

# What is AJAX and what are its purposes?

Author: Mihai Rotaru

Date: 1 Dec 2011

AJAX stands for Asynchronous JavaScript and XML, and is a group of technologies used to create dynamic web pages. The main objective of AJAX is to allow pages to have their content modified, without having to re-download the whole page again.

Before AJAX became widespread, most websites relied mostly on separate HTML files for displaying even dynamic content - such as constantly changing game scores, displayed by a betting company's website. This has proven troublesome both for the company, because each time the information to be displayed changed, the server had to send a new, updated HTML file - which was costly due to heavy bandwidth usage. It was also inconvenient for the users, since the whole page had to be reloaded.

Therefore, the need for methods such as AJAX became apparent. It was first implemented by Microsoft's Internet Explorer, taking the form of `iframe` elements - which allowed only for parts of a webpage to be updated by the server. This technology evolved into the `XMLHttpRequest` JavaScript object, which forms the backbone of current AJAX technologies and is supported by all modern browsers.

However, implementation by most browsers did not translate into wide adoption, and such technologies remained uncommon until their successful employment by Microsoft and Google in large-scale applications such as Outlook Web Access and Gmail. This proved the value and usefulness of AJAX technologies, and spurred their adoption for mainstream web development.

Taking a website which displays football scores of on-going games, without AJAX the users would have to re-download the whole HTML file in order to notice any changes in scores. But it would be much better to be able to re-download just the XML file containing the scores, and update the HTML file accordingly, and AJAX makes that possible.

An `XMLHttpRequest` object would be created from the JavaScript file, and a function would be called when the object's `readyState` property changes ( via the `xhr.onreadystatechange` property, assuming the `XMLHttpRequest` object is named `xhr` ) The function would parse the retrieved data, and update the HTML code currently loaded in the user's browser accordingly, without having to re-download all of the HTML code from the server.