Advanced JavaScript

Eng. Niveen Nasr El-Den
SD & Gaming CoE
iTi

> “ *These are the golden days of JavaScript*

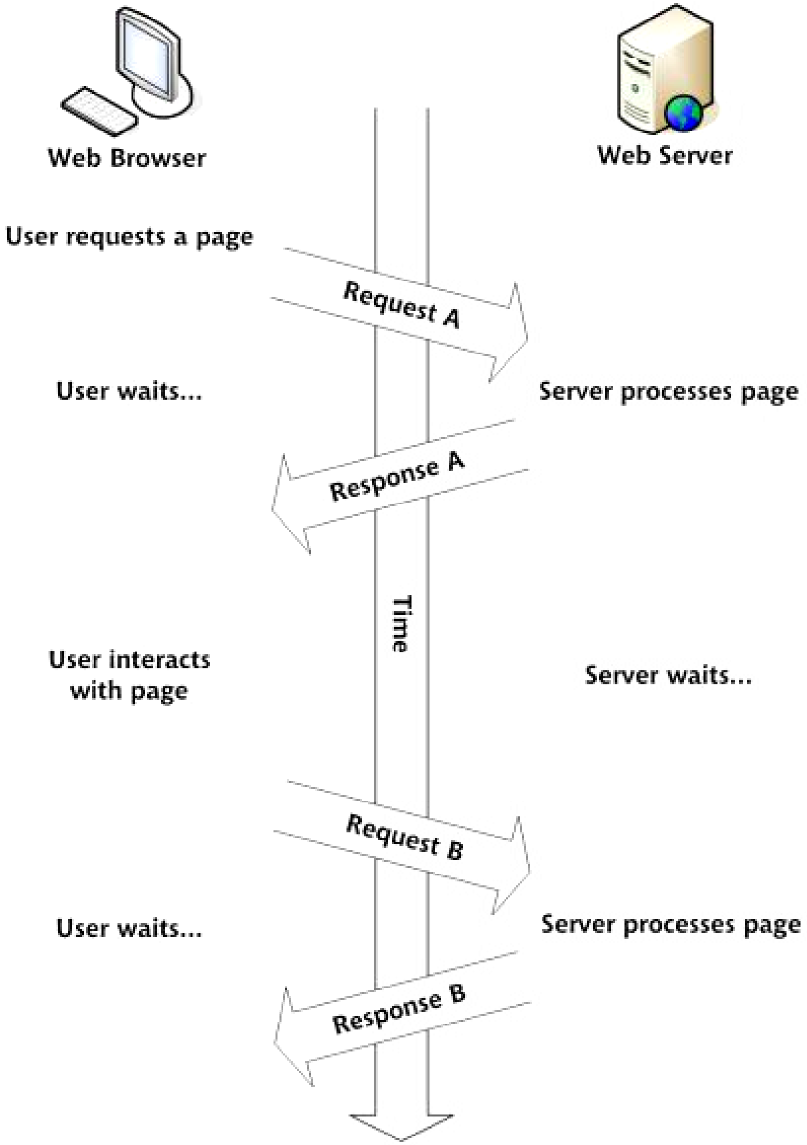
ajax

Ajax

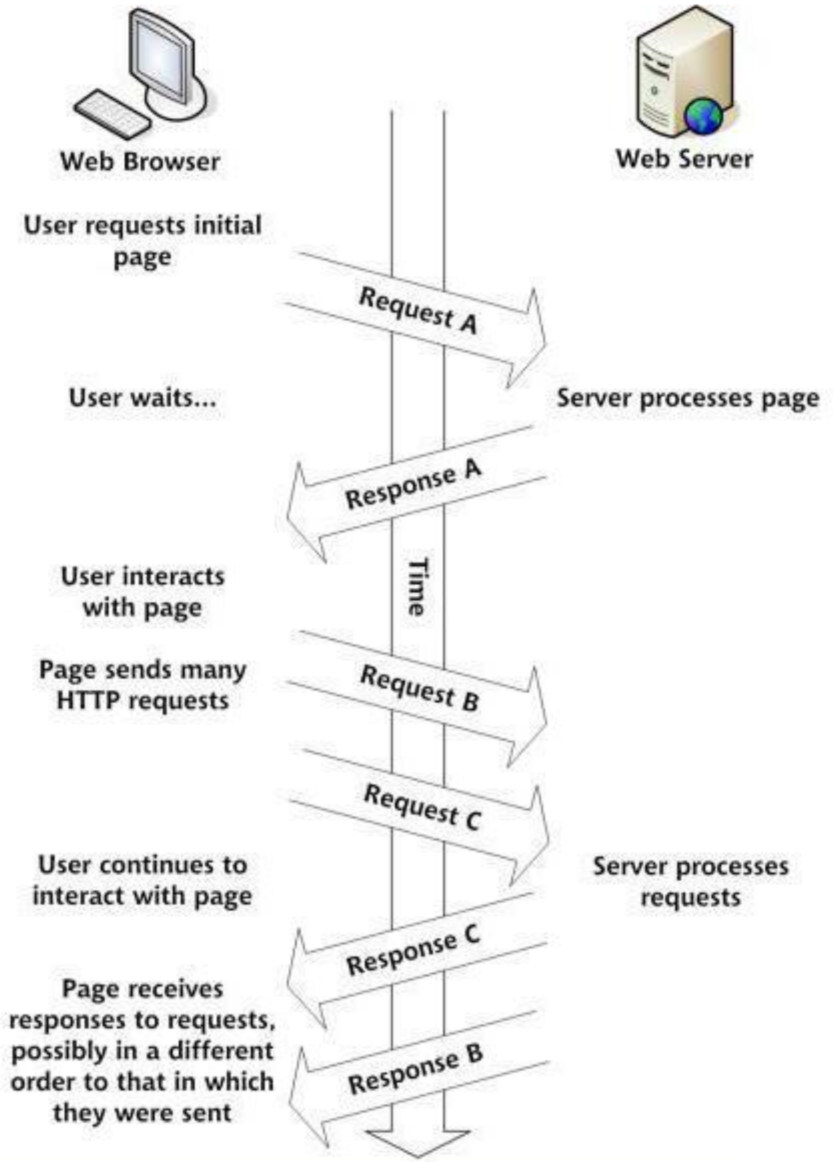
http://adaptivepath.org/ideas/ajax-new-approach-web-applications/

- *Ajax stands for Asynchronous JavaScript and XML.*
- A new paradigm which was introduced in 2005 by Jesse James Garrett.
- The purpose of Ajax is that of giving the illusion that websites are responsive.
- It achieves this by processing requests which involve the sending and receiving of small packets of data without refreshing the web browser.

