

Voorwoord bij de vierde editie

Van alle technieken die komen kijken bij het maken van webpagina's en webapps zijn HTML en CSS de belangrijkste. Zonder HTML is er sowieso geen webpagina en zonder CSS blijft het een kale boel. Met aanvullende technologieën zoals JavaScript kunnen apps en pagina's meer functionaliteit krijgen, maar uiteindelijk begint alles bij HTML en CSS.

HTML5 is sinds oktober 2014 een W3C Recommendation, wat betekent dat het een webstandaard is. Van ontwikkelaars wordt verwacht dat zij webpagina's bouwen op deze aanbevolen manier. Van browserfabrikanten wordt verwacht dat hun browser die pagina's op de aanbevolen manier verwerken.

Bij CSS ligt het wat ingewikkelder. Het uitgangspunt is nog steeds CSS 2.1, de webstandaard uit 2011. De ontwikkeling van CSS gaat sindsdien verder in modules. Daarin worden afzonderlijke onderdelen uitgewerkt, bijvoorbeeld achtergronden en randen, selectors of flexibele lay-out. Er is heel veel in ontwikkeling, waarvan een deel al in browsers is ingebouwd en een groter deel niet. De veelgebruikte term CSS3 is door deze ontwikkeling meer een verzamelnaam voor 'alles na CSS 2.1' dan een afgebakende standaard.

De eerste editie van dit boek verscheen in 2011, toen duidelijk was dat HTML5 en CSS3 veel veranderingen zouden brengen, maar nog niet 'klaar' waren. Sindsdien blijven de webstandaarden zich ontwikkelen en daarom is er nu de vierde editie, bijgewerkt in de zomer van 2017. De veranderingen zijn opnieuw spectaculair, dankzij de doorbraak van de lay-outtechnieken Flexible Box Layout en CSS Grid Layout en de mogelijkheid om CSS-variabelen te gebruiken. Verder is een hoofdstuk toegevoegd over technieken die in modern webdesign steeds belangrijker worden: transities, animaties en transformaties.

Natuurlijk komen nog steeds alle belangrijke basisonderwerpen aan bod, van het goed structureren en markeren van de content met HTML tot de vormgeving met CSS. Bovendien leert u niet alleen dát dingen werken, maar vooral waaróm ze werken.

Ik wens u veel plezier bij het leren en bouwen!

Peter Doolaard
Schoonhoven, juni 2017

reacties@handboek-html-css.nl

Inhoud

1	Webtalen, browsers en editors	1
	Introductie	2
	Een snelle terugblik	2
	Afspraken over webtalen	3
	Twee smaken HTML	5
	Wat HTML5 is	7
	Kennismaken met HTML en CSS	8
	Aan de serverkant	9
	HTML – Hypertext Markup Language	9
	CSS – Cascading Style Sheets	10
	Browsers	12
	Google Chrome	13
	Microsoft Edge (Internet Explorer)	14
	Mozilla Firefox	15
	Mac OS X: Safari	16
	HTML-editors	16
	Brackets	17
	Aanvullende hulpmiddelen	19
	Hulpmiddelen voor validatie	19
	Hulpmiddelen voor ontwikkeling	20
	Samenvatting	26
	Oefeningen	26
2	De basis van HTML5	29
	Kenmerken van HTML-documenten	30
	Elementen, tags en tekst	30
	Opbouw van een element	32
	Elementen nesten	33
	Kleine letters	34
	Sluiten in de goede volgorde	34
	De HTML-code van webpagina's bekijken	35

Attributen	37
Kenmerken van attributen	37
Globale attributen	39
Microdata	43
WAI-ARIA	44
Kenmerken van HTML5	44
Omringende technieken	46
Vervallen onderdelen	48
De basis van een HTML5-document	49
<!DOCTYPE html>	49
<html lang="nl">	50
<head>	50
<meta charset="utf-8">	50
<title>	51
<link>	52
<body>	52
Paginastructuur	52
Document Object Model – HTML DOM	53
De outline van een document	54
Koppen in HTML-documenten	54
Het getal bepaalt de keus, niet de tekstgrootte	56
Titel website is niet automatisch <h1>	56
Het contentmodel van HTML5	57
Secties markeren – theorie	59
<article> (sectioning content)	61
<section> (sectioning content)	62
<nav> (sectioning content)	63
<aside> (sectioning content)	64
<header> (flow content)	65
<footer> (flow content)	66
<address> (flow content)	68
<main> (flow content)	69
<div> (flow content)	70
Samenvatting	71
Secties markeren – praktijk	71
Eenvoudige webpagina	72
Productenpresentatie	73
Blog	76
Samenvatting	78
Oefeningen	79

