anatom5 perception marketing
anatom5 perception marketing GmbH
fon: +49.211.420.944.1 / fax: +49.211.420.944.3
mail: info@anatom5.de / web: www.anatom5.de

Javascript und das Web der Zukunft

Kaum eine andere Skript-Sprache hat so viele Höhen und Tiefen durchlaufen wie JavaScript: Mal Buhmann, mal Held. Und nun womöglich sogar elementares Bindeglied für unterschiedliche Technologien und Standards im Web. Wohin entwickelt sich das Web 3.0 unter Federführung von JavaScript?

Ob sich Brendan Eich bei der Erfindung von JavaScript jemals 3D-Objekte, asynchrone Datenübertragung und mobile Web-Apps vorgestellt hat? Unlängst antwortete Eich auf eine vergleichbare Frage, dass man sich früher Dinge, wie Canvas, nicht habe vorstellen können, weil JavaScript nicht schnell und leistungsfähig genug war. Früher meint den Zeitraum der zweiten Browser-Kriege von 2008 bis 2009, vor Einführung der kompilierenden JavaScript-Engines. Eine Zeit, in der JavaScript im Browser ein Schattendasein führte und von langsamen Interpretern ausgebremst wurde. Heute haben JavaScript-Engines so klangvolle Namen wie Carakan, Chakra, SquirrelFish oder SpiderMonkey. Sie alle eint das Bestreben nach noch mehr Geschwindigkeit und Leistung. Beides ist notwendig, denn Online-Anwendungen wie Bing Maps, Yahoo Mail oder Google Docs bestehen zu einem Großteil aus JavaScript und sind entsprechend leistungshungrig. Vor allem das Erscheinen der kompilierenden JavaScript-Engine V8 von Google und die daraus resultierenden Performance-Steigerungen haben das Zeitalter der Webanwendungen möglich gemacht. Wie stark sich der Leistungszuwachs auswirkt, zeigen vor allem Benchmark-Ergebnisse aktueller Browser, wie beispielsweise der SunSpider-Benchmark, der seit langem den Maßstab für die Leistungsfähigkeit von JavaScript darstellt. Die neuen JavaScript-Engines verlangen nach neuen Benchmarks, die zumeist von den Browser-Herstellern für ihre eigenen Engines entwickelt werden, so wie im Fall von Mozilla Kraken, das für die eigene Engine SpiderMonkey entwickelt wurde. Mit Dromaeo existiert allerdings auch ein einheitlicher JavaScript-Benchmark für alle Browser, der ebenfalls von John Resig (jQuery) initiiert wurde. Dadurch werden Testergebnisse wieder vergleichbarer.

Während sich Chrome und Firefox in nahezu allen Benchmarks auf Augenhöhe duellieren, liegen die Verfolger zum Teil deutlich zurück. Allerdings hat das Thema JavaScript-Performance bei allen Browser-Herstellern in den letzten Jahren eine deutliche Priorisierung erfahren und so wird es nicht lange dauern, bis das Feld wieder ausgeglichener ist.

Schnell und vielseitig

Warum so viele Webanwendungen auf JavaScript aufsetzen ist leicht erklärt: Es gibt schlichtweg keine Alternative, auch wenn Google jüngst versucht hat, Dart als neue Programmiersprache zu etablieren, dafür allerdings viel Kritik erntete.

Der Vorteil: JavaScript ist sehr vielfältig einsetzbar. Während HTML und CSS nur rudimentäre Möglichkeiten zur Nutzerinteraktion bieten, können Sie mit JavaScript nahezu beliebig Inhalte generieren bzw. nachladen. Dadurch wird die Kommunikation mit dem Server auf das Nötigste eingeschränkt. Heute ist dieser Aspekt unter dem Begriff AJAX in den Alltagssprachgebrauch der meisten Webworker übergegangen. Eng mit AJAX ist auch der Aufstieg von JavaScript zur beliebtesten Skriptsprache verbunden. In einem Blogbeitrag von John Resig, dem Schöpfer von jQuery, wird die Evolution von JavaScript durch vier wesentliche Phasen skizziert:

- Phase 1: „Wir benötigen Scripting für Webseiten" (Netscape, 1995)
- Phase 2: „Wir sollten das standardisieren" (ECMAScript, 1997)
- Phase 3: „JavaScript ist mehr als ein Spielzeug" (AJAX, 2005)
- Phase 4: „JavaScript ist eine Programmiersprache" (2007)

