

# Front-end development, onderschat maar van onschatbare waarde

**Dit is geen technisch artikel! Zo, dat is eruit. Want dit artikel is juist bedoeld voor mensen die niet geïnteresseerd zijn in de technische kant van front-end development maar in de 'wat kan ik er nu mee' kant van het verhaal. Timothy van Sas en ik zien veel voorbeelden om ons heen van websites waar het fout gaat. Ook, en misschien juist wel bij grote corporate websites. Vaak door gebrek aan kennis. Want zeg nu zelf, als eindverantwoordelijke van een website kun je niet alles inhoudelijk weten! En bovendien, daarvoor huur je toch mensen in?**

## Front-end development, wat is het?

Een kleine zoektocht op Google en de niet technici onder ons haken af. Woorden als XML, HTML, webstandaarden, CSS, jQuery en Javascript vliegen voorbij zonder dat je enig idee hebt waar het over gaat. Maar één ding is zeker, iedere website en die van jou dus ook, krijgt ermee te maken.

Bij het ontwikkelen van een nieuwe website komt front-end development in beeld zodra het grafisch ontwerp gereed is. Een front-ender zorgt ervoor dat het ontwerp van de website (als statische afbeeldingen) wordt omgezet naar een werkende website, inclusief de volledige interactie.

(x)HTML en CSS bestanden zijn de deliverables van een Front end developer. Vanuit deze bestanden nemen de 'echt zware' programmeurs het vaak over om het CMS dan wel technische koppelingen en integraties te realiseren.

## Op weg naar standaarden

Binnen het vakgebied zijn enorm veel ontwikkelingen. HTML 4.0, XHTML, HTML 5.0, eigenlijk allemaal sets van codes, volgen elkaar in rap tempo op. Het is dan ook een hele kunst om binnen het vakgebied bij te blijven.

Een belangrijke trend binnen het vakgebied is het standaardiseren van code. Dit valt misschien nog wel het beste te vergelijken met de opkomst van de videoband in de jaren '80. Bij de lancering van de tape werden er in no-time honderden verschillende systemen geïntroduceerd. Alleen maar onhandig voor gebruikers. Immers iedere videoband paste niet in iedere videorecorder. Uiteindelijk is VHS ontstaan als een 'standaard' binnen deze wereld waar verreweg de meeste recorders mee om konden gaan.

Deze videorecorder van toen is in dit artikel te vergelijken met de webbrowser van nu. Want of je nu Internet Explorer, Firefox of Safari gebruikt, een website moet er in alle gevallen hetzelfde uitzien, toch? De realiteit is anders! Doordat browsers verschillend omgaan met de opgelegde standaarden zijn er grote verschillen in hoe een website wordt getoond. Of we willen of niet, dit is iets waar absoluut rekening mee moet houden.

Een goede front-ender doet dit. Hij controleert tijdens, maar zeker ook bij afronding van zijn werk de website in verschillende browsers om te zien of de website goed wordt weergegeven.

## Scheiding van presentatie, opmaak en gedrag

In een ideale wereld is alle informatie deelbaar. Immers één keer je content invoeren en vervolgens delen is veel praktischer dan iedere keer opnieuw je content plaatsen. Wil je iets wijzigen, dan doe je dat op één plaats en niet op 50 verschillende.

Daarnaast zou het niet uit moeten maken waar en hoe je deze content tot je neemt. Wil jij de teksten uit dit artikel lezen op je plasma TV, je mobiel of wat mij betreft op het schermje van je magnetron? Mij best! Maar 1 ding weet ik zeker, dat gaat niet werken als ik het in een opgemaakt PDF bestand plaats. Dat leest immers niet zo lekker op je mobiel, laat staan op het schermje van je magnetron.

Een front-end developer bepaalt door middel van (x)HTML hoe de informatie wordt gepresenteerd. Los daarvan bepaalt hij door middel van CSS en Javascript hoe deze informatie eruit moet zien (vet gedrukt, groen en met een fraaie tussenkop) en hoe deze zich moet gedragen (denk aan het direct tonen van een foutmelding in een formulier).

Heel praktisch kun je dus met 1 HTML bestand en 1 CSS bestanden dezelfde informatie op zowel een mobiele telefoon als een laptop juist tonen. Op deze manier is informatie heel eenvoudig deelbaar wat ontelbare kansen biedt voor alle mogelijke nieuwe ontwikkelingen die ons nog staan te wachten.

Note, voor alle front-enders die dit lezen: ik ben me ervan bewust dat ik het wat gechargeerd doe overkomen maar zoals in de inleiding beloofd, dit is geen technisch artikel.

## Snelheid, minder kosten en blijge gebruikers

Hoe je het ook wendt of keert, als je over front-end praat dan praat je over code. Regels en regels aan code. En hoe meer code en afbeeldingen een website bevat, hoe langer hij erover doet om in te laden. Iets wat website bezoekers over het algemeen niet als plezierig ervaren. Snelle websites zijn nu éénmaal plezieriger dan langzame websites.

Het streven zou dan ook moeten zijn om de code zo kort en goed mogelijk te schrijven. Zonder dat dit een doel op zich wordt. In sommige gevallen is het denkbaar om juist een paar extra regels code te schrijven zodat het op andere plaatsen in de website hergebruikt kan worden. Dit komt de snelheid van de website juist alleen maar weer ten goede.

