Sander Karsten 568999| Ege Saraydar 547873|Niels Kooij 579211|Ruben Jansen 540427|Pasa Uzun 570974

*Docent: Peter Cornelissen (CNLP)*

*Klas: i1tb*

*Locatie: HAN-Arnhem*

*Datum: 7-11-2016*

*Versie: 1.10*

Onderzoeksrapport CSS-Preprocessors

Webdevs

Inhoudsopgave:

[Hoofdstuk 1 inleiding 4](#_Toc447189886)

[Hoofdstuk 2 Wat is een CSS Preprocessor? 5](#_Toc447189887)

[Hoofdstuk 3 Welke elementen bij Front End Development zijn arbeidsintensief / herhalend / complex? 6](#_Toc447189888)

[Hoofdstuk 4 Voor welk probleem is een CSS preprocessor een oplossing? 7](#_Toc447189889)

[Hoofdstuk 5 Wat is het verschil tussen een CSS preprocessor en CSS? 8](#_Toc447189892)

[Hoofdstuk 6 Welke CSS preprocessoren zijn er te verkrijgen? 9](#_Toc447189893)

[Hoofdstuk 7 Welke criteria zijn er te stellen aan CSS preprocessors? 10](#_Toc447189895)

[Hoofdstuk 8 Welke werkwijze wordt er verondersteld/vereist? 12](#_Toc447189896)

[Hoofdstuk 9 Wat zijn de tekortkomingen van de huidige CSS preprocessor? 13](#_Toc447189897)

[Hoofdstuk 10 Hoe is het ‘gebruikersgemak’ vanuit het perspectief van een ontwikkelaar? 14](#_Toc447189898)

[Hoofdstuk 11 Wat zijn de verschillen/overeenkomsten tussen de gevonden bronnen uit je categorie? 15](#_Toc447189899)

[Conclusies & Aanbevelingen 16](#_Toc447189900)

[Literatuurlijst 17](#_Toc447189901)

[Bijlagen 18](#_Toc447189902)

# Hoofdstuk 1 inleiding

Dit onderzoeksrapport is geschreven in opdracht van het vak webdevs. In dit rapport zullen wij onderzoek doen naar welke CSS preprocessor het meeste effect heeft op de productiviteit van FED bij ICA.

FED, oftewel front end development is het produceren van HTML, CSS en JavaScript voor een website. Het produceren van CSS kan met verschillende tools en technologieën versimpeld worden. Een CSS preprocessor is een van die tools/technologieën, wij zullen de voor- en nadelen belichten van een CSS preprocessor. Na verschillende preprocessors met elkaar te hebben vergeleken beslissen we welke het meeste effect heeft op de productiviteit van FED bij de ICA. Onze hoofdvraag luidt dan ook:

***“Welke css preprocessor heeft het meeste effect op de productiviteit van FED bij ICA?”***

# Hoofdstuk 2 Wat is een CSS Preprocessor?

CSS is primitief en incompleet. Het maken van functies, het hergebruiken van definities en overerving zijn moeilijk voor elkaar te krijgen. Voor grote projecten en complexe systemen wordt dit al snel heel ingewikkeld te bouwen en onderhouden. Ook gaan ontwikkelingen in de web development razendsnel waardoor je vaak browser specifieke code moet schrijven om je website in alle browsers goed te weergeven. Dit vergt een hoop tijd en veroorzaakt een hoop frustratie.

Om beter CSS te kunnen maken kan men gebruik maken van een CSS preprocessor. Een CSS preprocessor is een soort scripting taal die je mixt met normale CSS. Het geschreven script wordt gecompileerd en zo vertaald naar normale CSS die je browser kan gebruiken. Op deze manier kun je op een flexibele manier code schrijven die door elk type browser ondersteund wordt. Ook is het makkelijker om leesbare code te schrijven en je kunt gemakkelijk stukjes code delen.

CSS preprocessors geven ook de mogelijkheid om if-else statements te gebruiken in je CSS en zelfs wiskunde. Ook kunnen bijvoorbeeld kleuren dynamisch worden aangepast afhankelijk van bepaalde eigenschappen in plaats van het in de CSS hard te moeten coderen. Verder zijn er vele snippets en bibliotheken voor veel voorkomende problemen die je kunt importeren en gebruiken in je eigen project.

Kortom een CSS preprocessor is eigenlijk CSS op steroïden. De mogelijkheden worden gigantisch vergroot en het maakt het een stuk makkelijker om dynamisch en voor alle browsers css te maken.

# Hoofdstuk 3 Welke elementen bij Front End Development zijn arbeidsintensief / herhalend / complex?

Bij front end web development komt een hoop kijken. Zo zijn er een heleboel verschillende script talen, frameworks, processors, etc. die binnen een project gebruikt worden. Ook moet worden gelet op de verschillende browsers en schermgroottes die de gebruiker kan hebben.

Projecten kunnen een framework of een scripttaal gebruiken die je als ontwikkelaar nog niet eerder gebruikt hebt. Van tevoren weet je vaak niet hoeveel tijd en moeite dit gaat kosten. De ene taal, framework of tool is een stuk makkelijker te leren dan de ander. Hier moet dan ook redelijk wat tijd in worden gestoken om goed onder de knie te krijgen en te kunnen toepassen in het project.

Ook de manier waarop dingen worden toegepast is niet altijd hetzelfde en kan enige tijd vergen om onder de knie te krijgen. Je kunt bijvoorbeeld twee systemen hebben die je samen moet gebruiken terwijl dat nog niet eerder is gedaan.

