

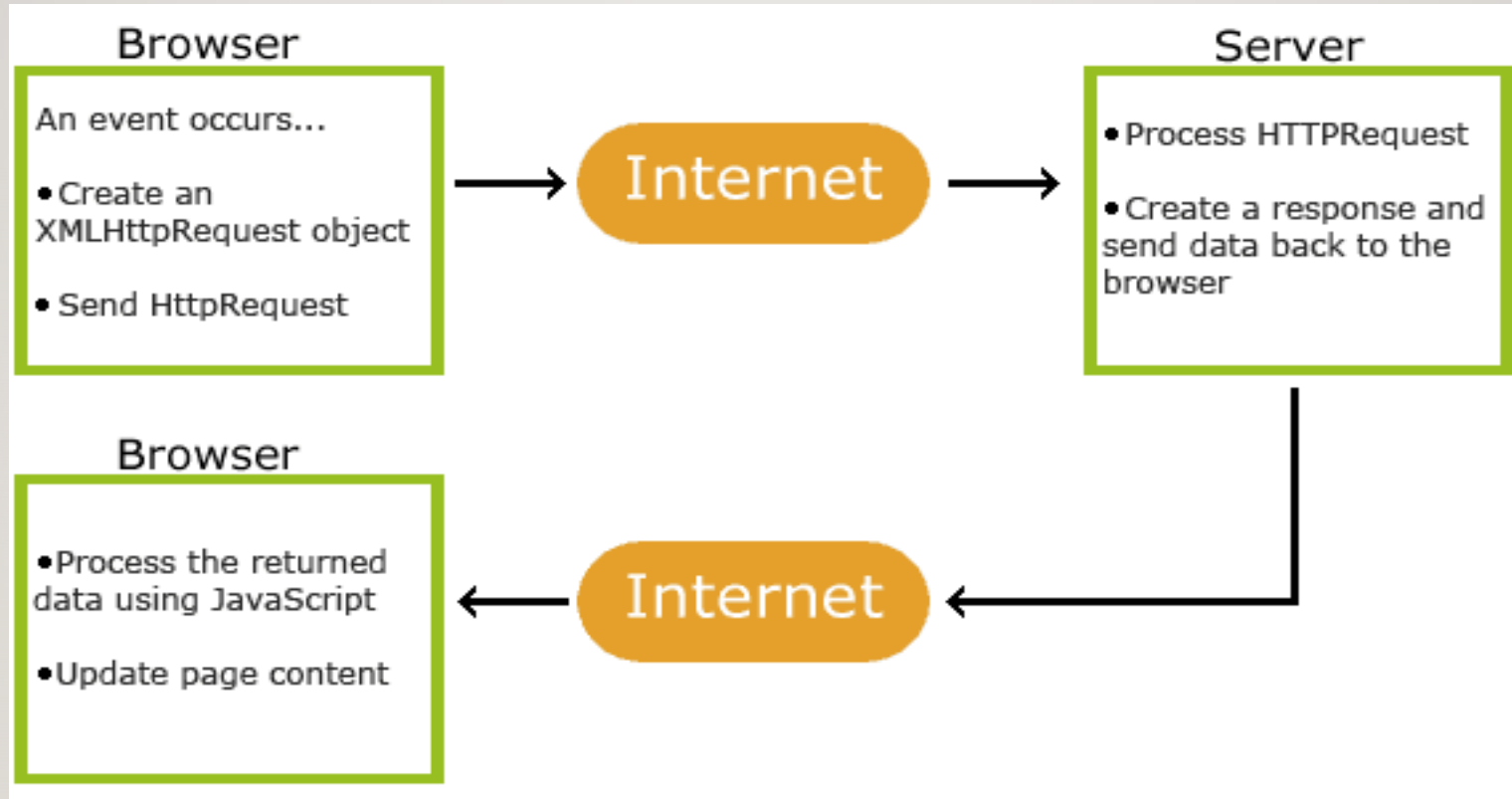
LECTURE – 6

- AJAX
- Node.js

AJAX – ASYNCHRONOUS JAVASCRIPT AND XML

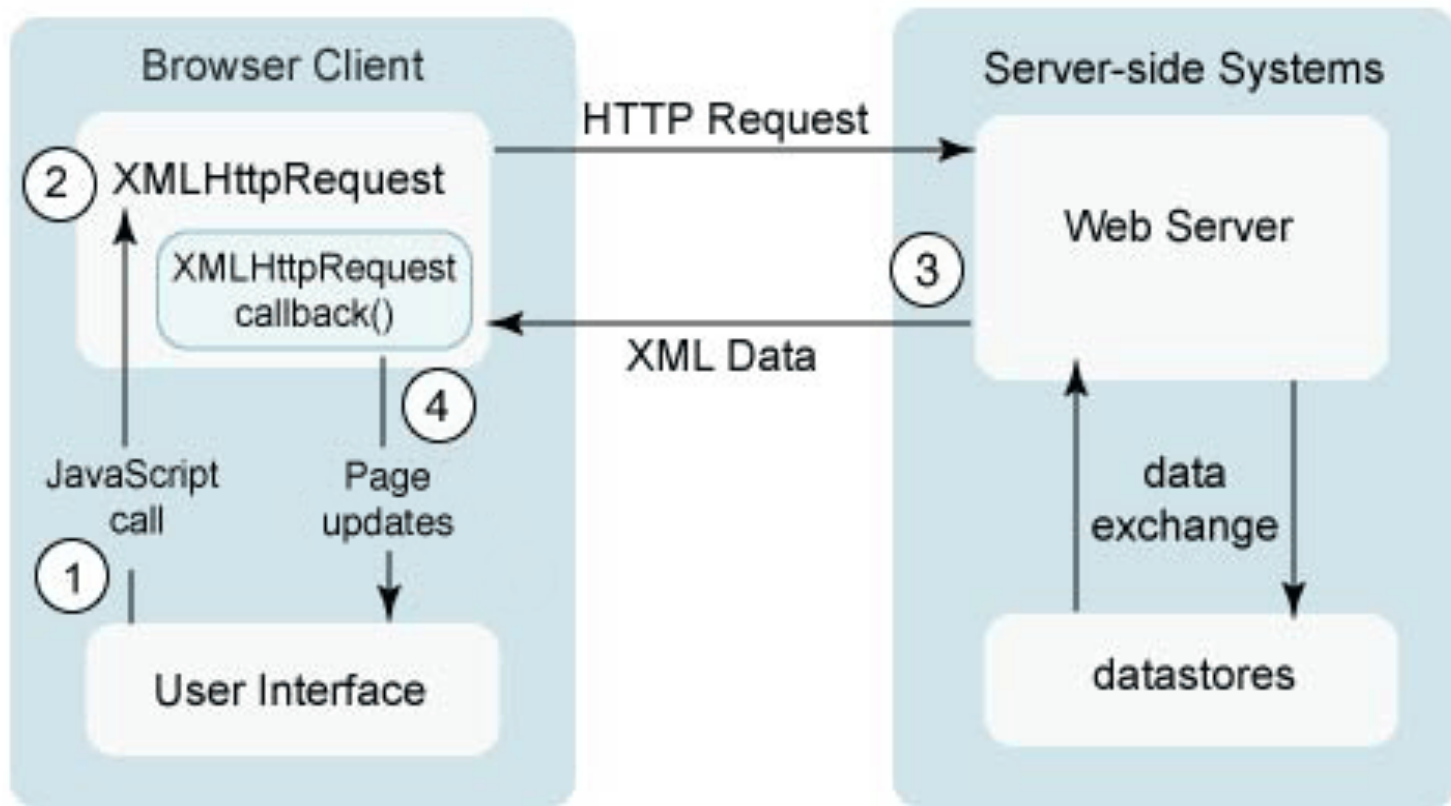
- Made popular by Google (with Google Suggest).
- NOT a new programming language
 - A new way to use existing standards.
- Based on JavaScript and HTTP requests.
- With AJAX, JavaScript communicates
 - Directly with the (web) server
 - using XMLHttpRequest object
 - To retrieve data as needed
 - Using Javascript events (e.g., keyPressed)
 - **WITHOUT** refreshing the page.

HOW DOES AJAX WORK ... CONTD.



Source: W3Schools

HOW DOES AJAX WORK



Source: SUN's JAVA web page

AJAX ... CONTD.

- Note: Data is typically sent in XML format
- XMLHttpRequest
 - The basic data structure interfacing the client with server.
 - Sends a request to a server (e.g., Google suggest server) on any events
 - Like “onKeyUp(..)” when the user types any character search key.
 - Receives data from the server
 - Updates the required fields with data received from server.

REQUESTS AND RESPONSES

- XMLHttpRequest
 - open(..) // open a connection to ...
 - setRequestHeader(..) // Set the request header
 - send(..) // Send the request
 - status // response 200 OK, or other values
 - readState // Store the state information
 - onreadystatechange // callback function for status change
 - responseText // response in Text
 - responseXML // response in XML

XMLHTTPREQUEST FUNCTIONS

- `open(method, url, async, user, psw)`
 - method: Get/Post
 - url: address of the server
 - sync – true or false (asynchronous call or not)