Daarnaast heeft Google aangegeven binnenkort ook de snelheid van een website mee te nemen in de beoordeling en daarmee ranking van een website. De zogenaamde Google serp. Best interessant om te weten dat een snelle website je dus hoger in de resultaten van Google brengt!

Voor een snelle website is niet alleen een front-end developer verantwoordelijk. Met name ook de keuze voor welke techniek, op welk platform een site wordt gebouwd en de instellingen van de hosting provider zijn hierbij echt van belang.

Dan nog het kostenaspect. Iedere website wordt gehost bij een provider. Providers rekenen af op bandbreedte. Websites met heel veel verkeer betalen dus meer geld voor hun webhosting, dan websites met heel weinig verkeer. Klinkt logisch toch? Dan is het dus ook logisch dat een website met minder regels code en daarmee dus minder kb's in bandbreedte, eenvoudigweg minder geld kost. En geloof me dat dit voor grote websites met veel verkeer echt heel veel geld kan opleveren. Naast de snelheid sla je hier dus echt twee vliegen in één klap.

## Gemakkelijker structureel verbeteren

Je markt verandert, je bedrijf verandert, je doelgroep verandert. Eigenlijk verandert alles voortdurend. Dat is het enige wat je echt zeker weet. In tegenstelling tot bij het produceren van een product biedt het medium internet je de mogelijkheid om redelijk snel en laagdrempelig een verandering door te voeren. Ten minste, als je website qua code goed in elkaar zit en niet bij de kleinste verandering uit elkaar dreigt te vallen.

Zeker in het doorontwikkelen van je website is goede code van zeer groot belang. Bij goede code is het doorvoeren van wijzigingen zo gedaan. Vergelijk het met een goede fundering van een huis. Een dakkapel plaatsen is in een dag prima te doen, maar niet als eerst de hele fundering opnieuw moet worden gezet!

Daarnaast is online experimenteren sterk in opmars. Met gratis tools als Google optimizer is het redelijk eenvoudig om twee of meerdere varianten van je website live te testen. Als ik de bestelbutton nu rechts zet in plaats van links, zouden de mensen dan meer klikken? Dit is een vraag die redelijk makkelijk online te testen is, tenminste als je daarvoor niet je complete website hoeft om te bouwen! Een goede front-ender houdt hier vanaf de start rekening mee. Wijzigingen doorvoeren is ineens een kwestie van uren geworden in plaats van dagen. Nu ga je pas echt het belang van goede weldoordachte code inzien.

## 10 punten waaraan je een goede front ender herkent

Het belang van goede front-end zal inmiddels duidelijk zijn. Maar hoe herken je nu of je eigen front-ender zijn of haar werk goed gedaan heeft? En waar moet je nu eigenlijk allemaal op letten? Daarom, in willekeurige volgorde, 8 praktische tips waarmee je een goede front-ender herkent.

Een goede front-ender:

1. communiceert altijd intensief met de interactie ontwerpers en programmeurs van de website. Het is een samenspel waarbij iedereen samen tot de beste resultaten komt.
2. test voortdurend zijn eigen werk, zeker in alle verschillende browsers. (Dit is iets wat voor je eigen website natuurlijk ook redelijk snel nu zelf te testen is).
3. is in staat het hele project goed te overzien alvorens hij start met zijn werk. Goede voorbereidingen zijn hierin echt het halve werk.
4. levert zijn werk altijd volledig en in een onderbouwd overdrachtsdocument op zodat programmeurs hier goed mee uit de voeten kunnen.
5. houdt rekening met website bezoekers en confirmeert zich daarmee aan het user centered design principe. Uiteindelijk doen we het allemaal voor de bezoeker. Alles wordt dus in het werk gezet om zijn of haar online experience te vergroten.
6. is in staat een goed grafisch ontwerp neer te zetten. Er is heel veel voor te zeggen om het grafisch ontwerp en de (x)HTML door één persoon te laten maken.
7. is een vakidoot die kennis heeft van de ontwikkelingen in zijn vakgebied en overziet waar de (online) wereld naartoe gaat.

8. scheidt presentatie, opmaak en gedrag van elkaar zodat content eenvoudig deelbaar en op verschillende devices (mobiel, laptop, TV) opvraagbaar is.
9. schrijft correcte en semantisch kloppende code zonder dat dat dit een doel op zich wordt.
10. denkt na over de toekomst van de website en maakt een website voor de toekomst eenvoudig aanpasbaar.

Concluderend werkt een goede front-ender volgens het 'know why instead of know how' principe. Hij weet waarom hij dingen doet en doet niet zomaar dingen omdat ze moeten. Dit zorgt ervoor dat hij echt begrijpt waar het om draait wat uiteindelijk de kwaliteit van een website enorm opschroeft. Een goede front-ender is zijn geld dan ook dubbel en dwars waard!

## Auteur en publicatie

Raymond Klompsma en Timothy van Sas zijn werkzaam bij Concept7 ([www.concept7.nl](http://www.concept7.nl)). Dit artikel is gepubliceerd op [www.raymondklompsma.nl](http://www.raymondklompsma.nl), [www.frankwatching.com](http://www.frankwatching.com) en in Usabilityweb magazine.

## Bedankt mensen!

- <http://searchengineland.com/site-speed-googles-next-ranking-factor-29793>
- <http://www.raymondklompsma.nl/2009/06/structureel-verbeteren/>
- [http://en.wikipedia.org/wiki/A/B\\_testing](http://en.wikipedia.org/wiki/A/B_testing)