AJAX is an acronym that stands for "**Asynchronous JavaScript and XML**".

It is a group of inter-related technologies like JavaScript, DOM, XML, HTML/XHTML, CSS,

It is used to communicate with the server without refreshing the web page and thus increasing the user experience and better performance.

## Synchronous vs Asynchronous

**Synchronous:** We can send one request at one time and have to wait for the response before send the second request. i.e. It blocks the client browser until operation completes.

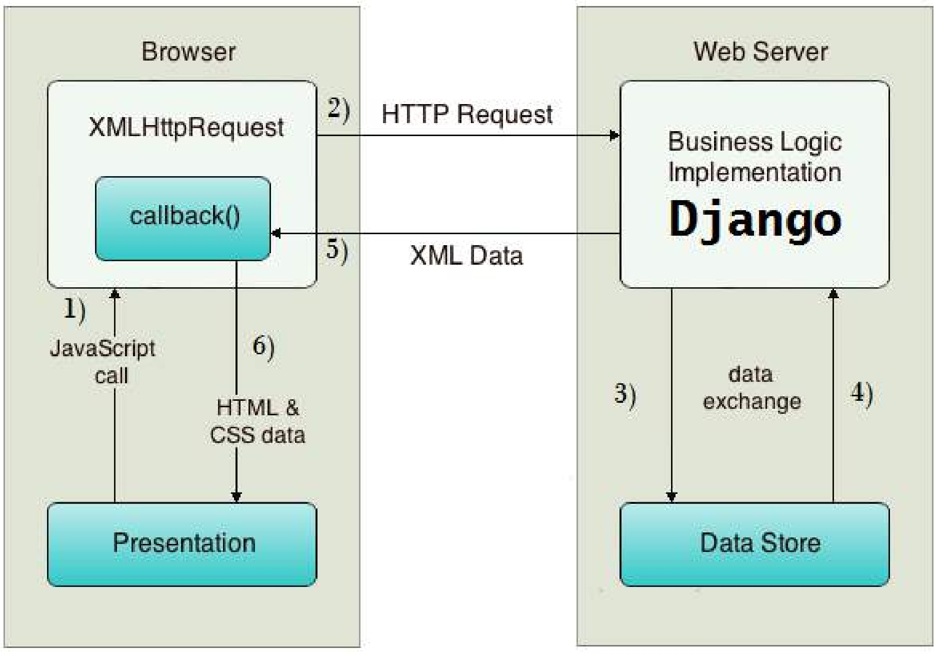
**Asynchronous:** We can send other request before getting the response of first request. i.e. It does not blocks the client browser.

## AJAX Technologies

As describe earlier, ajax is not a technology but group of inter-related technologies.

- **HTML / XHTML and CSS** : These technologies are used for displaying content and style. It is mainly used for presentation.
- **DOM** : It is used for dynamic display and interaction with data.
- **XML or JSON** : For carrying data to and from server. JSON (Javascript Object Notation) is a light weight data interchange language.
- **XMLHttpRequest** : For asynchronous communication between client and server.

## AJAX Work Flow



1. User sends a request from the UI and a javascript call goes to XMLHttpRequest object.
2. HTTP Request is sent to the server by XMLHttpRequest object.
3. Server interacts with the database using JSP, PHP, Servlet, ASP.net etc.
4. Data is retrieved.
5. Server sends XML data or JSON data to the XMLHttpRequest callback function.
6. HTML and CSS data is displayed on the browser.

Note: All modern browsers (Chrome, Firefox, IE7+, Edge, Safari Opera) have a built-in XMLHttpRequest object.

## Create an XMLHttpRequest Object

*Syntax :*   *variable* = new XMLHttpRequest();

## XMLHttpRequest Object Methods

| Method | Description |
|---|---|
| new XMLHttpRequest() | Creates a new XMLHttpRequest object |
| abort() | Cancels the current request |
| getAllResponseHeaders() | Returns header information |
| getResponseHeader() | Returns specific header information |
| open(*method,url,async,user,psw*) | Specifies the request<br><br>*method*: the request type GET or POST<br>*url*: the file location<br>*async*: true (asynchronous) or false (synchronous)<br>*user*: optional user name<br>*psw*: optional password |
| send() | Sends the request to the server<br>Used for GET requests |
| send(*string*) | Sends the request to the server.<br>Used for POST requests |
| setRequestHeader() | Adds a label/value pair to the header to be sent |

