

Inhoud

1	Webtalen, browsers en editors	1
	Introductie	2
	Een korte terugblik	2
	Afspraken over webtalen	3
	Twee smaken HTML	5
	Wat HTML 5 is	6
	Kennismaken met HTML en CSS	7
	Aan de serverkant	7
	De functie van HTML	8
	De functie van CSS	9
	Browsers	11
	Google Chrome	12
	Microsoft Edge	13
	Mozilla Firefox	13
	MacOS: Safari	14
	HTML-editors	14
	Visual Studio Code	15
	Commerciële editors/IDE's	17
	Aanvullende hulpmiddelen	17
	Hulpmiddelen voor validatie	17
	Hulpmiddelen voor ontwikkeling	18
	Over dit boek	22
	Voorbeelden online	22
	Samenvatting	23
	Oefeningen	23
2	De basis van HTML	25
	Kenmerken van HTML-documenten	26
	Elementen, tags en tekst	26
	Opbouw van een element	27
	Elementen nesten	29

Inhoud

Kleine letters	29
Sluiten in de goede volgorde	29
De HTML-code van webpagina's bekijken	30
Attributen	33
Kenmerken van attributen	33
Globale attributen	34
Microdata	38
WAI-ARIA	38
Kenmerken van moderne HTML	39
Omringende technieken	40
De basis van een HTML-document	42
<!DOCTYPE html>	43
<html lang="nl">	43
<head>	43
<meta charset="utf-8">	43
<title>	44
<link>	44
<body>	45
Paginastructuur	45
Document Object Model – HTML DOM	45
Het contentmodel van HTML	46
Secties markeren	49
Toegankelijkheid	50
<article> (sectioning content)	51
<nav> (sectioning content)	51
<aside> (sectioning content)	52
<section> (sectioning content)	54
Koppen: <h1> t/m <h6>	55
<header> (flow content)	56
<footer> (flow content)	57
<main> (flow content)	59
Samenvatting	60
Oefeningen	61
3 Tekst markeren	63
Inleiding	64
Koppen in HTML-documenten	64
Titel van de website is niet automatisch <h1>	65
Toegankelijkheid	65
Koppen markeren	66
Koppen in een header	66
Alineatekst markeren	67
Alinea's: het element <p>	68

Het regeleinde 	69
Afbreken van woorden	70
Speciale betekenissen aangeven	70
Klemtoon leggen met 	71
Nadruk leggen met 	71
Aanvullende informatie is <small>	71
Ongeldig maken met <s>	71
Verwijzing naar boek of film: <cite> en <q>	72
Afkortingen en definities: <abbr> en <dfn>	72
Programmacode en dergelijke: <code>, <samp> en <kbd>	73
Subscript en superscript: <sub> en <sup>	73
Tekst markeren met <mark>	73
Het containerelement 	73
Toegevoegd of verwijderd: <ins> en 	74
Inhoud groeperen	74
<address>	74
Vooraf opgemaakte inhoud: <pre>	75
Citaten gebruiken: <blockquote>	75
Lijsten maken: , en 	76
Illustraties en bijschriften: <figure> en <figcaption>	80
FAQ, metadata en definities: <dl>, <dt>, <dd>	83
Nieuw thema markeren: <hr>	84
Het laatste redmiddel: <div>	85
Speciale tekens in webpagina's	85
Van ASCII naar UTF-8	86
Referenties gebruiken	87
Samenvatting	88
Oefeningen	89
4 Koppelingen maken	91
Inleiding	92
Verbindingen leggen met <a>	93
Toegankelijkheid	94
Attributen van <a>	95
Bladwijzers maken	98
Link naar een grote foto	99
Link naar het vervolg van een artikel	100
Link naar een e-mailadres	100
Link naar een telefoonnummer	101
Een pdf-bestand weergeven of downloaden	101
Klik hier voor richtlijnen	101
Externe bestanden koppelen met <link>	104
Stylesheet koppelen	105