3	Tekst markeren	81
	Inleiding	82
	Koppen markeren	82
	Koppen in een header	83
	Tekst markeren	85
	Alinea's: het element <p>	86
	Het regeleinde 	87
	Afbreken van woorden	87
	Speciale betekenis aangeven	88
	Inhoud groeperen	94
	Vooraf opgemaakte inhoud: <pre>	94
	Citaten gebruiken: <blockquote>	95
	Lijsten maken: , en 	96
	Illustraties en bijschriften: <figure> en <figcaption>	101
	FAQ, metadata en definities: <dl>, <dt>, <dd>	106
	Uitvouwbare tekst: <details>	107
	Nieuw thema markeren: <hr>	107
	Speciale tekens in webpagina's	108
	Van ASCII naar UTF-8	109
	Referenties gebruiken	110
	Samenvatting	110
	Oefeningen	111
4	Koppelingen maken	113
	Inleiding	114
	Verbindingen leggen met <a>	115
	Attributen van <a>	116
	Bladwijzers maken	119
	Link naar een grote foto	120
	Link naar het vervolg van een artikel	121
	Link naar een e-mailadres	123
	Link naar een telefoonnummer	123
	Klik hier voor richtlijnen	123
	Externe bestanden koppelen met <link>	126
	Stylesheet koppelen	126
	Een favicon gebruiken	127
	Opeenvolgende pagina's	129
	Vooraf laden: prefetch, prerender en dns-prefetch	130
	Overige waarden van rel	131
	Samenvatting	131
	Oefeningen	132

5	Beeld, geluid en andere externe inhoud	133
	Inleiding	134
	URL's en structuur van de website	134
	Lokale site	136
	Soorten afbeeldingen	137
	Bitmaps	137
	Vectorafbeeldingen: SVG	138
	Bestandsformaten voor bitmaps	140
	GIF	140
	JPEG	141
	PNG	142
	De juiste afbeelding voor het scherm	143
	Afbeeldingen plaatsen met 	144
	De bron: src	144
	Beschrijvende tekst: alt	145
	Afmetingen: width en height	146
	Klikbare gebieden: usemap	147
	Responsive images	148
	Srcset voor schermen met hoge resolutie	148
	Srcset voor variabele afmetingen	149
	Bijgesneden afbeeldingen: <picture>	150
	Het type afbeelding selecteren	151
	Het verschil tussen en <picture>	152
	Externe HTML-inhoud: <iframe>	152
	Attributen	153
	Beveiliging	154
	Insluiten met <embed> of <object>	155
	Video in een webpagina	157
	Bestandstypen voor video	157
	Een codec kiezen	158
	Video coderen	158
	Het element <video>	158
	De bron: <source>	160
	Ondertiteling met <track>	161
	Titelbestanden: WebVTT	162
	Audio op de website gebruiken	163
	Samenvatting	165
	Oefeningen	165
6	Tabellen maken	167
	Inleiding	168
	De structuur van een HTML-tabel	168

De basis: <table>	169
Rijen en kolommen: <tr> en <td>	170
Koptekst: <thead>	171
Een bijschrift: <caption>	172
Overige elementen	173
De tabelinhoud: <tbody>	173
Voettekst: <tfoot>	174
Kolomgroepen: <colgroup> en <col>	174
Voorbeelden van tabellen	175
Samenvatting	178
Oefeningen	179
7 Formulieren maken	181
Formulieren in HTML5	182
Verbeteringen aan de clientkant	182
De rol van de webserver	184
De basis van een formulier: <form>	184
Buiten <form> mag ook	184
De verwerking: action	185
Verzendwijze: method	185
Codering: enctype	186
Automatisch aanvullen: autocomplete	186
Niet valideren: novalidate	186
Direct invoeren: autofocus	187
De naam: name	187
Het doel: target	188
Buitenspelelementen: form	188
Het element <label>	189
De tabvolgorde	190
Het element <input>	191
Attributen van <input>	191
Soorten invoer: het attribuut type	199
Telefoonnummers: <input type="tel">	199
E-mailadressen: <input type="email">	200
Webadressen: <input type="url">	200
Wachtwoorden: <input type="password">	201
Getallen: <input type="number">	201
Een bereik: <input type="range">	202
Kleuren: <input type="color">	203
Datum en tijd	205
Selectievakjes: <input type="checkbox">	206
Keuzerondjes: <input type="radio">	207
Bestanden uploaden: <input type="file">	208

Verzendknop: <code><input type="submit"></code>	209
Afbeeldingknop: <code><input type="image"></code>	209
Resetknop: <code><input type="reset"></code>	210
Functieloze knop: <code><input type="button"></code>	210
Verborgene waarden: <code><input type="hidden"></code>	210
Het element <code><button></code>	211
Kiezen uit een lijst: <code><select></code>, <code><datalist></code>, <code><option></code>	212
<code><option></code>	213
<code><select></code>	214
<code><datalist></code>	215
Uitgebreide tekst invoer: <code><textarea></code>	217
Groeperen met <code><fieldset></code> en <code><legend></code>	218
Voortgang tonen: <code><progress></code>	220
Schaal: <code><meter></code>	222
Resultaat: <code><output></code>	223
Samenvatting	224
Oefeningen	225
8 De basis van CSS	227
Wat CSS is	228
Voorgeschiedenis	228
En toen was er CSS3	229
Waarom CSS zo handig is	231
De taal CSS	231
Waarden en eenheden	233
Globale waarden	234
Lengte	234
CSS-verwerking door de browser	238
De boomstructuur	238
Waarden toewijzen	239
De cascade	240
Specificiteit berekenen	241
Overerving	243
Percentages werken door	244
De plaats van de CSS-declaraties	245
Het attribuut <code>style</code>	245
Het element <code><style></code>	246
Extern bestand: <code><link></code>	247
Het CSS-bestand	247
Commentaar toevoegen	247
Andere stijlbladen importen	248
De ingebouwde stijlen aanpassen	249
Elementen benaderen met selectors	250