 - user – username (optional)
 - psw – Password (optional)
- `setRequestHeader()` // Sets label/value pairs in the header
- `send()` or `send(string)` // Sends the request to the server
- `abort()` // Cancel the current request
- `getResponseHeader()` // Get specific header information
- `getAllResponseHeaders()` // Get all headers' information

`XMLHttpRequest` OBJECT PROPERTIES

- `readyState` – Holds the status of `XMLHttpRequest` object
 - 0: Request Not initialized
 - 1: Server connection established
 - 2: Request received
 - 3: Processing request
 - 4: Request finished, response is ready
- `Status` – Returns the status number of a request
 - 200: OK
 - 403: Forbidden
 - 404: Not Found
 - ..Others
- `statusText` – Status text of a response (e.g., “OK”, “Not Found”, etc.)
- `responseText` – response data as a string
- `responseXML` – response as XML data

NODE.JS

- Open source cross-platform runtime environment
- Runs Javascript engine
- Node.js app runs in a single process
- Uses asynchronous non-blocking calls to resources
- Can handle many (thousands) of requests on a server.

WHAT CAN NODE.JS DO?

- Server side Javascript
- Serves content to browsers
- Generate dynamic page content
- Can create/open/read/write/delete/close files on server
- Can interact with databases on server
- Can handle forms, events, emails, etc.

ADVANTAGES OF NODE.JS

- Fast and makes good use of resources (CPU, memory, etc.)
- Can handle multiple requests simultaneously
- Open source
- Thousands of open source packages are available.

NODE.JS – HIGH LEVEL ARCHITECTURE

