

# CACS 205: Script Language



# Ajax.

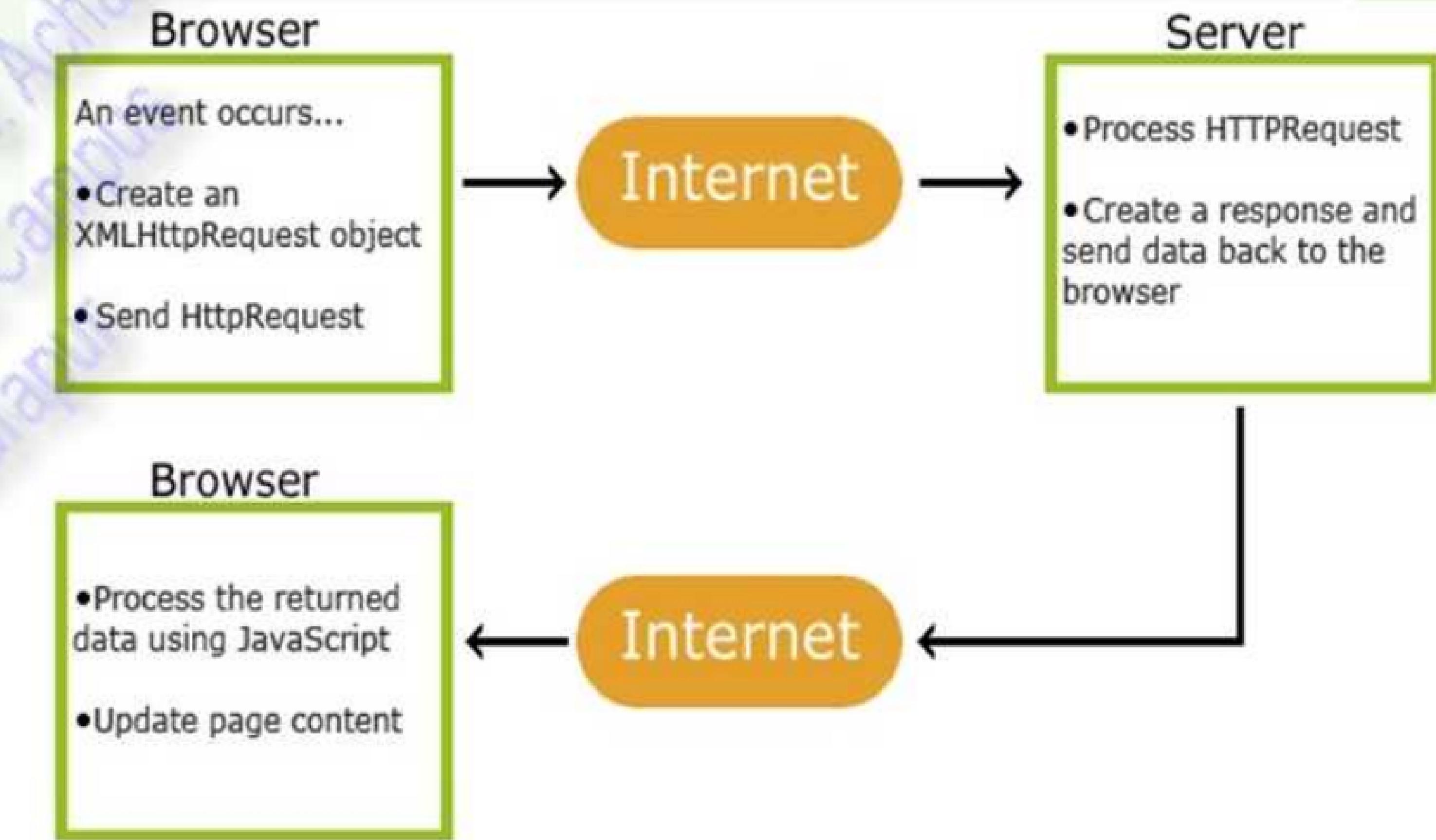
- Introduction of Ajax.
- Working mechanism of ajax in web application.
- Advantages and Disadvantages of Ajax.
- Ajax working with PHP
- Ajax and XML
- Ajax with PHP along with mySql