Verder is het lastig om een product te maken dat op alle verschillende browsers en verschillende schermgroottes werkt. Hiervoor moet je vaak nagaan of wat je aan het doen bent ook echt werkt op elke browser en misschien zelfs een browser specifieke oplossing implementeren.

Het moeilijke aan front end web development is dat het constant aan het ontwikkelen is. Hierdoor moet je veel leren en de nieuwste trends blijven volgen om te kunnen concurreren.

# Hoofdstuk 4 Voor welk probleem is een CSS preprocessor een oplossing?

Een CSS preprocessor is een verandering/verbetering aan CSS, maar wat is nou eigenlijk het probleem met CSS?

## Herhaling variabelen:

CSS bevat te veel herhalingen, bij programmeren is het de bedoeling zo min mogelijk te herhalen. Er wordt bijvoorbeeld een keer een variabele gedeclareerd die steeds kan worden hergebruikt. Stel je hebt een kleur gevonden en wil deze meerdere keren in je CSS gebruiken dan moet je steeds de hele kleurcode opschrijven. Als je deze kleurcode nu weer wil veranderen moet je op zoek gaan naar de oude kleurcodes in de CSS en deze allemaal weer gaan vervangen naar de nieuwe kleurcode. Met een CSS preprocessor hoef je maar een keer een variabele te declareren en kan deze steeds weer hergebruikt worden, dit voorkomt herhaling van dezelfde waardes.

## Herhaling stukjes CSS:

Een andere vorm van herhaling die veel in CSS optreedt, is het steeds weer declareren van eigenschappen. In CSS preprocessors kan je stukjes CSS steeds weer hergebruiken.

Voorbeeld:

CSS:

.title {

Font-weight: bold;

Color: red;

Font-size: 23px;

}

.sub-title {

Font-weight: bold;

Color: red;

Font-size: 14px;

}

CSS preprocessor:

.title {

Font-weight: bold;

Color: red;

Font-size: 23px;

}

.sub-title {

.title;

Font-size: 14px;

}

(De sub-title zal alle eigenschappen van title overnemen alleen de font-size zal nu overschreden worden met een kleinere font-size)

In CSS zouden alle eigenschappen opnieuw gedeclareerd moeten worden. Het probleem dat een CSS preprocessor voorkomt, is het feit dat CSS te veel herhalingen bevat.

# Hoofdstuk 5 Wat is het verschil tussen een CSS preprocessor en CSS?

Om het globaal te houden, kunnen we zeggen dat een CSS preprocessor in principe een uitbreiding is op de huidige CSS in de vorm van een applicatie of programma. Deze applicatie of programma zorgt voor het omzetten van de “scripted” taal naar de normale CSS syntax wat de browser verwacht te zien zodat de browser de omgezette CSS kan laten zien. Kort gezegd is er nauwelijks verschil tussen CSS en de CSS preprocessor. Het enige wat de CSS preprocessor doet is het compileren van de code naar bruikbare CSS syntax.

Er zijn natuurlijk wel verschillen tussen de CSS preprocessors die nu verkrijgbaar zijn. In bron 10 staan alle verschillen tussen de CSS preprocessors uitgewerkt.

# Hoofdstuk 6 Welke CSS preprocessoren zijn er te verkrijgen?

Een CSS-preprocessor is handig om mee te werken, maar welke CSS-preprocessors bestaan er nou eigenlijk?

Bij deze vraag is het in principe heel simpel om tot een conclusie te komen. De processors die hier behandeld zullen worden zijn tevens de populairste CSS-preprocessors die er op dit moment verkrijgbaar zijn.

## CSS-Preprocessors:

1. LESS CSS

Less is waarschijnlijk de meest bekende CSS preprocessor. Less maakt het gebruikt van vereenvoudigde syntax en variabelen. Ook bevat het een uitbreiding voor het creëren van het cachegeheugen stylesheets voor eventuele PHP projecten.

1. SASS  
   Op de website, beweert SASS CSS weer leuk te maken. Het feit dat SASS veelbelovend is, maakt het een stuk interessanter. Zoals minder CSS en het gebruik van variabelen en een vereenvoudigde syntax. Sass is zeker een geweldig hulpmiddel maar helaas alleen beschikbaar voor de programmeertaal Ruby.
2. TURBINE

Dit is een geschikte processor voor mensen die regelmatig met PHP aan het werk gaan. Turbine ziet er goed uit. Deze preprocessor ondersteund minimale syntax, automatische gzip, en het oplossen van kwesties met cross-browser. Turbine is dus zeer geschikt voor PHP gebruikers.

1. SWITCH CSS  
   Switch is een volledig uitgeruste, productie klaar CSS preprocessor. Het draait onder Apache met de mod: python, of als een environment agnostic command line tool.
2. STYLUS

Stylus is een ander CSS preprocessor dat al een sterke concurrent is tegen Sass en dus ook een kenmerk-rijke uitbreiding van CSS is. Het heeft meer dan 60 aangepaste functies, waarmee u wiskundig de verzadiging van de CSS code kunt aanpassen. Bij de Stylus syntaxis kunt u onder andere accolades weglaten ({} ) , dubbele punten (:) en puntkomma's ( ;), of je kunt gewoon normale CSS gebruiken wat het gebruik van Stylus een stuk efficiënter maakt.

