BIT 2nd Year
Semester 3
IT 3505

Web Application Development II

Fundamentals of Asynchronous JavaScript and XML (AJAX) – Part 3

jQuery Introduction

Introduction

- jQuery is a lightweight javaScript library developed to simplify the development of dynamic Web pages.
- Advantages in using jQuery
  - Easy to use than raw javascript commands.
  - Supports almost all browsers.
  - Availability of good documentation.
  - Has a large development community.
  - Extensibility.

jQuery selectors

- JQuery selectors enable you to access HTML DOM elements easily.
- Selectors can be used to access DOM elements by
  - Tag Name
  - ID
  - Class
  - Attributes
  - Attribute values
  - …….
- All selectors return a collection of JQuery objects.
- You can access an individual object in this collection by specifying its index inside square brackets.

JQuery tag selector

- The JQuery tag selector allows you to access HTML elements based on their tags.
- Syntax

  $('tag')

  Where **tag** is the tag assigned for the HTML element.

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>JQuery Examples</title>
<script src="jquery_2.2.0.min.js"></script>
</head>
<body>

<script type="text/javascript">
  $(document).ready(function() {
      alert($('div')[0].getAttribute('name'));
      alert($('div')[1].getAttribute('name'));

})
</script>
```

```
<div name="first">
  first division
</div>
<div name="second">
  second division
</div>


</script>
</body>
</html>
```

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>JQuery Examples</title>
<script src="jquery_2.2.0.min.js"></script>
</head>
<body>

<script type="text/javascript">
  $(document).ready(function() {
      var objs = $('div');
      alert(objs[0].getAttribute('name'));
      alert(objs[1].getAttribute('name'));

})
</script>
```

```
<div name="first">
  first division
</div>
<div name="second">
  second division
</div>


</script>
</body>
</html>
```

Iterating through a JQuery object collection

.each construct can be used to iterate through a collection of JQuery objects.

Syntax :

.each(callback,arguments)

Semantics:

For each JQuery object call the anonymous function **callback** with the specified **arguments**.

Callback function of .each

- The callback function of .each is an anonymous function with two optional parameters.
  - The first parameter is the index of the object in the collection.
  - The second parameter is the DOM element in the JQuery object collection.

Example

```
<title>JQuery Examples</title>
<script src="jquery_2.2.0.min.js"></script>
</head>
<body>

<script type="text/javascript">
  $(document).ready(function() {
      $('div').each(printProperties);

  })

function printProperties(index,item){
  alert("Object index = "+ index + " name = "+
          item.getAttribute('name'));
}
</script>
```

```
<div name="first">
  first division
</div>
<div name="second">
  second division
</div>

</script>
</body>
</html>
```

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>JQuery Examples</title>
<script src="jquery_2.2.0.min.js"></script>
</head>
<body>

<script type="text/javascript">
  $(document).ready(function() {
      $('div').each(function(index,item){
          alert("Object index = "+ index + "
name = "+
                  item.getAttribute('name'));
       });
  })
</script>
```

```
<div name="first">
  first division
</div>
<div name="second">
  second division
</div>


</script>
</body>
</html>
```

Example

this keyword can be used to access the current DOM element in the iteration.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>JQuery Examples</title>
<script src="jquery_2.2.0.min.js"></script>
</head>
<body>
<script type="text/javascript">
  $(document).ready(function() {
      $('div').each(function(index){
              alert("Object index = "+ index + "
name = "+
                          this.getAttribute('name'));
       });
  })
</script>
```

```
<div name="first">
  first division
</div>
<div name="second">
  second division
</div>


</script>
</body>
</html>
```

Example

The second parameter of the callback can also be omitted if it is not needed in the body of the function.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>JQuery Examples</title>
<script src="jquery_2.2.0.min.js"></script>
</head>
<body>
<script type="text/javascript">
  $(document).ready(function() {
      $('div').each(function(){
              alert( " name = "+
this.getAttribute('name'));
       });
  })
</script>
```

```
<div name="first">
  first division
</div>
<div name="second">
  second division
</div>