## What is Ajax?

- Ajax is an acronym for Asynchronous JavaScript and XML. It is used to communicate with the server without refreshing the web page and thus increasing the user experience and better performance. Ajax is just a means of loading data from the server and selectively updating parts of a web page without reloading the whole page.
- Basically, what Ajax does is make use of the browser's built-in **XMLHttpRequest** (XHR) object to send and receive information to and from a web server asynchronously, in the background, without blocking the page or interfering with the user's experience.
- Ajax is not a new technology, in fact, Ajax is not even really a technology at all. Ajax is just a term to describe the process of exchanging data from a web server asynchronously through JavaScript, without refreshing the page.
- Ajax has become so popular that you hardly find an application that doesn't use Ajax to some extent. The example of some large-scale Ajax-driven online applications are: Gmail, Google Maps, Google Docs, YouTube, Facebook, Flickr, and so many other applications.
- Don't get confused by the term X (i.e. XML) in AJAX. It is only there for historical reasons. Other data exchange format such as JSON, HTML, or plain text can be used instead of XML.

# How ajax works?

1. An event occurs in a web page (the page is loaded, a button is clicked)
2. An XMLHttpRequest object is created by JavaScript
3. The XMLHttpRequest object sends a request to a web server
4. The server processes the request
5. The server sends a response back to the web page
6. The response is read by JavaScript
7. Proper action (like page update) is performed by JavaScript



# Advantage and Disadvantage of ajax?

## Advantages:

1. Speed is enhanced as there is no need to reload the page again.
2. AJAX make asynchronous calls to a web server, this means client browsers avoid waiting for all the data to arrive before starting of rendering.
3. Form validation can be done successfully through it.
4. Bandwidth utilization – It saves memory when the data is fetched from the same page.
5. More interactive.

## Disadvantages:

1. Ajax is dependent on JavaScript. If there is some JavaScript problem with the browser or in the OS, Ajax will not support.
2. Ajax can be problematic in Search engines as it uses JavaScript for most of its parts.
3. Source code written in AJAX is easily human readable. There will be some security issues in Ajax.
4. Debugging is difficult.
5. Problem with browser back button when using AJAX enabled pages.

## Basic Syantax of Ajax.

- Basic Syntax: The syntax of creating the object is given below

```
req = new XMLHttpRequest();
```

- There are two types of methods open() and send(). Uses of these methods explained below.

```
req.open("GET", "abc.php", true);  
req.send();
```

- The GET parameter is as usual one of two types of methods to send the request. Use POST as well depending upon whether send the data through POST or GET method.
- The second parameter being the name of the file which actually handles the requests and processes them. The third parameter is true, it tells that whether the requests are processed asynchronously or synchronously. It is by default true which means that requests are asynchronous.
- The open() method prepares the request before sending it to the server. The send method is used to send the request to the server.

## Basic Syntax of Ajax.

```
req.open("GET", "abc.php?x=25&y=100", true);  
req.send();
```

- In the above lines of code, the specified query in the form of URL followed by ? which is further followed by the name of the variable then = and then the corresponding value. If sending two or more variables use ampersand(&) sign between the two variables. The above method as shown applies for GET request. Sending the data through the POST, then send it in the send method as shown below.

```
req.send("name=ankit&marks=99");
```

- The XMLHttpRequest object has a property called as readyState whose value changes in the complete request-response journey i.e when a request is prepared, sent, resolves, processed and when the response comes. That's why it is called onreadystatechange.
- The onreadystatechange stores a function (or the name of the function) to be called automatically each time the readyState property changes.
- The readyState holds different values ranging from 0 to 4.