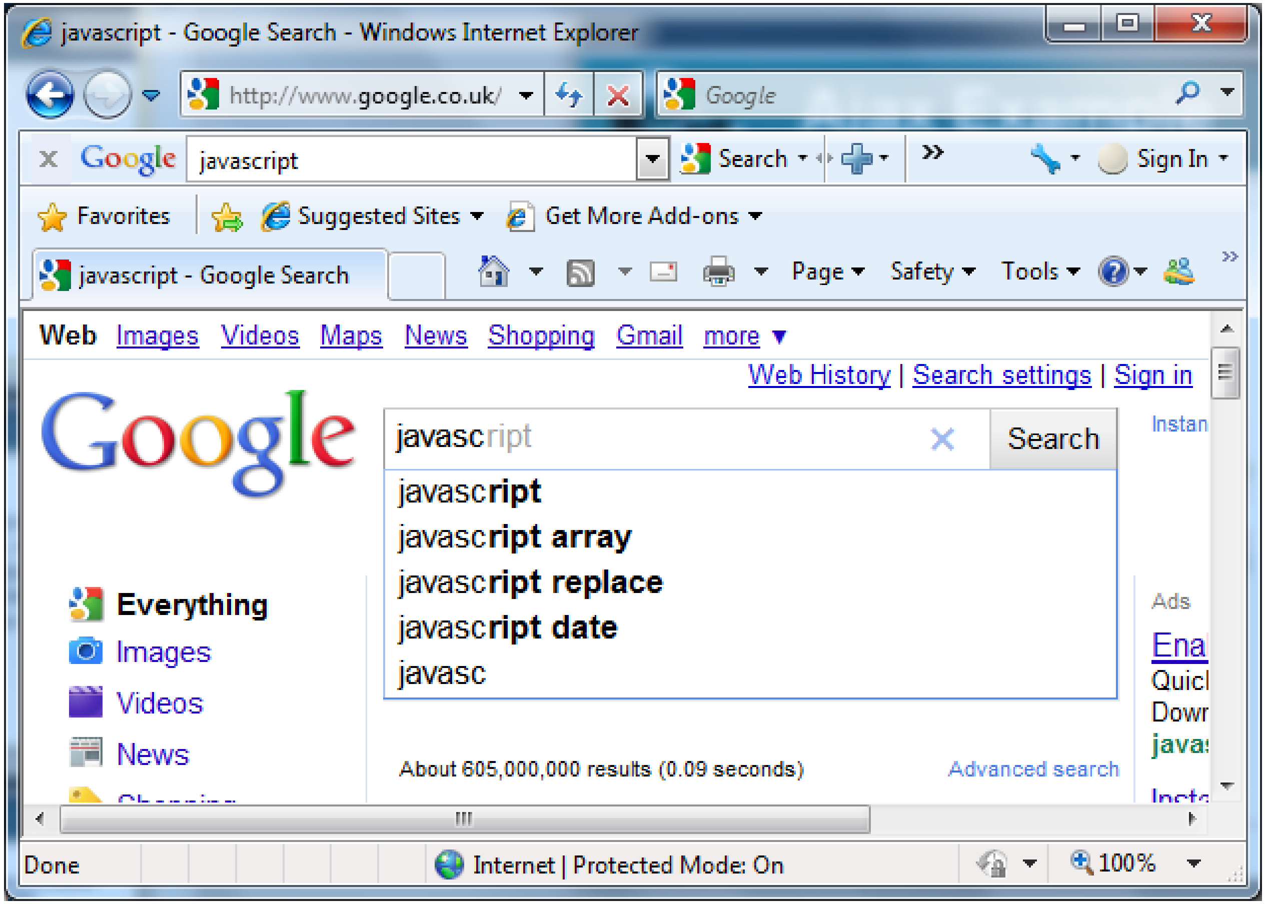
Synchronous



ASynchronous


Web Browser
Web Server
User requests initial page
Request A
User waits...
Server processes page
Response A
Time
User interacts with page
Page sends many HTTP requests
Request B
Request C
User continues to interact with page
Server processes requests
Response C
Page receives responses to requests, possibly in a different order to that in which they were sent
Response B

Ajax Example



javascript

javascript
javascript array
javascript replace
javascript date
javasc

About 605,000,000 results (0.09 seconds)

Ajax

- Ajax is not a new technology, nor a new programming language.
- It is a methodology for creating fast, rich, user-friendly, and interactive Web applications by allowing a Web page to request small fragments of information from the server instead of an entire page.
- XML is not a requirement for writing Ajax applications

Ajax is founded on

- **JavaScript**
  - With JavaScript, you can make an HTTP request to the server, get the response and update only a part of the page.
- **XHTML**
  - A rigid subset of html which is used to mark-up and style the information.
- **DOM**
  - The Document Object Model which can be accessed by the client browsers.
- **XHR Object**
  - The object used to exchange the information asynchronously.
- **XML**
  - The format used to transfer the data from the server to the client.

XMLHttpRequest Object

XHR

XMLHttpRequest Object

- The XMLHttpRequest object is not specified in any W3C recommendation. (**XHR** for short)

- It is an object (a constructor function) that allows you to send HTTP requests from JavaScript.

- Historically, was introduced in IE and was initially implemented as an **ActiveX** object.

- The XHR object is supported in all modern browsers.

XHR Object

- XHR object across browsers gave birth to the so-called AJAX applications, where it is no longer necessary to refresh the whole page every time you need new content.

- With JavaScript, you can make an HTTP request to the server, get the response and update only a part of the page.

