JSON and AJAX

JSON

- ❑ JSON: A common way of storing and working (exchanging) with data in javaScript.
- ❑ Stands for JavaScript Object Notation
- ❑ JSON is a text written with JavaScript Object notation.
- ❑ Data must be a text when exchanging between browser and server.
- ❑ We can convert JavaScript Object into JSON and send to the server.
  - ❑ JSON.stringify(object);
- ❑ Also convert JSON received from server to javascript object.
  - ❑ JSON.parse(json);

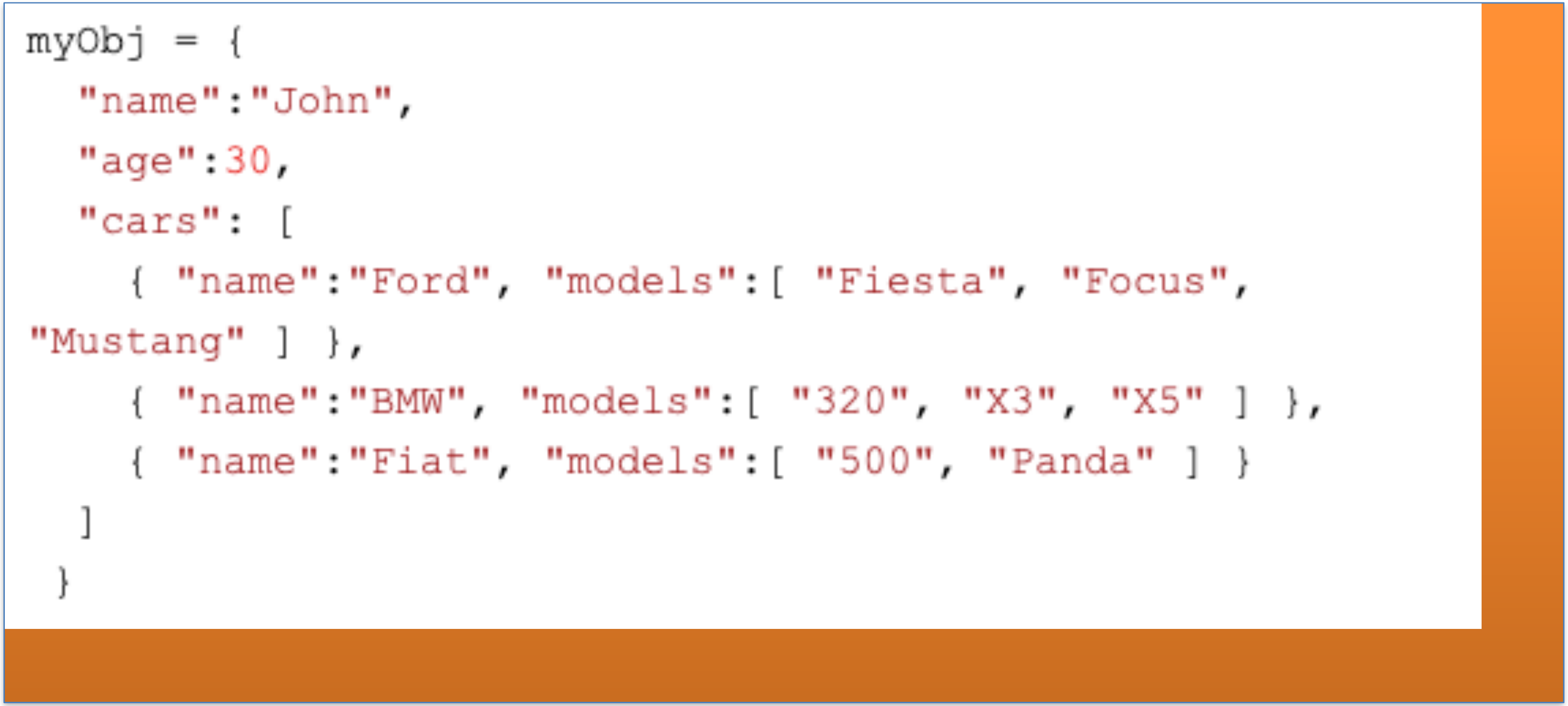
JSON Syntax

- Since JSON is derived from JavaScript, JSON uses syntax similar to JavaScript
  - Data is in name/value pairs
  - Data is separated by commas
  - Curly braces used to enclosed objects
  - Square brackets to hold arrays.
- Strings in JSON are identified by double quotes. JSON keys must also be enclosed by double quotes.

