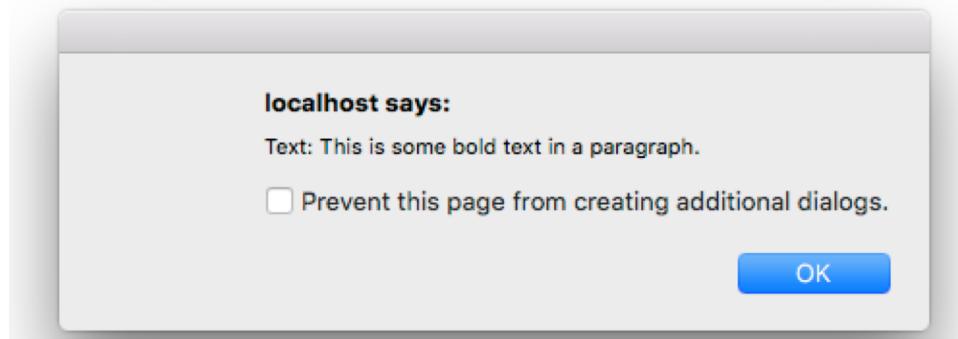


# Sample Code Get Content Result



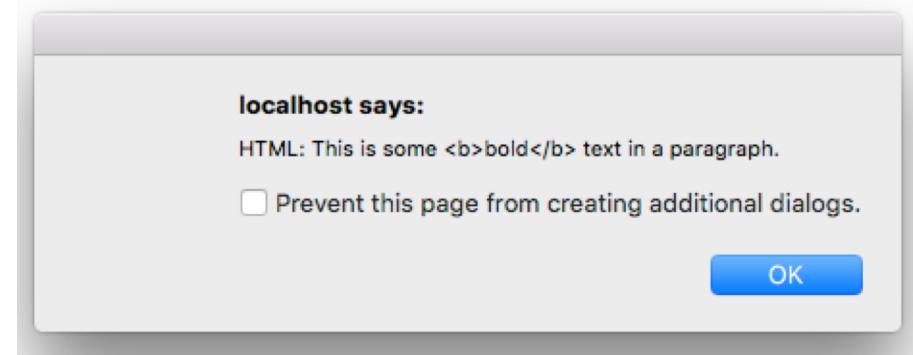
This is some **bold** text in a paragraph.

[Show Text](#) [Show HTML](#)



This is some **bold** text in a paragraph.

[Show Text](#) [Show HTML](#)



# jQuery – AJAX Introduction

- AJAX = Asynchronous JavaScript and XML
- AJAX is the art of exchanging data with a server, and updating parts of a web page – without reloading the whole page
- Examples of applications using AJAX
  - Gmail, Google Maps, Youtube, and Facebook



# jQuery and AJAX

- 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
- You can load the external data directly into the selected HTML elements of your web page



# jQuery load() Method

- The jQuery load() method is a simple, but powerful AJAX method
- The load() method loads data from a server and puts the returned data into the selected element



# jQuery load Syntax

- Syntax:

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

- The required URL parameter specifies the URL you wish to load
- The optional data parameter specifies a set of query string key/value pairs to send along with the request
- The optional callback parameter is the name of a function to be executed after the load() method is completed



# File Content and Code

- File “demo\_text.txt”

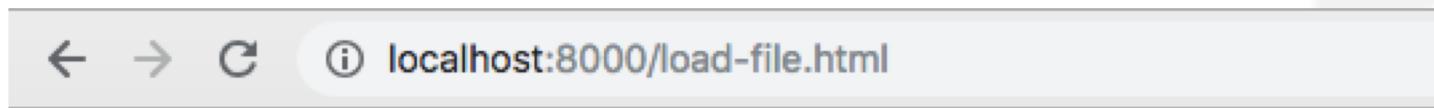
```
<h2>jQuery and AJAX is FUN!!!</h2>
<p id="p1">This is some text in a paragraph.</p>
```

- Code

```
<script>
$(document).ready(function(){
 $("button").click(function(){
 $("#div1").load("demo_text.txt");
 });
});
</script>
```



# jQuery and AJAX



## Let jQuery AJAX Change This Text

[Get External Content](#)

A screenshot of a web browser window. The address bar at the top shows the URL "localhost:8000/load-file.html". Below the address bar, there is a large, empty white area representing the content of the page.

## jQuery and AJAX is FUN!!!

This is some text in a paragraph.

