CS3201: Computer Network Technology TY Div A n B AY 2020-21

Study Material for Section-II-Part-IV- jQuery

(Resource: *www.w3schools.com*)

Learning jQuery

jQuery is a lightweight, "write less, do more", JavaScript library.

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 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

jQuery has plugins for almost any task out there.

jQuery greatly simplifies JavaScript programming.

jQuery is easy to learn.

What You Should Already Know

Before you start studying jQuery, you should have a basic knowledge of:

HTML, CSS, JavaScript

Why jQuery?

There are lots of other JavaScript libraries out there, but jQuery is probably the most popular, and also the most extendable.

Many of the biggest companies on the Web use jQuery, such as:

- Google
- Microsoft
- IBM
- Netflix

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

```
<head>
<script src="jquery-3.5.1.min.js"></script>
</head>
```

- Include jQuery from a CDN, like Google

```
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min
.js"></script>
</head>
```

jQuery Syntax

With jQuery you select (query) HTML elements and perform "actions" on them.

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)

- 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 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 name, id, classes, types, attributes, values of attributes**.

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")
```

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

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

The #id Selector

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

To find an element with a specific id, write a hash character, followed by the id of the HTML element:

```
$("#test")
```

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

The .class Selector

The jQuery *.class* selector finds elements with a specific class.

To find elements with a specific class, write a period character, followed by the name of the class:

```
$(".test")
```

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

$("*")	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"
$("a[target!='_blank']")	Selects all <a> elements with a target attribute value NOT equal to "_blank"
$(":button")	Selects all <button> elements and <input> elements of type="button"

$("tr:even")	Selects all even <tr> elements
$("tr:odd")	Selects all odd <tr> elements

Functions In a Separate File

If your website contains a lot of pages, and you want your jQuery functions to be easy to maintain, you can put your jQuery functions in a separate .js file.

```
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min
.js"></script>
<script src="my_jquery_functions.js"></script>
</head>
```

What are Events?

All the different visitors' actions that a web page can respond to are called events.

An event represents the precise moment when something happens.

Examples:

- moving a mouse over an element
- selecting a radio button
- clicking on an element

The term **"fires/fired"** is often used with events. Example: "The keypress event is fired, the moment you press a key".

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 Syntax For Event Methods

In jQuery, most DOM events have an equivalent jQuery method.

To assign a click event to all paragraphs on a page, you can do this:

```
$("p").click();
```

The next step is to define what should happen when the event fires. You must pass a function to the event:

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

The Document Ready Event

```
$(document).ready(function(){

  // jQuery methods go here...

});
```

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.

click()

The `click()` method attaches an event handler function to an HTML element.

The function is executed when the user clicks on the HTML element.

```
$("p").click(function(){
  $(this).hide();
});
```

dblclick()

The `dblclick()` method attaches an event handler function to an HTML element.

The function is executed when the user double-clicks on the HTML element:

```
$("p").dblclick(function(){
  $(this).hide();
});
```

mouseenter()

The `mouseenter()` method attaches an event handler function to an HTML element.

The function is executed when the mouse pointer enters the HTML element:

```
$("#p1").mouseenter(function(){
  alert("You entered p1!");
});
```

mouseleave()

The `mouseleave()` method attaches an event handler function to an HTML element.

The function is executed when the mouse pointer leaves the HTML element:

```
$("#p1").mouseleave(function(){
  alert("Bye! You now leave p1!");
});
```

mousedown()

The `mousedown()` method attaches an event handler function to an HTML element.

The function is executed, when the left, middle or right mouse button is pressed down, while the mouse is over the HTML element:

```
$("#p1").mousedown(function(){
  alert("Mouse down over p1!");
});
```

mouseup()

The `mouseup()` method attaches an event handler function to an HTML element.

The function is executed, when the left, middle or right mouse button is released, while the mouse is over the HTML element:

```
$("#p1").mouseup(function(){
  alert("Mouse up over p1!");
});
```

hover()

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

The first function is executed when the mouse enters the HTML element, and the second function is executed when the mouse leaves the HTML element:

```
$("#p1").hover(function(){
  alert("You entered p1!");
},
function(){
  alert("Bye! You now leave p1!");
});
```

focus()

The `focus()` method attaches an event handler function to an HTML form field.

The function is executed when the form field gets focus:

```
$("input").focus(function(){
  $(this).css("background-color", "#cccccc");
});
```

blur()

The `blur()` method attaches an event handler function to an HTML form field.

The function is executed when the form field loses focus:

```
$("input").blur(function(){
  $(this).css("background-color", "#ffffff");
});
```

The on() Method

The `on()` method attaches one or more event handlers for the selected elements.

Attach a click event to a `<p>` element:

Example

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

Attach multiple event handlers to a <p> element:

Example

```
$("p").on({
  mouseenter: function(){
    $(this).css("background-color", "lightgray");
  },
  mouseleave: function(){
    $(this).css("background-color", "lightblue");
  },
  click: function(){
    $(this).css("background-color", "yellow");
  }
});
```

jQuery Effects - Hide and Show

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

$("#show").click(function(){
  $("p").show();
});
```

jQuery toggle()

You can also toggle between hiding and showing an element with the `toggle()` method.

Shown elements are hidden and hidden elements are shown:

```
$("button").click(function(){
  $("p").toggle();
});
```

jQuery fadeIn() Method

The jQuery `fadeIn()` method is used to fade in a hidden element.

Syntax:

```
$(selector).fadeIn(speed,callback);
```

```
$("button").click(function(){
  $("#div1").fadeIn();
  $("#div2").fadeIn("slow");
  $("#div3").fadeIn(3000);
});
```

jQuery fadeOut() Method

The jQuery `fadeOut()` method is used to fade out a visible element.

```
$("button").click(function(){
  $("#div1").fadeOut();
  $("#div2").fadeOut("slow");
  $("#div3").fadeOut(3000);
});
```

jQuery fadeToggle() Method

The jQuery `fadeToggle()` method toggles between the `fadeIn()` and `fadeOut()` methods.

If the elements are faded out, `fadeToggle()` will fade them in.

If the elements are faded in, `fadeToggle()` will fade them out.

jQuery fadeTo() Method

The jQuery `fadeTo()` method allows fading to a given opacity (value between 0 and 1).

```
$("button").click(function(){
  $("#div1").fadeTo("slow", 0.15);
  $("#div2").fadeTo("slow", 0.4);
  $("#div3").fadeTo("slow", 0.7);
});
```

jQuery Sliding Methods

With jQuery you can create a sliding effect on elements.

jQuery has the following slide methods:

- `slideDown()`
- `slideUp()`
- `slideToggle()`

jQuery Animations - The animate() Method

The jQuery `animate()` method is used to create custom animations.

```
$("button").click(function(){
  $("div").animate({left: '250px'});
});
```

jQuery animate() - Manipulate Multiple Properties

```
$("button").click(function(){
  $("div").animate({
    left: '250px',
    opacity: '0.5',
    height: '150px',
    width: '150px'
  });
});
```

jQuery Stop Animations

The jQuery stop() method is used to stop animations or effects before it is finished.

```
$("#stop").click(function(){
  $("#panel").stop();
});
```

jQuery Callback Functions

A callback function is executed after the current effect is finished.

Typical syntax: **$(*selector*).hide(*speed,callback*);**

```
$("button").click(function(){
  $("p").hide("slow", function(){
    alert("The paragraph is now hidden");
  });
});
```

jQuery Method Chaining

```
$("#p1").css("color", "red").slideUp(2000).slideDown(2000);
```

jQuery DOM Manipulation

One very important part of jQuery is the possibility to manipulate the DOM.