JSON Values

- JSON values must be of following types
    - String
    - Number
    - Object
    - Array
    - Boolean
    - Null
- Javascript function , date and undefined values cannot be used as JSON values.

JSON Example



```
myObj = {
  "name":"John",
  "age":30,
  "cars": [
    { "name":"Ford", "models":[ "Fiesta", "Focus",
"Mustang" ] },
    { "name":"BMW", "models":[ "320", "X3", "X5" ] },
    { "name":"Fiat", "models":[ "500", "Panda" ] }
  ]
 }
```

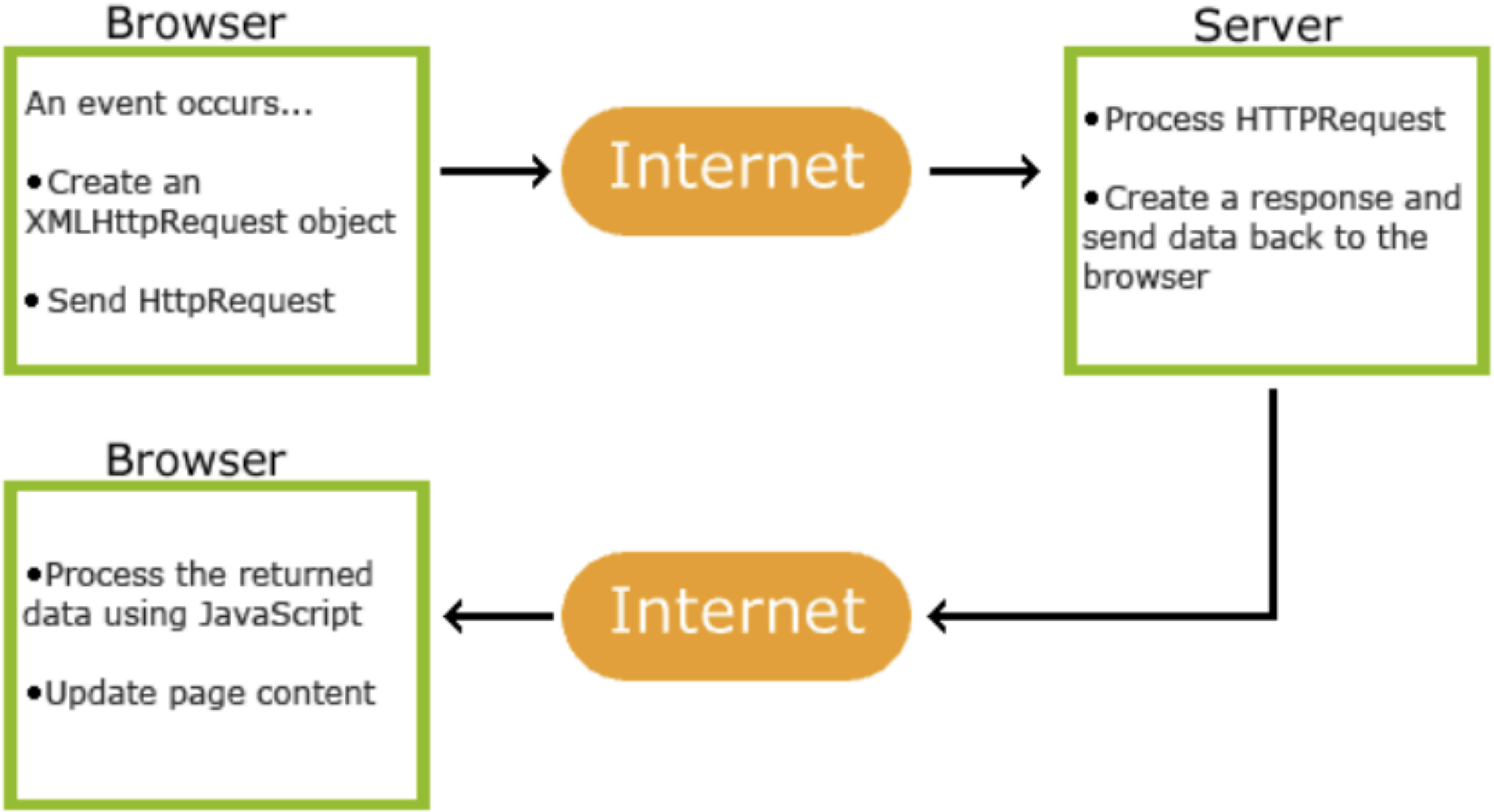
AJAX