Inhoud

Een favicon gebruiken	106
Opeenvolgende pagina's	107
Vooraf laden: prefetch, preload, prerender en dns-prefetch	107
Knoppen en hyperlinks	108
Samenvatting	110
Oefeningen	110
5 Beeld, geluid en andere externe inhoud	111
Inleiding	112
URL's en structuur van de website	112
Lokale site	114
Soorten afbeeldingen	114
Bitmaps	115
Vectorafbeeldingen: SVG	116
Bestandsformaten voor bitmaps	117
GIF	117
JPEG	118
PNG	119
WebP	120
AVIF	120
De juiste afbeelding voor het scherm	120
Afbeeldingen plaatsen met 	121
De bron: src	122
Beschrijvende tekst: alt	122
Het laden uitstellen	123
Afmetingen: width en height	123
Klikbare gebieden: usemap	124
Responsive images	125
Width, height en beeldverhouding	125
Srcset voor schermen met hoge resolutie	126
Srcset voor variabele afmetingen	128
Bijgesneden afbeeldingen: <picture>	130
Resolutie en afmetingen combineren	131
Het type afbeelding selecteren	131
Het verschil tussen en <picture>	131
Externe HTML-inhoud: <iframe>	132
Attributen van <iframe>	132
Beveiliging	133
Insluiten met <embed> of <object>	134
Video in een webpagina	134
Bestandstypen voor video	134
Een codec kiezen	135
Video coderen	135

Het element <video>	135
De bron: <source>	137
Ondertiteling met <track>	138
Titelbestanden: WebVTT	139
Audio op de website gebruiken	140
Samenvatting	141
Oefeningen	141
6 Formulieren maken	143
De rol van formulieren	144
Formulieren aan de clientkant	144
De rol van de webserver	145
De basis van een formulier: <form>	146
Buiten <form> mag ook	146
De verwerking: action	146
Verzendwijze: method	147
Codering: enctype	147
Automatisch aanvullen: autocomplete	148
Niet valideren: novalidate	148
Direct invoeren: autofocus	149
De naam: name	149
Het doel: target	149
Buitenspelelementen: form	150
Labels bij invoervelden	151
Het element <button>	152
De tabvolgorde	153
Het element <input>	153
Attributen van <input>	154
Soorten invoer: het attribuut type	161
Telefoonnummers: type="tel"	161
E-mailadressen: type="email"	161
Webadressen: type="url"	162
Wachtwoorden: type="password"	162
Getallen: type="number"	163
Een bereik: type="range"	163
Kleuren: type="color"	164
Datum en tijd	164
Selectievakjes: type="checkbox"	165
Keuzerondjes: type="radio"	166
Bestanden uploaden: <type="file">	168
Verzendknop: type="submit"	168
Resetknop: type="reset"	169
Afbeeldingknop: type="image"	169

Inhoud

Functieloze knop: type="button"	170	
Verborgen waarden: type="hidden"	170	
Kiezen uit een lijst: <select>, <datalist>, <option>	170	
<option>	171	
<select>	171	
<datalist>	172	
Uitgebreide tekstinput: <textarea>	174	
Groeperen met <fieldset> en <legend>	174	
Voortgang tonen: <progress>	177	
Toegankelijkheid	178	
Schaal: <meter>	179	
Resultaat: <output>	179	
Samenvatting	180	
Oefeningen	181	
7	Tabellen maken	183
Inleiding	184	
De structuur van een HTML-tabel	184	
De basis: <table>	186	
Rijen en kolommen: <tr> en <td>	186	
Koptekst: <thead>	187	
Een bijschrift: <caption>	188	
Overige elementen	189	
De tabelinhoud: <tbody>	189	
Voettekst: <tfoot>	190	
Kolomgroepen: <colgroup> en <col>	190	
Voorbeelden van tabellen	191	
Samenvatting	194	
Oefeningen	194	
8	De basis van CSS	195
Wat is CSS?	196	
Voorgeschiedenis	196	
En toen was er CSS3	197	
CSS3 is gewoon CSS	197	
Waarom CSS zo handig is	198	
De taal CSS	199	
De plaats van de CSS-declaraties	201	
Het attribuut style	201	
Het element <style>	202	
Extern bestand: <link>	202	
Andere stylesheets importen	203	

Het CSS-bestand	203
Commentaar toevoegen	203
Indeling van CSS-bestanden	204
De boomstructuur	204
Elementen benaderen met selectors	206
Elementnaam, klasse of beide?	207
Elementnamen en selectors	207
Klassen	209
Van alles wat	210
En de winnaar is...	210
Basisselectors	211
Typeselector	211
Universele selector *	212
Klasseselector	213
ID-selector	214
Attribuutselectors	214
Pseudoklassen	215
Pseudo-elementen	219
Combinatieselectors: afstamming, kind en sibling	220
Geen combinator: komma	220
Algemene afstamming: spatie	220
Kindselector	221
Aangrenzend: +	221
Op hetzelfde niveau: ~	221
Best practices	222
CSS-verwerking door de browser	223
De cascade	223
Specificiteit berekenen	224
De selector doet het niet. En nu?	226
Overschrijven met !important	227
Overerving	228
Nieuw: cascade layers (@layer)	229
Achtergrond	230
Lagen declareren	231
De volgorde van lagen	231
De ingebouwde stijlen aanpassen	234
Waarden en eenheden	236
Globale waarden	237
Lengte	237
De pixel	237
De eenheid em	238
De makkelijke em: rem	238
Handig voor tekstvakken: ch	239
Eenheden gebaseerd op de viewport	239

