Website Development Cutting Edge Technology

Contents

- Storing data persistently
- JQuery

Contents

- Storing data persistently
- JQuery

Storing data persistently

- Cookies
- Local Storage
- What about object data?
- JSON (JavaScript Object Notation)
- Using JSON to store JavaScript Objects to Local Storage

Cookies

- It is called by many name, e.g., HTTP cookie, web cookie.
- It is a vary small data that is download from a web server when you visit a web site. It is stored locally by web browser.
- Normally, it is used to send with requests to the same web server. In addition, it is used to tell a web server where requests come from. E.g., keeping a user logged-in.
- These cookies contain key/value pairs of data

Cookies (cont.)

- It allows us to maintain session state over several page requests. For example, keeping track of items in your shopping cart.
- As mentioned, cookies are limited to small amount of text (up to 4KB)
- In addition, cookies have an expiry.

Cookies (cont.)

- The main purpose of cookies:
  - Session management
  - Personalization
  - Tracking
- Example of cookies:
  - "username=Wudhichart; expires=Thu, 18 Dec 2016 12:00:00 UTC";
- See example how to manipulate cookies

Local Storage

- Local Storage is used in the similar as Cookies. i.e., Storing data.
- Its properties:
  - Storage allocated up to 5MB per web domain
  - Web applications on the same web domain share the same Local Storage
  - Data in Local Storage is not transferred to a server

Local Storage (cont.)

- The differences between Local Storage and Cookies are:
  - Storage limitation is larger than that of Cookies (up to 5MB)
  - Data in Local Storage is not transferred to a server
  - Data stored in Local Storage has no expiration time

Local Storage (cont.)

- See example to use Local Storage
  - Store data
  - Get data
  - Set data
  - Remove data
  - Use GUI to manipulate Local Storage

What about object data?

- We normally implement and design our application following the object-orientated paradigm.
- It means that most of our data is in the form of objects
- In order to use Local Storage, we need to convert those objects to Strings since Local Storage accepts on String.
- The only condition is that the conversion needs to keep data and the structure of objects. Therefore, they can be retrieved and reconstituted later.

What about object data? (cont.)

- Generally, the converting process is called serialisation or stringifying and the reverse process is called deserialisation or parsing.

JSON

- Remind about JavaScript's object literal notation
- With the idea of JavaScript's object literal notation, it lead to a data format standard used extensively to transmit structured data such as object data.
- It is called JavaScript Object Notation (JSON).
- The important difference between JSON and JavaScript's object literal notation are
  - JSON is a String
  - JSON does not support properties
  - Property names in JSON are enclosed by quotes.

JSON (cont.)

- JSON basic types
  - Number
  - String
  - Boolean
  - Array
  - Object (JSON object)
  - null

JSON (cont.)

- Recall: JavaScript's object literal

```
var wudhichart = {
    firstName: "Wudhichart",
    lastName: "sawangphol",
    age: 30,
    university: "Mahidol",
    feesPaid: true
};
```

- How about JSON

```
{
    "wudhichart" : {"firstName": "Wudhichart",
                    "lastName": "sawangphol",
                    "age": 30,
                    "university": "Mahidol",
                    "feesPaid": true}
}
```

JSON (cont.)

- JSON object

```
var jsonobject = {
    "wudhichart" : {"firstName":
"Wudhichart",
                    "lastName":
"Sawangphol",
                    "age": 30,
                    "university": "Mahidol",
                    "feesPaid": true}
};
```

- See example.

JSON (cont.)

- JSON Array

```
var jsonarray = {
    "people": [{
        "firstName": "Wudhichart",
        "lastName": "Sawangphol",
        "age": 30,
        "university": "Mahidol",
        "feesPaid": true
    }, {
        "firstName": "Pawitra",
        "lastName": "Chiravirakul",
        "age": 30,
        "university": "Mahidol",
        "feesPaid": true
    }]
};
```

- See example

JSON (cont.)

- JSON Editor
  - http://jsoneditoronline.org/

Using JSON to store JavaScript Objects to Local Storage

- As mentioned earlier, if we can convert an object to a string, the object can be stored in Local Storage.
- In this case, JSON is very helpful since:
  - JSON is String
  - JSON can keep all structures of an object
- JavaScript Functions
  - Object -> JSON: JSON.stringify(object)
  - JSON -> Object: JSON.parse(JSON)

Contents

- Storing data persistently
- JQuery

What is jQuery?

- jQuery is a lightweight JavaScript library, based on the concept '*write less, do more*'
- 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.

What is jQuery? (cont.)

- jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.
- The jQuery library contains the following features:
  - HTML/DOM manipulation
  - CSS manipulation
  - HTML event methods
  - Effects and animations
  - AJAX
  - Utilities

Using jQuery

- There are two versions of jQuery available for downloading:
  - 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)
- Both versions can be downloaded from jQuery.com

Using jQuery (cont.)

- 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-XXX.min.js"></script>
  </head>

jQuery Content delivery network (CDN)

- Google CDN (https://developers.google.com/speed/libraries/#jquery)
  - <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
- Microsoft CDN (https://docs.microsoft.com/en-us/aspnet/ajax/cdn/overview#jQuery_Releases_on_the_CDN_0)
  - <script src="https://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.3.1.min.js"></script>

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 (cont.)