## Basic Syntax of Ajax

- The readyState holds different values ranging from 0 to 4.
  - 0. request not initialized(UNSENT)
  - 1. server connection established(OPENED)
  - 2. request received(HEADERS\_RECEIVED)
  - 3. processing request(LOADING)
  - 4. request finished and response is ready(DONE)
- XMLHttpRequest also has a property named as status. The status has following values
  - 1. 200: "OK"
  - 2. 404: "Page not found"

Now remember it always that when readyState is 4 and status is 200, the response is ready.

```
<p id = "dis"></p>
req.onreadystatechange = function(){
if(req.readyState == 4 && req.status == 200){
    document.getElementById("dis").innerHTML = req.responseText;
}
}
```

# Basic Syntax of Ajax

The status property returns the numerical HTTP status code of the XMLHttpRequest's response.

200 — OK. The server successfully processed the request.

404 — Not Found. The server can't find the requested page.

503 — Service Unavailable. The server is temporarily unavailable.

responseText

get the response data as a string

responseXML

get the response data as XML data

```
if (window.XMLHttpRequest) {  
    // code for modern browsers  
    xmlhttp = new XMLHttpRequest();  
} else {  
    // code for old IE browsers  
    xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");  
}
```

# Ajax with PHP using get method

```
<!DOCTYPE html>
]<html lang="en">
]<head>
<title>JavaScript Ajax Demo</title>
]<script>
function displayFullName() {
    var request = new XMLHttpRequest();
    request.open("GET", "Ajax.php?fname=Ankit&lname=Siwakoti");
    request.onreadystatechange = function() {
        if(this.readyState === 4 && this.status === 200)
            document.getElementById("result").innerHTML = this.responseText;
    };
    request.send();
}
</script>
]</head>
]<body>
    <div id="result">
        <p>Content of the result DIV box will be replaced by the server response</p>
    </div>
    <button type="button" onclick="displayFullName()">Display Full Name</button>
]</body>
]</html>
```

Ajax.html

# Ajax with PHP using get method

```
]<?php  
if(isset($_GET["fname"]) && isset($_GET["lname"])) {  
    $fname = htmlspecialchars($_GET["fname"]);  
    $lname = htmlspecialchars($_GET["lname"]);  
    $fullname = $fname . " " . $lname;  
  
    echo "Hello, $fullname! Welcome to BCA Programme";  
} else {  
    echo "Hi there! Welcome to our website.";  
}  
?>
```

Ajax.php

# Ajax with PHP usine of Post method

```
<!DOCTYPE html>
<head>
<title>JavaScript Ajax POST Demo</title>
<script>
function postComment() {
    var request = new XMLHttpRequest();
    request.open("POST", "Ajax1.php");
    request.onreadystatechange = function() {
        if(this.readyState === 4 && this.status === 200)
            document.getElementById("result").innerHTML = this.responseText;
    };
    var myForm = document.getElementById("myForm");
    var formData = new FormData(myForm);
    request.send(formData);
}
</script>
</head>
<body>
    <form id="myForm">
        <label>Name:</label>
        <div><input type="text" name="name"></div>
        <br>
        <label>Comment:</label>
        <div><textarea name="comment"></textarea></div>
        <p><button type="button" onclick="postComment()">Post Comment</button></p>
    </form>
    <div id="result">
        <p>Content of the result DIV box will be replaced by the server response</p>
    </div>
</body>
</html>
```

Ajax1.html

# Ajax with PHP usine of Post method

Ajax1.php

```
<?php  
if($_SERVER["REQUEST_METHOD"] == "POST") {  
    $name = htmlspecialchars(trim($_POST["name"]));  
    $comment = htmlspecialchars(trim($_POST["comment"]));  
    if(!empty($name) && !empty($comment)) {  
        echo "<p>Hi, <b>$name</b>. Your comment has been received successfully.<p>";  
        echo "<p>Here's the comment that you've entered: <b>$comment</b></p>";  
    }  
    else  
        echo "<p>Please fill all the fields in the form!</p>";  
}  
else  
    echo "<p>Something went wrong. Please try again.</p>";  
?>
```