XHR Object Properties/Events

https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest

Properties	Description
onreadystatechange	An event handler for an event that fires at every state change
readyState	Returns the state of the object
responseText	Returns the response as a string
responseXML	Returns the response as XML.
status	Returns the status as a number (e.g. 404 for "Not Found" or 200 for "OK")
statusText	Returns the status as a string (e.g. "Not Found" or "OK")

readyState Values Description

Value	Description
0 XMLHttpRequest.UNSENT	uninitialized (object has been created but not initialized)
1 XMLHttpRequest.OPENED	loading (Loading, send method not yet called)
2 XMLHttpRequest.HEADERS_RECEIVED	loaded (Loaded, send method called, headers not available)
3 XMLHttpRequest.LOADING	interactive (Interactive, some data has been received, status and response headers not available)
4 XMLHttpRequest.DONE	complete (Complete, all data has been received and is available)

Status & Status Text

- 404: Not found → The server found nothing matching the URI given.
- 200: OK → The server successfully returned the page
- 400: Bad Request → Server didn’t understand the request due to malformed syntax.
- 401: Unauthorized → The request requires user authentication.
- 500: Internal Server Error → The server encountered an unexpected error and couldn’t fulfill the request.
- 503: Service Unavailable → The server is currently unable to handle the request due to temporary overloading or maintenance.

XHR Object Methods

Methods	Description
open("method","URL“ [,async [,"uname“[,"pswd“]]])	Specifies the method, URL, and other optional attributes of a request async default is true
send(content)	Sends the request •*content* is the posted in a POST request •*content* can be **null** or empty

XHR Object

- The *XMLHttpRequest* object has several important properties and methods that will give you information about
  - the status of the request response,
  - the data that was returned from the server, when the state of the server changes, and so on.
- There are methods to
  - initialize the *XMLHttpRequest* object,
  - send a request,
  - inform you about what is in the response headers,
  - how to cancel a request, and so on.