- 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".

jQuery Syntax (cont.)

- 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.
- Put jQuery methods inside a document ready event to prevent any jQuery code from running before the document is finished loading (is ready).

jQuery Syntax (cont.)

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

jQuery Selectors

- jQuery selectors are used to "find" (or select) HTML elements based on their *name*, *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: *$()*.
- See Example

jQuery Selectors (cont.)

Selector	Description	Example
Name	Selects all elements which match with the given element Name	$('tagname')
#ID	Selects a single element which matches with the given ID.	$('#elementid')
.Class	Selects all elements which match with the given Class.	$('.classid')
Universal (*)	Selects all elements available in a DOM.	$('*')
Multiple Elements	Selects the combined results of all the specified selectors	$('A, B, C,….')

JQuery Attributes

- As we know, each element HTML has attribute. JQuery can be used to manipulate those attributes.
- Some of common attributes:
  - className
  - Id
  - Title
  - Text
  - value

JQuery Attributes (cont.)

Methods	Description
attr(properties)	Set a key/value object as properties to all matched elements.
attr(key, fn)	Set a single property to a computed value, on all matched elements.
removeAttr(name)	Remove an attribute from each of the matched elements.
toggleClass(class)	Adds the specified class if it is not present, removes the specified class if it is present.
removeClass(class)	Removes all or the specified class(es) from the set of matched elements.
text( )	Get the combined text contents of all matched elements.
text(val)	Set the text contents of all matched elements.
val( )	Get the input value of the first matched element.
val(val)	Set the value attribute of every matched element

jQuery DOM

- jQuery provides library to manipulate DOM in efficient way. You can write very small chunk of code to modify elements or values of elements on HTML page.

jQuery DOM (cont.)

Methods	Description
html( )	Gets the html contents (innerHTML) of the first matched element.
replaceWith(content)	Replace a complete DOM element with the specified HTML or DOM elements
empty( )	Remove all child nodes from the set of matched elements
remove(expr)	Removes all matched elements from the DOM
after(content)	Insert content after each of the matched elements
before(content)	Inserts content before each of the matched elements
append(content)	Append content to the inside of every matched element

jQuery Event

- An event represents the precise moment when something happens.
- Examples:
  - moving a mouse over an element
  - selecting a radio button
  - clicking on an element

JQuery Event (cont.)

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

JQuery Event (cont.)

Methods	Description
$(document).ready()	Allows us to execute a function when the document is fully loaded.
click()	Attaches an event handler function to an HTML element when the element is clicked
dblclick()	Attaches an event handler function to an HTML element when the element is double-clicked
mouseenter()	Attaches an event handler function to an HTML element when the mouse pointer enters the HTML element
mouseleave()	Attaches an event handler function to an HTML element when the mouse pointer leaves the HTML element
on()	Attaches one or more event handlers for the selected elements

JQuery Effects

- JQuery provides methods to add effects to HTML pages.
- For example,
  - Showing and Hiding elements
  - Fading elements
  - Sliding elements up and down

JQuery Effects (cont.)

Methods	Description
hide([speed],[callback])	Hide HTML elements
show([speed],[callback])	Show HTML elements
fadeIn([speed],[callback])	Fade elements in
fadeOut([speed],[callback])	Fade elements out
slideDown([speed],[callback])	Slide down an element
slideUp([speed],[callback])	Slide up an element

- The optional *speed* parameter specifies the speed of the hiding/showing, and can take the following values: "slow", "fast", or milliseconds.
- The optional *callback* parameter is a function to be executed after the effect method completes

JQuery Effects (cont.)

- Animate method allows us to create custom animations.
- Syntax:
  - **$(*selector*).animate({*params*},*speed*,*callback*)**
  - The required *params* parameter defines the CSS properties to be animated.
  - The optional *speed* parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.
  - The optional *callback* parameter is a function to be executed after the animation completes.

JQuery AJAX

- AJAX (Asynchronous JavaScript and XML) is the approach to exchange data with server and update parts of webpages without reloading the whole page.
- The main methods that we will use are:
  - load(*URL,[data],[callback]*)
    - The URL of the server-side resource to which the request is sent. It could be a CGI, ASP, JSP, or PHP script
    - The optional data parameter specifies a set of querystring 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.

JQuery AJAX (cont.)

- getJSON(URL, [data], [callback])
  - The URL of the server-side resource to which the request is sent. It could be a CGI, ASP, JSP, or PHP script
  - The optional data parameter specifies a set of querystring 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.

JQuery Interaction

- Interactions allows us to add basic mouse-based behaviours to any element.
- We can create sortable lists, resizeable elements, drag & drop behaviours.
- In order to use Interactions, we need to have JQuery UI library. It can be download from https://jqueryui.com/
- CDN: https://ajax.googleapis.com/ajax/libs/jqueryui/1.11.3/jquery-ui.min.js

JQuery Interaction (cont.)

- Some interesting methods
  - Draggable
  - Droppable
  - Resizable
  - Selectable
  - Sortable

References

- https://www.w3schools.com