De meeste Preprocessors ondersteunen dezelfde kenmerken wat het dus ook identiek maakt. Naast deze CSS-preprocessors zijn er nog meer verkrijgbaar waaronder CSS-Crush, Myth en nog vele andere.

# Hoofdstuk 7 Welke criteria zijn er te stellen aan CSS preprocessors?

Met deze vraag willen we de criteria bepalen waarop we de CSS-preprocessors gaan beoordelen.

De criteria die wij bepaald hebben zijn:

Met betrekking tot het besturingssysteem:

• De CSS-preprocessor moet op verschillende besturingssystemen te gebruiken zijn.

Dit criterium is om de software die de opdrachtgever gebruikt zonder verdere aanpassingen te kunnen gebruiken.

Met betrekking tot het leren van de CSS-preprocessor

• Moet een steile leercurve hebben. (in 2 weken moet je ermee kunnen werken).

• Moet leermiddelen bevatten (voor als mensen er dieper in willen duiken).

Deze criteria zijn om te bepalen welke het makkelijkste te leren is en de minste aanpassingen in de huidige werkwijze vereist vanuit het perspectief van de personen die er mee moet gaan werken.

Met betrekking tot het onderhoud van het systeem.

• De processor moet elke 6 maanden geüpdatet worden.

• De hoeveelheid / functionaliteiten van de CSS-preprocessor.

• De support moet Nederlands- of Engelstalig zijn.

• De support moet op 1 plek te vinden zijn.

Deze criteria zijn opgenomen om de actualiteit te bepalen en om te bepalen of het op toekomstige veranderingen kan inspelen.

De ondersteuningscriteria zijn om te bepalen of er binnen de ICA (informatie communicatie academie)mee gewerkt kan worden. De ondersteuning op een locatie criterium is om te voorkomen dat het hele internet doorgespit moet worden.

Met betrekking tot code kwaliteit

• De code moet clean zijn (weinig tekens die het lezen bemoeilijken, logische structuur enz.).

• De uiteindelijke CSS code moet duidelijk zijn (geen vreemde structuren logische volgorde enz.).

Deze criteria zijn gesteld om de kwaliteit van de code te bepalen. In het schrijven en de uiteindelijke CSS code.

Met betrekking tot gebruikersgemak

• Er moet een API beschikbaar zijn.

• De API moet up to date zijn.

• Moet duidelijke foutmeldingen geven.

• De suport moet op 1 plek te vinden zijn

De criteria met betrekking op de API zijn gesteld om de gebruiker een mogelijkheid te geven om de syntax van functies op te zoeken.

Het criterium over foutmeldingen is gesteld om onderscheid te kunnen maken in hoe behulpzaam de foutmeldingen zijn als er een fout in de code zit.

Met betrekking tot gebruikseisen voor de ICA

• De CSS-preprocessor moet gratis zijn.

Dit criterium is gesteld omdat het financieel moeilijk is om voor een grote groep studenten een CSS preprocessor aan te schaffen, voor de relatief korte tijd die ze binnen de organisatie studeren.

• De CSS-preprocessor moet bij de ICA passen.

Reden voor dit criterium is dat de mensen die er gebruik van maken het idee moeten krijgen dat het bij de huisstijl van het bedrijf waar ze werken hoort.

• De CSS-preprocessor biedt mogelijkheden voor het gebruik van plug-ins.

Dit criterium is om te bepalen of de CSS-preprocessor gemakkelijk aangepast kan worden in de toekomst.

# Hoofdstuk 8 Welke werkwijze wordt er verondersteld/vereist?

Deze vraag is bedoeld om te kijken of het gaan gebruiken van een CSS-preprocessor een aanpassing van de huidige werkwijze vereist.

Omdat het onmogelijk is om in de beschikbare tijd voor elke CSS-preprocessor te bepalen of er een verandering in werkwijze nodig is, hebben we gekozen om naar 3 CSS-preprocessors te kijken. De drie die we gekozen hebben zijn: Less, SASS en Stylus. De reden om voor deze 3 CSS-preprocessors te kiezen is dat deze bij een google zoekopdracht hoog in de resultaten stonden.

Op de website van less (organisatie Less, Z.D.) onder Using less staat dat less gebruikt kan worden als command-line tool (e.g. MS-DOS (Bron Magaret,2015) via npm en in een groot aantal tools van derden.

Omdat less werkt met een groot aantal tools van derden is voor less is dus geen aanpassing in de werkwijze nodig.

In de file sass\_reference op de website van SASS (organisatie Sass, Z.D.) onder het kopje Using Sass staat dat het op 3 manieren gebruikt kan worden. Sass kan gebruikt worden als: commandline-tool, een standalone tool of als plug-in bij bepaalde frameworks.

Omdat er 3 manieren zijn om met Sass te werken is er net als voor less geen aanpassing in de werkwijze nodig

Op de stylus website (organisatie stylus, Z.D.) staat geen specifieke werkwijze vermeld.

Omdat er op de stylus website geen specifieke werkwijze vermeld wordt neem ik aan dat er ook geen specifieke werkwijze vereist wordt bij het werken met Stylus.

Omdat voor zowel less, sass als stylus geen werkwijze vereist word kan er volgens mij gezegd worden dat er voor de meeste CSS preprocessors geen specifieke werkwijze vereist wordt.