</script>
</body>
</html>
```

HTML elements, tags, attributes and properties

- An HTML element is an object in the hieratical model of a document. An "element" is an abstract representation of a node in the DOM. An element may have properties. A **property** of an element is a named characteristic of the element. An element may inherit properties from its parent element.
- An HTML element is represented by using a tag. Typically a tag is encoded in an HTML document by using the following syntax.

  tag_start tag_body tag_end
- The **tag_start** starts with the **<** symbol followed by the **element name**, one or more optional **attribute name/value pairs** separated by whitespace(s) and **>** or **/>** symbols.
  - The attributes of a tag represent the properties of the HTML element.
- The **tag_end** starts with **</** followed by **element name** and **>** symbol.

Listing all attributes of an HTML element by using only JavaScript

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>JQuery Examples</title>
<script src="jquery_2.2.0.min.js"></script>
</head>
<body>


<div id="parent" name="name_of_parent"
class="parentClass">
   This is the parent division
  <div id="child" name="name_of_child">
     THis is the child division
  </div>
</div>
```

```
<script type="text/javascript">
   window.onload = function(){
     var attrs =

document.getElementById("child").attributes;


     for(var i=0; i < attrs.length ; i++){
     alert("attribute name/value pair of child div
= " + attrs[i].nodeName
+","+attrs[i].nodeValue);
     }
};




</script>
</body>
</html>
```

Listing all attributes of an HTML element by using only Jquery : Example 1

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>JQuery Examples</title>
<script src="jquery_2.2.0.min.js"></script>
</head>
<body>


<div id="parent" name="name_of_parent" class="parentClass">
   This is the parent division
  <div id="child" name="name_of_child">
     THis is the child division
  </div>
</div>
```

```
<script type="text/javascript">
   $(document).ready(function(){
      $("#child").each(function(){
         var attrs = this.attributes;
         for(var i=0; i < attrs.length;i++){
               alert("attribute name/value pair of child div = " + attrs[i].nodeName +","+attrs[i].nodeValue);
                        }
      });
});
</script>
</body>
</html>
```

Note : $(“#child”) returns a collection of **JQuery objects** whereas this returns an **DOM object**.

Listing all attributes of an HTML element by using only Jquery : Example 2

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>JQuery Examples</title>
<script src="jquery_2.2.0.min.js"></script>
</head>
<body>


<div id="parent" name="name_of_parent"
class="parentClass">
   This is the parent division
  <div id="child" name="name_of_child">
     THis is the child division
  </div>
</div>
```

```
<script type="text/javascript">
   $(document).ready(function(){
      $("#child").each(function(){
         $(this.attributes).each(function(){
           alert("Attribure of div child = " +
this.nodeName +","+this.nodeValue);
         });
      });
});
</script>
</body>
</html>
```

Note : $(this.attributes) converts the DOM object to JQuery object. The each construct can be used on JQuery objects not on DOM objects.

JavaScript Objects

- JavaScript object is a collection of properties.
- A JavaScript object can be defined by using the following syntax

  {property_1, property_2,……., property_n}
- Each property is a name value pair separated by the : symbol.

Example :

{"name":"Saman","age":20}

Iterating through the properties of an object

- The JavaScript **for** construct can be used to iterate through all elements in an object.

Syntax :

```
for(property in object){
  // body
}
```

Iterating through the properties of an object: Example

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>JQuery Examples</title>
<script src="jquery_2.2.0.min.js"></script>
</head>
<body>
```

```
<script type="text/javascript">
  $(document).ready(function(){
    var obj ={"name":"Saman","age":20};
    for(prop in obj){
      alert("property name - " + prop + ",value - "+obj[prop]);
    };

});

</script>
</body>
</html>
```

Converting DOM objects to JQuery objects

- The following construct can be used to convert a DOM object to an JQuery object.

$(DOM_Object)

Listing all properties of a JQuery object

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>JQuery Examples</title>
<script src="jquery_2.2.0.min.js"></script>
</head>
<body>


<div id="parent" name="name_of_parent"
class="parentClass">
   This is the parent division
  <div id="child" name="name_of_child">
     THis is the child division
  </div>
</div>
```

```
<script type="text/javascript">
   $(document).ready(function(){
      $("#child").each(function(){
         obj = $(this);
         for(prop in obj){
            alert("property name - " + prop +
",value - "+obj[prop]);
         }
      });
});
</script>
</body>
</html>
```

jQuery selectors

- JQuery selectors enable you to access HTML DOM elements easily.
- Selectors can be used to access DOM elements by
  - Tag Name
  - ID
  - Class
  - Attributes
  - Attribute values
  - …….
- All selectors return a collection of JQuery objects.
- You can access an individual object in this collection by specifying its index inside square brackets.

