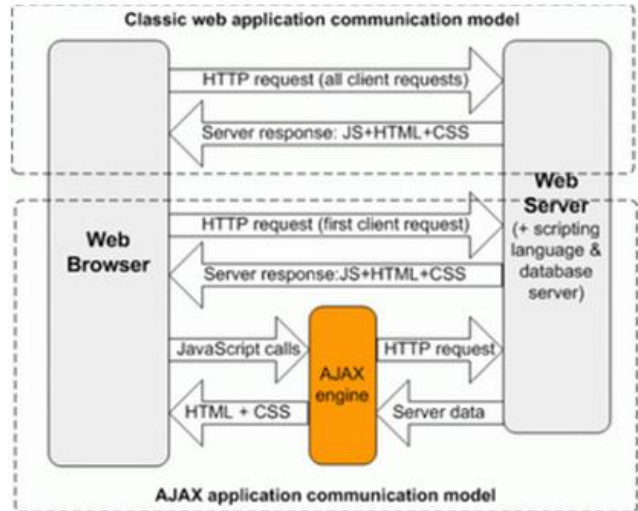


Die Grundidee hinter AJAX (Asynchronous JavaScript and XML) ist eine zusätzliche Schicht im Gegensatz zum traditionellen Modell von Webanwendungen. Diese Schicht ist die sogenannte AJAX-Engine (siehe Abb.). Sie soll die Interaktion der Besucher beobachten, bei Bedarf Anfragen an den Server schicken, dessen Antworten interpretieren und die Seitendarstellung entsprechend anpassen – und zwar direkt und im Hintergrund, ohne dass der Benutzer etwas von den Client-Server- Prozessen merkt. HTTP-Anfragen werden demnach durch Engine-Anfragen ersetzt, wobei Inhalte der Webseite immer automatisch aktualisiert werden, sodass ein erneuter Aufbauvorgang der kompletten Seite nicht mehr notwendig ist.



Quelle: <https://www.webdesign-podcast.de/2011/01/26/wie-funktioniert-ajax/>

Die AJAX-Technik besteht aus mehreren Komponenten, die zusammenarbeiten, um eine asynchrone Kommunikation zwischen Client und Server zu ermöglichen.

JavaScript	Stellt dynamische Inhalte dar und ermöglicht Benutzerinteraktionen
XMLHttpRequest	Initiiert und verarbeitet HTTP-Anfragen und -Antworten
DOM (Document Object Model)	Ändert die Struktur, den Stil oder den Inhalt der Webseite in Echtzeit
HTML und CSS	Strukturiert und stylt die Webseite
JSON oder XML-Datenformate	Zum Versenden von Daten von Server zu Client und umgekehrt

	Ajax Request (mit Datenübertragung GET ⑥ ⑦ oder POST ⑧ ⑨ ⑩)
①	<code>var xmlhttp = new XMLHttpRequest();</code>
②	<code>xmlhttp.onreadystatechange = function() {</code>
③	<code>if (this.readyState == 4 && this.status == 200) {</code>
④	<code>document.getElementById("xxx").innerHTML = this.responseText;</code>
⑤	<code>/* responseText is der Antwort-String des Server-Scriptes */</code>
⑥	<code>} };</code>
⑦	<code>xmlhttp.open("GET", "URI_ServerScript.php?var1=wert1&var2=wert2" , true);</code>
⑧	<code>xmlhttp.send();</code>
⑨	<code>xmlhttp.open("POST", "URI_ServerScript.php" , true);</code>
⑩	<code>xmlhttp.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');</code>
	<code>xmlhttp.send(encodeURIComponent(var1=wert1&var2=wert2));</code>

Typischerweise werden komplexere Daten zwischen Client und Server im JSON-Format (JavaScript Object Notation) übertragen. Sowohl in JavaScript als auch in php stehen Konvertierungs-Funktionen zur Verfügung.

 PHP <?>	String json_encode (object \$value) Gibt eine Zeichenkette zurück, die die JSON-Darstellung des übergebenen value beinhaltet	 JSON {}	object json_decode (String \$json) Konvertiert eine JSON-kodierte Zeichenkette \$json in einen PHP-Objekt
 JS f{}	String JSON.stringify (object daten) konvertiert daten in eine JSON-formatierte Zeichenkette		object JSON.parse (String daten) JSON.parse wandelt den String in ein Objekt