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 of form fields

```
$("#btn1").click(function(){
  alert("Text: " + $("#test").text());
});
$("#btn2").click(function(){
  alert("HTML: " + $("#test").html());
});
```

```
$("#btn1").click(function(){
  alert("Value: " + $("#test").val());
});
```

Get Attributes - attr()

The jQuery `attr()` method is used to get attribute values.

```
$("button").click(function(){
  alert($("#w3s").attr("href"));
});
```

Set Content - text(), html(), and val()

We will use the same three methods from the previous page to **set content**:

- `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

```
$("#btn1").click(function(){
  $("#test1").text("Hello world!");
});
$("#btn2").click(function(){
  $("#test2").html("<b>Hello world!</b>");
});
$("#btn3").click(function(){
  $("#test3").val("Dolly Duck");
});
```

Set Attributes - attr()

The jQuery `attr()` method is also used to set/change attribute values.

Add New HTML Content

We will look at four jQuery methods that are used to add new content:

- `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

The jQuery `append()` method inserts content AT THE END of the selected HTML elements.

```
$("p").append("Some appended text.");

$("p").prepend("Some prepended text.");

$("img").after("Some text after");

$("img").before("Some text before");
```

Add Several New Elements With after() and before()

Remove Elements/Content

To remove elements and content, there are mainly two jQuery methods:

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

```
$("#div1").remove();

$("#div1").empty();
```

jQuery Manipulating CSS

jQuery has several methods for CSS manipulation. We will look at the following 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 from the selected elements
- `css()` - Sets or returns the style attribute

jQuery css() Method

```
$("p").css("background-color");
```

Set Multiple CSS Properties

To set multiple CSS properties, use the following syntax:

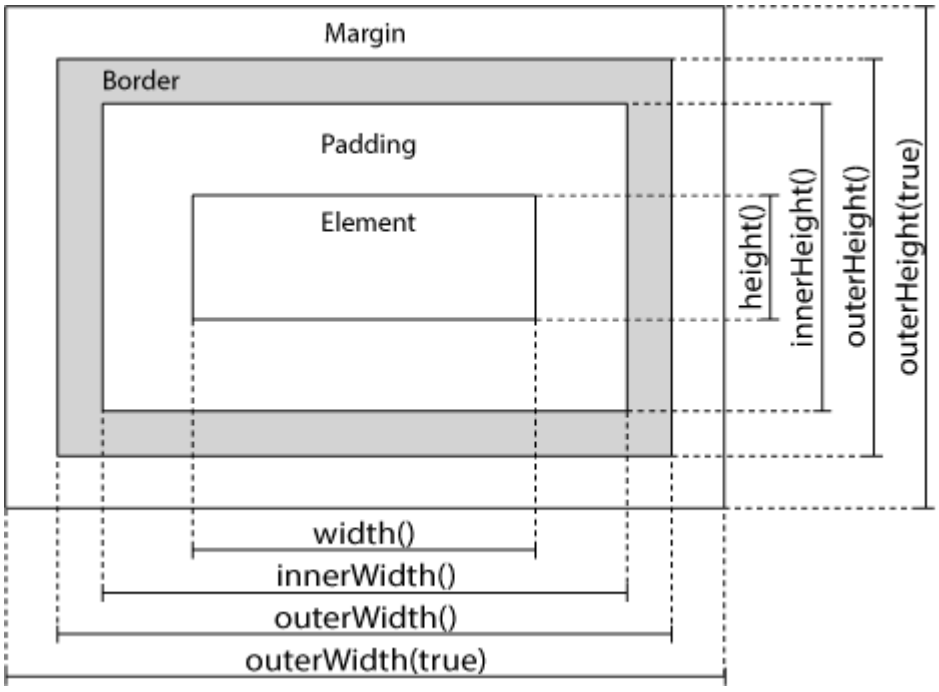
```
css({"propertyname":"value","propertyname":"value",...});
```

```
$("p").css({"background-color": "yellow", "font-size": "200%"});
```

jQuery Dimension Methods

jQuery has several important methods for working with dimensions:

- `width()`
- `height()`
- `innerWidth()`
- `innerHeight()`
- `outerWidth()`
- `outerHeight()`


Margin
Border
Padding
Element
height()
innerHeight()
outerHeight()
outerHeight(true)
width()
innerWidth()
outerWidth()
outerWidth(true)