Basiselectors	251
Typeselector	251
Universele selector *	252
Klassenselector	253
ID-selector	254
Attribuutselectors	255
Combinatieselectors: afstamming, kind en sibling	256
Geen combinator: komma	256
Algemene afstamming: spatie	256
Kindselector: >	257
Aangrenzend: +	257
Opeenvolgend: ~	257
Pseudoklassen	258
Pseudoklassen voor links	258
Pseudoklasse doel	259
Pseudoklasse taal	259
Pseudoklasse UI-toestand	259
Structurele pseudoklassen	259
Pseudoklasse ontkenning	261
Pseudo-elementen	261
Best practices	263
Samenvatting	265
Oefeningen	266
9	Boxmodel, weergavemodel en positionering
	269
De opbouw van pagina's	270
Browserstijlen zijn er niet voor niets	270
Het boxmodel	270
De eigenschap box-sizing	272
Begrensde breedte en hoogte	275
Waarden voor hoogte en breedte	275
Marges	276
Horizontaal centreren	276
Verticaal centreren	277
Negatieve marge	278
Ingeklapte marges	278
Padding	278
Randen	279
Randdikte	279
Randkleur	280
Randstijl	282
Korte notatie voor randen	283

Weergavemodel	283
Terminologie	284
De eigenschap display	285
Waarden voor binnen en buiten	285
Positionering – de lay-out	288
Het omvattende blok	288
Positioneringsschema's	289
Position	289
Verschuiven met top, right, bottom en left	289
Relatieve positionering	290
Absolute positionering	291
Vaste positie: fixed	293
Flexibele vaste positie: sticky	294
Float	295
Float als lay-outtechniek	297
Ouderwetse oplossing: clearfix	299
Moderne clearfix: flow-root	300
Meer eigenschappen van blokken	301
Overlopende inhoud: overflow	301
Speciaal voor tekst: text-overflow	302
Praktijk: overlopende tekst met fade-out	303
Formaat aanpassen met resize	305
Uitsnijden met clip-path	307
Zichtbaarheid: visibility	309
Niet verbergen voor screenreaders	310
De toekomst	310
Samenvatting	311
Oefeningen	312
10 Tekst en typografie	315
Inleiding	316
Lettertype: van systeem of online?	317
Web fonts	317
Lettertypen downloaden: @font-face	320
Eigenschappen van @font-face	320
Een embedcode gebruiken	323
Eigenschappen voor lettertype: font	326
Lettertype	326
Lettergrootte: waarden en overwegingen	327
Geavanceerd: responsive tekst	328
Toepassing van font-size	329
Grootte aanpassen: font-size-adjust	330
Regelhoogte: line-height	330

Letterstijl: font-style	332
Lettergewicht: font-weight	332
Uitrekken of indrukken: font-stretch	334
Kleinkapitaal: font-variant	334
De verzameleigenschap font	335
Eigenschappen voor tekst	336
Uitlijnen: text-align en text-align-last	336
Lijneffecten op tekst	337
Inspringen	338
Rechtopstaande tekst: text-orientation	339
Overlopende tekst: text-overflow	340
Tekst met schaduw	340
Tekst omzetten: text-transform	341
Witruimte behouden: white-space	342
Ruimte tussen woorden en letters	344
Afbreken: word-break en overflow-wrap	344
Opmaak van lijsten	345
Opsommingstekens	345
Lijst als navigatie	346
Verticaal menu	348
Uitklapbaar menu	349
Samenvatting	351
Oefeningen	352
11 Kleur, randen en achtergronden	353
Inleiding	354
Kleurwaarden	354
RGB en RGBA	355
HSL en HSLA	356
Transparantie	358
Kleur van tekst	358
De achtergrond	359
Achtergrondkleur	359
Achtergrondafbeelding	359
Verschillende achtergrondafbeeldingen	367
Kleurverloop als achtergrond	368
Parallax met CSS	368
Kleurverlopen	369
Lineair verloop	369
Radiaal verloop	371
Repeterende verlopen	372
Afgeronde hoeken	373

Randafbeelding	376
Voorbeeld van een randafbeelding	377
Schaduw	378
De werking van schaduw	379
Material design	380
Samenvatting	381
Oefeningen	382
12 CSS-variabelen, transformatie, animatie en meer	383
Rekenen met calc()	384
Voorbeelden	384
CSS-variabelen: de functie var()	385
Kenmerken van variabelen	386
Variabelen declareren	387
Toepassingen van variabelen	387
All reset alles, behalve variabelen	389
Beweging in webpagina's	390
Overgangen (transitions)	390
Kant-en-klare overgangen	392
Positie aanpassen	393
Animatie	394
Opbouw een animatie	394
Keyframes	396
Transformatie	398
Tweedimensionale transformaties	399
Schalend menu	401
Driedimensionale transformaties	402
Beeldfilters	409
Kleuren mengen	409
Samenvatting	411
Oefeningen	412
13 Media queries, flexbox en grid	413
Inleiding	414
De mogelijkheden	414
Responsive design en media queries	415
Responsive design en mobile first	418
Weergave op mobiele schermen: de metatag viewport	419
Media queries: breekpunten in de lay-out	421
Vensterbreedte en vensterhoogte	422
Schermresolutie	423
Portret of landschap	424

