

Präprozessoren für CSS

EXPOSÉ BACHELORARBEIT 2

Studierende/Studierender Barbara Huber

, 1010601010

MultimediaTechnology Fachhochschule Salzburg

Salzburg, am 01. Februar 2015

Inhaltsverzeichnis

1. Zielsetzung	1
2. Forschungsfrage	2
3. Methoden	2
4. vorläufige Gliederung	3
4.1. Basics zu Präprozessoren	3
4.2. Parser	4
4.3 Implementierung	4
4.3.1 Less	5
4.3.2 Sass	5
4.3.3 Stylus	5
Literaturverzeichnis	6
Abkürzungsverzeichnis	7

1. Zielsetzung

Ein wichtiger Teil von Webseiten, ist die Gestaltung und das Design. Um eine Seite nach dem Design zu gestalten, verwendet man im World Wide Web das CSS 3. Seit ein paar Jahren, werden sogenannte Präprozessoren für CSS entwickelt, die Erweiterungen zu CSS bereitstellen. Was sind nun Präprozessoren? Wie werden sie verwendet und welche Erweiterungen stellen Sie im Detail zur Verfügung?

Diese Fragen sollen in der Arbeit behandelt werden. Mit CSS3 ist schon eine große Vielfalt an Gestaltungen möglich. CSS3 bietet einfache Berechnungen und es gibt Funktionen, mit denen Beispielsweise Die Ecken einer Box abgerundet werden können oder ein Hintergrund einen Verlauf haben kann.

Die Notwendigkeit, der Verwendung von Präprozessoren für CSS, wird immer größer, da die Ansprüche an die Gestaltung von Webseiten immer mehr Bedeutung bekommt. Es reicht nicht mehr, dass eine Webseite verschiedene Hintergrundfarben hat oder in Boxen aufgeteilt ist. Die Verwendung dieser Präprozessoren, welche Präprozessoren es gibt und was Präprozessoren sind, möchte ich in meiner Bachelorarbeit 2 untersuchen. Ein wichtiger Bestandteil der Arbeit wird die Definition von Präprozessoren und wie diese funktionieren. Nach ersten Recherchen, fand ich heraus, dass es einige verschiedene CSS-Präprozessoren gibt, die das gemeinsame Ziel verfolgen, die Gestaltung einer Webseite zu erleichtern und Erweiterungen für CSS3 bereit zu stellen.

Im Zuge der Bachelorarbeit 2 werde ich im theoretischen Teil die aus meiner Recherche hervorgegangenen Tools, Less, Sass und Stylus und deren Implementierung genau erläutern. Die Anwendungen werden im praktischen Teil getestet und mithilfe eines Parsers, soll die Implementierung und die Funktion der Präprozessoren erläutert werden.

Ziel der Arbeit ist es, die genannten Möglichkeiten genau zu erforschen, um dadurch die Vor- und Nachteile dieser zu eruieren und bestimmte Kriterien auf Performance, Handhabung und Einfachheit zu vergleichen.

2. Forschungsfrage

In dieser Arbeit soll die Frage:

Welche Präprozessoren gibt es und wie funktionieren diese? Wie funktionieren Präprozessoren im generellen?

beantwortet werden.

Die Relevanz dieser Frage sehe ich im zunehmenden Interesse, Webseiten immer besser und stylisher zu gestalten. Die Webseiten sollen immer innovativer und kreativer werden und oft reicht es nicht mehr, mit den von CSS3 gebotenen Möglichkeiten auszukommen.

In dieser Arbeit soll nun, wie schon unter Punkt 1 erläutert, untersucht werden, welche Tools bereitgestellt werden, um CSS3 zu erweitern und den Ansprüchen zu entsprechen. Am Beginn der Arbeit möchte ich kurz erläutern wie sich Präprozessoren definieren und welchen Anteil am Gesamten die einzelnen Bestandteile, wie z.B. Mixins und Vererbung, haben und wie diese zusammenwirken. Im nächsten Schritt werde ich auf die einzelnen Methoden eingehen um im Anschluss, im praktischen Teil der Arbeit, die Vergleiche anzustellen.

Ich denke, dass sich der praktische Teil und der theoretische Teil vom Arbeitsaufwand in der Waage halten, da es auch beim praktischen Teil einiges zu recherchieren gibt.