#id selector

- Syntax

  $('#id')

  Where **id** is the value assigned to the **id** attribute of the HTML element.

  The name of the id must be preceded by the # symbol.

#id selector

- **id** of an HTML element must be unique on a page.
- #id selector returns only the first HTML element (collection with at most one item), if the page contain multiple elements with the same **id**.
- #id selector is the most efficient JQuery selector.
- If the requested id is not in the page JQuery will not generates an error.
  - Use the **length** property to check the existence of the requested element.
- **What returns by the #id selector is not the raw DOM object but a JQuery object that wraps the DOM element.**
  - You can obtain the raw DOM object by using the construct $('#id')[0]

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>JQuery Examples</title>
<script
src="jquery_2.2.0.min.js"></script>
</head>
<body>

<div id="id01" name="test">
   This is a division
</div>
```

```
<script type="text/javascript">
 $(document).ready(function(){
   if($('#id01').length > 0){
       alert("Element found");
   } else {
       alert("No element with the id =
id01 found");
   }
 });

</script>
</body>
</html>
```

Example .....

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>JQuery Examples</title>
<script
src="jquery_2.2.0.min.js"></script>
</head>
<body>

<div id="id01" name="test">
   This is a division
</div>
```

```
<script type="text/javascript">
  $(document).ready(function(){
   //this construct allow  you to access
a JQuery object
    alert($('#id01')[0]);
 });


 });

</script>
</body>
</html>
```

Example .....

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>JQuery Examples</title>
<script
src="jquery_2.2.0.min.js"></script>
</head>
<body>

<div id="id01" name="test">
   This is a div
</div>
```

```
<script type="text/javascript">
 $(document).ready(function(){
  //this construct allows you to print
the value of an
 // attribute of an JQuery object
   alert($('#id01')[0].getAttribute("id"));
 });

</script>
</body>
</html>
```

JQuery class selector

The class selector allows HTML elements to be selected on class names.

Syntax:

$(‘.class_name’)

– The name of the class must be preceded by the period(.) symbol.

Example:

$(‘.button’) //select all DOM element with the class name **button**

Example .....

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>JQuery Examples</title>
<script src="jquery_2.2.0.min.js"></script>
<script src="script.js"></script>
</head>
<body>
<div id="id01" class="class1" name="div1">
   This is the first division
</div>
<div id="id02" class="class1" name="div2">
   This is the second division
</div>
<script type="text/javascript">
$(document).ready(function(){
  ex01();
});
</script>
</body>
</html>
```

```
//Content of file script.js
//select all DOM elements with the
class name class1
function ex01(){
  $(".class1").each(function(){
     alert(this.getAttribute('id'));
   });
}
```

JQuery attribute selector

- The attribute selector allows HTML elements to be selected on specific attributes or their values.

Syntax:

$('[attribute]')

$('[attribute="value"]')

//value in double quote and attribute in single quote

$("[attribute='value']")

//value in single quote and attribute in double quote

$("[attribute=\"value\"]")

$('[attribute=\'value\']')

Example .....

Refer to the html page example01.php

```
//Content of file script.js
//Select all elements with the attribute
name
function ex01(){
  $('[name]').each(function(){
     alert(this.getAttribute('id'));
   });
}
```

Example .....

Refer to the html page example01.php

```
//Content of file script.js
//Select all elements with the attribute name and its
// value second
function ex01(){
  $('[name="second"]').each(function(){
     alert(this.getAttribute('id'));
   });
}
```

Example .....

Refer to the html page example01.php

```
//Content of file script.js
//Select all elements with the tag h2 and class attribute
function ex01(){
  $('h2[class]').each(function(){
     alert(this.getAttribute('id'));
   });
}
```

Example .....