Beeldverhouding	424
Flexibele lay-out met flexbox	425
Kenmerken	425
Flexcontainers en flexitems	426
Het assenstelsel van de flex-flow	427
De richting: flex-direction	428
Regelomloop: flex-wrap	430
De eigenschap flex-flow	431
De volgorde: order	431
Flexitems schalen	431
Instellingen voor flex	433
Flexitems uitlijnen	434
Uitlijnen met de marge	434
Uitlijnen op de hoofdas: justify-content	434
Uitlijnen op de kruisas: align-self	436
Regels uitlijnen: align content	436
Een lay-out maken met flexbox	437
Lay-out in een raster met CSS Grid	443
Kenmerken van gridlay-out	443
CSS-eigenschappen voor gridlay-out	445
Definitie van het raster	445
Plaatsing van de items	445
Uitlijning en tussenruimte	445
Workflow in vogelvlucht	446
Een raster maken	447
Auto-flow	450
Tussenruimte (gap)	451
Waarden voor tracks	451
De functie repeat()	452
Expliciet en impliciet raster	454
Een voorbeeld van gridlay-out	455
Smalle weergave	456
Breakpoint 37rem	458
Breakpoint op 47rem	459
Breakpoint 69rem en breder	461
Verder ontdekken	461
Samenvatting	462
Oefening	462
Index	463

Webtalen, browsers en editors

HTML is de belangrijkste taal op internet. De standaard waar we ook vandaag mee werken bestaat sinds 2011 en heet HTML5. Dit hoofdstuk beschrijft het ontstaan en de actuele situatie. Er wordt kort uitgelegd wat HTML en CSS zijn en wat HTML5 is, want daar wordt verschillend over gedacht. Er is een overzicht van browsers en hulpmiddelen voor het schrijven van HTML-documenten en u maakt kennis met een handige gratis editor.

U leert in dit hoofdstuk:

Hoe we hier gekomen zijn.

De rol van HTML en van CSS, en wat HTML5 is.

Welke browsers niet mogen ontbreken.

Handige hulpmiddelen: editors, validators en developer tools.

Introductie

Elke pagina op het world wide web heeft als basis de taal HTML, de afkorting van Hypertext Markup Language. Het kan er nog zo flitsend en grafisch uitzien en er kunnen allerlei aanvullende talen en technieken zijn gebruikt, de basis, de kern, is altijd HTML. Dit boek gaat over HTML5. Het is de actuele versie van de taal, vastgelegd in oktober 2014.

HTML5 kan niet los worden gezien van de voorgangers, HTML4.01 en XHTML. HTML5 is daar een voortzetting van, waarbij onderdelen zijn aangepast, toegevoegd en afgeschaft, maar alle oude webpagina's blijven werken. Met HTML5 beginnen we dus niet aan een nieuw web. Afschaffen betekent in dit verband dan ook dat auteurs worden aangemoedigd nieuwe webpagina's volgens nieuwe standaarden te coderen en oude pagina's te moderniseren. Gebeurt dat niet, dan blijven de webpagina's toch toegankelijk voor bezoekers. Browsers moeten verouderde pagina's blijven weergeven.

De kans dat uw ervaring ophoudt bij HTML4 is intussen flink afgenomen, maar als dat uw achtergrond is, zult in dit boek veel elementen herkennen. Is dit uw eerste kennismaking met HTML, dan leert u – niet gehinderd door soms verouderde kennis – direct hoe het moderne HTML kan worden toegepast.

Het is bijna standaard, maar daarom niet minder relevant, dat in boeken als dit een overzicht van de historie wordt geschetst. We houden het kort. Maar het is bijvoorbeeld handig om te weten dat er verschillende partijen werken aan HTML5.

Een snelle terugblik

Toen de eerste editie van dit boek verscheen – voorjaar 2011 – was de inkt van het logo van HTML5 nog nat. In 2014 heeft de beheerder van de webtalen, het World Wide Web Consortium (W3C) de specificatie 'definitief' vastgesteld (in W3C-termen is het nu de aanbevolen specificatie, gelabeld als *recommendation*). Definitief staat niet voor niets tussen aanhalingstekens, want intussen wordt nog steeds aan de specificatie gewerkt. De huidige versie is HTML5.1 en die is sinds 1 november 2016 gelabeld als recommendation. Het betreft een *minor upgrade* en de verschillen zijn daarom klein. Ze zijn verwerkt in deze vierde editie van het boek. Eind 2017 wordt HTML 5.2 verwacht, opnieuw een minor upgrade. Om het makkelijk te houden spreken we in dit boek gewoon van HTML5.



Afbeelding 1.1 *Het logo van HTML5, in 2011 gepresenteerd door het W3C.*

Naast de versie van het W3C is er een tweede HTML-specificatie en die gaat door het leven als *HTML Living Standard*. Een updatedatum geeft aan welke versie het betreft.