Inhoud

Overige eenheden	240
Wiskundige expressies	241
Samenvatting waarden en eenheden	242
Samenvatting	243
Oefeningen	243
9 Lay-out: boxmodel en weergavemodel	245
De opbouw van pagina's	246
Browserstijlen zijn er niet voor niets	246
Het boxmodel	247
De eigenschap box-sizing	248
Eigenschappen voor breedte en hoogte	249
Waarden voor breedte en hoogte	251
Marges	251
Horizontaal centreren	252
Negatieve marge	253
Samengevoegde marges	253
Opvulling	254
Randen	255
Standaardranden	255
Randen met logische eigenschappen	256
Breedte, stijl of kleur apart instellen	256
Overlopende inhoud: overflow	257
Overloop van de scrol	259
Weergavemodel	260
Achter de schermen	260
De eigenschap display	261
Waarden voor binnen en buiten	261
Inhoud (niet) weergeven: visibility	264
Niet verbergen voor screenreaders	264
Samenvatting	265
Oefeningen	266
10 Lay-outs maken met CSS	267
Inleiding	268
Responsive design	268
Media queries	269
Andere media features	271
Rekening houden met de bezoeker	272
Weergave op mobiele schermen: de metatag viewport	273
CSS-pixels en hardwarepixels (device pixels)	274

Lay-out: positionering	275
Het omvattende blok	275
Positioneringsschema's	276
De eigenschap position	276
Een element verschuiven	276
Position: static	277
Position: relative	277
Position: absolute	278
Position: fixed	280
Position: sticky	280
Float en clear	282
De block formatting context	285
Lay-out: multi-column	285
Kolommen instellen	285
Marges worden niet samengevoegd	287
Kolommen overspannen	288
Ruimte tussen de kolommen	289
Lijn tussen de kolommen	290
Kolommen vullen	291
Afbreken in kolommen	291
Wanneer multi-column (niet) nuttig is	294
Toepassing: portfolio	295
Lay-out in een raster: gridlay-out	296
Kenmerken van gridlay-out	297
Hulpmiddelen voor gridlay-out en flexbox	298
Een raster maken	300
De functie repeat()	301
De eenheid fr	301
Rasteritems plaatsen op lijnummers	302
Kolommen en rijen samen in grid-area	305
De lijnen een naam geven	305
Items plaatsen in gebieden met een naam	307
Afmetingen van tracks instellen	310
Ruimte tussen tracks	311
Het raster zichzelf laten bouwen	311
Het verschil tussen auto-fill en auto-fit	314
Van richting veranderen met grid-auto-flow	314
De gaten in het raster opvullen	315
Uitlijnen in gridlay-out	316
Toepassing: nos.nl	319
De lay-out tot 760px	323
De lay-out van 760px tot 900px	327
De lay-out vanaf 900px	328
De lay-out vanaf 1216px	330
Een achtergrond voor leesbare koptekst	330

Lay-out: flexbox	331
Kenmerken van flexbox	331
Het assenstelsel van flexbox	331
Flex-direction	332
Flex-wrap	333
Flex-flow	333
Flexcontainers en flexitems	333
Items flexibel maken met de eigenschap flex	334
Instellingen voor flex	335
Samenvatting: het formaat van flexitems	336
Items uitlijnen met automarge	337
Uitlijnen op de hoofdas: justify-content	338
Uitlijnen op de kruisas: align-items en align-self	340
Regels uitlijnen: align-content	341
Toepassing: navigatiebalk	342
Toepassing: kaartlay-out	344
Opdracht: nos.nl	348
De toekomst: box alignment	348
Nieuw: container queries	349
Voorbeeld: kaartlay-out	350
Samenvatting	353
Oefeningen	355
11 Tekst en typografie	359
Inleiding	360
Lettertype: van systeem of online?	360
Web fonts	361
Lettertypen downloaden: @font-face	362
Zelf webfonts hosten	363
Eigenschappen van @font-face	364
Een embedcode gebruiken	366
Eigenschappen voor lettertype: font	367
Lettertype	367
Lettergrootte: waarden en overwegingen	368
Geavanceerd: responsieve tekst	369
Toepassing van font-size	370
Grootte aanpassen: font-size-adjust	370
Regelhoogte: line-height	370
Letterstijl: font-style	372
Lettergewicht: font-weight	372
De korte eigenschap font	373
Uitrekken of indrukken: font-stretch	374
Kleinkapitaal: font-variant	374