# Hoofdstuk 9 Wat zijn de tekortkomingen van de huidige CSS preprocessor?

Over het algemeen wordt de CSS preprocessor beschouwt als een tool met veel voordelen. Toch zijn er een aantal tekortkomingen aan de huidige CSS preprocessor.

In een CSS preprocessor is het moeilijk te overzien hoe de CSS eruit gaat zien. Zodra de CSS wordt gegenereerd zal het veel groter en langer worden dan verwacht. Doordat dit overzicht er niet is, kan je ook niet weten hoeveel impact het kan hebben op de performance, meer CSS zorgt immers voor meer werk voor de computer.

Debugging wordt lastiger, de browser geeft bij foutmeldingen de regelnummers van de CSS. De foute regels in de CSS preprocessor worden niet gegeven. Doordat de foute regels moeilijker gevonden kunnen worden wordt debuggen lastiger.

Zodra je hebt gewerkt in een CSS preprocessor moet dit ook weer gecompileerd worden en compileren kost tijd.

Tekortkomingen in CSS zijn dus het niet kunnen inschatten van de bestandsgrootte van de CSS en problemen met het debuggen en compileren.

# Hoofdstuk 10 Hoe is het ‘gebruikersgemak’ vanuit het perspectief van een ontwikkelaar?

CSS preprocessors bieden veel extra functionaliteit en overzicht toe aan het programmeren met CSS, maar hoe makkelijk is het voor een ontwikkelaar om een CSS preprocessor taal te beheersen? Er komen veel dingen kijken bij het bepalen of een CSS preprocessor gemakkelijk te beheersen is, zoals support van de ontwikkelaars en gebruikers van de CSS preprocessor. Een handleiding in meerdere talen, en een overzichtelijke duidelijke lay-out voor de programmeeromgeving zijn ook erg belangrijke factoren. En natuurlijk de gemiddelde leercurve van de CSS preprocessor code.

Om het gebruiksgemak te bepalen gaan we naar de drie meest gebruikte preprocessors kijken: sass, less en sylus.

Na gebruik van alle drie de preprocessoren blijkt dat de code erg snel te beheersen valt bij alle drie de preprocessoren omdat de standaard CSS code ook gewoon zonder problemen ingevoerd kan worden in de preprocessoren. Hierdoor hebben preprocessoren een erg snelle leercurve voor ontwikkelaars die bekent zijn met standaard CSS code. Ontwikkelaars die nog geen of weinig ervaring hebben met CSS zullen natuurlijk een langere leercurve hebben, maar nog steeds binnen korte tijd de code kunnen beheersen.

De lay-outs van de preprocessoren zijn erg overzichtelijk en de navigatie tussen opties en schermen is snel te beheersen. Verder hebben de preprocessoren ook duidelijke handleidingen op hun websites staan die beginners goed kunnen helpen. Er zijn ook veel forums waar je terecht kan bij het leren van een nieuwe preprocessortalen, en vooral voor de populaire preprocessors is er veel te vinden.

We kunnen na ons onderzoek concluderen dat het gebruiksgemak erg goed is voor ontwikkelaars door de uitgebreide handleidingen op het web, de duidelijke design van de preprocessors, en de grote overeenkomsten met standaard CSS code.

# Hoofdstuk 11 Wat zijn de verschillen/overeenkomsten tussen de gevonden bronnen uit je categorie?

De overeenkomsten van onze bronnen zijn dat bij de meeste bronnen meerdere preprocessoren met elkaar vergelijkt worden. De meeste bronnen geven ook mee dat ontwikkelaars zelf een keuze moeten baseren op eigen prioriteiten en dat bij elke gebruiker andere dingen fijner werken. De meeste bronnen zijn geschreven met specificaties en experimenten als bron. De verschillen zijn dat er in sommige bronnen meer nadruk wordt gelegd op gebruiksvriendelijkheid van de preprocessors en in andere bronnen meer naar functionaliteit. De schrijvers van de bronnen zijn ook verschillend. De ontwikkelaars van de preprocessors hebben natuurlijk andere soorten belangen bij het maken van deze bronnen dan normale gebruikers van preprocessoren (het adverteren van hun product).

# Conclusies & Aanbevelingen

Een CSS preprocessor is dus een verbetering aan het huidige CSS dat veel te veel herhalingen bevat, ook is een CSS preprocessor erg gebruiksvriendelijk. Om te werken met CSS preprocessors moet je constant alert zijn op nieuwe veranderingen, FED en ook CSS preprocessors zijn constant aan het ontwikkelen. Voor de overstap van CSS naar CSS preprocessors moet er altijd wel op gelet worden dat debugging lastiger wordt en dat het moeilijk te overzien kan zijn hoe groot je CSS wordt.

Er zijn vijf populaire CSS preprocessors LESS CSS, SASS, TURBINE, SWITCH CSS en Stylus. Na verschillende criteria opgesteld te hebben als de leercurve, onderhoud, codekwaliteit enz. hebben wij geconcludeerd dat Stylus het meeste effect heeft op de productiviteit van FED bij de ICA. Dit is dus ook ons antwoord op de hoofdvraag: ***“Welke css preprocessor heeft het meeste effect op de productiviteit van FED bij ICA?”*** Stylus voldeed aan de meeste criteria en had technisch ook de meeste mogelijkheden.

