

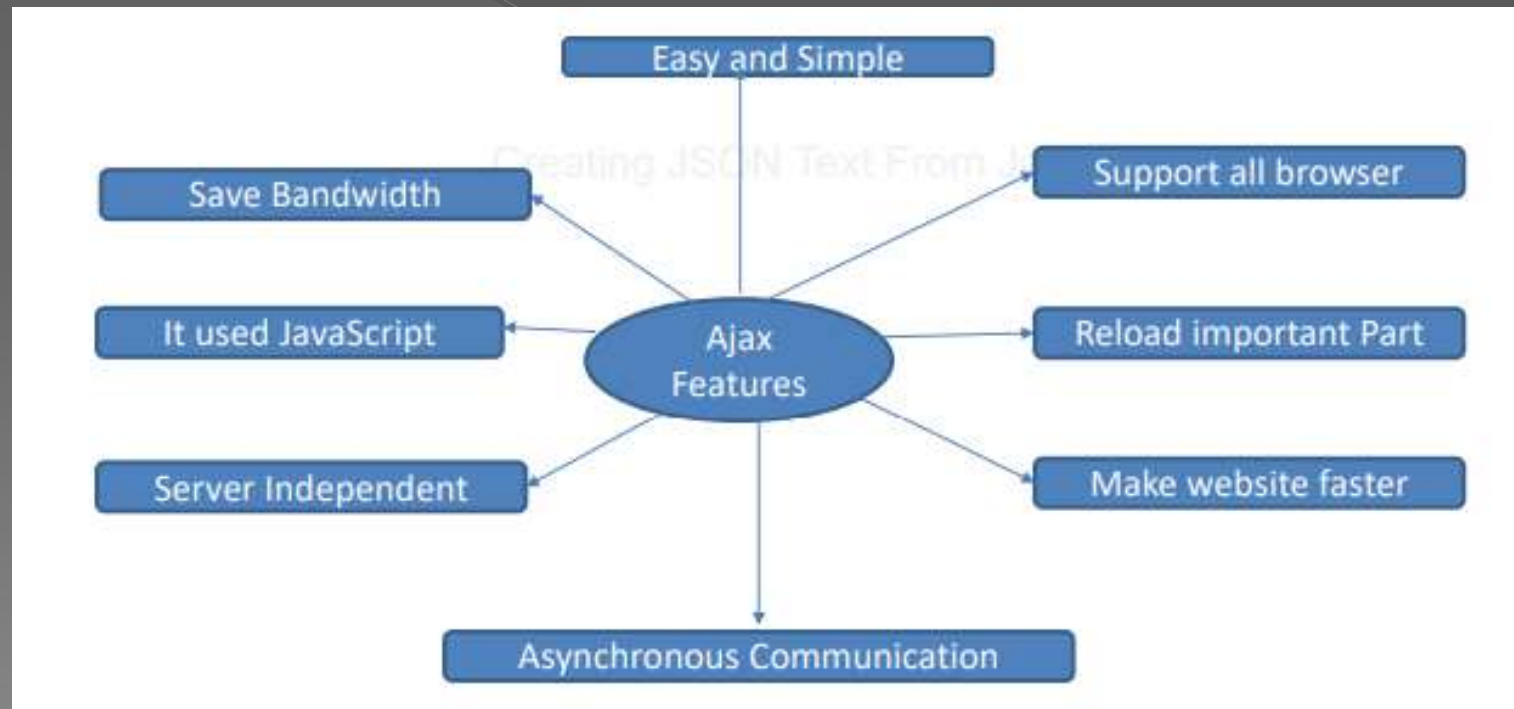
AJAX

Harshita Maheshwari

AJAX

Ajax stands for Asynchronous JavaScript and XML

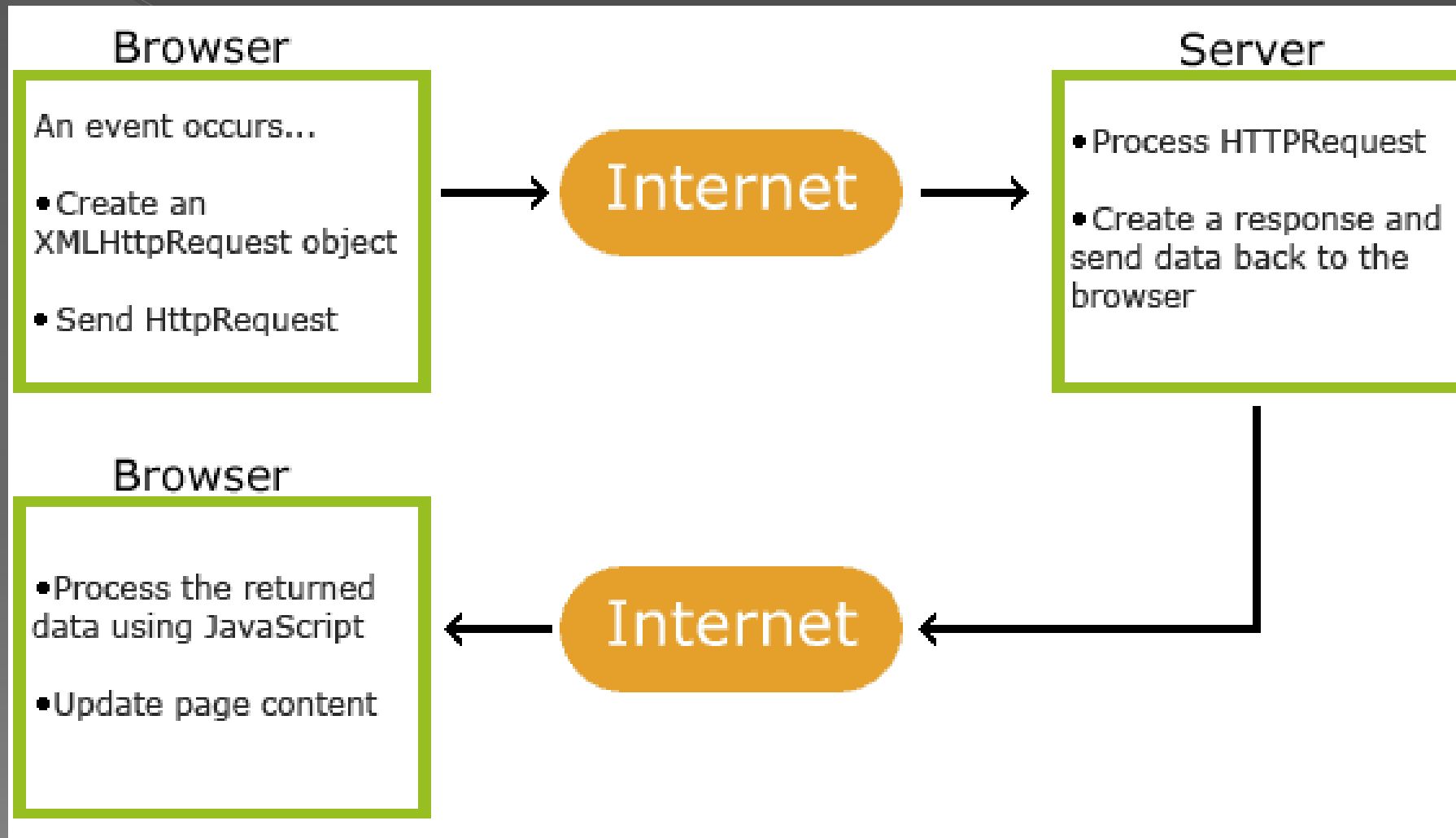
AJAX was made popular in 2005 by Google, with Google Suggest



Where should use Ajax?

- Form Validation
- Sort or Filter
- Vote or Rating
- Chat Websites
- Blog Comments
- Captcha

AJAX Process Cycle



Step-1

Create a XMLHttpRequest Object

All modern browsers support the XMLHttpRequest object. The XMLHttpRequest object can be used to exchange data with a server behind the scenes. This means that it is possible to update parts of a web page without reloading the whole webpage.

Syntax:

```
Variable=new XMLHttpRequest();  
var xhttp= new XMLHttpRequest()
```

Step-2

Making a request to the server

The XMLHttpRequest object is used to exchange data with a server.

To send a request to a server, we use the open() and send() methods of the XMLHttpRequest object.

```
Xhttp.open("GET","ajax_info.txt",true);
```

Method used in request
it can be get or post

This is the url of the
file that we want to load
into our web page

true signifies that we are making
asynchronous request if we set it to false
then it will not to be ajax request

OnReadyStateChange Property

The readyState property holds the status of the XMLHttpRequest.