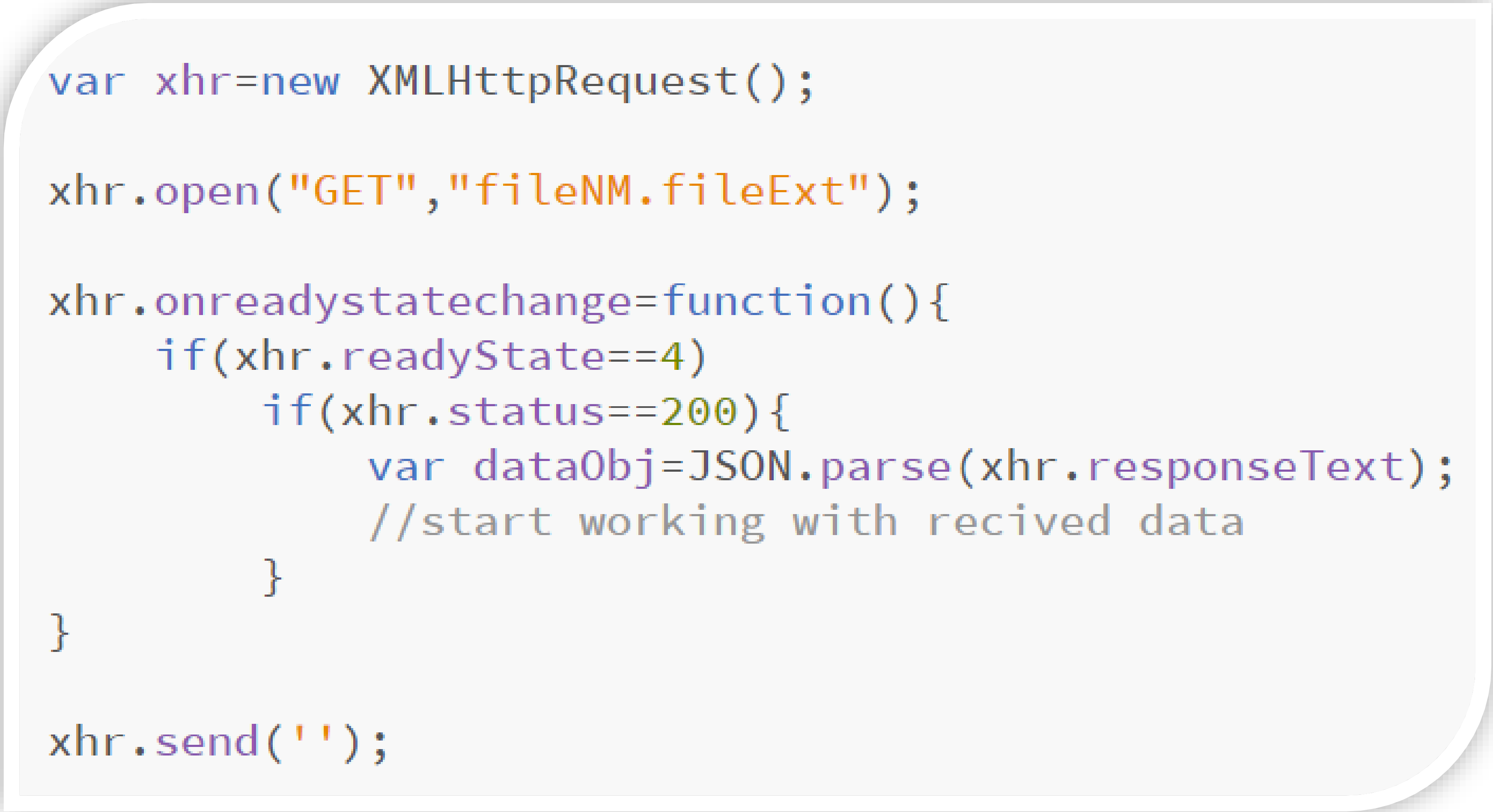
XHR Object

- There are effectively two steps to using the XMLHttpRequest:
  - Send the request:
    - this includes creating an XMLHttpRequest object and attaching an event listener
  - Process the response:
    - your event listener gets notified that the response has arrived and your code gets busy doing something useful with the response

The Steps for Creating Ajax Communication

- Create an ***XMLHttpRequest*** object.
- Use the object's ***open()*** method to initialize the object
- Send the request to the server with the ***send()*** method.
- Track the state of the request object by ***readyState*** property of ***XMLHttpRequest*** object.

Example!



```
var xhr=new XMLHttpRequest();

xhr.open("GET","fileNM.fileExt");

xhr.onreadystatechange=function(){
    if(xhr.readyState==4)
        if(xhr.status==200){
            var dataObj=JSON.parse(xhr.responseText);