- AJAX : Asynchronous JavaScript And XML

- AJAX uses combination of browser built in ***XMLHttpRequest*** object and JavaScript and HTML DOM.

- Allows web pages to be updated asynchronously by exchanging data with a web server behind the scenes. i.e possible to update parts of a web page, without reloading the whole page.

AJAX working



Browser

An event occurs...

- Create an XMLHttpRequest object
- Send HttpRequest

Server

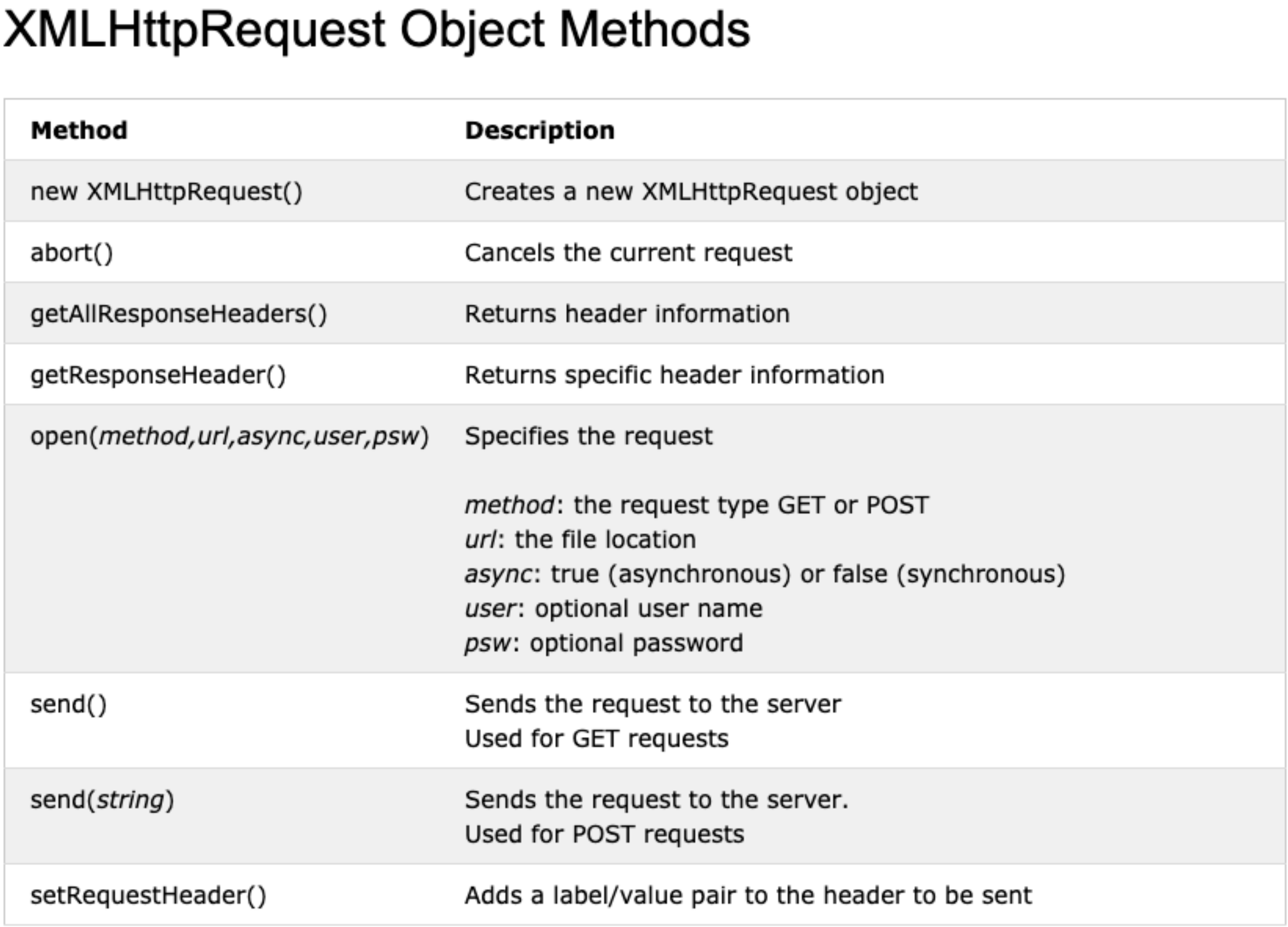
- Process HTTPRequest
- Create a response and send data back to the browser