Dat er twee standaarden bestaan klinkt misschien vreemd (en dat is het ook), maar het komt doordat HTML een stoornis in de ontwikkeling heeft gehad. Om dat te kunnen begrijpen, moet u iets weten van de geschiedenis.

HTML

Living Standard — Last Updated 18 May 2017

Afbeelding 1.2 *HTML Living Standard, beheerd door WHATWG.*

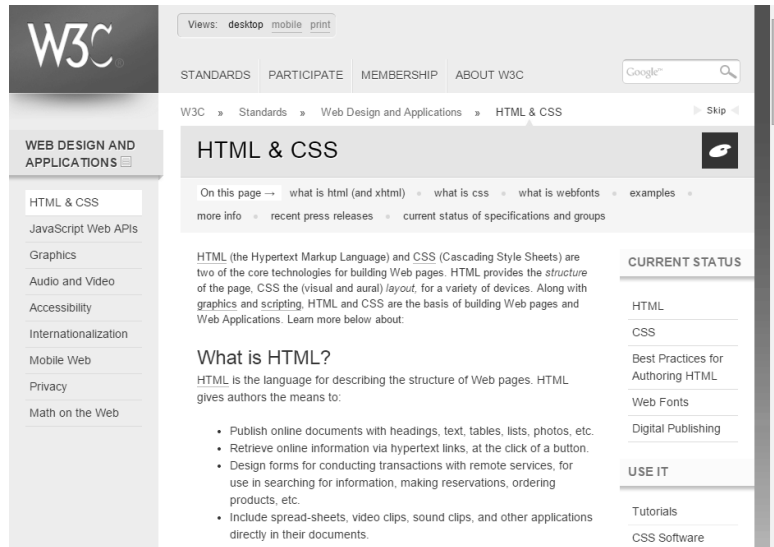
Afspraken over webtalen

Sinds 1994 wordt geprobeerd lijn te brengen in de talen waarmee websites (en andere webtoepassingen) worden gemaakt. Die talen zijn onder meer Hyper-text Markup Language (HTML) en Cascading Style Sheets (CSS). Daarvoor is het World Wide Web Consortium, kortweg W3C, opgericht. In verschillende werkgroepen overleggen browserfabrikanten zoals Microsoft, Google, Mozilla en Opera. Omdat W3C ook allerlei andere webstandaarden beheert, vindt u onder de leden ook bedrijven als Nokia, Rabobank Nederland, Walt Disney Company en Motion Picture Association of America. Het doel van bijvoorbeeld de werkgroep Web Platform is afspraken maken over hoe een auteur in een webdocument aangeeft dat iets een hyperlink, een kop of een opsomming is, en hoe een browser daarmee zou moeten omgaan.



World Wide Web Consortium

Kijk voor meer informatie over wat het W3C is en doet op w3.org. Dat is de thuishaven van deze in Amerika gevestigde organisatie. U vindt er ook informatie over de hierna genoemde talen XHTML en XML.



Afbeelding 1.3 *Het onderdeel HTML en CSS op de website van W3C.*

Het is allerm minst vanzelfsprekend dat er nu een W3C-standaard van HTML5 is. Eind jaren negentig, de standaard HTML4.01 was klaar, besloot de organisatie zich volledig te richten op XHTML, een vorm van HTML gemodelleerd naar de strikte regels van XML. Het begon met het herschrijven van HTML naar XML (XHTML 1.0) en tegelijk werd gewerkt aan een geheel nieuwe versie (XHTML2) die niet uitwisselbaar zou zijn met eerdere HTML- en XHTML-talen. Er was daarmee een route gekozen waarbij het bestaande web zou blijven zoals het was (HTML4) en er daarnaast een niet-uitwisselbaar nieuw web zou ontstaan met op XML gebaseerde applicaties.

Tijdens dat ontwikkelproces zagen Mozilla en Opera in 2004 nieuwe mogelijkheden voor HTML en zij hebben voorgesteld om toch door te gaan met de ontwikkeling van HTML, maar daar waren bij het W3C onvoldoende medestanders voor. In reactie daarop besloten Apple, Mozilla en Opera dan maar zelf aan de slag te gaan met HTML5. Daarvoor richtten zij in 2004 de Web Hypertext Application Technology Working Group op, kortweg WHATWG. Hun voornaamste uitgangspunt was dat webontwikkelaars een webtaal gebaseerd op het oude HTML4 tot hun beschikking zouden krijgen, dus uitwisselbaar, maar met nieuwe mogelijkheden om een volwaardig alternatief te kunnen bieden voor native webapps (iOS en Android) en gepatenteerde technieken zoals Flash van Adobe en Silverlight van Microsoft. Dit initiatief viel bij de gemeenschap van webontwikkelaars in goede aarde. Omdat die de ontwikkelingen met gejuich ontvingen en omdat fabrikanten de nieuwe ideeën gingen inbouwen in hun

browsers, ging uiteindelijk ook bij het W3C het roer om. De werkgroep XHTML is eind 2009 opgeheven en de aandacht gaat weer uit naar HTML.



