Parul® University

Web Programming

Prof. Mosam Patel, Assistant Professor
CSE

Parul®
University

DIGITAL LEARNING CONTENT

CHAPTER-5

JSON And JQUERY

Introduction of jQuery

- **The goal of jQuery is to make it very accessible to use JavaScript on your website.**

Before proceeding with jQuery, you should have basic knowledge of **HTML, CSS, JavaScript, Document Object Model (DOM)** and any text editor. As we are going to reveal a web-based application using jQuery, it would be fine if you have a knowledge of how the internet and web-based applications work.

Introduction of jQuery

- jQuery is a quick and compact JavaScript library created by John Resig in 2006 with a great motto: Write less, do more.
- jQuery clarifies HTML document traversing, event handling, animating, and Ajax interactions for accelerated web development.
- jQuery is a JavaScript toolkit that aims to simplify many tasks by lettering less code.

Introduction of jQuery

- **Here is a list of essential core features supported by jQuery**
  - DOM Manipulation
  - Event handling
  - AJAX Support
  - Animations
  - Lightweight
  - Cross Browser Support

Embed in page

```
<html>
        <head>
                <script src="path/to/jquery-x.x.js"></script>
                <script>
                        $(document).ready(function(){
                        // Start here
                        });
                </script>
        </head>
        <body> … </body>
</html>
```

Standard jQuery Syntax

- With jQuery, you find (query) HTML components and perform "actions" on them.
- The jQuery syntax is a class marker for choosing HTML components and delivering 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 philosophy

```
$("div").addClass("xyz");
```

Do something with them

jQuery Object

jQuery Syntax - Examples

- $(this).hide() - hides the current element.
- $(“table").hide() - hides all <table> elements.
- $(".myclass").hide() - hides all elements with class=" myclass".
- $("#myid").hide() - hides the element with id=" myid".
- jQuery uses CSS syntax to select elements

jQuery Selectors

- Just pass a selector to $().
- jQuery selectors are used to "discover" (or select) HTML components based on their names, IDs, classes, types, attributes, attributes values, and many more.
- All jQuery selectors begin with a dollar sign and parentheses: $().

jQuery Selectors

- **Think about your simplest CSS file.**

```
#header{
            margin : 0 auto;
      }
      div{
            margin : 0px;
            padding : 0px
      }
      ul.menu li{
      .....
      }
```

jQuery Selectors

- **The highlighted items are selectors**

```
#header{
        margin : 0 auto;
}
div{
        margin : 0px;
        padding : 0px
}
ul.menu li{
.....
}
```

jQuery Selectors

- Selecting using selectors
- Selecting By Id

  $(“#header”)
- Selecting By Class

  $(“.updated”)
- Selecting by tag name

  $(“table”)
- Combine them

  $(“table.user-list”), $(“#footer ul.menu li”)

jQuery Selectors Example

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

```
<head>
<script   src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
</script>
 <script> $(document).ready(function(){
       $("button").click(function(){
         $("p").hide();
      });
   }); </script>
</head>  <body>
    <h2>This is my heading</h2>
    <p>This is my paragraph</p>
    <p>This paragraph contain my name.</p>
   <button>Click me to hide paragraphs</button></body>
```

jQuery Selectors Example

When a user clicks on a button, the element with id=“myid" will be hidden:

```
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
</script> <script>
    $(document).ready(function(){
      $("button").click(function(){
        $("#myid").hide();
       });  });
   </script> </head>
    <body>
     <h2>Parul University</h2>
     <p>This is normal peragraph</p>
      <p id=“myid">This paragraph is with myid.</p>
     <button>Click me</button></body>
```

jQuery Selectors Example

Example: **When a user clicks on a button, the elements with class="myclass" will be hidden:**

```
<head>
<scriptsrc="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></scrip>
  <script>
     $(document).ready(function(){
           $("button").click(function(){
              $(".myclass").hide();
          });  });
    </script>  </head> <body>
     <h2 class="myclass">Parul University</h2>
     <p class=" myclass">My class paragraph.</p>
     <p>This is another paragraph.</p>
      <button>Click me</button></body>
```

jQuery – Events

We can build dynamic web pages using events. Event is a function that can be identified by your web application.

Following are the examples of events −

- **A mouse click**
- **A web page loading**
- **Taking mouse over an element**
- **Submitting an HTML form**
- **A keystroke on your keyboard, etc.**

When these events are triggered, you can use a custom function to do whatever you want with this event. Its called a custom function event handler.

jQuery Syntax For Event Methods

- Using the jQuery event model, we can install event handlers on DOM elements.
- jQuery simplifies it to set event-driven responses of on-page elements.
- These events are often triggered by the end user's interaction with the page, such as when text is entered into a form element or a mouse pointer is moved.
- In some cases, such as page load and unload events, the browser itself will trigger the event.

jQuery Syntax For Event Methods

- **jQuery offers convenience methods for most native browser events.**

These methods — including .click(), .focus(), .blur(), .change(),

```
// Event setup using a convenience method
$( "p" ).click(function() {
    console.log( "You clicked a paragraph!" );
});
```

```
// Equivalent event setup using the `.on()` method
$( "p" ).on( "click", function() {
    console.log( "click" );
});
```

Regularly Used jQuery Event Functions

1. $(document).ready()

- The $(document).ready() function enable us to execute a function when the document is fully loaded.

2. 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.

