

Interview mit Jonas Hellwig

# CSS und SVG – “Mittlerweile denkt man in Komponenten und Design-Modulen”



© S&S Me

Auf der **webinale** geben vom 29. Mai. bis 2. Juni international renommierte Digital-Experten und Vordenker ihr Know-How an die Teilnehmer weiter – in über 80 Workshops, Sessions und Keynotes. Als Konferenz für digitale Professionals, Trendsetter und Macher im World Wide Web ist die **webinale** die optimale Crossover-Plattform für Wissensvermittlung, Inspiration sowie Erfahrungsaustausch über alle Branchen hinweg.

Die Teilnehmer der **webinale** haben neben den klassischen **webinale**-Vorträgen auch Zugang zum gesamten Angebot der parallel stattfindenden International PHP Conference (IPC).

Da auf der webinale die Trends von morgen ebenso auf dem Programm stehen wie praxistaugliches Wissen für erfolgreiche Web- und Mobile-Projekte von heute, haben wir uns mit Jonas Hellwig unterhalten. Er ist Webdesigner, Fachbuchautor, Dozent und Speaker auf der webinale/IPC. Bezugnehmend auf seine Session "[Flat ist nicht genug! Bildbearbeitung mit CSS](#)" spricht Jonas im Interview mit Tom Wießeckel über den Trend, grafische Bestandteile einer Website mit Code umzusetzen.

**entwickler.de: Jonas, das Jahr 2015 ist fast vorbei – und die Browser nähern sich langsam einander an, was die Unterstützung der Funktionalitäten angeht. Wie siehst du die Situation aus deinem Tagesgeschäft heraus betrachtet?**

**Jonas Hellwig:** Auf den ersten Blick scheinen die neuen Browser-Generationen in der Tat die wichtigsten CSS-Eigenschaften gleich oder zumindest recht ähnlich zu interpretieren. Aber wenn man sehr moderne Techniken wie Blend-Modes, Shapes, Filter, Grid-Module-Layouts etc. einsetzen möchte, dann ist die Kompatibilität nach wie vor recht unterschiedlich.

Die Zeiten in denen die Anpassung für ältere Browsergenerationen sehr zeitaufwändig und ärgerlich war sind allerdings vorbei. Das hat verschiedene Gründe. Einerseits werden die wirklich problembehafteten Browser wie der Internet Explorer 8 oder älter nur noch selten im Projekt berücksichtigt. Viel entscheidender ist aber, dass der Workflow sich verändert hat. Wir entwickeln Websites mittlerweile aus Sicht schwacher und kleiner Geräte und erweitern dann mittels Feature Detection nach und nach neuere Funktionen und mittels Media Queries größere Layout-Varianten (Stichwort Mobile & First Progressive Enhancement). Diese Herangehensweise führt zu weniger Problemen, da wir am Anfang eines Projekts merken, wenn etwas nicht funktioniert.

Auch unsere Kunden verstehen viel besser als früher, dass eine Website nicht in jedem Browser gleich aussehen muss. Wir achten allerdings auch penibel darauf, dass sie diesen Punkt zum Projektbeginn verstanden und akzeptiert haben. Aber das macht vieles leichter. Wir können moderne

verstanden wurde, dass moderne Systeme dadurch einen Vorteil haben. Früher wurde es eher so verstanden, dass ältere Systeme einen Nachteil haben.

**entwickler.de: Auf der webinale/IPC hältst du einen Workshop mit dem Titel "Flat ist nicht genug! Bildbearbeitung mit CSS", in dem du erklärst, wie sich grafische Bestandteile einer Website mit Code umsetzen lassen. Ist das nicht ein gefährliches Unterfangen – gerade wenn man an die Browservielfalt und deren zahlreiche Unterschiede im Umgang mit den einzelnen Technologien angeht?**

**Jonas:** Teils teils. Ein Webdesigner sollte wissen wie kompatibel die CSS- oder SVG-Eigenschaften sind, die verwendet werden müssen um die gewünschte Gestaltung zu erreichen. Im Idealfall schreibt er den CSS-Code natürlich direkt selbst. Aber wenn er nicht in der Lage ist einzuschätzen, wie umsetzbar sein Design ist, dann sehe ich da ehrlich gesagt ein Problem.