Browser

- Process the returned data using JavaScript
- Update page content

XMLHttpRequest Object

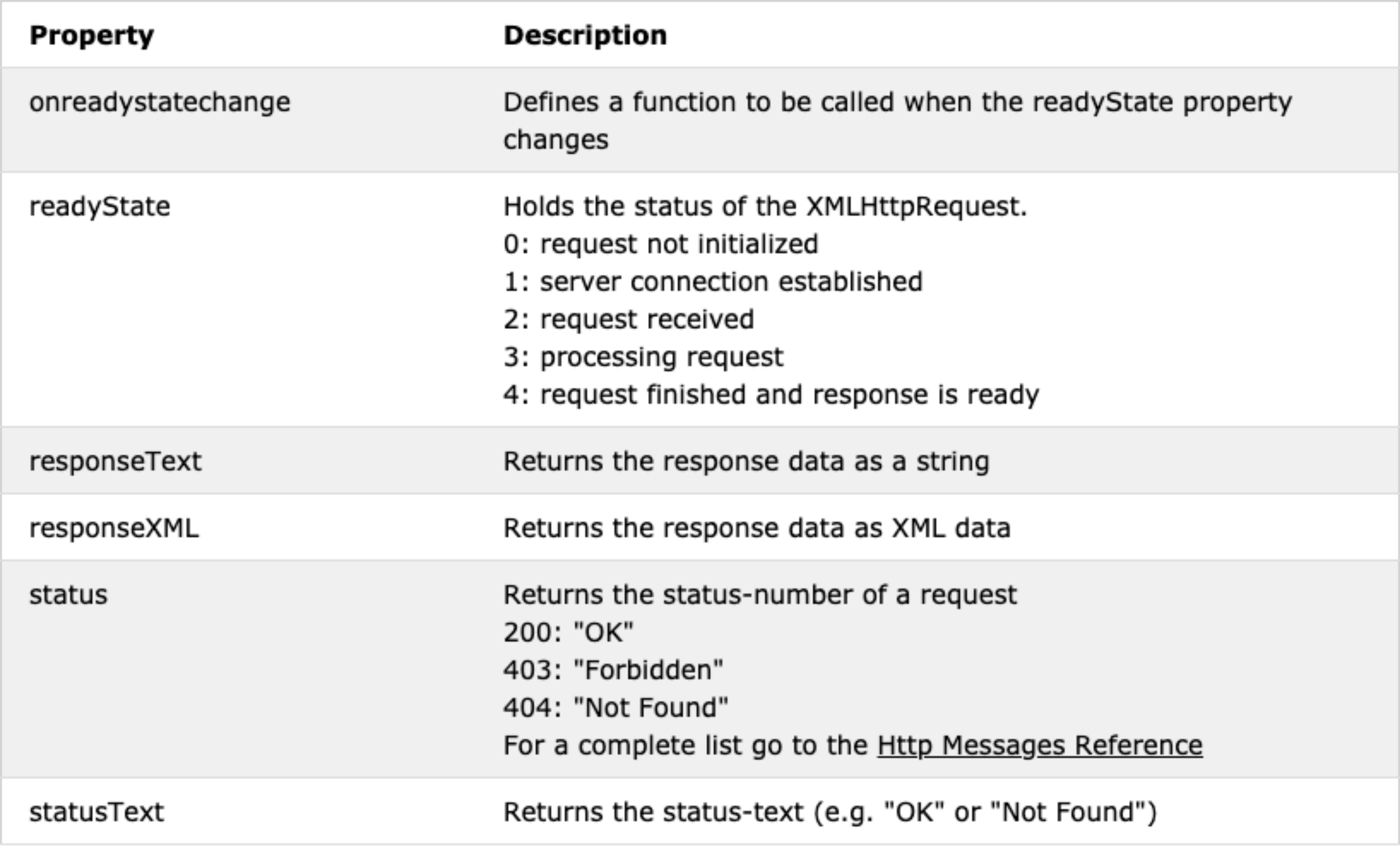
- Used to exchange data with a server behind the scenes.
- Syntax
- var reqObjName = new XMLHttpRequest();