# Literatuurlijst

1. Hawe, S. (z.d.) Advanced html-css preprocessors. Geraadpleegd op 30 maart 2016, van: <http://learn.shayhowe.com/advanced-html-css/preprocessors/>
2. Bilal Cinarli (21 maart 2014). An Introduction to CSS Pre-Preocessors: SASS, LESS and Stylus. Geraadpleegd op 27 maart 2015, van  
   <http://htmlmag.com/article/an-introduction-to-css-preprocessors-sass-less-stylus>
3. Futekov, A. (z.d.).Comparison of support for selected features. Geraadpleegd op 29 maart 2016, van: [https://www.csspre.com/compare/](https://www.csspre.com/compare/%20)
4. Fred (29 januari 2015). Complexity in Web Development: A simple guide  
   Geraadpleegd op 27 maart 2015, van  
   <http://wpbusinesstips.com/2015/01/complexity-web-development-simple-guide/>
5. Trent, J. (14 november 2012). Cons of CSS Preprocessors. Geraadpleegd op 28 maart 2016, van <http://jaketrent.com/post/cons-css-preprocessors/>
6. Jean. (9 augustus 2010). 8 CSS PREPROCESSORS TO SPEED UP DEVELOPMENT TIME. geraadpleegd op 28 maart 2016 , van: <http://www.catswhocode.com/blog/8-css-preprocessors-to-speed-up-development-time>
7. Margaret Rouse. (2015). DEFINITION command line interface (CLI). Geraadpleegd op 25 maart 2016, van: <http://searchwindowsserver.techtarget.com/definition/command-line-interface-CLI>
8. Silver, A. (4 maart 2016). The disadvantages of CSS preprocessors. Geraadpleegd op 28 maart 2016, van <http://adamsilver.io/articles/the-disadvantages-of-css-preprocessors/>
9. Less website. (Z.D.). Geraadpleegd op 25 maart 2016, van: [http://lesscss.org/#](http://lesscss.org/)
10. Coyier, Chris. (11 juni 2012). Popularity of CSS Preprocessors

Geraadpleegd op 28 maart 2016, van: [https://css-tricks.com/poll-results-popularity-of-css-preprocessors/](https://css-tricks.com/poll-results-popularity-of-css-preprocessors/%20)

1. Borges Medeiros, M. H. (24 oktober 2012). The problem with CSS pre-processors. Geraadpleegd op 28 maart 2016, van <http://blog.millermedeiros.com/the-problem-with-css-pre-processors/>
2. Sass development team. (z.d.). SASS\_REFERENCE. Geraadpleegd op 24 maart 2016, van: http://sass-lang.com/documentation/file.SASS\_REFERENCE.html#using\_sass
3. Stylus website. (Z.D.).Geraadpleegd op 25 maart 2016, van: <http://stylus-lang.com/try.html#?code=body%20%7B%0A%20%20font%3A%2014px%2F1.5%20Helvetica%2C%20arial%2C%20sans-serif%3B%0A%20%20%23logo%20%7B%0A%20%20%20%20border-radius%3A%205px%3B%0A%20%20%7D%0A%7D>
4. Loos, C. (12 april 2012). Ten Reasons You Should Be Using a CSS Preprocessor. geraadpleegd op 28 maart 2016, van <http://blog.millermedeiros.com/the-problem-with-css-pre-processors/>
5. Drupalize. (z.d.) What is a CSS Preprocessor?  
   Geraadpleegd op 27 maart 2015, van  
   <https://drupalize.me/videos/what-css-preprocessor?p=1175>

# Bijlagen

### Bijlage 1: Zoekplan: Wat is een CSS preprocessor

### Trefwoorden/zoektermen:

* Css preprocessor

### In welke Ruimte vindt dit onderzoek plaats?

Om deze deelvraag te beantwoorden kan het beste een literatuurondezoek worden gedaan. Hierin worden verschillende bronnen over het onderwerp vergeleken wat een totaalbeeld schept van wat we zoeken. Hiervoor kunnen de bronnen worden gebruikt die al eerder gevonden zijn. De meeste bronnen komen van sites die de CSS preprocessor hebben gemaakt dus die zijn waarschijnlijk wel betrouwbaar als het gaat om wat een CSS preprocessor is.

### Bronnen:

* <http://htmlmag.com/article/an-introduction-to-css-preprocessors-sass-less-stylus>
* <https://drupalize.me/videos/what-css-preprocessor?p=1175>
* <https://www.urbaninsight.com/2012/04/12/ten-reasons-you-should-be-using-css-preprocessor>
* <http://vanseodesign.com/css/css-preprocessors/>
* https://2015.miami.wordcamp.org/files/2015/05/The-Beauty-of-CSS-Pre-Processors.pdf

## Experimenten:

Nvt.

### Bijlage 2: Zoekplan: Welke elementen bij Front End Development zijn arbeidsintensief / herhalend / complex?

### Trefwoorden/zoektermen:

* Front end Web Development
* Complex
* Repetative
* Most work

### In welke Ruimte vindt dit onderzoek plaats?

Deze deelvraag is het best te beantwoorden door middel van een aantal interviews of een enquête met mensen die zelf front end webdeveloper zijn en hier enige ervaring in hebben. Zij kunnen vanuit ervaring een antwoord geven op deze vraag. Maar door gebrek aan tijd om een enquête af te nemen is gekozen voor een literatuur onderzoek.

### Bronnen:

* <http://wpbusinesstips.com/2015/01/complexity-web-development-simple-guide/>
* <https://www.urbaninsight.com/2012/04/12/ten-reasons-you-should-be-using-css-preprocessor>

## Experimenten:

Nvt.