Nun einmal angenommen er weiß, dass eine benötigte CSS-Eigenschaft nicht voll kompatibel mit allen gewünschten Browsern ist. Jetzt geht es darum abzuwägen wie schlimm das ist und welche Fallback-Lösungen zur Verfügung stehen. Wenn ich beispielsweise die Trennstriche in einer horizontalen Navigation mit Hilfe von `transform:skew(-45deg);` schräg stelle, weiß ich, dass der Internet Explorer 8 keine Transformationen versteht. Wenn man nun nach dem Prinzip »Progressive Enhancement« vorgeht, dann ist die Navigation in inkompatiblen Browsern gerade ausgerichtet und in Browsern die CSS-Transforms verstehen eben schräg. Das finde ich in absolut in Ordnung, solange beides für sich gut aussieht und gut funktioniert.

Bauchschmerzen habe ich bei unterschiedlichen Ergebnissen in verschiedenen Systemen, wenn das Element stilprägend ist, oder fest im Corporate Design des Kunden verankert ist. Aber das muss je nach Projekt abgewogen werden. Die Kunst besteht nicht nur darin moderne Effekte einzusetzen, sondern saubere Fallback-Lösungen einzuplanen.

**entwickler.de: Um wie vieles wird es komplizierter, wenn man dann noch das Thema Responsive Webdesign einbezieht?**

**Jonas:** Responsive Design verkompliziert die Sache zwar noch einmal – für

mich aber nicht erheblich. Es ist auch hier eine Frage der Herangehensweise und des Workflows. Da bei uns Websites größtenteils im Browser gestaltet werden, testen wir regelmäßig, wie die Gestaltung sich in den verschiedenen Systemen verhält. Dazu gehört dann natürlich auch, dass von vorne herein alle Displaygrößen und Bedienkonzepte berücksichtigt werden. Und da können sich dann schon ein paar Kombinationsmöglichkeiten ergeben. Am gerade besprochenen Beispiel der schräg gestellten Navigation gäbe es beispielsweise folgende denkbare Szenarien:

1. Der Browser versteht CSS-Transformationen. Die Navigation ist folglich schräg ausgerichtet.
2. Der Browser versteht keine CSS-Transformationen. Die Navigation ist gerade ausgerichtet.
3. Der Anwender nutzt einen Touch-Screen. Mouse-Over Effekte können nicht eingesetzt werden.
4. Der Anwender nutzt eine Mouse. Wir können Mouse-Over-Effekte verwenden.
5. Die »mobile Navigation« wird angezeigt.
6. Die »Desktop Navigation« wird angezeigt.

Wenn ich nun alle Kombinationen in Photoshop gestaltet müsste wäre der Aufwand viel zu hoch und das Projekt ruck zuck in den roten Zahlen. Wenn direkt im Browser gearbeitet wird, müssen nur die einzelnen Zustände entwickelt werden. Kombinationen ergeben sich dann automatisch, was die Arbeit erheblich erleichtert. Und darüber hinaus sehen wir sofort, wenn etwas nicht wie geplant funktioniert.

**entwickler.de: Wie erklärst du dir, dass Technologien wie SVG, die schon einige Jahre alt sind, erst jetzt durchstarten?**

**Jonas:** Ich denke das hat verschiedene Gründe. Einerseits war der Prozess, ausgehend von wenigen kompatiblen Browser hin zum Status Quo mit einer sehr guten SVG-Kompatibilität langwierig und schleichend. Ich denke viele Designer und Webentwickler kennen das Format SVG namentlich schon lange, hatten aber immer im Hinterkopf, dass SVG zu inkompatibel für den

berücksichtigt werden muss, kann SVG allerdings mehr oder weniger problemlos eingesetzt werden. Aber es gab nie die große Ankündigung »Hört her – ab sofort kann SVG verwendet werden«. Das sah vor kurzem beim Thema responsive Images (<picture> & Co.) schon etwas anders aus.