Ganz aktuell ist JavaScript in Phase 5 eingetreten und erobert nach der Client-Seite auch die Server-Seite, obwohl Server Side JavaScript bereits ebenso alt ist wie die bekannte Client-Side-Variante. Allerdings gibt es erst seit dem Erscheinen von Googles V8 eine Engine, die leistungsfähig genug für den Server-Betrieb ist und die JavaScript-Code bei der Ausführung direkt in Maschinen-Code übersetzt, statt zunächst Bytecode auszuführen oder ein Skript direkt zu interpretieren. Kurzum: Alles wird deutlich schneller. Insbesondere durch Anwendungen wie node.js, die diese neuen

anatom5 perception marketing
anatom5 perception marketing GmbH
fon: +49.211.420.944.1 / fax: +49.211.420.944.3
mail: info@anatom5.de / web: www.anatom5.de

Performance-Möglichkeiten nutzen, erarbeitet sich JavaScript immer neue Anwendungsfelder, die zuvor den Hochsprachen C/C++ vorbehalten waren. Wie einflussreich die Entwicklungen rund um node.js sind, zeigen entsprechende APIs und SDKs von Diensten wie Dropbox. Dass JavaScript auf dem Vormarsch ist, belegen die dokumentbasierten und sehr populären Datenbankformate CouchDB und MongoDB, bei denen JavaScript die Rolle von SQL übernimmt – in Form von JSON-Abfragen, einer weiteren Triebfeder der fünften Phase.

Neue Standards, neue Funktionen

Durch die Standardisierung des Kerns von JavaScript in Form von ECMAScript wurde der Grundstein für den Erfolg der Skriptsprache gelegt, doch auch ActionScript oder JScript nutzen den ECMA-262-Standard. Während die führenden Browser-Hersteller bereits ECMAScript Version 5 (wenngleich nicht vollständig implementiert) einsetzen, nutzen zahlreiche andere Anwendungen, wie die Adobe Creative Suite oder das Microsoft .NET-Framework immer noch den Stand der Version 3. Die vierte Version des Standards wurde wegen großer Differenzen der am Standard beteiligten Unternehmen gar nicht erst in die Praxis umgesetzt. Parallelen zum HTML-Standard sind hier nicht zufällig, denn es geht um viel Geld und proprietäre Funktionserweiterungen und damit verbundene Patente. Im Zuge des Hypes rund um HTML5 und CSS3 ist die Runderneuerung von JavaScript in Form von ECMAScript 5 in der Berichterstattung aus dem Sommer 2011 leider etwas zu kurz gekommen. Neben der Unterstützung von JSON als Datenaustauschformat wurde vor allem versucht, die Sprache praxistauglicher zu gestalten, ohne die Kompatibilität zu brechen. Ein notwendiger Schritt, betrachten Sie die enormen Unterschiede zwischen dem ursprünglichen Aufgabengebiet von JavaScript und den heutigen Anforderungen im Zuge von immer komplexer werdenden Anwendungen, die im Browser ablaufen. Oberstes Ziel von ECMAScript 5 ist es, die Sprache von unnötigem und unsinnigem Ballast zu befreien, sie robuster zu machen und dadurch besser für zukünftige Aufgaben gerüstet zu sein. Wichtigste Neuerung: ein Strict Mode, in dem bestimmte Altlasten nicht mehr funktionieren.

JQuery als Vater des Erfolgs

In den meisten Fällen werden Webworker aber wohl kaum etwas von ECMAScript 5 mitbekommen. Kein Wunder, sorgt bereits die Erwähnung des ECMA-Standards für Stirnrunzeln und Irritationen. Faktisch ist an die Stelle von JavaScript eine Mittelschicht getreten, die wie keine andere Entwicklung dazu beigetragen hat, dass JavaScript sich aktuell so großer Beliebtheit erfreut: die Rede ist von JavaScript-Bibliotheken, wie jQuery, Prototype oder Dojo. Ähnlich wie im Zeitalter der Videorecorder hat sich auch im Web eine Bibliothek gegenüber allen anderen besonders durchgesetzt. Das von John Resig entwickelte jQuery ist in seinen Grundfunktionen auch für HTML- und CSS-affine Webworker verständlich, und mit geringem Aufwand lassen sich einfache Funktionen in eine Webseite integrieren. Zudem gibt es eine kaum zu überblickende Ansammlung von Plug-Ins, mit denen der Funktionsumfang von jQuery nahezu beliebig erweitert werden kann – von der Bildergalerie bis hin zum Styleswitcher.