### Bijlage 3: Zoekplan: Voor welk probleem is een CSS preprocessor een oplossing?

### Trefwoorden/zoektermen:

* Css vs css preprocessors
* Improvements of css preprocessors
* Problems with css
* Solutions css preprocessors
* Things wrong with css

### In welke Ruimte vindt dit onderzoek plaats?

We zullen hier vooral gebruik maken van de Bieb-Ruimte op internet zijn er veel bronnen te vinden over de problemen die een CSS preprocessor oplost.

### Bronnen:

* <https://www.urbaninsight.com/2012/04/12/ten-reasons-you-should-be-using-css-preprocessor>
* <http://www.sitepoint.com/the-problem-with-css-is/>
* <http://www.pcmag.com/article2/0,2817,1987181,00.asp>
* <http://blog.codinghorror.com/whats-wrong-with-css/>
* <http://www.stubbornella.org/content/2009/02/12/css-doesn%E2%80%99t-suck-you%E2%80%99re-just-doing-it-wrong/>

### Experimenten:

Nvt.

### Bijlage 4: Zoekplan: Wat is het verschil tussen een CSS preprocessor en CSS?

### Trefwoorden/zoektermen:

* difference between css and css preprocessor
* difference between basic css and css preprocessor

### In welke Ruimte vindt dit onderzoek plaats?

Er is hier sprake van de veld-ruimte. Op internet is er tabel te vinden over de huidige verkrijgbare CSS preprocessors en de verschillen ervan.

### Bronnen:

* <https://www.quora.com/What-is-the-difference-between-LESS-and-CSS-Which-is-a-better-one>
* <https://css-tricks.com/poll-results-popularity-of-css-preprocessors/>
* <https://csspre.com/compare/>

### Aanpak:

Om deze deelvraag te beantwoorden kan het beste onderzoek worden gedaan naar algemene informatie over beide processors wat meestal te vinden is op forums/websites waarbij er wordt toegelicht wat de belangrijkste verschillen zijn tussen de ''Basic css'' en de css preprocessor. Deze vraag zou ook beantwoord kunnen worden door op zoek te gaan naar objectieve tabellen/grafieken die de basis verschillen tussen de 2 systemen weergeven.

### Bijlage 5: Zoekplan: Welke CSS preprocessoren zijn er te verkrijgen?

### Trefwoorden/zoektermen:

* All css preprocessors statistics
* All css preprocessors list

### In welke Ruimte vindt dit onderzoek plaats?

Er is hier sprake van de veld-ruimte. Op internet is er lijst te vinden over de huidige verkrijgbare CSS preprocessors . Er moet ook onderzoek worden gedaan naar de algemene ervaring van mensen en wat zij denken wat verbeter punten kunnen zijn.

### Bronnen:

* http://www.sitepoint.com/6-current-options-css-preprocessors/
* http://www.catswhocode.com/blog/8-css-preprocessors-to-speed-up-development-time Experimenten:

### Aanpak:

De beste manier om deze deelvraag te beantwoorden is door onderzoek te doen naar bronnen die relevante statistieken/grafieken of een lijst bevat waaruit je de verkrijgbare CSS preprocessors boven water kunt halen

Bijlage 6 Zoekplan: Welke werkwijze wordt er verondersteld/vereist?

Omdat in de onderzoeksbriefing stond dat we aannames mochten doen met betrekking tot de eisen van de opdrachtgever dat is dus wat we gedaan hebben.

Had die opdracht niet zo in de briefing gestaan dan hadden we een gesprek moeten aangaan met de opdrachtgever om te bepalen wat de opdrachtgever belangrijk vind in de CSS-preprocessor.

### Bijlage 8: Zoekplan: Welke werkwijze wordt er verondersteld/vereist?

### Trefwoorden/zoektermen:

* CSS-preprocessor werkwijzen
* CSS-preprocessor workmethod
* < naam van de CSS-preprocessor> werkwijze
* < naam van de CSS-preprocessor> work method

### In welke Ruimte vindt dit onderzoek plaats?

Er is hier sprake van de bieb-ruimte en de veld-ruimte of elke andere ruimte met internet.

### Bronnen:

* [www.sass-lang.com](http://www.sass-lang.com)
* [www.less.org](http://www.less.org)
* [www.stylus-lang.com](http://www.stylus-lang.com)

### Onderzoeksmethode

Op het internet zijn voor veel CSS-preprocessors websites te vinden. Op die websites zal wel staan als ze een bepaalde werkwijze vereisen.

### Bijlage 7: Zoekplan: Welke werkwijze wordt er verondersteld/vereist?

Omdat in de onderzoeksbriefing stond dat we aannames mochten doen met betrekking tot de eisen van de opdrachtgever dat is dus wat we gedaan hebben.

Had die opdracht niet zo in de briefing gestaan dan hadden we een gesprek moeten aangaan met de opdrachtgever om te bepalen wat de opdrachtgever belangrijk vind in de CSS-preprocessor.

### In welke Ruimte vindt dit onderzoek plaats?

Dit onderzoek heeft geen aparte ruimte nodig de enige eis die gesteld kan worden is dat iedereen van de groep in de ruimte past.