Eigenschappen voor tekst	375
Uitlijnen: text-align en text-align-last	376
Lijneffecten op tekst	376
Inspringen	378
Rechtopstaande tekst: text-orientation	378
Overlopende tekst: text-overflow	379
Tekst met schaduw	379
Tekst omzetten: text-transform	380
Witruimte behouden: white-space	381
Ruimte tussen woorden en letters	382
Afbreken: word-break en overflow-wrap	382
Opmaak van lijsten	384
Opsommingstekens	384
Lijst als navigatie	385
Samenvatting	390
Oefeningen	391
12 Kleur, randen en achtergronden	393
Inleiding	394
Kleurwaarden	394
RGB	395
HSL en HSLA	396
Transparantie	398
Kleur van tekst	398
De achtergrond	399
Achtergrondkleur	399
Achtergrondafbeelding	399
background-image	399
background-repeat	400
background-attachment	401
background-position	401
background-clip	403
background-origin	404
background-size	404
background	405
Verschillende achtergrondafbeeldingen	406
Kleurverloop als achtergrond	407
Kleurverlopen	408
Lineair verloop	408
Radiaal verloop	409
Repeterende verlopen	410
Afgeronde hoeken	411

Randafbeelding	414
border-image-source	414
border-image-slice	414
border-image-width	415
border-image-outset	415
border-image-repeat	415
Voorbeeld van een randafbeelding	415
Schaduw	416
De werking van schaduw	417
Material design	418
Afbeeldingen uitsnijden met clip-path	419
Toepassing: header met schuine kanten	421
Beeldfilters	423
Kleuren mengen	424
Isolation	425
Samenvatting	426
Oefeningen	427
13 Overgangen en animaties, attr() en CSS-variabelen	429
Beweging in webpagina's	430
Overgangen (transitions)	430
Kant-en-klare overgangen	431
Positie aanpassen	432
Animatie	433
Opbouw een animatie	433
Keyframes	435
Transformatie	436
Tweedimensionale transformaties	437
Driedimensionale transformaties	439
Perspectief	440
Kijkrichting	441
3D-ruimte	441
CSS-variabelen: de functie var()	445
Kenmerken van variabelen	445
Variabelen declareren	446
Toepassingen van variabelen	446
Attributen uitlezen met attr()	448
Samenvatting	449
Oefeningen	449
1Index	451

Webtalen, browsers en editors

HTML is de belangrijkste taal op internet. Sinds 2011 werken we met HTML 5 en daar wordt nog dagelijks aan gesleuteld. Dit hoofdstuk beschrijft het ontstaan van HTML 5 en de actuele situatie. Er wordt kort uitgelegd wat HTML en CSS zijn en wat HTML 5 is. Er is een overzicht van browsers en hulpmiddelen voor het schrijven van HTML-documenten en u maakt kennis met handige gereedschappen.

U leert in dit hoofdstuk:

Hoe we hier gekomen zijn.

De rol van HTML en van CSS, en wat HTML 5 is.

Welke browsers niet mogen ontbreken in uw gereedschapskist.

Handige gereedschappen: editors, validators en hulpmiddelen voor ontwikkelaars.

Introductie

Het world wide web met zijn blogs, games, webwinkels, videosites, sociale media, magazines en al die andere soorten sites en apps zou niet bestaan zonder de taal HTML, de afkorting van Hypertext Markup Language. Het fraaie uiterlijk wordt gemaakt met CSS (Cascading Style Sheets) en de logica wordt geprogrammeerd met JavaScript, maar zonder HTML was er niets.

Dit boek gaat over HTML 5. Geloof het of niet: er wordt nog elke dag aan gewerkt, al is het resultaat daarvan niet dagelijks als vernieuwing terug te vinden in de browser.