[Get External Content](#)

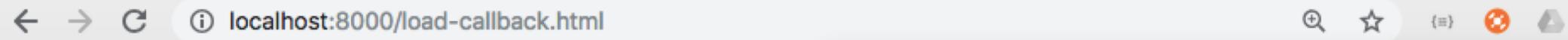
# The load Optional Callback Parameters

- The optional callback parameter specifies a callback function to run when the load() method is completed
- The callback function can have different parameters
  - responseTxt: contains the resulting content if the call succeeds
  - statusTxt: contains the status of the call
  - xhr – contains the XMLHttpRequest object



# Sample Callback load Function

```
<script>
$(document).ready(function(){
 $("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);
 });
 });
});
</script>
```



Let jQuery AJAX Change This Text

Get External Content

localhost:8000 says  
External content loaded successfully!

OK

# jQuery – AJAX get() and post() Methods

- The jQuery get() and post() are used to request data from the server with an HTTP GET or POST request
- GET – Requests data from a specified resource
- POST – Submits data to be processed to a specified resource
- GET is basically used for just getting (retrieving) some data from the server
- POST can also be used to get some data from the server but it never cache data



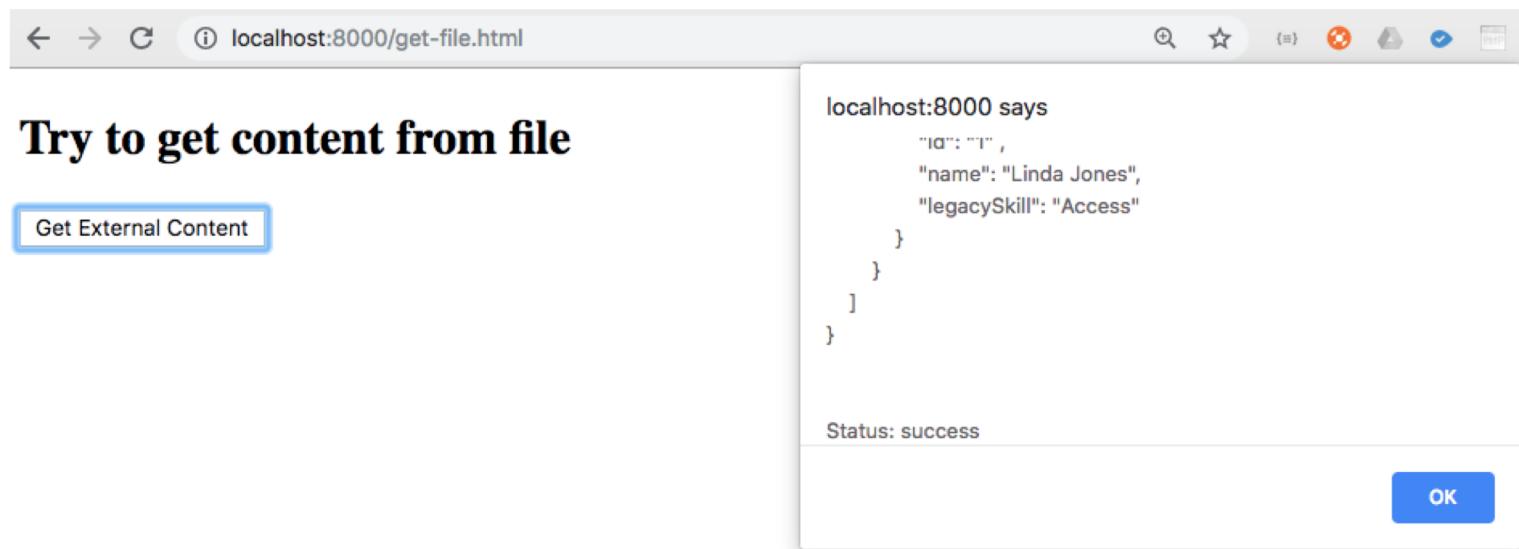
# jQuery \$.get() Method

- Syntax: `$.get(URL, callback)`
- The required URL parameter specifies the URL you wish to request
- The optional callback parameter is the name of a function to be executed if the request succeeds



# Sample `$.get()` Code

```
<script>
$(document).ready(function(){
 $("button").click(function(){
 $.get("json/json_web.txt", function(data, status){
 alert("Data: " + data + "\nStatus: " + status);
 });
 });
});
</script>
```