### Bijlage 8: Zoekplan: Wat zijn de tekortkomingen van de huidige CSS preprocessors?

### Trefwoorden/zoektermen:

* Problems with CSS preprocessors
* Advantages of <CSS preprocessor name>
* Disadvantages of <CSS preprocessor name>
* Things missing in CSS preprocessor
* New improvements on CSS preprocessors
* CSS preprocessors

### In welke Ruimte vindt dit onderzoek plaats?

Er is hier sprake van de bieb-ruimte en de veld-ruimte op internet is er literatuur te vinden over de huidige tekortkomingen maar er moet ook onderzoek worden gedaan naar de algemene ervaring van mensen en wat zij denken wat verbeter punten kunnen zijn.

### Bronnen:

* <https://csspre.com/compare/>
* <http://blog.millermedeiros.com/the-problem-with-css-pre-processors/>
* <http://blog.blakesimpson.co.uk/read/37-less-sass-the-advantages-of-css-preprocessing-explained>
* <http://stackoverflow.com/questions/28570752/what-are-the-advantages-disadvantages-of-using-css-preprocessors-e-g-sass-less>
* <http://jaketrent.com/post/cons-css-preprocessors/>
* <http://adamsilver.io/articles/the-disadvantages-of-css-preprocessors/>

### Experimenten:

Op internet zijn er veel fora hier delen veel gebruikers hun ervaringen over css preprocessors uit. Via dit experiment zoeken wij uit wat over het algemeen de nadelen/tekortkomingen van de huidige CSS preprocessors zijn.

### Bijlage 9: Zoekplan: Hoe is het gebruikersgemak vanuit het perspectief van een ontwikkelaar?

### Trefwoorden/zoektermen:

Sass reviews, sass, less reviews, less, stylus reviews, stylus, most popular CSS pre-processors, leer curve pre-processors.

### In welke Ruimte vindt dit onderzoek plaats?

Er is hier sprake van de bieb-ruimte en de veld-ruimte of elke andere ruimte met internet.

### Bronnen:

<http://htmlmag.com/article/an-introduction-to-css-preprocessors-sass-less-stylus>

<http://learn.shayhowe.com/advanced-html-css/preprocessors/>

[http://lesscss.org/#](http://lesscss.org/)

<http://sass-lang.com/>

<http://stylus-lang.com/>

### Experimenten:

Het downloaden en gebruiken van de preprocessoren LESS, Sass en stylus.

Op internet zoeken naar ervaringen van andere ontwikkelaars.

### Bijlage 10: Zoekplan: Wat zijn de verschillen/overeenkomsten tussen de gevonden bronnen uit je categorie?

### Trefwoorden/zoektermen:

Geen. Alle bronnen die we van te voren hebben verzameld zijn gebruikt.

### In welke Ruimte vindt dit onderzoek plaats?

Er is hier sprake van de bieb-ruimte en de veld-ruimte of elke andere ruimte met internet.

### Bronnen:

Al onze bronnen waar we gebruik van gemaakt hebben

### Experimenten:

Alle bronnen kritisch lezen en met elkaar vergelijken.

# Bijlage 11: Zelfreflecties:

S. Karsten:

Voor webdevs gingen wij onderzoek doen naar CSS preprocessors. We verdeelden de deelvragen die we samen hadden bedacht onder 5 personen, 2 deelvragen per persoon dus. Mijn taak was om 2 deelvragen te beantwoorden, Wat zijn de tekortkomingen aan de huidige CSS preprocessors en voor welk probleem is een CSS preprocessor een oplossing? Ook heb ik de inleiding en conclusies in overlegd samengesteld. Mijn doel was een goed stuk tekst neer te zetten en ik denk dat ik uiteindelijk alle relevante informatie heb kunnen geven.

In de praktijk bleek dat de deadlines niet altijd aangehouden werden. Veel problemen hebben wij hier niet aan ondervonden het werk is nog op tijd afgekomen. Misschien kunnen we bij een vervolgonderzoek de CSS preprocessors nog iets beter met elkaar vergelijken. Nu keken we nog veel naar wat gebruikers vonden, maar we konden ook proberen zelf verschillende CSS preprocessors te leren en hierop te baseren welke technische eigenschappen ze bevatten.

P. Uzun:

Ik vindt dat het project goed is verlopen, en dat de samenwerking ook goed ging. We hebben samen een goede basis gemaakt en daarna alles goed verdeeld. Ik heb alles kunnen vinden uit het onderzoeksplan. Mogelijke verbeteringen voor het volgende onderzoek zijn beter zoeken naar bronnen met de " " en andere zoektools waar ik niet gebruik van heb gemaakt.

N. Kooij:

Over het algemeen ging het werken aan het onderzoek erg goed. Iedereen wist wanneer hij wat af moest hebben en dat werd door de meesten ook gedaan. Het werken aan het verslag ging ook best wel goed. Het verzamelen van informatie was niet heel moeilijk want er is veel over het onderwerp te vinden op het internet.  
De meeste dingen die af moesten werden wel op het laatste moment gedaan. Dit hadden we beter anders kunnen doen. Het was ook niet altijd mogelijk een bepaalde groepsgenoot te bereiken wat het voor de rest lastig maakte om alles af te maken.

Zelfreflectie Ruben Janssen:

Ik vond het de afgelopen paar weken de samenwerking moeilijk verlopen. Dit kwam omdat niet iedereen actief communiceerde en daar van mijn kant veel energie in is gaan zitten. Was de communicatie op gang dan was de communicatie vlot en to the point.