HTML 5 kan niet los worden gezien van de voorgangers, HTML 4.01 en XHTML. HTML 5 is daar een voortzetting van, waarbij onderdelen zijn aangepast, toegevoegd en afgeschaft, maar alle oude webpagina's blijven werken. 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. Doen ze dat niet, dan blijven de webpagina's toch toegankelijk voor bezoekers. Ze zien er waarschijnlijk niet meer precies hetzelfde uit als in 2000 werd bedoeld, maar de browser moet in elk geval de inhoud weergeven.



Afbeelding 1.1 *Het logo van HTML 5, in 2011 gepresenteerd door het W3C.*

Een korte terugblik

De ontwikkeling van HTML 5 begon al in 2007. Sinds 2011 is het beschikbaar in de webbrowsers. De specificatie was toen nog volop in ontwikkeling, maar de belangrijkste elementen waren al omarmd door de gemeenschap van webbouwers en door browserfabrikanten. In 2014 heeft de beheerder van de webtalen, het World Wide Web Consortium (W3C) de specificatie 'definitief' vastgesteld (in W3C-termen is het dan de aanbevolen specificatie, gelabeld als *recommendation*). Definitief staat niet voor niets tussen aanhalingstekens, want het werk aan de specificatie gaat gewoon door. Het betekent alleen dat die versie wordt bevroren en dat het werk verder gaat in de volgende versie. Die kwamen er in 2016, en zelfs twee keer in 2017. Daarna is de ontwikkeling van de HTML-specifi-

catie overgedragen aan een andere organisatie: WHATWG. Die doet niet aan versienummers en werkt dus dagelijks aan *HTML, the living standard*. Elke revisie bevat kleine veranderingen. Vaak zijn het aanvullingen en verbeteringen gebaseerd op hoe HTML wordt gebruikt en op reacties uit de gebruikersgemeenschap. Soms wordt een element afgekeurd (omdat het nauwelijks wordt gebruikt) of verandert de toepassing ervan. *Breaking changes* zijn in elk geval niet te verwachten, omdat het web nu eenmaal niet mag ‘breken’.

HTML

Living Standard — Last Updated 1 March 2022

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

Afspraken over webtalen

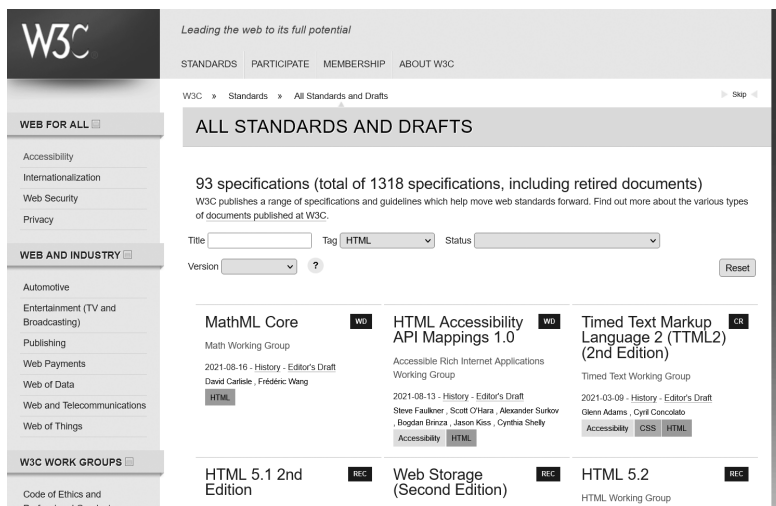
Sinds 1994 wordt geprobeerd lijn te brengen in de talen waarmee websites (en andere webtoepassingen) worden gemaakt. Die talen zijn onder meer Hypertext Markup Language (HTML) en Cascading Style Sheets (CSS). Een standaard voor deze talen is geen overbodige luxe. Toen de belangstelling voor internet groeide en er meer browsers op de markt werden gebracht, deed elke fabrikant wat hem leuk leek. De HTML-standaard werd wel ingebouwd, maar daar werd een eigen draai aangegeven. Zo ontstond een situatie waarin browsers naast standaard-HTML ook eigen tags hadden (berucht waren `<b1 ink>` voor knipperende tekst en `<bgsound>` voor achtergrondgeluid). Bij CSS was de situatie vergelijkbaar. Zo kon een website met dezelfde code er in elke browser anders uitzien, anders werken of compleet uit elkaar vallen. Vooral Microsoft heeft met eigen interpretaties van de standaarden in Internet Explorer ontwikkelaars tot wanhoop gedreven. Intussen zijn dergelijke problemen grotendeels opgelost en werken de talen in alle browsers grotendeels hetzelfde.