3. Methoden

Wie in Punkt 2 beschrieben werde ich in der Arbeit zuerst auf die Grundthematik Präprozessoren eingehen und beschreiben, wie Präprozessoren im Gesamten und die Bestandteile im Einzelnen entstehen und welche Auswirkungen Präprozessoren auf die Erstellung eines Webitenstylung haben. Das Kapitel dient zur Information der Leser/innen und um die technischen Weiterführungen der Präprozessoren in den weiteren Kapiteln besser verstehen zu können. Im Weiteren wird in der Arbeit erläutert, was Parsing ist und wie das im speziellen bei Präprozessoren funktioniert. Nach dieser Einführung werden die, in Punkt 1 genannten Tools genauer betrachtet und deren Benützung und Implementierung beschrieben. Dieser Teil der Arbeit wird zum größten Teil aus Recherche bestehen. Schwierigkeiten bei der Recherche sehe ich darin, dass die meisten meiner Quellen auf Englisch verfasst sind und ich daher sehr aufpassen muss, dass ich die Literatur richtig verstehe und dementsprechend verwende. Durch die Tatsache, dass auch für die Umsetzung des praktischen Teiles viel Recherche notwendig ist, wird dies den Großteil des Zeitmanagements in Anspruch nehmen.

Nach dem derzeitigen Stand meiner Recherche und genaueren Überlegungen anhand meines praktischen Teils kam ich zu dem Entschluss, mich in dieser Arbeit vor allem auf die Funktion von Parsing und Linting zu konzentrieren.

Ziel der Arbeit ist es, herauszufinden, was Präprozessoren sind und wie sie funktionieren. Um das herauszufinden muss ich wissen, wie aus den CSS-Präprozessoren normales CSS wird und wie die CSS-Präprozessoren arbeiten.

4. Vorläufige Gliederung

Nach bisherigen Recherchen zufolge, setzt sich meine Gliederung wie folgt zusammen: Kurze Einführung in die Thematik von Präprozessoren, Erläuterung des Begriffes Parsing und Beschreibung der Präprozessoren in Bezug auf CSS. Danach folgt eine Erklärung von Less und Sass. Im Besonderen deren Verwendung und Relevanz für meine Arbeit. Im weiteren Verlauf der Arbeit wird zu den genannten Tools noch Stylus genau betrachtet.

4.1 Präprozessoren

Präprozessoren sind Computerprogramme, welche Daten vorbereiten und zur Weiterverarbeitung an ein anderes Programm weitergeben. In den meisten Fällen, wird ein Präprozessor dazu verwendet, Eingabedaten, im Falle dieser Arbeit CSS-Styles, zu konvertieren. (Peter2012)

Präprozessoren werden verwendet um beispielsweise Variabilität zu schaffen. In dieser Arbeit geht es um die CSS-Präprozessoren Sass, Less und Stylus. Diese Präprozessoren werden verwendet, um das Schreiben des Codes zu erleichtern. Die Präprozessoren erleichtern die Syntax und stellen Funktionen und Variablen zur Verfügung. Mithilfe der genannten Präprozessoren können Aufgaben automatisiert werden. Es gibt die Möglichkeit Variablen zu erstellen und so die Bearbeitung von Stylings wie z.B. der Farbe, um ein Vielfaches zu erleichtern. Möchte man in einem CSS file die Farbe der Schrift verändern, muss man alle Stellen suchen, an denen diese Farbe zugeordnet wird. Verwendet man beispielsweise mit less eine Variable für die Farbe, so muss nur an einer Stelle, dort wo die Farbe der Variablen zugewiesen wird, die Farbe geändert werden.

Für die weitere Arbeit soll dieses Kapitel als Information zu den technischen Gegebenheiten eines Präprozessors dienen.

4.2 Parser

Der Parser ist ein Programm zur Zerlegung und Umwandlung einer beliebigen Eingabe, welche zur Weiterverarbeitung in ein brauchbares Format umgewandelt wird. Ein Parser erzeugt zusätzliche Strukturbeschreibungen. Parsing bedeutet Syntaxanalyse.

Der Parser verwendet zur Analyse von Text einen lexikalischen Scanner, auch Lexer genannt. Ein Lexer zerlegt die Eingabe in sogenannte Tokens (beispielsweise Wörter oder Eingabesymbole) die der Parser versteht.