Einer Statistik von Pingdom zufolge wird jQuery aktuell von mehr als 60% aller Webseiten weltweit eingesetzt. Diese Werte werden auch von builtwith.com gestützt, die ähnlich wie Pingdom für Prototype als nächstem „Verfolger" rund 8% Nutzungsquote sehen. Ist jQuery so etwas wie ein neuer De-facto-Standard? Ja und nein. Ja, weil es inzwischen selbst in den Lehrplänen für Auszubildende und Studenten im Bereich Mediendesign auftaucht, wohingegen JavaScript zumeist nur in der Informatik gelehrt wird. Nein, weil sich mit dem mobilen Internet neue Anforderungen ergeben, die andere Frameworks, wie extJS, derzeit besser abbilden. Zwar gibt es mit jQuery Mobile und jQ-Touch auch hier jQuery-Ableger, aber Qualitätsführer sind andere Bibliotheken.

Das mobile Internet

Der Desktop-Bereich bietet bekanntlich eine sehr gute Unterstützung für JavaScript. Anbieter mobiler Browser und

anatom5 perception marketing
anatom5 perception marketing GmbH
fon: +49.211.420.944.1 / fax: +49.211.420.944.3
mail: info@anatom5.de / web: www.anatom5.de

Betriebssysteme sind derweil noch nicht so weit. Sie müssen noch zahlreiche Klimmzüge unternehmen, um den immensen Funktionsumfang auch bei mäßiger bis schwacher Prozessorleistung abzubilden und dem User das bekannte Nutzererlebnis zuteilwerden zu lassen.

Während Apple mit dem mobilen Safari für iOS und Google mit einem abgespeckten Chrome-Browser für Android nahezu Desktop-Komfort in Bezug auf die Unterstützung von JavaScript, HTML5 und CSS3 bieten, müssen Sie in Bezug auf andere Plattformen deutliche Einbußen in Kauf nehmen. In der Folge kommt es zu erheblichen Darstellungsfehlern und Funktionseinbußen, gerade BlackBerry liegt hier weit abgeschlagen zurück. Interessanter sieht es, wie gesagt, bei iOS und Android aus, die in den jeweils aktuellsten Versionen einen nahezu identischen Support für die verschiedenen Webstandards bieten. Kein Wunder also, dass die Anzahl sogenannter Web-Apps stetig zunimmt. Mithilfe von PhoneGap lassen sich mittels HTML, CSS und JavaScript erstellte Anwendungen als App zusammenpacken und in die jeweiligen Stores einstellen. Der Kreativität und dem Funktionsumfang sind dabei kaum Grenzen gesetzt. Eindeutiger Vorteil: Mit nur einer App-Version können die zwei nutzerstärksten Betriebssysteme am Markt bedient werden. Es gibt aber auch eine Reihe von Nachteilen. Gerade hinsichtlich Performance, Sicherheit und Funktionsumfang bieten native Apps hier aktuell noch einen deutlichen Vorsprung. Gut möglich jedoch, dass JavaScript auch diese Bereiche eines Tages komplett abdecken wird – ohne Wenn und Aber.

Desktop-Programme

Zahlreiche Anwendungen bauen auf ECMAScript auf bzw. verwenden JavaScript, um die Funktionalität zu erweitern. Allen voran nutzt Adobe JavaScript in Produkten wie InDesign, Photoshop oder Acrobat. Das sind nicht eben typische Webanwendungen, ganz im Gegensatz zu Flex oder AIR, die ebenfalls auf JavaScript zurückgreifen. Wozu aber JavaScript in einer Anwendung wie InDesign oder Photoshop? Die Fähigkeiten der Programme können dadurch um eine leistungsfähige Skriptsprache erweitert werden. Damit lassen sich beispielsweise aus InDesign heraus mithilfe von XML und JavaScript vollautomatisch Dokumentvorlagen mit Datenbankinhalten befüllen. Diese werden direkt zur Weiterverarbeitung als PDF an Adobe Acrobat übergeben, wo ein weiteres Skript für die Weiterverarbeitung, beispielsweise zum interaktiven PDF-Formular, sorgt. Google Chrome und Opera verwenden JavaScript zur Programmierung von Erweiterungen, stets in Verbindung mit HTML und CSS. Zukünftige Standards, wie beispielsweise zur Erstellung von Widgets, setzen ebenfalls auf JavaScript als funktionalen Baustein. Dadurch verschmelzen Online- und Offline-Welt immer mehr, das Web wird zum Desktop – was Anwendungen wie Google Docs und Co. schon heute eindrucksvoll unter Beweis stellen. Erst durch JavaScript sind Cloud-Anwendungen wie beispielsweise Google Docs überhaupt möglich geworden. Ein Markt mit nahezu unbegrenztem Wachstumspotential – nicht umsonst werben Telekom, Microsoft und Apple bereits heute im Fernsehen für ihre Cloud-Dienste.