Voor het beheer van de talen is het World Wide Web Consortium opgericht, kortweg W3C. In verschillende werkgroepen wordt overlegd door browserfabrikanten zoals Google, Microsoft, Mozilla en Opera. Omdat W3C ook allerlei andere webstandaarden beheert, vindt u onder de leden ook bedrijven als Bloomberg, Vrije Universiteit Amsterdam, Nokia en Walt Disney Company. 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 informatie over elke webstandaard. Daar hoort JavaScript niet bij, dat wordt beheerd door Ecma (**www.ecma-international.org**).



The screenshot shows the W3C website's 'ALL STANDARDS AND DRAFTS' page. The header includes the W3C logo and the tagline 'Leading the web to its full potential'. Navigation links for 'STANDARDS', 'PARTICIPATE', 'MEMBERSHIP', and 'ABOUT W3C' are visible. The main content area displays '93 specifications (total of 1318 specifications, including retired documents)'. Below this, there are search filters for 'Title', 'Tag' (set to 'HTML'), 'Status', and 'Version'. A 'Reset' button is also present. The main list of specifications includes:

- MathML Core** (WD) - Math Working Group, 2021-08-16 - History - Editor's Draft, David Carlsie, Frédéric Wang, HTML
- HTML Accessibility API Mappings 1.0** (WD) - Accessible Rich Internet Applications Working Group, 2021-08-13 - History - Editor's Draft, Steve Faulkner, Scott O'Hara, Alexander Surkov, Bogdan Brinza, Jason Kiss, Cynthia Shelly, Accessibility, HTML
- Timed Text Markup Language 2 (TTML2) (2nd Edition)** (ES) - Timed Text Working Group, 2021-03-09 - History - Editor's Draft, Glenn Adams, Cyril Concolato, Accessibility, CSS, HTML
- HTML 5.1 2nd Edition** (REC)
- Web Storage (Second Edition)** (REC)
- HTML 5.2** (REC) - HTML Working Group

Afbeelding 1.3 Een (heel) klein deel van de standaarden en concepten op de website van W3C.



Alle hyperlinks staan online

Op handboek-html-css.nl vindt u alle hyperlinks die in het boek worden genoemd, gerangschikt per hoofdstuk.

Het is allerminst vanzelfsprekend dat er nu een W3C-standaard van HTML 5 is. Eind jaren 1990, de standaard HTML 4.01 was klaar, besloot het W3C 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 (XHTML 2) 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 (HTML 4) 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 niet genoeg medestanders voor. In reactie daarop besloten Apple, Mozilla en Opera dan maar zelf aan de slag te gaan met HTML 5. 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 HTML 4 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 mediatechnieken zoals Flash van Adobe en Silverlight van Microsoft.

Dit initiatief viel bij de gemeenschap van webontwikkelaars in goede aarde. Ontwikkelaars ontvingen de ontwikkelingen met gejuich en browserfabrikanten gingen de nieuwe ideeën inbouwen in hun browsers, en daardoor ging uiteindelijk ook bij het W3C het roer om. De werkgroep XHTML is eind 2009 opgeheven.

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), Apple (Safari), Mozilla (Firefox) en Microsoft (Edge) brengen in hoog tempo updates voor de browser uit. Het resultaat is dat de nieuwste versies van de 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.

Twee smaken HTML

Gedurende enkele jaren waren er in theorie twee smaken HTML. Hoewel de oorspronkelijke HTML 5-aanbeveling op de website van het W3C het resultaat is van samenwerking tussen W3C en WHATWG, zijn de partijen na 'HTML 5.0' weer apart verder gegaan. Eerst leek het erop dat het W3C HTML 5 helemaal niet verder zou ontwikkelen, maar dat bleek mee te vallen. WHATWG ging ook gewoon door en zo liepen er een aantal jaren twee HTML-specificaties naast elkaar. Webontwikkelaars merkten daar nauwelijks iets van. Hun enige zorg is of een bepaalde optie goed werkt in de browser en het is niet moeilijk om daar achter te komen. (U mag alle websites vergeten, maar onthoud altijd deze twee: caniuse.com en developer.mozilla.org.)

Intussen heeft W3C de ontwikkeling van HTML overgedragen aan WHATWG en daar vindt u nu ook de enige echte specificatie: html.spec.whatwg.org/multipage/.

4 The elements of HTML

4.1 The document element

4.1.1 The `html` element

Categories:

None.

Contexts in which this element can be used:

As document's document element.

Wherever a subdocument fragment is allowed in a compound document.