Ein weiterer Grund ist wahrscheinlich auch der Zeitpunkt zu dem SVG das Licht der Welt erblickt hat. Damals (2001) gab es Probleme wie hochauflösende Displays oder responsive Design noch nicht. Mit den Pixelbasierten Bildformaten konnten man eigentlich alles machen, was man wollte. Warum sollte man sich als Webdesigner also mit einem neuen, damals inkompatiblen Format herumschlagen, wenn es keine wirkliche Notwendigkeit dafür gab.

Zuletzt möchte ich auch nicht ausschließen, dass die Erstellung und die Bearbeitung von SVGs für einige Designer damals eine Herausforderung darstellte oder auch heute noch darstellt. Photoshop konnte lange Zeit mit SVGs gar nichts anfangen, Illustrator hatte früher deutlich schlechtere Export-Funktionen als heute. Man musste daher fast zwangsläufig nach dem Export aus Illustrator, Anpassungen am Code der SVG-Datei vornehmen. Das ist auch bei aktuellen Projekten häufig notwendig. Zu einer Zeit in der Design und Entwicklung noch stärker getrennt waren als das heute der Fall ist, war die notwendige Anpassung des Codes mit Sicherheit kein Pluspunkt für die Verwendung von SVG. Denn die Erstellung der Grafiken fiel damals klar ins Ressort des Designers.

**entwickler.de: In deinem Tagesgeschäft arbeitest du mit Illustrator und Co. – liefern die Tools mittlerweile sauberen Code, den man bedenkenlos übernehmen kann?**

**Jonas:** Es hängt zwar sehr stark von der Gestaltung ab, aber im Normalfall muss man SVG-Files nach dem Export händisch anpassen und optimieren. Man kann zwar in Illustrator mittlerweile recht viel einstellen, aber oft entstehen bei der Bearbeitung leere Gruppen (<g>) o.ä. Dargestellt werden die Grafiken allerdings auch ohne Korrektur zuverlässig.

Der SVG-Export von Photoshop erlaubt aktuell gar keine Einstellungen. Man erzeugt in Photoshop daher SVG-Grafiken die deutlich größer sind als nötig

Ebene werden beispielsweise zu einem Objekt zusammengefasst, Kurvenpunkte werden immer mit mehreren Nachkommastellen exportiert usw. Aber Photoshop ist halt auch eine Bildbearbeitungssoftware.

Beim CSS-Export sieht es folgendermaßen aus: Der CSS-Export von Photoshop selbst ist unbrauchbar. Der CSS-Export von Illustrator schon besser. Adobe Extract führt ebenfalls zu ganz passablen Ergebnissen. Sehr gute Ergebnisse erzielt man mit kommerziellen Plugins von Drittanbietern.

Ein guter CSS-Export setzt allerdings – unabhängig vom Export-Tool – voraus, dass der Designer alles so gestaltet hat, dass es sich umsetzen lässt. Auch sollte man Elemente einzeln exportieren und nicht das ganze Photoshop-Layout in den Export schicken und erwarten, dass wie von Zauberhand sauberer CSS-Code daraus wird.

### **entwickler.de: ... und wie steht es mit dem Mehraufwand, wenn man sein Design modular aufbauen möchte?**

**Jonas:** Hier ist es eher der Prozess der Umstellung, der Designern Schwierigkeiten bereitet. Früher haben wir Layouts in Unterseiten und in Geräteklassen betrachtet. Da gab es eine Startseite und eine Unterseite und diese beiden Seitentypen für Smartphone, Tablet und Desktop. Schon war man bei sechs Layouts, was bekanntlich dazu geführt hat, dass dieser Workflow heute nicht mehr effizient ist.