Refer to the html page example01.php

```
//Content of file script.js
//Select all  child elements of the elements with id ul2
// having the attribute name
function ex01(){
  $('#ul2 [name]').each(function(){
     alert(this.getAttribute('id'));
   });
}
```

Selecting descendent elements of an HTML element

Spaces can be used between selectors to define descendent elements of the preceding selectors.

Example :

$('div .c1')

- All child elements of div elements with the class name **c1**

$('div ul .names')

-all elements with the class name **names** which are children of **ul** elements which in turn are children of **div** elements.

Example .....

Refer to the html page example01.php

```
//Content of file script.js
//Select all elements with the class name item
// which are children of elements with the class
// name names which in turn are the child elements of
// div elements.
function ex01(){
  $('div .names .item').each(function(){
     alert(this.getAttribute('id'));
   });
}
```

Example .....

Refer to the html page example01.php

```
//Content of file script.js
//Select all
//1) chid elements with the class name fruits and id l12 or
// 2) elements with id l21
function ex01(){
  $('.fruits #l12,#l21').each(function(){
     alert(this.getAttribute('id'));
   });
}
```

Combining selectors together (or condition)

Comma (,) in the selector list indicates the or condition

Example:

$('.button, .option')

//select all DOM elements with the class name **button** or **option**.

$('#id01, .option')

//select all DOM elements with the id **id01** or class name **option**

Example .....

Refer to the html page example01.php

```
 //Content of file script.js
//Select elements with ids l11 or l21
function ex01(){
  $('#l11,#l21').each(function(){
     alert(this.getAttribute('id'));
   });
}
```

Example .....

Refer to the html page example01.php

```
//Content of file script.js
//select all elements with id id01 or
class name class2
function ex01(){
  $("#id01,.class2").each(function(){
     alert(this.getAttribute('id'));
   });
}
```

Combining selectors together (and condition)

The selectors appended together one after the other (without embedded spaces) define a **and** condition

Example:

$('div.option')

//select all **div** elements with the class name **option**.

$('#id01.option')

//select all DOM elements with the id **id01** and the class name **option**

Example .....

Refer to the html page example01.php

```
//Content of file script.js
//Select all elements with h2 tag and
class name title
function ex01(){
  $('h2.title').each(function(){
     alert(this.getAttribute('id'));
   });
}
```

Class handling in JQuery

Adding a CSS class to an element

The Jquery function addClass can be used to add a CSS classes to an HTML element.

Syntax :

addClass(“className1 className2 …… className_n”)

Example

Refer Example02.php file.

```
//Content of file script01.js
function ex01(){
  $("#button1").click(function(){
    if($("#div1“).hasClass(“class1){
       alert(“Div has the class
       $("#div1").addClass("class1");
  });

}
```

Checking the existence of a class

JQuery **hasClass** function can be used to check whether a particular HTML element has a given class or not.

Syntax:

hasClass(“className”)

Example

Refer Example02.php file.

```
//Content of file script01.js
function ex01(){
  $("#button1").click(function(){
     if($("#div1").hasClass("class1")){
        alert("Div has class1 class");
     }else {
       $("#div1").addClass("class1");
     }
  });

}
```

Deleting a class of an HTML element

JQuery **removeClass** function can be used to remove a class/es from an HTML element.

Syntax:

removeClass("className1 className1 className_n")

The function call **removeClass()** will remove all classes assigned to the HTML element.

Example

Refer Example02.php file.

```
//Content of file script01.js
function ex01(){
  $("#button1").click(function(){
     if($("#div1").hasClass("class1")){

$("#div1").removeClass("class1");
     }else {
       $("#div1").addClass("class1");
     }
  });

}
```

Toggle a class of an HTML element

JQuery **toggleClass** function can be used add class/es to an HTML element if the element has not assigned the class/es or to remove the classes if they are already assigned.

Syntax:

toggleClass("className1 className1 className_n")

Example

Refer Example02.php file.

```
//Content of file script01.js
function ex01(){
  $("#button1").click(function(){
       $("#div1").toggleClass("class1");

  });

}
```