Living Standard

In een dynamische omgeving als internet moeten de mogelijkheden niet beperkt worden door starre regels. Met dat uitgangspunt propageert WHATWG dat HTML een levende standaard is, constant in ontwikkeling. Daar past geen versienummer bij. Lees de FAQ van WHATWG voor achtergrondinformatie over deze organisatie op wiki.whatwg.org/wiki/FAQ.

Het zijn de gedrevenheid van WHATWG en de snelheid waarmee in browsers nieuwe mogelijkheden worden ingebouwd die de ontwikkeling van HTML in een stroomversnelling hebben gebracht. Google (Chrome), Mozilla (Firefox), Opera en Microsoft (Edge) brengen in hoog tempo updates voor de browser uit. Het resultaat is dat de nieuwste versies van alle toonaangevende browsers veel nieuwe mogelijkheden ondersteunen. Het werkt als een vliegwiel: het feit dat nieuwe opties snel worden opgenomen in browsers stimuleert de ontwikkeling.



Automatische updates

Er worden niet alleen sneller nieuwe versies van browsers uitgebracht, ze worden ook veel sneller op grote schaal gebruikt door het internetpubliek. Dat komt doordat browserupdates automatisch worden gedownload en geïnstalleerd, zonder tussenkomst van de gebruiker. Dit principe staat bekend als Evergreen browsers (zie ook <http://eisenbergeffect.bluespire.com/evergreen-browsers/>).

Twee smaken HTML

Zijn er nu twee smaken HTML? In zeker zin is dat zo. Hoewel de HTML5-recommendation op de website van het W3C het resultaat is van samenwerking tussen W3C en WHATWG, hebben de wegen van de groepen zich gescheiden. Wat W3C betreft is HTML5 'af', hoewel de organisatie sinds 2012 ook werkt aan upgrades van HTML5. Daarbij gaat het om zaken die in de huidige specificatie nog onvoldoende ontwikkeld zijn. In grote lijnen is de HTML5-specificatie van W3C te zien als een stabiel geheel dat wordt ondersteund door een groot aantal marktpartijen. De Living Standard van WHATWG daarentegen is veel dynamischer. Er wordt permanent gewerkt aan nieuwe of verbeterde mogelijkheden op basis van wat websitebouwers nodig hebben.

Doordat er twee specificaties bestaan, rijst uiteraard de vraag welke u als websitebouwer het best kunt hanteren. Daarbij telt uiteraard welke specificatie door de browserfabrikanten wordt gevolgd en dat is de W3C-versie. Gebruik die als stabiele standaard om websites te bouwen die door alle moderne browsers worden ondersteund. Volg de 'levende standaard' van WHATWG om te zien wat de toekomst kan brengen. Want zo werkt het globaal: uit de gemeenschap van webontwikkelaars komt naar voren wat een gewenste techniek, die wordt uitgewerkt door WHATWG, wat kan leiden tot (experimentele) implementatie in browsers, waarna uiteindelijk in de W3C Web Platform-werkgroep zo'n techniek als aanbeveling wordt opgenomen in de HTML-specificatie.

U hebt als ontwikkelaar in elk geval geen last van de meningsverschillen tussen de specificatiemakers. Integendeel. WHATWG denkt vooral na over wat er mogelijk zou moeten zijn; het zijn de creatieven. Het W3C is een stuk behoudender, maar neemt bij voldoende draagvlak nieuwe ideeën wel over. Daar worden de mogelijkheden van HTML alleen maar groter van.

Er bestaat overigens geen overzicht van de verschillen.



Bekijk de specificaties


Uiteraard zijn alle specificaties online beschikbaar. De W3C-recommendation is te vinden op www.w3.org/TR/html. Deze specificatie toont wat er is. De specificatie van WHATWG is te vinden op <https://html.spec.whatwg.org/multipage/> en toont wat er mogelijk komt. Daardoor is de kans ook groter dat er zaken in staan die nog niet in webbrowsers zijn opgenomen.

Weliswaar is het getalsmatig een hele stap van 4.01 naar 5, dat betekent niet dat alles van HTML4 ongeldig en overbodig is nu HTML5 er is. Integendeel. Er zijn wel dingen veranderd, vervallen en toegevoegd, maar het is nog steeds

W3C Recommendation

HTML 5.1

W3C Recommendation, 1 November 2016



This version:
<https://www.w3.org/TR/2016/REC-html51-20161101/>

Latest published version:
<https://www.w3.org/TR/html51/>

Latest version of HTML:
<https://www.w3.org/TR/html/>

Editor's Draft:
<https://w3c.github.io/html/>

Afbeelding 1.4 Hier gaan we het uitgebreid over hebben.

HTML. HTML5 is zogezegd *backward compatible* oftewel uitwisselbaar met voorgaande versies. Hebt u ervaring met HTML4, dan kunt u de overstap naar HTML5 beschouwen als bijscholing.



Oude HTML werkt nog

De achterwaartse uitwisselbaarheid (*backward compatibility*) van HTML5 gaat ver. Ook ‘verboden’ opmaaktags zoals `<center>` of `` doen het nog in de allernieuwste browsers. Dat moet ook wel, want er zijn talloze webpagina’s die met deze tags zijn gemaakt en ook die pagina’s moeten toegankelijk blijven voor bezoekers. Wel worden web-sitebouwers aangemoedigd dergelijke tags niet meer te gebruiken. HTML is voor structuur en inhoud, niet voor opmaak. Verderop wordt dit uitgelegd.

