

Web Design & Programming

JavaScript and AJAX

Xavier Belanger

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

<http://creativecommons.org/licenses/by-sa/4.0/>



You are free to:

- **Share** — copy and redistribute the material in any medium or format
- **Adapt** — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

- **Attribution** — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
- **ShareAlike** — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.
- **No additional restrictions** — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

History

- The JavaScript language was created by Brendan Eich, working at Netscape Communication Corp. in 1995.
- The goal was to add scripting and interactivity to static web pages. The language creation was influenced by Java and other programming languages, but it is a distinct language of its own.
- The formal specification of JavaScript is ECMAScript, standardized by the European Computer Manufacturers Association (ECMA). Various web browsers and vendors have their own implementation.

JavaScript as a language

- Interpreted, not compiled
- Case sensitive
- Variables are weakly typed
- General syntax close to C/C++/Java
- Object-oriented

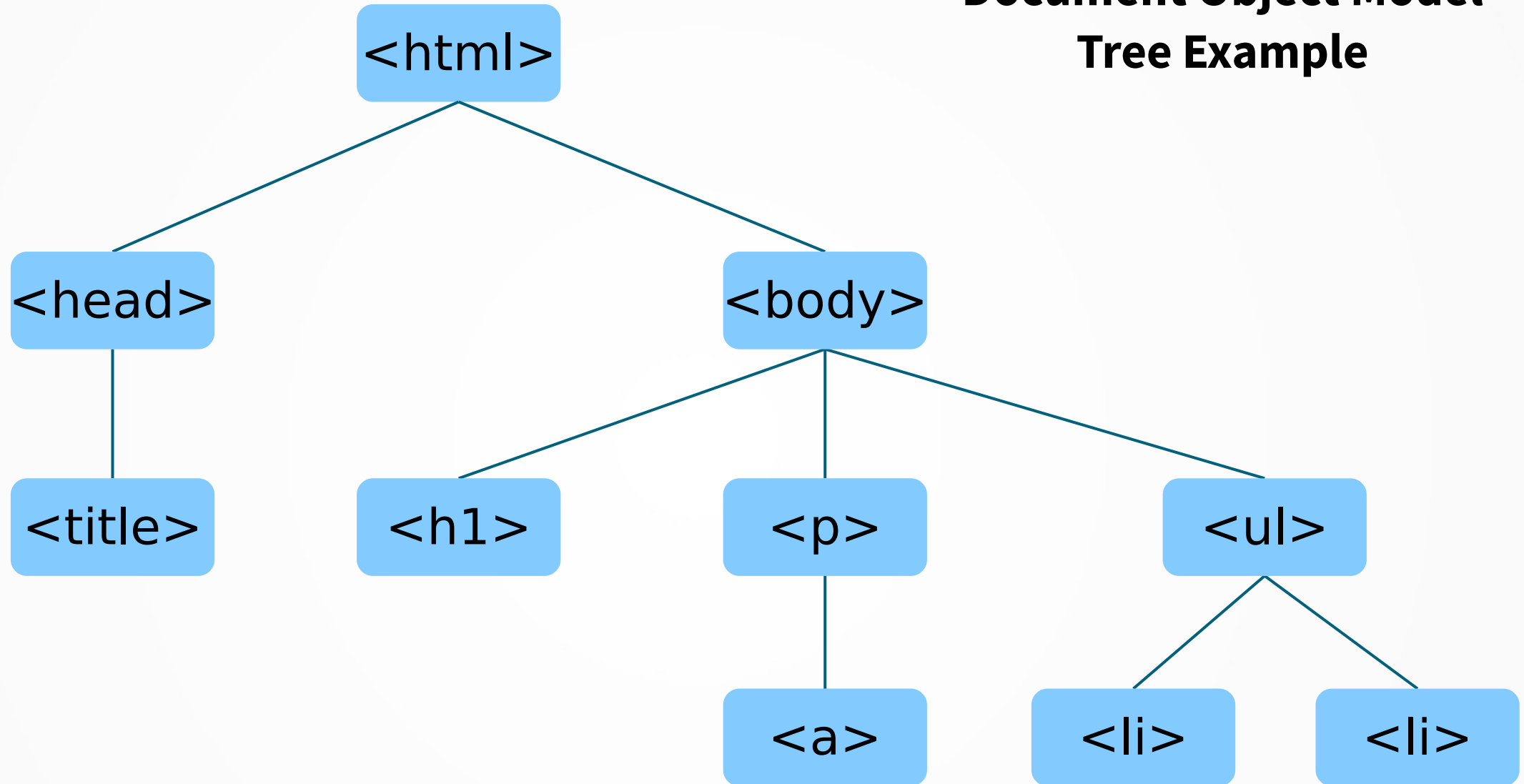
Using JavaScript

- Scripts can be declared:
 - inline, in a `<script>` block, in the head or body of a HTML document
 - in an external file (.js) included in a HTML document
- The second solution is the most common, either with one or multiple files used for an entire website.

The Document Object Model

- The DOM provides an interface to manipulate HTML elements in a page.
- Elements are organized in a tree, with different types of nodes:
 - element nodes
 - attribute nodes
 - text nodes
- Node properties can be modified, nodes can be selected, modified, added or removed by using different methods.

Document Object Model Tree Example



Libraries and Frameworks

- In general few developers are using “vanilla” JavaScript, many web applications are based on a library or on a framework:
 - Angular
 - JQuery
 - React
 - Vue.js

JavaScript on the Server Side

- Node.js is a JavaScript library that can be used to generate web content on the server side (outside of a web browser).
- The advantages are the execution speed (using non-blocking functions) and the language unicity between the front end and the back end.

JSON

- JavaScript Object Notation
- Standardized text-based file, human readable used to store and exchange information.
- Used with JavaScript and many other programming languages.
- Doesn't allow comments.
- File extension: .json

Asynchronous JavaScript and XML

- AJAX is used to create asynchronous communication between a web browser and a server; this allow the web browser to receive pieces of information without reloading a whole web page.
- This is used to bring more interactivity into web applications; the user-experience is closer to what regular applications provides. This also helps with network bandwidth usage.

Implementation

- The original AJAX implementation was using Microsoft ActiveX; as of today JavaScript is used instead.
- Using the *XMLHttpRequest* object you can create a new request (targeting a file on the web server), tracking its progress and processing the result.
- Despite his name, XMLHttpRequest is not limited to XML files, other files can be used (HTML, JSON, ...).

The XMLHttpRequest Object

- Properties
 - **onreadystatechange**: used to call a function when the status of the object changes
 - **readyState**: specify the status of the object (0/1/2/3/4, from 'unsent' to 'done')
 - **responseText**: data returned by the server
 - **status**: HTTP status code
- Methods
 - **open**: define the HTTP Request
 - **send**: initiate the HTTP Request
- Reference: <https://xhr.spec.whatwg.org/>

JavaScript Code

```
var request = new XMLHttpRequest();
request.open("GET", "data.html");
request.onreadystatechange = function()
{
    if (this.readyState === 4 && this.status === 200)
    {
        document.getElementById("targetElement").innerHTML = this.responseText;
    }
};
request.send();
```