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 *method*: the request type GET or POST *url*: the file location *async*: true (asynchronous) or false (synchronous) *user*: optional user name *psw*: optional password
send()	Sends the request to the server Used for GET requests
send(*string*)	Sends the request to the server. 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. 0: request not initialized 1: server connection established 2: request received 3: processing request 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 200: "OK" 403: "Forbidden" 404: "Not Found" For a complete list go to the Http Messages Reference
statusText	Returns the status-text (e.g. "OK" or "Not Found")

AJAX request

- XMLHttpRequest object us used to exchange data with a server.
- To send request to server, use open() and send() methods of XMLHttpRequest object.

Eg.

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

GET/POST: specifies the type of request

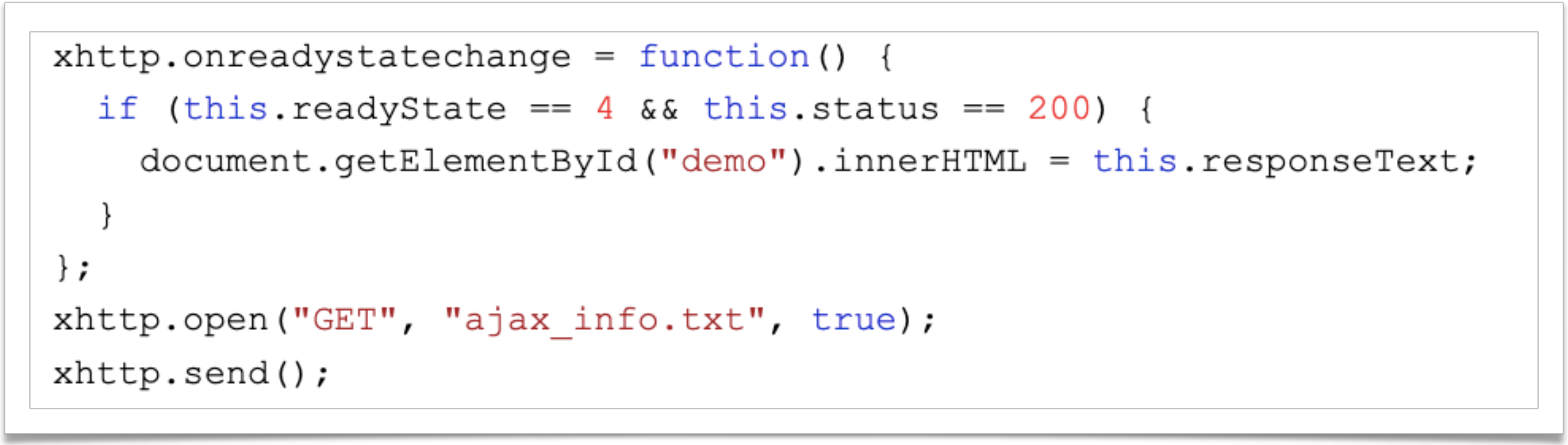
URL:file location on a server

Async(true/false): must be set to true. The javascript does not have to wait for the server response but can execute other scripts while waiting for response and deal with response when ready.

onreadystatechange property

❑ With the XMLHttpRequest object you can define a function to be executed when the response was received.