# Sample: load1.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>The jQuery Example</title>
5 <script type = "text/javascript"
6 src =
7 "https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js">
8 </script>
9 <script>
10 $(document).ready(function() {
11 $("#driver").click(function(event){
12 $('#stage').load('result.html');
13 });
14 });
15 </script>
16 </head>
17 <body>
18 <p>Click on the button to load result.html file</p>
19 <div id="stage" style="background-color:yellow;">
20 STAGE
21 </div>
22 <input type = "button" id = "driver" value = "Load Data" />
23 </body>
</html>
```

localhost:8000/load1.html

Click on the button to load result.html file

STAGE

Load Data

localhost:8000/load1.html

Click on the button to load result.html file

**THIS IS RESULT...**

Load Data



# Getting JSON data

- There would be a situation when server would return JSON string against your request
- jQuery utility function `getJSON()` parses the returned JSON string
- jQuery makes the resulting string available to the callback function as first parameter to take further action



# Syntax for getJSON() method

- [selector].getJSON(URL, [data], [callback]);
  - URL: the URL of the server-side resource contacted via the GET method
  - data: an object whose properties serve as the name/value pairs used to construct a query string to be appended to the URL
  - callback: a function invoked when the request completes
    - The value resulting from digesting the response body as a JSON string is passed as the first parameter
    - The status as the second parameter



# Sample: getjson1.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>The jQuery Example</title>
5 <script type = "text/javascript"
6 src = "https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js">
7 </script>
8 <script>
9 $(document).ready(function() {
10 $("#driver").click(function(event){
11 $.getJSON('result.json', function(jd) {
12 $('#stage').html('<p> Name: ' + jd.name + '</p>');
13 $('#stage').append('<p>Age : ' + jd.age+ '</p>');
14 $('#stage').append('<p> Gender: ' + jd.gender+ '</p>');
15 });
16 });
17 });
18 </script>
19 </head>
20 <body>
21 <p>Click on the button to load result.json file</p>
22 <div id = "stage" style = "background-color:yellow">STAGE</div>
23 <input type = "button" id = "driver" value = "Load Data" />
24 </body>
</html>
```

Click on the button to load result.json file

STAGE

Load Data

Click on the button to load result.json file

Name: Manee

Age : 39

Gender: female

Load Data



# Sample: Getting JSON using jQuery

```
4 <title>The jQuery Example</title>
5 <script type = "text/javascript"
6 src =
7 "https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"
8 >
9 </script>
10 </head>
11 <body>
12 <p>Click on the button to load books.json file</p>
13 <div id="stage" style="background-color:yellow;">
14 STAGE
15 </div>
16 <input type = "button" id = "driver" value = "Load Data" />
17 <script>
18 $(document).ready(function() {
19 $("#driver").click(function(event){
20 $.getJSON('json/books.json',
21 function(jd, index) {
22 var result = "";
23 for (i = 0; i < jd.length; i++) {
24 result += ' Title: ' + jd[i].title +
25 '';
26 }
27 result += "";
28 $('#stage').html(result);
29 });
30 });
31 </script>
32 </body>
```

localhost:8000/getbooks.html

localhost:8000/load1.html

Click on the button to load result.html file

STAGE

Load Data

Click on the button to load books.json file

1. Title: Professional JavaScript
2. Title: JavaScript: The Definite Guide

Load Data

# Sample: Getting XML using jQuery

```
10 <body>
11 <p>Click on the button to parse an RSS file</p>
12 <div id="stage" style="background-color:yellow;">
13 </div>
14 <input type = "button" id = "driver" value = "Load Data" />
15 <script>
16 $(document).ready(function () {
17 $('#driver').click(function() {
18 $.ajax({
19 url: 'xml/rss.xml',
20 type: 'GET',
21 dataType: "xml",
22 success: function(data) {
23 parseXml(data);
24 }
25 });
26 });
27 });
28 function parseXml(xml) {
29 var item = $(xml).find("item");
30
31 $(item).each(function() {
32 $("#stage").append($(this).find("title").text() + "
");
33 });
34 }
35 </script>
```

← → C ⓘ localhost:8000/xml/rss.xml

Search by name or XPath

```
<rss version="2.0">
 <channel>
 <title> W3Schools Home Page </title>
 <link> http://www.w3schools.com </link>
 <description> Free web building tutorials </description>
 <item>
 <title> RSS Tutorial </title>
 <link> http://www.w3schools.com/xml/xml_rss.asp </link>
 <description> New RSS tutorial on W3Schools </description>
 </item>
 <item>
 <title> XML Tutorial </title>
 <link> http://www.w3schools.com/xml </link>
 <description> New XML tutorial on W3Schools </description>
 </item>
</channel>
</rss>
```