Mittlerweile denkt man in Komponenten und Design-Modulen. Diese werden dann auf verschiedenen Geräten oder Seitentypen unterschiedlich kombiniert. Anstelle weniger großer Dateien, arbeiten wir mit vielen kleinen Dateien und Sinnabschnitten.

Diesen Wandel hin zur modularen Denkweise spüren wir auch in fast allen Tools: Photoshop exportiert längst Image-Assets, CSS-Präprozessoren erlauben die Verschachtelung von Dateien usw. Das mag auf den ersten Blick unübersichtlicher sein, langfristig führt es aber dazu, dass man viel organisierter arbeiten kann, Aufgaben besser delegieren kann und im Team besser gemeinsam an einem Projekt arbeiten kann. Wenn der Workflow also einmal sitzt, ist man schneller als bisher.

**experimentell – wann denkst du, wird die Browserlandschaft so weit sein, dass man mit einer einheitlichen Basis ohne Hacks und Workarounds identische – oder zumindest gleichwertige – Ergebnisse erreicht?**

**Jonas:** Ich denke nicht, dass die Browser irgendwann auf einem einheitlichen Stand der Technik sind. Wir werden um Workarounds, Hacks, Vendor-Präfixe etc. also nicht herumkommen. Doch bereits heute ist es ja so, dass es zahlreiche Tools und Task-Runner gibt, die diese notwendigen Anpassungen für uns übernehmen. Früher musste man beispielsweise noch nachschauen welche Vendor-Präfixe für eine bestimmte CSS-Eigenschaft benötigt werden. Anschließend musste man sie schreiben.

Heute übernimmt diese Aufgabe ein Postprozessor der – wie im Falle des Autoprefixers den CSS-Code mit der Datenbank von CanIUse abgleicht und die Präfixe automatisch bei der Kompilierung des Sass/Less-Files hinzufügt. Schöne automatisierte Welt.

**entwickler.de: Abschließend eine Frage, die vielen unter den Nägeln brennt: Ist Microsoft noch immer das größte Sorgenkind für Webdesigner?**

**Jonas:** Webdesigner lieben es gegen den verhassten Internet Explorer zu wettern. Da ginge also ein Stück Kultur verloren, wenn dieser Browser plötzlich keine Probleme mehr bereiten würde. Aber Spaß beiseite. Es ist zwar enttäuschend, dass ein moderner Browser wie Edge keine SVG-Animationen mit SMIL unterstützt, aber gebessert hat sich die Microsoft-Fraktion in den letzten Jahren schon.

Ich persönlich habe mit dem Internet Explorer bis einschließlich Version 8 wenig Probleme. Vorausgesetzt man hat nicht den Anspruch, dass alles pixelgenau überall gleich aussieht.

Mobile iOS- oder Android-Browser sind aus meiner Erfahrung viel schlimmer und auch moderne Versionen von Chrome, Firefox und vor allem Safari können sehr lästige Bugs besitzen. Fehler in diesen Browsern sind viel schwieriger zu beheben, da sich die Systeme nicht so leicht gezielt ansprechen lassen



*(Foto: Robert Kleine, Rheinwerk Verlag)*

**Zur Person:** Jonas Hellwig arbeitet als Webdesigner, Videotrainer, Fachbuchautor und Dozent in Berlin. Neben Responsive Design hat er sich auf WordPress, Photoshop und Workflowoptimierungen im Webdesign spezialisiert und ist Gründer der Agentur kulturbanause®. Auf

[blog.kulturbanause.de](http://blog.kulturbanause.de) veröffentlicht er interessante Artikel, Tipps und Tutorials zum Thema Webdesign. Jonas Hellwig ist regelmäßig als Sprecher auf Veranstaltungen zum Thema Webdesign anzutreffen und unterstützt Webagenturen und Unternehmen bei der Planung und bei der technischen Umsetzung von Webprojekten.