❑ Function is defined on **onreadystatechange** property.



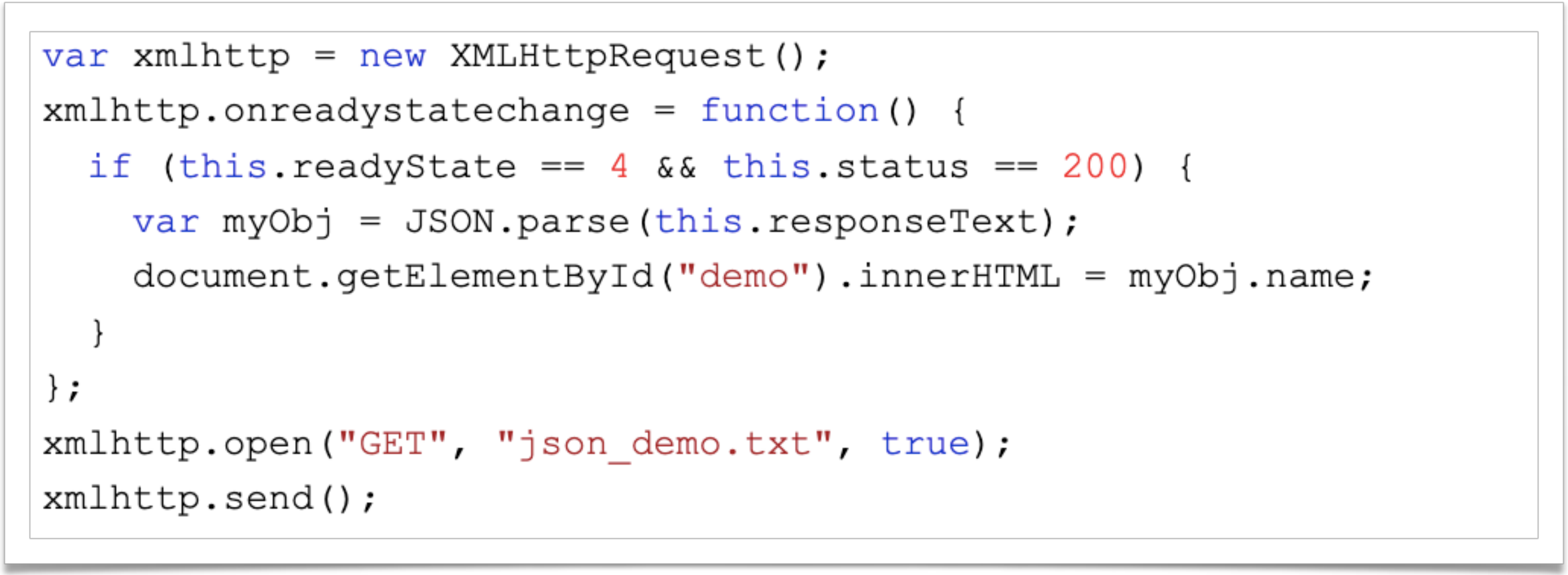
```
xhttp.onreadystatechange = function() {
  if (this.readyState == 4 && this.status == 200) {
    document.getElementById("demo").innerHTML = this.responseText;
  }
};
xhttp.open("GET", "ajax_info.txt", true);
xhttp.send();
```

AJAX response

- ❑ The **readyState** property holds the status fo the XMLHttpRequest
- ❑ **Status** and **statusText** property holds the status of the XMLHttpRequest object.
- ❑ The onreadystatechange function is called every time the readyState changes.
- ❑ When readyState is 4 and status is 200, the response is ready.

JSON from sever with AJAX

❑ You can request JSON from the server by using an AJAX request.



```
var xmlhttp = new XMLHttpRequest();
xmlhttp.onreadystatechange = function() {
  if (this.readyState == 4 && this.status == 200) {
    var myObj = JSON.parse(this.responseText);
    document.getElementById("demo").innerHTML = myObj.name;
  }
};
xmlhttp.open("GET", "json_demo.txt", true);
xmlhttp.send();
```

Thank you.