Das alte Mantra vom bösen JavaScript greift bei modernen Anwendungen nicht mehr. An die Stelle dröger Dialogboxen sind modale Fenster und Lightboxen getreten. Was geblieben ist, sind die Sicherheitsbedenken – denn wo licht ist, ist auch Schatten. Gerade AJAX und seine XMLHttpRequests bieten bei unsachgemäßer Programmierung eine Schwachstelle, einen Angriffspunkt für Hacker. Doch mit ECMAScript 5 gibt es hier einige Optimierungen, und auch in HTML5 sind bereits spürbare Verbesserungen zur Sicherheit zu finden.

Barrierefreiheit

Während JavaScript und Multimedia lange Zeit alles andere als barrierefrei waren und entsprechend stigmatisiert wurden, zeichnet sich heute ein ganz anderes Bild. Das liegt auch an der schnellen Reaktion des W3C in Form von W3C-ARIA als Antwort auf die rasanten Entwicklungen rund um AJAX und RIAs sowie die mangelnde Unterstützung assistiver Technologien. Inzwischen gibt es zahlreiche herausragende Beispiele, wie moderne Webanwendungen durch den zielgerichteten Einsatz von JavaScript und ARIA noch besser zugänglich werden können.

anatom5 perception marketing
anatom5 perception marketing GmbH
fon: +49.211.420.944.1 / fax: +49.211.420.944.3
mail: info@anatom5.de / web: www.anatom5.de

Vielseitig und Multimedial

Mash-Ups verzahnen schon heute die unterschiedlichsten Anwendungen miteinander. Zukünftig wird sich dieser Trend noch weiter verstärken. Erste Vorboten sind Anwendungen, wie Popcorn.js, mit deren Hilfe sich neue Anwendungsfälle für multimediale Inhalte und deren Anreicherung mit Zusatzinformationen erstellen lassen. Und das umso mehr, je mehr Daten in cloudbasierten Anwendungen vorgehalten werden und je mehr Cloud-Services es gibt.

Die Zukunft von JavaScript liegt wohl aber unter anderem in Sprachen und Tools, wie CoffeeScript oder Traceur, die die Skriptsprache um die besten Aspekte von Python und Ruby erweitern und somit das Programmieren für Entwickler leichter und effizienter machen. Und dennoch ist das Ergebnis am Ende JavaScript – nur eben moderner erstellt, sozusagen ein Vorgriff auf die Zukunft. Unter altjs.org ist eine Sammlung von Sprachen und Werkzeugen entstanden, die nur ein Ziel verfolgen: JavaScript besser machen und die Macht zur Weiterentwicklung von JavaScript nicht nur in die Hände von Browser-Herstellern und Großunternehmen zu legen.

Fazit

jQuery, JSON, node.js und CoffeeScript haben den Weg für das JavaScript der Zukunft bereitet: Einfacher, schlanker, spezialisierter. Dadurch werden neue Anwendungsfelder und Nutzungsszenarien entstehen, die wiederum von den Browser-Herstellern durch immer neue und immer bessere JavaScript-Engines angetrieben werden. Umgekehrt treibt die Entwicklungen im JavaScript-Umfeld die Browserhersteller weiter zu Höchstleistungen an. Erste Vorboten davon sind Anwendungen wie Hummingbird oder Word², die eine Vielzahl von Daten in Echtzeit aufbereiten – ohne Wartezeiten, ohne Reload.

Bleibt noch der Sicherheitsaspekt. JavaScript wird von zahlreichen Schnittstellen und Templating-Systemen genutzt, beispielsweise bei Facebook. Daraus resultieren natürlich Sicherheitsprobleme, gerade auch aufgrund der Fülle an Minianwendungen. Schon jetzt nutzen zahlreiche Nutzer eine Vielzahl von Facebook- oder Google+ Apps, von denen es zukünftig in den verschiedenen Portalen und sozialen Netzwerken immer mehr geben wird. Möglicherweise schlägt dann die große Stunde von Caja, einer Art Schutzschild (Sandbox) für JavaScript. Auf diese Weise können auf JavaScript aufbauende Dritt-Anwendungen sicher eingebunden werden – und damit die Akzeptanz und wohl auch Verbreitung der Skriptsprache noch weiter steigern. JavaScript ist auf dem Weg zur Lingua Franca der (Web-) Entwickler. Und das nicht, weil es die beste Sprache ist, sondern weil sie die meisten Möglichkeiten für die unterschiedlichsten Anwendungsfälle bietet und am weitesten verbreitet ist, nämlich in jedem grafischen Browser. Ob es nun der One-Pager mit sanftem Scrolling und Lightbox-Funktion für den Webdesigner ist oder die komplexe serverseitige node.js-Anwendung – JavaScript gehört ganz sicher die Zukunft.

Text: Ansgar Hein, bis Ende 2013 Mitinhaber von anatom5. (Erschienen im Screenguide Magazin Nr. 13)