            //start working with recived data
        }
}

xhr.send('');
```

JavaScript Object Notation

JSON

JSON

- JSON stands for JavaScript Object Notation
- Like Ajax, it is not a programming language
- It is a special object notational construct, that is a subset of JavaScript.
- It is not a markup language.
- JSON is based on two data structures:
  - arrays and
  - objects.

JSON

- A Data Interchange Format
- Text-based
- Light-weight
- Simple
- Easy to parse

JSON Structure “Object”

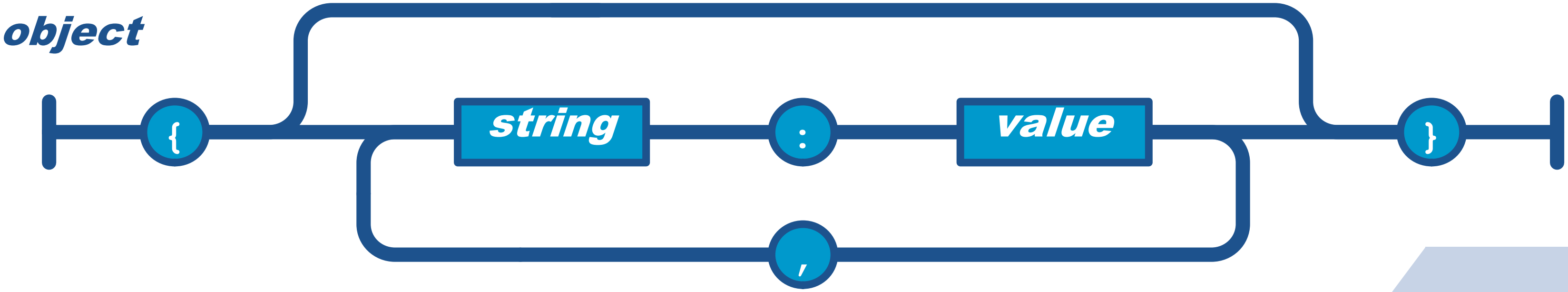
- JSON can be either an Object or an Array.
- The Object is an **unordered** set of name/value pairs.
- An Object begins with ({) and end with (})
  - A pair is a name followed by (:) and the value
    - The colon is called the name separator
  - The key/value pairs are separated with a (,)
    - The comma is called value separator.
  - Values are JSON values