De taken die verdeeld zijn, zijn over het algemeen goed uitgevoerd en volgens de groepsplanning gevolgd. Een opvallende uitzondering hierop was Pasa die altijd te laat en na een hoop pushen van de rest pas zijn stukken inleverden. Ik werd ontzettend moe van het feit dat ik naast mijn eigenwerk en plan voor uitvoering ook iemand anders ertoe moest bewegen om hetzelfde te doen. Dat heeft een weerslag gehad op mij en de rest van de groep.

Met de rest van de groep verliep de samenwerking beter hier was het eigen initiatief in de communicatie niet altijd even duidelijk maar ik kreeg wel vlot antwoord op de vragen die ik stelde.

Zelfreflectie Ege

Over het algemeen is de samenwerking binnen in de groep goed verlopen. Het toe verdeelde werk is vrijwel op tijd afgekomen ook al was het niet voor de opgestelde deadline. Dus het opstellen van de deadlines had beter gekund. Er werd dus van ieder individu verwacht zijn werk op tijd af te hebben en dat was dus ook het geval. We hebben ons aan het stappenplan gehouden dus het meeste werk was al af. De taken werden eerlijk toe verdeeld dus daar hadden we ook geen problemen mee.

Zelfreflectie Niels

Over het algemeen ging het werken aan het onderzoek erg goed. Iedereen wist wanneer hij wat af moest hebben en dat werd door de meesten ook gedaan. Het werken aan het verslag ging ook best wel goed. Het verzamelen van informatie was niet heel moeilijk want er is veel over het onderwerp te vinden op het internet.  
De meeste dingen die af moesten werden wel op het laatste moment gedaan. Dit hadden we beter anders kunnen doen. Het was ook niet altijd mogelijk een bepaalde groepsgenoot te bereiken wat het voor de rest lastig maakte om alles af te maken.

# Bijlage 12: Feedback groep Maarten, Daniël, Michiel en Tjieco

|  |  |  |
| --- | --- | --- |
| Documentstructuur | Ja / Nee | Toelichting |
| Correct Nederlands?  (Waaronder <3 taal- en spelfouten op een willekeurige pagina) | ja |  |
| Geschreven in zakelijke stijl? | Ja |  |
| Netjes en verzorgd? (geen vegen, onleesbaarheid, onnodige wisseling van lettertypes) | Ja |  |
| Concreet, to-the-point geschreven? | Ja |  |
| Alinea’s goed gebruikt? | Ja |  |
| Taalgebruik actief? (geen lijdende zinnen) | Ja |  |
| Voorblad bevat alle relevante gegevens? | Ja |  |
| Sprekende titel? | Nee |  |
| Bevat referenties naar de opdracht, het doel, de aanpak, resultaten en conclusies? | Ja |  |
| Alle bijlagen (Zoekplan, Trello bord, Reflectie, feedback van medestudenten) aanwezig? | Ja |  |
| Letterlijk overgenomen tekst tussen quotes en met bronvermelding? | Ja |  |
| Literatuurlijst aanwezig volgens APA? | Ja | Tussendoor misschien nog verwijzen naar bronnen |
| Pagina’s genummerd? | Ja |  |
| Hoofdstukken en paragrafen genummerd? | Ja |  |
| Kop- en voettekst aanwezig? | Ja |  |
| Figuren/tabellen/illustraties hebben nummer en naam en worden naar gerefereerd in de tekst. | ja | Geen naam |
| Document bevat 5-7 pagina’s inhoudelijke tekst. | Ja |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| Onderzoeksvragen | Ja / Nee | Toelichting |
| Sluiten de onderzoeksvragen aan op de hoofdvraag? | Ja |  |
| Dekken alle deelvragen de hoofdvraag af? | Ja |  |
| Zijn de deelvragen correct geformuleerd?  (denk aan onderzoeksles 2) | Ja |  |

|  |  |  |
| --- | --- | --- |
| Bronnen | Ja / Nee | Toelichting |
| Meerdere type bronnen gebruikt?  (boek, artikel, tijdschrift, blog, vendor) | Ja |  |
| Zijn de bronnen op kwaliteit gecheckt?  (denk aan onderzoeksles 2) | Ja |  |
| Is correct en volledig naar bronnen gerefereerd volgens APA? | Ja | Zie eerder punt over APA |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| Aanpak | Ja / Nee | Toelichting |
| Is er voor iedere onderzoeksvraag een passende aanpak gekozen? (bv opzoeken, experiment) | Ja |  |
| Zijn de gebruikte selectiecriteria gemotiveerd en correct gebruikt? | ja |  |

|  |  |  |
| --- | --- | --- |
| Resultaten | Ja / Nee | Toelichting |
| Komen de resultaten logischerwijs voort uit de aanpak? | Ja |  |
| Leiden de selectiecriteria logischerwijs tot de resultaten? | Ja |  |
| De conclusie refereert aan de resultaten? | Nee | De hoofdvraag wordt niet herhaald, verder weinig gerefereerd aan andere deelvragen |

|  |  |  |
| --- | --- | --- |
| Terugblik (groepsgewijs) | Ja / Nee | Toelichting |
| Aangegeven hoe de onderzoekscyclus is gebruikt en welke onderzoeksgebieden er zijn geraakt? (denk aan onderzoeksles 1) | Ja |  |
| Kijkt evaluerend en kritisch terug op het gelopen proces en de onderlinge samenwerking? | Ja |  |
| Formuleert een duidelijke mening over het doen van onderzoek, het maken van een rapportage en het vakgebied in het algemeen? | ja |  |