The onreadystatechange property defines a function to be executed when the readyState changes.

The status property and statusText property holds the status of the XMLHttpRequest object.

readyState: Holds the status of XMLHttpRequest

0: request not initialized

1: server connection established

2: request received

3: processing request

4: request finished and response ready

Status: 200 "OK"

403 "forbidden"

404 "Page not found"


```
xhttp.onreadystatechange=function(){  
If(this.readyState==4 && this.status==200){  
Document.getElementById("demo").innerHTML=this.responseText;  
}  
};
```

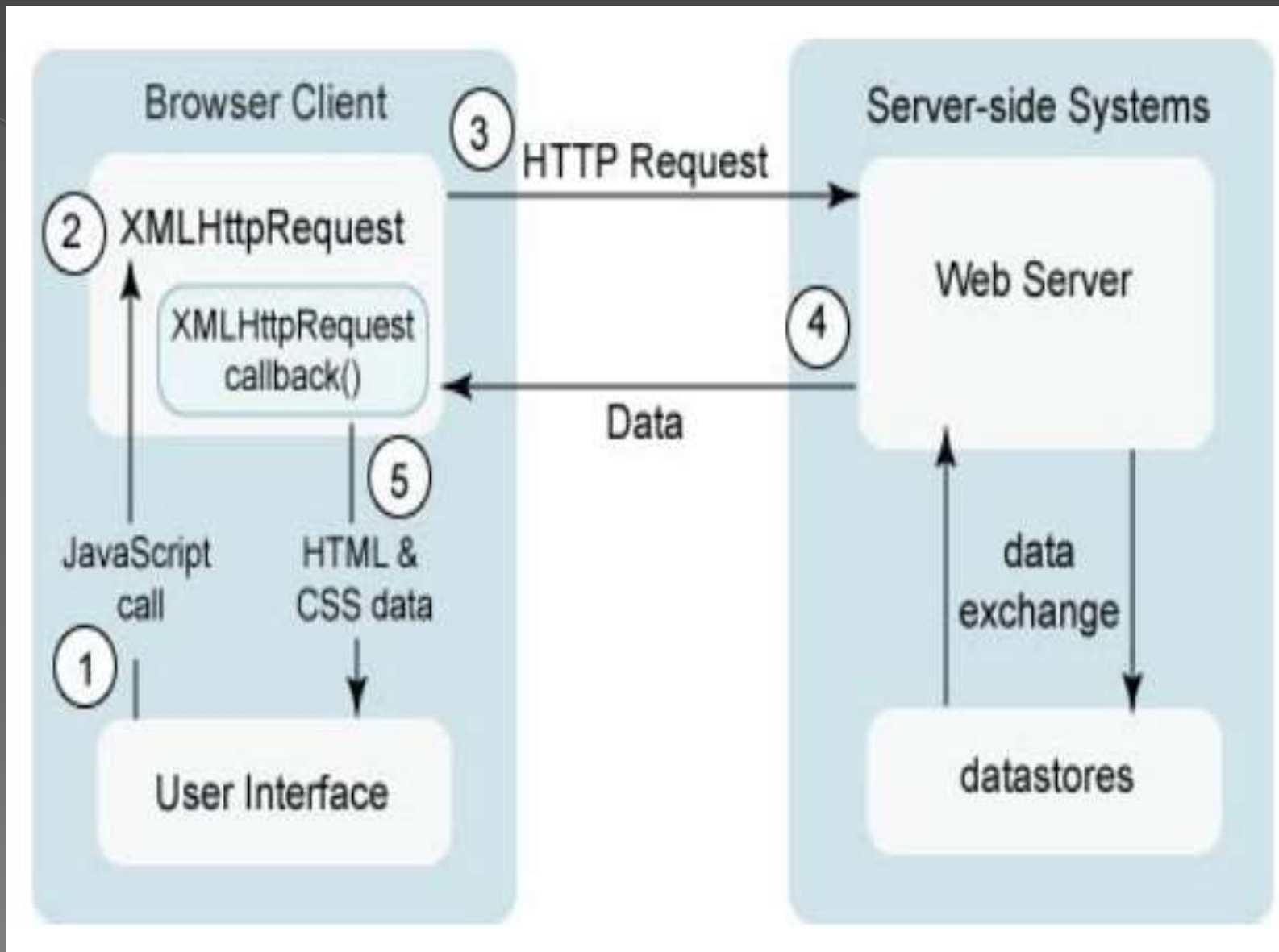
We are executing this anonymous function when this readystatechange event trigger or readystate value changes if the value of readystate is 4 and status is 200, we are just modifying the inner html of id demo with the response that is returned from the server.