## XMLHttpRequest Object Properties

| Property | Description |
|---|---|
| onreadystatechange | Defines a function to be called when the readyState property changes |
| readyState | Holds the status of the XMLHttpRequest.<br>0: request not initialized<br>1: server connection established<br>2: request received<br>3: processing request<br>4: request finished and response is ready |
| responseText | Returns the response data as a string |
| responseXML | Returns the response data as XML data |
| status | Returns the status-number of a request<br>200: "OK"<br>403: "Forbidden"<br>404: "Not Found"<br>For a complete list go to the Http Messages Reference |
| statusText | Returns the status-text (e.g. "OK" or "Not Found") |

## GET request Example Using Javascript

### HTML

```
<html lang="en">
<head>
   <script>
   function display()
   {
     var xhttp = new XMLHttpRequest();
     xhttp.onreadystatechange = show;
     xhttp.open("GET","http://127.0.0.1:8000/main",true);
     xhttp.send()
      function show()   // callback function
      {
         if(xhttp.readyState==4)
         {
             var v = xhttp.responseText;
             alert(v)
         }
      }
   }
</script>
</head>
<body bgcolor="yellow">
<input type="text" onkeyup="display()">
</body>
</html>
```

## Urls.py

```
path('main/',views.one),
```

## Views.py

```
from django.http import JsonResponse

def one(request):
        d1 = {"idno": 101, "name": "Ravi"}
        return JsonResponse(d1)
```

## POST request Example Using Javascripts

## js/sample.js

```
function display()
{
   alert("pp");
   var name = document.getElementById("i2").value;
   var param = 'cname='+name
   var req = new XMLHttpRequest();

   req.onreadystatechange = show;
   req.open("POST","http://127.0.0.1:8000/two/",true);
   req.setRequestHeader('Content-Type', 'application/x-www-form-
urlencoded');
   req.send(param);
 alert("pp");
   function show()
   {
        if(req.readyState == 4 && req.status == 200)
        {
           var v = req.responseText;
           alert(v)
        }   }  }
```

## HTML

```
{% load static %}
<html>
<head>
   <script type="text/javascript" src="{% static 'js/sample.js'
%}"></script>
</head>
<body>
<h1>POST</h1>
<input type="text" placeholder="Name" required name="t1"
id="i2" onblur="display()"><br><br>
<input type="number" placeholder="Age" required name="t2">

</body>
</html>
```

## Urls.py

```
        path('two/',views.showTwo),
```

## Views.py

```
        from django.views.decorators.csrf import csrf_exempt
        from django.http import JsonResponse
```

```
@csrf_exempt
def showTwo(request):
    name = request.POST["cname"]
    print(name)

    data = {"k1": name}
    return JsonResponse(data)
```

## Reading JSON in Javascript

```
var v = xhttp.responseText;
var j = JSON.parse(v);
alert(j.idno)
alert(j.name)
```

jQuery

write less, do more.

jQuery is a fast, small, and feature-rich JavaScript library.

It makes things like HTML document manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a huge number of browsers.

With a combination of flexibility and extensibility, jQuery has changed the way that millions of people write JavaScript.

You can download the jQuery file from jquery.com or use the absolute URL of jQuery file.

Get the Link From : https://developers.google.com/speed/libraries

To load a hosted library, copy and paste the HTML snippet for that library (shown below) in your web page.

For instance, to load jQuery, embed the

**<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"> </script>**

snippet in your web page.

1. **<script** type="text/javascript" **>**
2. $(document).ready(function() {
3. $("p").css("background-color", "yellow");
4. });
5. **</script>**

1. **<body>**
2. **<p>**Python.**</p>**
3. **<p>**Django**</p>**
4. **<p>**Rest-API**</p>**
5. **</body>**

## The $() factory function

Every jQuery selector start with this sign $().

This sign is known as the factory function. It uses the three basic building blocks while selecting an element in a given document.

| No. | Selector | Example |
|---|---|---|
| 1) | Tag Name: | $('p') selects all paragraphs 'p' in the document. |
| 2) | Tag ID: | $('#real-id') selects a specific element in the document. |
| 3) | Tag Class: | $('real-class') selects all elements in the document that have a class. |
| class,.class | $(".primary,.secondary") | It will select all elements with the class "primary" or "secondary" |
| el1,el2,el3 | $("h1,div,p") | It will select all h1, div, and p elements. |
| :first | $("p:first") | This will select the first p element |
| :last | $("p:last") | This will select he last p element |
| :even | $("tr:even") | This will select all even tr |