Content model:

A `head` element followed by a `body` element.



Afbeelding 1.4 *Hier gaan we het uitgebreid over hebben.*

Wat HTML 5 is

HTML 5 is eigenlijk niet meer dan een upgrade van HTML 4. We hebben het dan alleen over de taal HTML. Maar HTML 5 wordt ook gebruikt als verzamelnaam voor webtechnieken om moderne (mobiele) webapplicaties te bouwen. Naast HTML gaat het dan ook over een verzameling definities waarmee de ontwikkelaar toegang krijgt tot onderdelen van de browser of smartphone (API's, zie de opmerking hierna) en aanvullende technieken zoals CSS, JavaScript, XML, JSON en SVG. Maar deze horen niet bij de HTML 5-standaard zelf. Ze hebben wel een sterke relatie en het is bijna niet mogelijk om moderne websites en webapps te bouwen zonder er een of meer te gebruiken. Op www.w3.org/TR/ kunt u zoeken met de tag Web API om een overzicht te bekijken. Ook op spec.whatwg.org vindt u een overzicht van technologieën voor het webplatform.

Over HTML en CSS leest u alles in dit boek. Voor de andere technieken moet u andere bronnen raadplegen. Een korte toelichting op de functie ervan:

- JavaScript is een programmeertaal (scripttaal) om onder meer webpagina's interactief te maken: (onder)delen van de pagina updaten, gegevens lezen uit een databron (JSON), een winkelwagen bijhouden, cookies plaatsen en lezen enzovoort.
- XML staat voor Extensible Markup Language en is een markeertaal (zoals HTML). XML 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 veelgebruikt alternatief voor XML. De functie is hetzelfde: uitwisseling van gegevens tussen een databron en een applicatie.
- 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).



Toegangspunt voor ontwikkelaars: API

De afkorting API staat voor application programming interface. Het is een toegangspunt voor aanvullende functies waarbij de ontwikkelaar eenvoudig toegang heeft tot die functies zonder dat hij precies hoeft te weten hoe die werken. Hij hoeft alleen te weten hoe hij toegang krijgt. Een vergelijking kan zijn het bedienen van software met een grafische interface of met getypte opdrachten in een terminalvenster. Voorbeelden zijn API's voor het opnemen van beeld en geluid met een telefoon, voor het werken met een muis, pen of aanraakscherm, voor het werken met de geografische locatie van de gebruiker, of voor het opslaan van gegevens op het apparaat van de gebruiker. Om een API te gebruiken is altijd JavaScript nodig.

Kennismaken met HTML en CSS

Lees zeker de volgende paragrafen als HTML en het maken van webpagina's voor u volledig onbekend terrein zijn. Hebt u al wat ervaring met HTML en CSS, dan kunt u ze eventueel overslaan.

Het eerste deel van dit boek gaat over HTML. Maar tenzij u kale tekstpagina'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 gemarkeerde teksten (en afbeeldingen en meer) eruit moeten zien en hoe de webpagina moet worden ingedeeld, de lay-out. Het tweede deel van het boek gaat over CSS.

Naast HTML en CSS kunnen andere programmeertalen worden gebruikt om webpagina's tot leven te brengen. JavaScript is zo'n taal. 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, React, Vue.js of Svelte. Het is belangrijk om u ook in JavaScript te verdiepen als u verder wilt met webontwikkeling.

HTML, CSS en ook JavaScript doen hun werk in de browser 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.

Aan de serverkant

Er zijn ook talen en technieken waarbij het werk op de webserver wordt gedaan, server-side dus, 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 content-managementsysteem (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.

Andere talen zijn C#, Perl, Python, Ruby, Java, Kotlin en Go.



JavaScript op de server

Met de opkomst van JavaScript-frameworks zoals Angular, React en Vue.js is ook de rol van JavaScript als servertaal belangrijker geworden. Daarbij spelen het platform Node.js (www.nodejs.org) en de pakketbeheerder npm (www.npmjs.com) een centrale rol. Met Node.js kunnen toepassingen worden geschreven in JavaScript (of TypeScript). Node.js bevat onder meer een webserverfunctie, waarmee op de server webpagina's en apps kunnen worden samengesteld uit componenten die zijn gebouwd met de genoemde frameworks (*server side rendering*). Dit in tegenstelling tot pure HTML-pagina's die u kant en klaar op de server zet. In dit boek kunnen al die mogelijkheden niet worden uitgelegd, maar u weet nu dat er na HTML en CSS nog veel meer te ontdekken is.