# Ajax use of xml

xml.html

```
<!DOCTYPE html>
<html>
<style>
table,th,td {
    border : 1px solid black;
    border-collapse: collapse;
}
th,td {
    padding: 5px;
}
</style>
<body>
<h2>The XMLHttpRequest Object</h2>
<button type="button" onclick="loadDoc()">Get Student Details</button>
<br><br>
<table id="demo"></table>
<script>
function loadDoc() {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200)
            myFunction(this);
    };
    xhttp.open("GET", "student.xml", true);
    xhttp.send();
}
</script>
```

```
function myFunction(xml) {
    var i;
    var xmlDoc = xml.responseXML;
    var table = "<tr><th>Title</th><th>Name</th></tr>";
    var x = xmlDoc.getElementsByTagName("bca");
    for (i = 0; i < x.length; i++) {
        table += "<tr><td>" +
        x[i].getElementsByTagName("title")[0].childNodes[0].nodeValue +
        "</td><td>" +
        x[i].getElementsByTagName("name")[0].childNodes[0].nodeValue +
        "</td></tr>";
    }
    document.getElementById("demo").innerHTML = table;
}
</script>
</body>
</html>
```

## The XMLHttpRequest Object

Get Student Details

Title	Name
Mr.	Ankit Siwakoti
Miss	Smirti Khanal
Miss	Poonam Tamang

# Ajax.

Student.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<student>
    <bca>
        <titile>Mr.</titile>
        <name>Ankit Siwakoti</name>
        <address>Birtamode</address>
        <semester>Fourth</semester>
        <bath>First</bath>
    </bca>

    <bca>
        <titile>Miss</titile>
        <name>Smirty Khanal</name>
        <address>Bhadrapur</address>
        <semester>Fourth</semester>
        <bath>First</bath>
    </bca>

    <bca>
        <titile>Miss</titile>
        <name>Poonam Tamang</name>
        <address>Bhadrapur</address>
        <semester>Second</semester>
        <bath>Second</bath>
    </bca>
</student>
```

# Ajax PHP and MySql

search.html

```
<!DOCTYPE html>
]<html lang="en">
]<head>
<meta charset="UTF-8">
]<script>

function postComment() {
    var request = new XMLHttpRequest();
    request.open("POST", "GetPhpAjaxInfo.php");
} request.onreadystatechange = function() {
} if(this.readyState === 4 && this.status === 200){
    var status = this.responseText;
    if (status=='Available')
        document.getElementById('error').style.color = 'green';
    else
        document.getElementById('error').style.color = 'red';
    document.getElementById("error").innerHTML =status;
}
};

var myForm = document.getElementById("myForm");
var formData = new FormData(myForm);
request.send(formData);
}

-</script>
-</head>
]<body>
]<form id="myForm">
    <input type="text" autocomplete="on" name="user" placeholder="User Name" onkeyup="postComment();"/>
-</form>
    <div id="error"></div>
-</body>
-</html>
```

# Ajax PHP and MySql

```
<?php  
if($_SERVER["REQUEST_METHOD"] == "POST") {  
    $name = htmlspecialchars(trim($_POST["user"]));  
    if(!empty($name)) {  
        $mysqli = new mysqli("localhost", "root", "", "bca");  
        if($mysqli === false)  
            die("ERROR: Could not connect. " . $mysqli->connect_error);  
        $sql = "select * from users where username='$name'";  
        if($result = $mysqli->query($sql)) {  
            if($result->num_rows > 0)  
                echo "Available";  
            else  
                echo "Not Available";  
            $result->free();  
        }  
    }  
}?  
?>
```

GetPhpAjaxInfo.php