Wat HTML5 is

HTML5 is in essentie niet meer dan een upgrade van HTML4. We hebben het dan alleen over de markeertaal HTML. HTML5 is ook een verzameling van web-technieken om moderne (mobiele) webapplicaties te bouwen. Naast HTML gaat het dan ook over een verzameling API’s en aanvullende technieken zoals CSS, JavaScript, XML, JSON en SVG. Bedenk dat deze geen deel uitmaken van de HTML5-standaard van W3C. U zult ze daar niet in terugvinden. Dat neemt niet weg dat ze in het leven van de webontwikkelaar een onmisbare rol spelen. Een korte toelichting:

- CSS staat voor Cascading Style Sheets en is de taal voor de vormgeving van HTML-pagina’s. CSS komt in dit boek uitgebreid aan bod.
- JavaScript is een programmeertaal (scripttaal) om onder meer webpagina’s interactief te maken. JavaScript wordt in dit boek niet behandeld.
- XML staat voor Extensible Markup Language en is een markeringstaal (zoals HTML). Het is ontwikkeld om gestructureerde gegevens uit te wisselen tussen (web)toepassingen op een manier die ook voor mensen leesbaar is.
- JSON is de afkorting van JavaScript Object Notation en is een alternatief voor XML. De functie is hetzelfde: uitwisseling van gegevens tussen een server en een applicatie. JSON wordt verder niet besproken.
- SVG staat voor Scalable Vector Graphics. Dit is een op XML gebaseerd bestandsformaat voor vectorafbeeldingen (vectorafbeeldingen kunnen worden vergroot of verkleind zonder dat ze rafelig of blokkerig worden zoals bitmapafbeeldingen). Ook SVG valt buiten het bestek van dit boek.



Toegangspunt voor ontwikkelaars: API

De afkorting API staat voor application programming interface. Het is een toegangspunt tot aanvullende functies waarbij de ontwikkelaar eenvoudig toegang heeft tot die functies zonder dat hij precies hoeft te weten hoe die functies werken. Hij hoeft alleen te weten hoe hij toegang krijgt. Een vergelijking kan zijn het bedienen van software met een grafische interface of via een terminalvenster. Voorbeelden zijn API's voor slepen en neerzetten van bestanden in webpagina's, voor het grafische HTML-canvas of voor het aanspreken van de camera in de computer. Hierbij is altijd JavaScript nodig en dat valt buiten het bestek van dit boek.

Kennismaken met HTML en CSS

De volgende paragrafen veronderstellen dat HTML en het maken van webpagina's voor u volledig onbekend terrein zijn. Hebt u al ervaring met HTML en CSS, dan kunt u doorbladeren naar de paragrafen *Browsers* en *HTML-editors*.

Dit boek gaat allereerst over HTML. Maar tenzij u kale pagina's wilt maken, kunt u niet zonder CSS, wat staat voor Cascading Style Sheets. Deze taal is ontwikkeld om aan te geven hoe de met HTML gecodeerde teksten (en afbeeldingen en meer) eruit moeten zien. Ook CSS komt daarom uitgebreid aan bod.

Naast HTML en CSS kunnen programmeertalen worden gebruikt om webpagina's tot leven te brengen. JavaScript is zo'n programmeertaal. Deze wordt bijvoorbeeld gebruikt om acties na een klik op een knop uit te voeren of om onderdelen van een webpagina te manipuleren. Maar ook complete apps worden gebouwd met JavaScript-frameworks zoals Angular of React. Het is zeker een goed idee om u ook in JavaScript te verdiepen als u verder wilt met webontwikkeling.

HTML, CSS en ook JavaScript doen hun werk op het apparaat waarmee een bezoeker uw pagina's bekijkt. Er is natuurlijk communicatie met de webserver, maar het werk wordt gedaan door de browser. Daarmee vormen uw bezoeker, zijn computer/tablet/telefoon en vooral zijn browser de client-side. Dan vermoedt u dat er waarschijnlijk ook een ander kant is; dat is ook zo.

Aan de serverkant

Er zijn ook talen en technieken waarbij het werk op de webserver wordt gedaan, bijvoorbeeld het bij elkaar zoeken van componenten waaruit een webpagina wordt samengesteld. De server stuurt dan een kant-en-klare webpagina naar de client (de browser). Een veelgebruikte programmeertaal op webserver is PHP. Deze taal wordt bijvoorbeeld toegepast om een contentmanagementsysteem (kortweg cms) zoals WordPress (www.wordpress.org) of Joomla! (www.joomla.org) aan te sturen. PHP wordt daarbij onder meer gebruikt om pagina's samen te stellen aan de hand van informatie die in een database is opgeslagen. PHP blijft in dit boek verder buiten beschouwing.



Nog meer talen

Er zijn nog veel meer talen voor het web dan de hier genoemde, maar tenzij u specialistische toepassingen wilt ontwerpen, krijgt u daar niet mee te maken. Op w3c.org kunt u er echter van alles over vinden.