| | | |
|---|---|---|
| | | elements |
| :odd | $("tr:odd") | This will select all odd tr elements |
| :first-child | $("p:first-child") | It will select all p elements that are the first child of their parent |
| :first-of-type | $("p:first-of-type") | It will select all p elements that are the first p element of their parent |
| :last-child | $("p:last-child") | It will select all p elements that are the last child of their parent |
| :last-of-type | $("p:last-of-type") | It will select all p elements that are the last p element of their parent |
| :nth-child(n) | $("p:nth-child(2)") | This will select all p elements that are the 2nd child of their parent |
| :nth-last-child(n) | $("p:nth-last-child(2)") | This will select all p elements that are the 2nd child of their parent, counting from the last |

| | | |
|---|---|---|
| | | child |
| :nth-of-type(n) | $("p:nth-of-type(2)") | It will select all p elements that are the 2nd p element of their parent |
| :nth-last-of-type(n) | $("p:nth-last-of-type(2)") | This will select all p elements that are the 2nd p element of their parent, counting from the last child |
| :only-child | $("p:only-child") | It will select all p elements that are the only child of their parent |
| :only-of-type | $("p:only-of-type") | It will select all p elements that are the only child, of its type, of their parent |
| parent > child | $("div > p") | It will select all p elements that are a direct child of a div element |
| parent descendant | $("div p") | It will select all p elements that are descendants of a div element |

| | | |
|---|---|---|
| element + next | $("div + p") | It selects the p element that are next to each div elements |
| element ~ siblings | $("div ~ p") | It selects all p elements that are siblings of a div element |
| :eq(index) | $("ul li:eq(3)") | It will select the fourth element in a list (index starts at 0) |
| :gt(no) | $("ul li:gt(3)") | Select the list elements with an index greater than 3 |
| :lt(no) | $("ul li:lt(3)") | Select the list elements with an index less than 3 |
| :not(selector) | $("input:not(:empty)") | Select all input elements that are not empty |
| :header | $(":header") | Select all header elements h1, h2 ... |
| :animated | $(":animated") | Select all animated elements |
| :focus | $(":focus") | Select the element that |

| | | |
|---|---|---|
| | | currently has focus |
| :contains(text) | $(":contains('Hello')") | Select all elements which contains the text "Hello" |
| :has(selector) | $("div:has(p)") | Select all div elements that have a p element |
| :empty | $(":empty") | Select all elements that are empty |
| :parent | $(":parent") | Select all elements that are a parent of another element |
| :hidden | $("p:hidden") | Select all hidden p elements |
| :visible | $("table:visible") | Select all visible tables |
| :root | $(":root") | It will select the document's root element |
| :lang(language) | $("p:lang(de)") | Select all p elements with a lang attribute value starting with "de" |
| [attribute] | $("[href]") | Select all elements with a href attribute |

| | | |
|---|---|---|
| [attribute=value] | $("[href='default.htm']") | Select all elements with a href attribute value equal to "default.htm" |
| [attribute!=value] | $("[href!='default.htm']") | It will select all elements with a href attribute value not equal to "default.htm" |
| [attribute$=value] | $("[href$='.jpg']") | It will select all elements with a href attribute value ending with ".jpg" |
| [attribute\|=value] | $("[title\|='Tomorrow']") | Select all elements with a title attribute value equal to 'Tomorrow', or starting with 'Tomorrow' followed by a hyphen |
| [attribute^=value] | $("[title^='Tom']") | Select all elements with a title attribute value starting with "Tom" |
| [attribute~=value] | $("[title~='hello']") | Select all elements with a title attribute value containing the specific word "hello" |
| [attribute*=value] | $("[title*='hello']") | Select all elements with a |

| | | |
|---|---|---|
| | | title attribute value containing the word "hello" |
| :input | $(":input") | It will select all input elements |
| :text | $(":text") | It will select all input elements with type="text" |
| :password | $(":password") | It will select all input elements with type="password" |
| :radio | $(":radio") | It will select all input elements with type="radio" |
| :checkbox | $(":checkbox") | Itwill select all input elements with type="checkbox" |
| :submit | $(":submit") | It will select all input elements with type="submit" |
| :reset | $(":reset") | It will select all input elements with |

| | | |
|---|---|---|
| | | type="reset" |
| :button | $(":button") | It will select all input elements with type="button" |
| :image | $(":image") | It will select all input elements with type="image" |
| :file | $(":file") | It will select all input elements with type="file" |
| :enabled | $(":enabled") | Select all enabled input elements |
| :disabled | $(":disabled") | It will select all disabled input elements |
| :selected | $(":selected") | It will select all selected input elements |
| :checked | $(":checked") | It will select all checked input elements |

## jQuery hide()

The jQuery hide() method is used to hide the selected elements.