← → C ⓘ localhost:8000/parse-rss.html

Click on the button to parse an RSS file

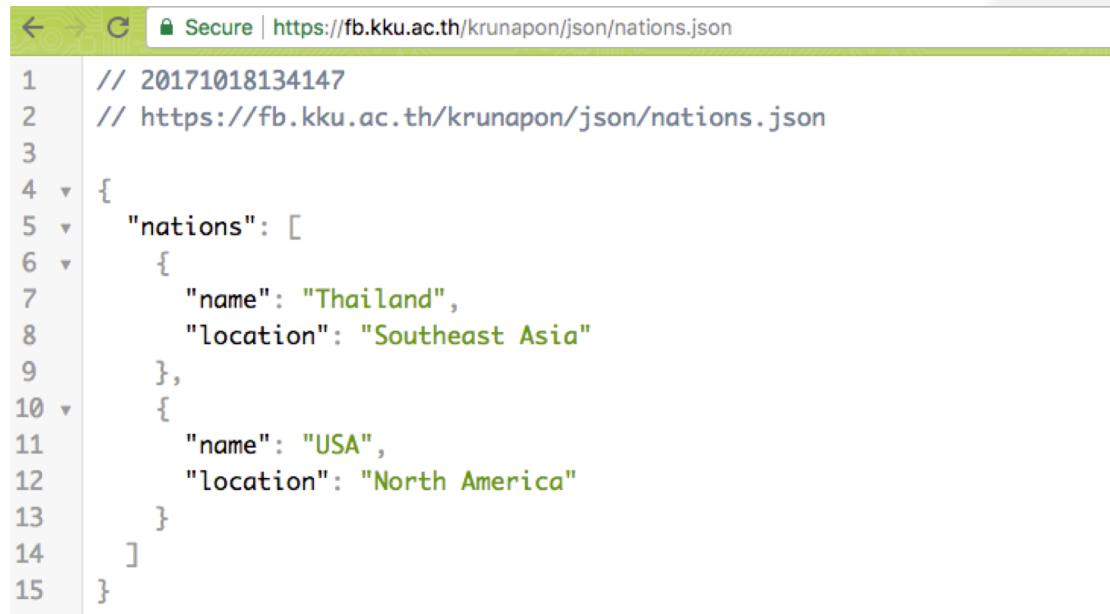
RSS Tutorial  
XML Tutorial

Load Data



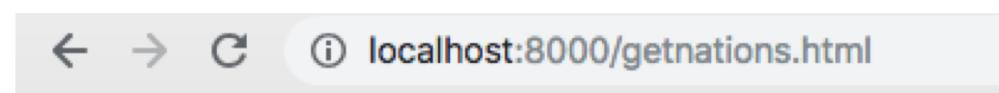
# Exercise 1

- Getting nation name and location from nations.json



```
// 20171018134147
// https://fb.kku.ac.th/krunapon/json/nations.json

1
2
3
4 {
5 "nations": [
6 {
7 "name": "Thailand",
8 "location": "Southeast Asia"
9 },
10 {
11 "name": "USA",
12 "location": "North America"
13 }
14]
15 }
```



Click on the button to load nations.json file

STAGE

Load Data

Click on the button to load nations.json file

- Thailand is in Southeast Asia
- USA is in North America

Load Data



# Exercise 2

- Getting nation name and location from nations.xml

localhost:8000/xml/nations.xml

Search by name or XPath

```
<nations>
 <nation id="th">
 <name> Thailand </name>
 <location> Southeast Asia </location>
 </nation>
 <nation id="us">
 <name> The United States of America </name>
 <location> America </location>
 </nation>
</nations>
```

localhost:8000/getnations.html

```
< > ⌂ ⓘ localhost:8000/getnations.html
```

Click on the button to load nations.json file

STAGE

Load Data

Click on the button to load nations.json file

- Thailand is in Southeast Asia
- USA is in North America

Load Data



# References

- <https://www.tutorialspoint.com/jquery/jquery-ajax.htm>
- <http://www.w3schools.com/jquery/default.asp>
- <http://try.jquery.com>
- <http://jqueryui.com/>
- <http://www.tutorialspoint.com/jquery/ajax-jquery-getjson.htm>