Bei einem HTML Code würde der Lexer die Datei in HTML-Tags und Fließtext zerteilen und so an den Parser weiterleiten. Den Lexer interessiert nur das Aussehen der Syntaxelemente wie z.B. die spitzen Klammern eines Tags. Der Parser verarbeitet dann die syntaktischen Zusammenhänge.

Dieses Kapitel soll veranschaulichen und erklären, wie aus einer style.less eine style.css werden kann.

4.3 Die Implementierung

Um einen CSS-Präprozessor verwenden zu können, benötigt man bestimmte Tools oder Programme. Für die Verwendung von Präprozessoren und deren Erweiterungen werden einige Tools bereitgestellt, die das ermöglichen. In dieser Arbeit werden 3 dieser Tools näher betrachtet. Less, Sass und Stylus. Wie diese Tools anzuwenden sind und funktionieren, wird in den folgenden Kapiteln genauer betrachtet.

4.3.1 Less

Less ist einer der genannten CSS-Präprozessoren und wird in diesem Kapitel genauer erklärt.

Entwickelt wurde und wird Less von Alexis Sellier. Die Entwicklung begann im Jahr 2010 und geht immer weiter. Less ist wie auch die Syntax-Variante SCSS von Sass ein Superset und kann somit ohne Probleme in eine CSS Datei eingefügt werden. Verwendet man Less, kann man also bei einer CSS Datei einfach die Endung in .less umändern und man hat eine funktionstüchtige Less-Datei in der man die zusätzlichen Eigenschaften von Less einbauen und verwenden kann.

Less verwendet dieselbe Syntax wie CSS und bietet wie auch Stylus und Sass die Möglichkeit der Verschachtelung und der Verwendung von Features wie Variablen, Mixins, Vererbung und Funktionen.

4.3.2 Sass

Sass ist eine Erweiterung von CSS3, welche Variablen, Mixins, Selectoren, Funktionen und andere Erweiterungen anbietet. Somit ist Sass ein Präprozessor, welcher schon in einem früheren Kapitel erklärt wurde, von CSS.

Einer der größten Vorteile eines CSS-Präprozessors ist, dass man in der Entwicklung verschiedene Dateien haben kann, ohne Performance einzubüßen.

Um Sass in CSS umzuwandeln, verwendet man entweder die Kommandozeile oder eines von verschiedenen web-frameworks welche die notwendigen Funktionen bereitstellen.

Für Sass gibt es zwei verschiedenen Syntaxen. Die Ursprüngliche Syntax verwendet die Dateierweiterung `.sass` und verwendet Einrückungen statt der geschweiften Klammern um die Verschachtelung der Selektoren anzuzeigen und Zeilenumbrüche statt eines Semicolons um die Eigenschaften zu trennen.

Die neuere Syntax verwendet die Dateierweiterung `.scss`. Hier werden im Gegensatz zur alten Syntax wieder Klammern und Semicolon verwendet. (Yard 2014)

4.3.3 Stylus

Stylus ist ein CSS-Präprozessor der mit Nodejs läuft. Die Dateierweiterung von Stylus ist `.styl`. Mit dem Befehl `npm install stylus -g`, lässt sich stylus über nodejs installieren und verwenden.

Die Syntax von Stylus ist ähnlich der von Sass. Wie auch bei Sass, können bei Stylus die Klammern und die Semicolons weggelassen werden.

Verwendet man diese Syntax ist Stylus kein Superset von CSS und nicht rückwärtskompatibel.

Die Schreibweise der Syntax ohne Klammern und Semicolons ist optional und es kann auch die normale CSS-Syntax angewandt werden.

Literaturverzeichnis

Peter, Christian (2012): C-Präprozessoren, 2012. Online verfügbar unter http://wr.informatik.uni-hamburg.de/_media/teaching/wintersemester_2012_2013/epc-1213-peter-praeprozessor-presentation.pdf, zuletzt geprüft am 26.11.2014.

Yard (2014): Sass. Documentation. Online verfügbar unter http://sass-lang.com/documentation/file.SASS_REFERENCE.html#syntax, zuletzt geprüft am 24.11.2014.

Abkürzungsverzeichnis

JS - JavaScript

MMP3 - Multimediaprojekt 3

Bzw. - Beziehungsweise