Regularly Used jQuery Event Functions

- **The following are cross platform and recommended event types which you can bind using JQuery –**

Sr.No.	Event Type & Description
1	**blur** Occurs when the element loses focus.
2	**change** Occurs when the element changes.
3	**click** Occurs when a mouse click.
4	**dblclick** Occurs when a mouse double-click.
5	**error** Occurs when there is an error in loading or unloading etc.
6	**focus** Occurs when the element gets focus.

Example

```
<head><script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
  <script>
    $(document).ready(function(){
        $("p").click(function(){
         $(this).hide();
         });
      });
   </script></head>
 <body>
   <p>If you click on me Button, it will disappear.</p>
   <p>Click me away!</p>
   <p>Click me too!</p>
 </body>
```

What is JSON

- JSON stands for JavaScript Object Notation
- JSON objects are used for server-client data transfer.
- JSON is a format for the data storage and exchange.
- JSON is a lightweight format for data-interchange
- JSON is text, written in JavaScript with object notation.

Features of JSON

- It is light-weight
- It is language independent
- Easy to read and write
- Text based, human readable data exchange format

Rules for JSON Syntax

- Data should be in name/value pairs
- Commas should separate data
- Curly braces should hold objects
- Square brackets hold arrays

Parul®
University

DIGITAL LEARNING CONTENT

Data Structure types of JSON and how to read them

- JSON objects
- JSON objects in the array
- Nesting of JSON objects

Parul®
University

DIGITAL LEARNING CONTENT

Data Structure types of JSON and how to read them

- JSON objects
- JSON objects in the array
- Nesting of JSON objects

JSON Objects

```
var chaitanya = {
  "name" : "Chaitanya Singh",
  "age" : "28",
  "website" : "paruluniversity.ac.in"
};
```

- **Access the information from such an object as JSON:**

```
document.writeln("The name is:  " +chaitanya.name);
document.writeln("his age is: " + chaitanya.age);
document.writeln("his website is: "+ chaitanya.website);
```

JSON objects in the array

In the example above, we have stored one person's information in a JSON object, assuming we want to store more than one person's information; in that case, we can have an array of objects.

```
var students = [{
  "name" : "ram",
  "age" :  "29",
  "gender" : "male"
},
{
  "name" : "Rahul",
  "age" : "32",
  "gender" : "male"
},
{
  "name" : "Rana",
  "age" : "23",
  "gender" : "male"}];
```

JSON objects in the array

- **We write the code like this to access the information from this:**

- array:document.writeln(students[0].age); //output would be: 29
- document.writeln(students[2].name); //output: Sophie

JSON Example for Sending Data

- When sending data to Server-side, the data has to be a string.
- We need to convert a JavaScript object into a string format with JSON.stringify().
- If you have data stored in a JavaScript object, you can convert the object into JSON, and send it to a server:

JSON Example for Sending Data

```
<body>
  <h2>Create JSON string from a JavaScript object.</h2>
  <p id="demo"></p>
 <script>
      var obj = { name: “Ram", age: 30, city: “Vadodara" };
      var myJSON = JSON.stringify(obj);
  document.getElementById("demo").innerHTML = myJSON;
  </script>
 </body>
```

JSON Example for Receiving Data

- When receiving data from Server-Side, the data is always a string.
- The data has to be parsed with JSON.parse (), making the data a JavaScript object.
- If you want to receive data in JSON format, you can convert it into a JavaScript object:

JSON Example for Receiving Data

```
<body>
   <h2>Create Object from JSON String</h2>
   <p id="demo"></p>
  <script>
      var txt = '{"name":“Ram", "age":30, "city":“Vadodara"}'
      var obj = JSON.parse(txt);
      document.getElementById("demo").innerHTML = obj.name + ", " +
obj.age;
  </script>
</body>
```

JSON Example for Storing Data

- While storing data, it needs to be in a fixed format, and text is always one of the legal formats where you want to store it.

- JSON helps to store JavaScript objects as text.

JSON Example for Storing Data

```
<body>
     <h2>Store and retrieve data from local storage.</h2>
     <p id="demo"></p>
      <script>
   var myObj, myJSON, text, obj;
 //Storing data:
     myObj = { name: “Ram", age: 31, city: “Vadodara" };
     myJSON = JSON.stringify(myObj);
     localStorage.setItem("testJSON", myJSON);
 //Retrieving data:
     text = localStorage.getItem("testJSON");
     obj = JSON.parse(text);
     document.getElementById("demo").innerHTML = obj.name;
   </script></body>
```

Use of JSON

- Helps you to forward data from a server
- JSON format helps in all types of structured data being transmitted and serialized.
- Enables you to make asynchronous data calls without needing to refresh a page
- It helps you to transfer data between web applications and a server.
- It is widely used for application based on JavaScript, which includes the browser extension and websites.
- Using JSON, you can forward data between the server and the web application.
- With modern programming languages, we can utilize JSON.

JSON vs. XML

JSON	XML
JSON object has a type	XML data is typeless
JSON types: string, number, array, Boolean	All XML data should be the string
Data is readily accessible as JSON objects	XML data needs to be parsed.
JSON files are more human-readable.	XML files are less human-readable.
Most browsers support JSON.	Cross-browser XML parsing can be tricky
JSON has no display capabilities.	XML provides a capability to display data because it is a markup language.
Retrieving value is easy	It is difficult to get value back
Supported by many Ajax toolkit	Unfulfilled with the Toolkit of Ajax
A automated way of deserializing/serializing JavaScript.	To serialize/deserialize from XML, developers must write JavaScript code

DIGITAL LEARNING CONTENT

Parul® University

www.paruluniversity.ac.in