HTML – Hypertext Markup Language

HTML staat voor Hypertext Markup Language. Met HTML wordt de *structuur* van de pagina en de *betekenis* van de elementen in die pagina aangegeven. (In Engelstalige literatuur wordt gesproken over *structure* en *semantics*; onthoud deze termen). Er wordt met tags (labels) beschreven wat de kopteksten en alinea's van een pagina zijn en welke afbeeldingen in de pagina moeten worden geladen, en met hyperlinks wordt aangegeven wat de onderlinge relatie tussen pagina's is. HTML heeft niets te maken met het uiterlijk van een pagina.

HTML is daarom geen programmeertaal. Het is een markeertaal. Vandaar ook de afkorting Hypertext Markup Language, oftewel 'een markeertaal die is gebaseerd op hypertext'. Het principe van hypertext kent ondertussen iedereen. Dit zijn de tekstdelen waarmee relaties (ook wel koppelingen of links) tussen de pagina's onderling zijn aangegeven. Vrijwel elke webpagina bevat wel een of meer hyperlinks.

De hyperlinks, maar ook alle andere kenmerken, zijn de elementen van HTML en die worden gemarkeerd met tags. Om met HTML te kunnen werken, moet u weten welke elementen er zijn en hoe ze worden gemarkeerd. Stel dat u een stuk tekst op een pagina als koptekst wilt aanmerken. De code hiervoor is:

```
<h1>Dit is een koptekst</h1>
```

De code `<h1>` is de openingstag. Alles wat hierna volgt (in dit voorbeeld de tekst 'Dit is een kopstekst') wordt als kopstekst aangemerkt. Dit gaat zo door tot dat de sluittag `</h1>` wordt geschreven. Een stuk gemarkeerde tekst wordt altijd afgesloten met een sluittag. Een sluittag heeft dezelfde naam als de openingstag, maar begint met een voorwaartse slash `</...>`.



Afbeelding 1.5 Een kopstekst met daaronder de HTML-code.

In de browser ziet de kopstekst er zo uit als in de afbeelding. HTML bevat meer dan honderd elementen. In dit boek worden de meeste ervan besproken.

CSS – Cascading Style Sheets

Met CSS wordt het uiterlijk van de pagina aangegeven. CSS-code beschrijft hoe de kopsteksten, alineateksten en afbeeldingen worden opgemaakt. In CSS wordt aangegeven welk lettertype wordt gebruikt, welke lettergrootte, letterkleur, uitlijning, regelafstand, witruimte tot andere onderdelen en meer. Met CSS is ook de opmaak van de pagina in te stellen in kolommen, kop- en voetsteksten en kaders. Zelfs transformaties, overgangen (*transitions*) en animaties zijn mogelijk. CSS heeft echter niets te maken met de inhoud van de pagina.

Neem nog even het voorbeeld van de kopstekst uit de voorgaande paragraaf in gedachten. In HTML hebt u aangegeven dat een bepaalde tekst een kopstekst moet zijn, maar dat zegt nog niets over het uiterlijk. Wilt u de kopstekst bijvoorbeeld in een ander lettertype dan de gewone tekst, met een lijn eronder of een kleurvlak? Dit is nu juist de taak van CSS. Onthoud het volgende:

- Met HTML wordt de *structuur* van de webpagina en de *betekenis* van de inhoud gemarkeerd (kopsteksten, tussenkoppen, hyperlinks, alinea's en dergelijke).
- Met CSS wordt het *uiterlijk* van de webpagina ingesteld (lettertype, kleur, regelafstand, marges, en dat alles desgewenst per element van de pagina).

Om de kopstekst uit de afbeelding 1.5 te voorzien van een blauwe achtergrond met donkerblauwe onderrand en een cursief lettertype, kan de volgende stijlregel worden geschreven:

```
h1 {
  font-family: Arial, Helvetica, sans-serif;
  font-style: italic;
  background-color: rgba(150, 200, 255, 0.5);
  border-bottom-width: 4px;
  border-bottom-style: solid;
  border-bottom-color: rgb(150, 200, 255);
  padding-left: 15px;
}
```

Dit is een koptekst

```
h1 {
  font-family: Arial, Helvetica, sans-serif;
  font-style: italic;
  background-color: rgba(150, 200, 255, 0.5);
  border-bottom-width: 4px;
  border-bottom-style: solid;
  border-bottom-color: rgb(150, 200, 255);
  padding-left: 15px;
}
```

Afbeelding 1.6 Een koptekst opgemaakt met een stylesheet. In de HTML-code is niets gewijzigd. Het uiterlijk is aangepast met een stijlregel.

De complete CSS-familie bestaat ook uit enkele honderden opdrachten die gezamenlijk alle mogelijke manieren bieden om een webpagina op te maken. Hoe meer CSS-regels u kunt toepassen, hoe fraaier het eindresultaat in de browser kan zijn.



CSS3

Net als de structuurtaal HTML is de opmaaktaal CSS sterk in ontwikkeling. Er wordt al jaren gewerkt aan de nieuwste standaard: CSS3, hoewel die naam eigenlijk niet klopt. De specificatie bestaat uit losse modules, is zeer omvangrijk en (daardoor) nog lang niet af, zo die al ooit 'af' zal zijn. Veel onderdelen (modules) zijn wel bruikbaar en die komt u tegen in de hoofdstukken over CSS.