JSON Structure “Array”

- JSON Array is an **ordered** sequence of values
- JSON Array begins with ([) and ends with (])
- Values are separated by a comma (,),
- The values can be any of the primitive types as well as the two structures, JSON objects and JSON arrays.

Object

- Objects are unordered containers of key/value pairs
- Objects are wrapped in { }
- , separates key/value pairs
- : separates keys and values



Object Example

```
{
    "name": “json",
    "at large": true,
    "grade": "A",
    "format": {
"type": "rect",
"width": 1920,
"height": 1080,
"framerate": 24
            }
}
```

```
var personObj={
    “firstName": “json",
    “lastName": true,
    “age": "A",
    “tellYourAge": function {
            alert(“This age is”+this.age);
          }
     }
```

Note: Accessing Data In JSON

Dot Notation→

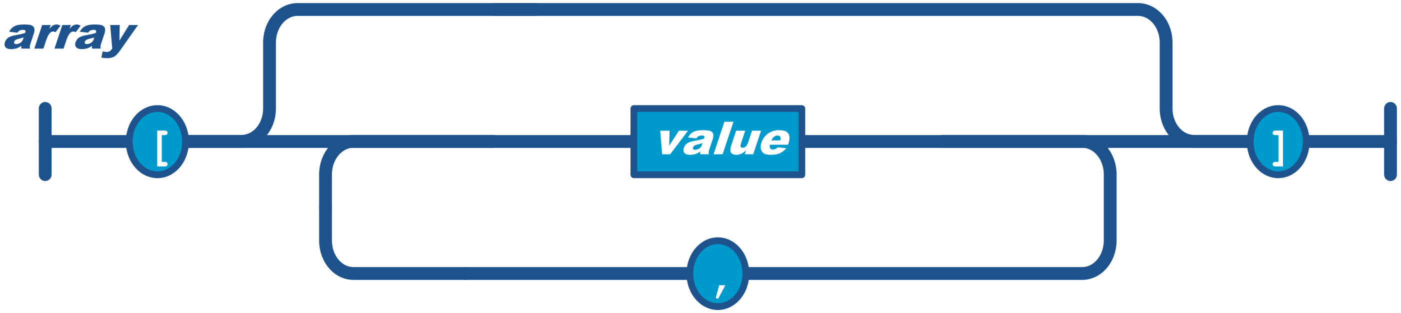
```
    myObj.myField=“something”
```

Subscript Notation(as Associative Array) →

```
    myObj[‘myField’]=“something”
```

Array

- Arrays are ordered sequences of values
- Arrays are wrapped in `[]`
- `,` separates values



```
["Sunday", "Monday",
"Tuesday", "Wednesday",
"Thursday", "Friday",
"Saturday"]
```

```
[
    [0, -1, 0],
    [1, 0, 0],
    [0, 0, 1]
]
```