**Syntax**: $(selector).hide();

### Example

```
<head>
   <script src =
"https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"><
/script>
   <script type="text/javascript">
     $(document).ready(function ()
     {
      $("#b1").click(function () {
         $("h1").hide()
      })
     })
   </script>
</head>

<body>
  <h1>Click on Button to Hide</h1>
   <br>
   <button id="b1">Click</button>
</body>
```

## jQuery show()

The jQuery show() method is used to show the selected elements.

**Syntax**: $(selector).show();

**Example**

```
<head>
      <script src =
"https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"><
/script>
      <script type="text/javascript">
        $(document).ready(function ()
        {
         $("#b1").click(function () {
             $("h1").hide()
         });
           $("#b2").click(function () {
              $("h1").show()
           })
        })
      </script>
   </head>
<body>
  <h1>Click on Button to Hide</h1>
   <br>
   <button id="b1">Hide</button>
   <button id="b2">Show</button>
</body>
```

## jQuery toggle()

The jQuery toggle() method is used to hide and show element.

**Syntax**: $(selector).toggle();

**Example**

```
<head>
      <script src =
"https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"
></script>
      <script type="text/javascript">
        $(document).ready(function () {
           $("#b1").click(function () {
              $("h1").toggle()
           })
        })
      </script>
   </head>
<body>
  <h1>Click on Button to Hide and show</h1>
   <br>
   <button id="b1">Toggle</button>
</body>
```

## jQuery fadeIn()

jQuery fadeIn() method is used to fade in the element.

**Syntax**: $(selector).fadein();

**Example**

```
<head>
      <script src =
"https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"><
/script>
      <script type="text/javascript">
        $(document).ready(function () {
            $("img").hide()
            $("#b1").click(function () {
                $("img").fadeIn(30000)
            })
        })
      </script>
   </head>
<body>
   <img src="1.jpg">
    <br>
    <button id="b1">Show Image</button>
</body>
```

## jQuery fadeOut()

The jQuery fadeOut() method is used to fade out the element.

**Syntax**: $(selector).fadeOut();

## jQuery fadeToggle()

jQuery fadeToggle() method is used to fadein and fadeout element.

**Syntax**: $(selector).fadeToggle();

## jQuery slideDown()

jQuery slideDown() method is used to slide down an element.

**Syntax**: $(selector).slideDown(speed);

## jQuery slideUp()

jQuery slideDown() method is used to slide up an element.

**Syntax**: $(selector).slideUp(speed);

## jQuery slideToggle()

jQuery slideToggle () method is used to slideUp and slideDown element.

**Syntax**: $(selector).slideToggle(speed);

## jQuery animate()

The jQuery animate() method animate an element.

**Syntax**: $(selector).animate({params}, speed);

**Example :** $("div").animate({left: '450px'},10000);

## jQuery serialize()

jQuery serialize() method is used to create a text string in standard URL-encoded notation.

It is used in form controls like <input>, <textarea>, <select> etc.

It serializes the form values so that its serialized values can be used in the URL query string while making an AJAX request.

**Syntax**: $ (selector).serialize()

**Example :**

```
<head>
      <script src =
"https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"><
/script>
      <script type="text/javascript">
        $(document).ready(function () {
           $("button").click(function () {
              var data = $("form").serialize();
              alert(data)
           })
        })
      </script>
   </head>
<body>
  <form action="">
      <input type="text" placeholder="Name"
name="name"><br><br>
```

```
        <input type="number" placeholder="age"
name="age"><br><br>
        Male <input type="radio" name="gender"
value="male"><br><br>
        Female <input type="radio" name="gender"
value="female"><br><br>
        <button type="submit" >Serialize</button>
    </form>
</body>
```

# jQuery Events

jQuery events are the actions that can be done on a web application.

These are some examples of events.

- A mouse click
- An HTML form submission
- A web page loading
- A keystroke on the keyboard
- Scrolling of the web page etc.

| **Mouse Events** | **Keyboard Events** |
|---|---|
| ○ **click** | ○ **keyup** |
| ○ **dblclick** | ○ **keydown** |
| ○ **mouseenter** | ○ **keypress** |
| ○ **mouseleave** | |

| **Form Events** | **Document/Window Events** |
|---|---|
| ○ **submit** | ○ **load** |

- change
- blur
- focus
- unload
- scroll
- resize

## Jquery Ajax Call Example

```
$("#i1").blur(function () {
   $.ajax(
      {
         url :'http://127.0.0.1:8000/check',
         type : 'get',
         data : {"pid":$("#i1").val()},
         success : function (data)
         {
            console.log(data);
            alert(data.k1)

         },
         failure : function (data) {
             alert('Got an error dude');
         }
      }
   )
})
```