Send the request



```
xhttp.send()
```

```
var xhttp=new XMLHttpRequest();
xhttp.onreadystatechange= ()=>{
if(this.readyState==4 && this.status==200){
document.getElementById("demo").innerHTML=this.responseText;
}
};
xhttp.open("GET","ajax_info.txt",true);
xhttp.send();
```



Handling AJAX Requests using jQuery

jQuery library provides you with various methods, known as jQuery AJAX methods, that allow you to make a call to the AJAX code. These methods allow you to perform various tasks.

The following list depicts the jQuery AJAX methods:

- load()
- get()
- post()
- ajax()

jQuery - AJAX load() Method

The load() method is used to load or fetch data from a Web server into a selected HTML element.

Syntax :

```
$(selector).load(URL[,data][,complete])
```

URL: Is used to specify the URL from where you want to load the data. The URL can be used to refer to any resource, such as text file or html file.

data: Is used to pass an object of a key/value pair along with the request to the server.

complete: Is used to refer to a callback method that will be executed after the successful execution of load() method.

```
<script>
$(document).ready(function(){
$("button").click(function(){
$("#div1").load("demo.txt");
});
});
</script>
```

jQuery \$.get() Method

The `get()` method is used to load data from a Web server using the HTTP GET request.

Syntax: `$.get(url[,data][, callback],[datatype])`

`datatype`: Is used to specify the type of data, such as text, JSON, or XML, that is expected in return from the server

```
$("button").click(function(){  
    $.get("demo.html", function(data, status){  
        alert("Data: " + data + "\nStatus: " + status);  
    });  
});
```


jQuery \$.post() Method

The post() method is:

- Similar to the get() method.
- Used to load data from a Web server using the HTTP POST request.
- Used when the requested is large in amount.
- Used to send the data in an encrypted format.

Syntax: \$.post(url[,data][, callback],[datatype])

```
$("#button").click(function(){
    $.post("demo_post.php",
    {
        name: "Infoway Technologies",
        city: "Pune"
    },
    function(data, status){
        alert("Data: " + data + "\nStatus: " + status);
    });
});
```

jQuery \$.ajax() Method

Syntax : \$.ajax(options)

options: Is an optional parameter that helps in configuring the AJAX calls by using key/value pairs.

Option	Description
<i>async</i>	<i>A boolean value that indicates whether to execute the request asynchronously.</i>
<i>complete</i>	<i>A callback function that executes whenever the request finishes.</i>
<i>datatype</i>	<i>A string defining the type of data, such as XML, HTML, JSON, or script, that is expected back from the server.</i>
<i>success</i>	<i>A callback function that is executed if the request succeeds.</i>
<i>type</i>	<i>A string defining the HTTP method to be used for the request (GET or POST).</i>
<i>URL</i>	<i>A required option that refers to the string containing the URL to which the request</i>

```
$("#button").click(function(){  
    $.ajax({  
        url: "demo.txt",  
        type: 'GET',  
        success: function(result){  
            $("#div1").html(result);  
        }  
    });  
});
```

jQuery \$.getJSON() Method

The `getJSON()` method is used to get JSON data using an AJAX HTTP GET request.

Syntax: `$.getJSON(url,data,success(data,status,xhr))`

`success(data,status,xhr)`: Optional. Specifies the function to run if the request succeeds
`data` - contains the data returned from the server.

`status` - contains a string containing request status ("success", "notmodified", "error", "timeout", or "parsererror").

`xhr` - contains the XMLHttpRequest object.

```
$("#button").click(function(){
    $.getJSON("demo.js", function(result){
        $.each(result, function(i, field){
            $("#div").append(field + " ");
        });
    });
});
```

jQuery \$.get() Method

The `get()` method is used to load data from a Web server using the HTTP GET request.

Syntax: `$(selector).get(url[,data][, callback],[datatype])`

`datatype`: Is used to specify the type of data, such as text, JSON, or XML, that is expected in return from the server

Thank you !!