Object Array Example

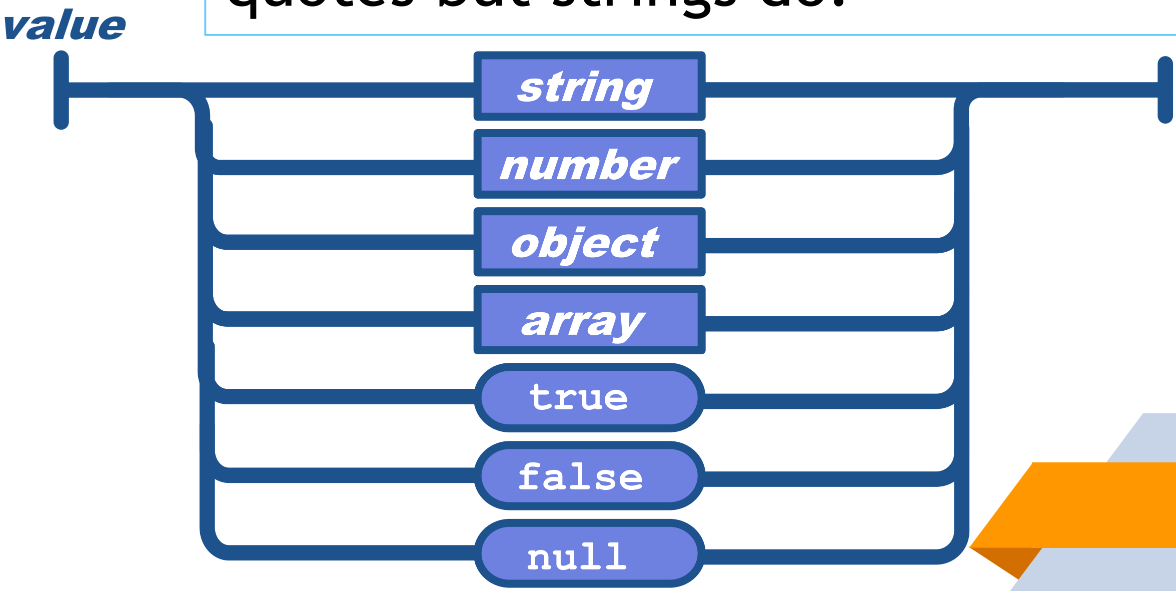
```
var myJSONObject =
{
    "web":[
        { "name": "html", "years": "5"},
        { "name": "css", "years": "3"}
        ]
    "db":[
          { "name": "sql", "years": "7"}
          ]
}
```

Example!

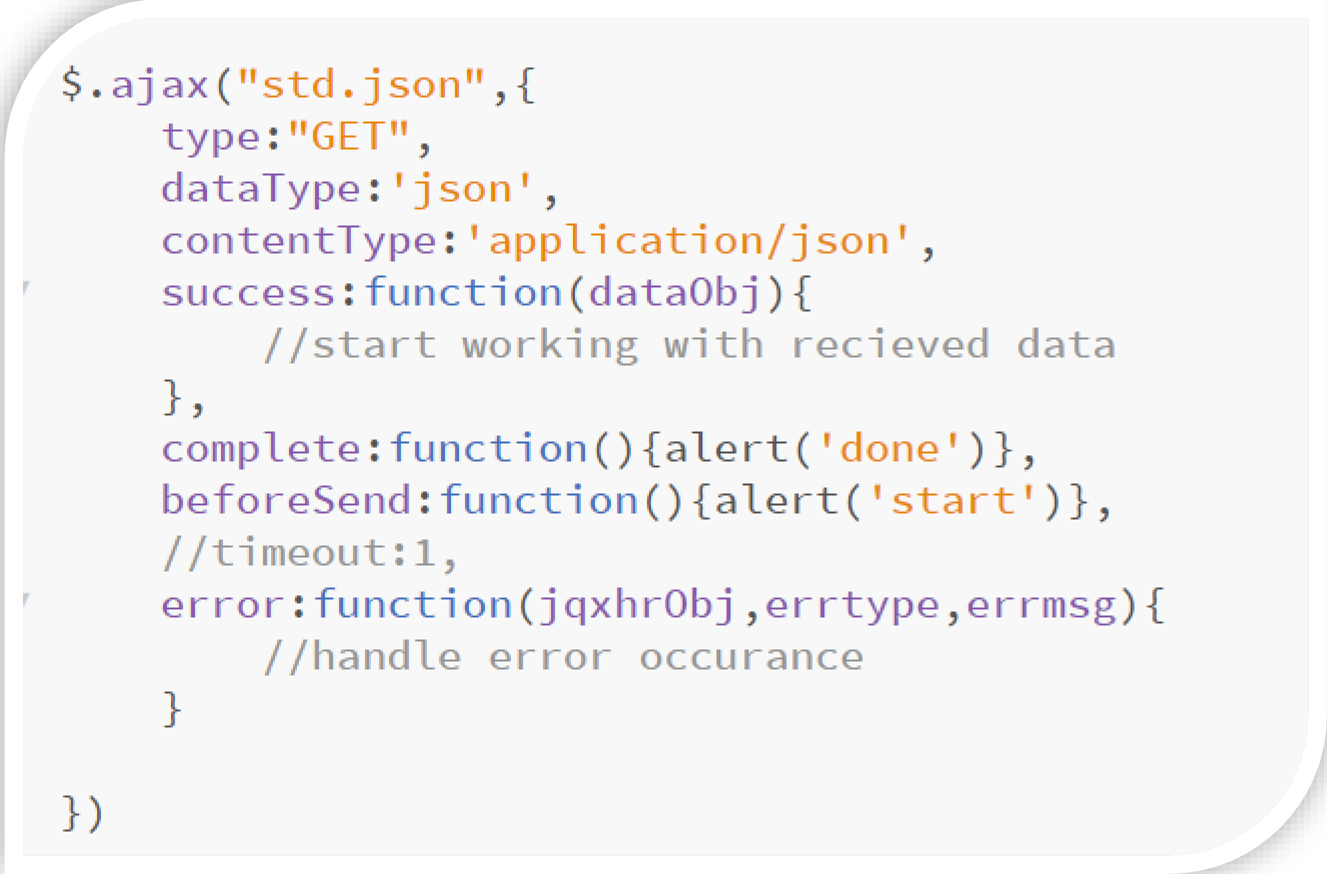
JSON Values

- Strings
- Numbers
- Booleans
  - true
  - false
- Objects
- Arrays
- **`null`**

Unlike JavaScript it does not support hexadecimal and octal representation of numbers, nor does it support NaN (not a number) and infinity.
It supports scientific notation like 6.023e+23. Also numbers do not need quotes but strings do.



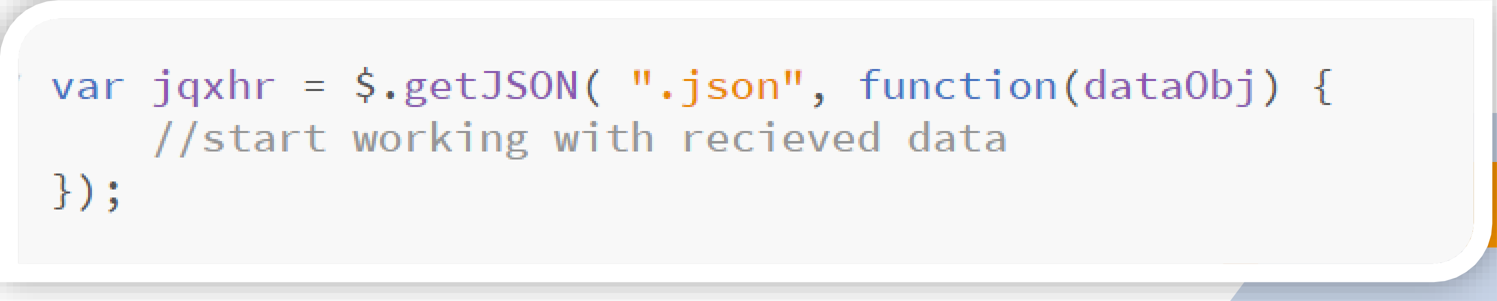
jQuery AJAX Methods



```
$.ajax("std.json",{
    type:"GET",
    dataType:'json',
    contentType:'application/json',
    success:function(dataObj){
        //start working with recieved data
    },
    complete:function(){alert('done')},
    beforeSend:function(){alert('start')},
    //timeout:1,
    error:function(jqxhrObj,errtype,errmsg){
        //handle error occurance
    }

})
```

http://api.jquery.com/category/ajax/



```
var jqxhr = $.getJSON( ".json", function(dataObj) {
    //start working with recieved data
});
```

Assignment