De functie van HTML

HTML staat voor Hypertext Markup Language. Met HTML worden de *structuur* van de pagina en de *betekenis* van de elementen in die pagina aangegeven. (In Engelstalige bronnen wordt gesproken over *structure* en *semantics*). Er wordt met tags (labels) beschreven wat de kopteksten en alineateksten 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 een programmeertaal om tekst te markeren (*markup*). Vandaar ook de afkorting Hypertext Markup Language, oftewel 'een markeertaal voor hypertekst'. Hypertekst is tekst met hyperlinks (koppelingen) naar andere tekst, afbeeldingen, audio, video en meer.

De hyperlinks, maar ook alle andere onderdelen van een webpagina, 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 welke tags daar bij horen. Stel dat u een artikel hebt met een kop en een alinea. De HTML-code hiervoor kan zijn:

```
<article>
  <h1>De functie van HTML</h1>
  <p>HTML staat voor Hypertext Markup Language. Met HTML worden de
  <i>structuur</i> van de pagina en de <i>betekenis</i> van de
  elementen in die pagina aangegeven.</p>
</article>
```

Een artikel wordt op een HTML-pagina aangeduid met het element `article`. Het begin van het artikel wordt gemarkeerd met de openingstag `<article>`. Voor de belangrijkste kop op de pagina heeft HTML het element `h1`. Het begin van de koptekst is gemarkeerd met de openingstag `<h1>`. Alles wat hierna volgt (in dit voorbeeld de tekst De functie van HTML) wordt gemarkeerd als koptekst

(met de h van *header*). De kop loopt door tot de sluittag `</h1>`. Een stuk gemarkeerde tekst wordt altijd afgesloten met een sluittag. Een sluittag heeft dezelfde naam als de openingstag en begint met een voorwaartse slash `</...>`.

Hetzelfde geldt voor de alineatekst. De openingstag `<p>` markeert het begin van de alinea (*paragraph*) en de sluittag `</p>` het einde ervan.

In de alinea ziet u nog meer tags, twee keer de openingstag `<i>` en de sluittag `</i>`. Deze markeren belangrijke begrippen. In dit boek zijn die woorden om dezelfde reden cursief. Toch staat `<i>` niet voor *italic* (cursief). ‘Vroeger’, toen HTML-tags ook opmaak markeerden, was dat wel zo, maar nu markeert het element `i` tekst die op de een of andere manier afwijkt van de rest, meer daarover in hoofdstuk 2. De tags `<i>` en `</i>` staan binnen de alineatags `<p>`. Dit insluiten of nesten van tags is een belangrijk principe in HTML. Overigens kunnen niet alle tags worden genest. Hoe dat zit komt ook in volgende hoofdstukken aan de orde.

Het artikel wordt afgesloten met de sluittag `</article>`. Alles wat tussen de begin- en eindtag `<article>` staat, hoort bij het HTML-element `article`. Dat ‘doet’ verder niets, het geeft alleen aan dat alles wat tussen de tags staat bij het artikel hoort. Op die manier is ook duidelijk dat de elementen `h1` en `p` bij elkaar horen. In de browser ziet het voorbeeld er zo uit als in de afbeelding.

HTML bevat meer dan honderd elementen. In dit boek worden de meeste ervan besproken, maar u zult ze zeker niet allemaal dagelijks gebruiken.

De functie van HTML

HTML staat voor Hypertext Markup Language. Met HTML wordt de *structuur* van de pagina en de *betekenis* van de elementen in die pagina aangegeven.

```
<article>
  <h1>De functie van HTML</h1>
  <p>HTML staat voor Hypertext Markup Language. Met
    HTML wordt de <i>structuur</i> van de pagina
    en de <i>betekenis</i> van de elementen in die
    pagina aangegeven.</p>
</article>
```

Afbeelding 1.5 Een artikel met een kop en een alinea met benadrukte woorden. Het uiterlijk is bepaald door de browser. Deze bevat een basisopmaak voor elk HTML-element: zwarte tekst met een schreef en ruime marges.

De functie van CSS

Met CSS wordt het uiterlijk van de pagina bepaald. CSS-code beschrijft hoe de kopteksten, alineateksten en alle andere elementen op de pagina worden opge-maakt. In CSS wordt aangegeven welk lettertype wordt gebruikt, hoe groot de tekst is, welke kleur die heeft, de uitlijning, regelafstand, witruimte tussen

onderdelen en veel meer. Met CSS wordt ook de lay-out van de pagina ingesteld: kolommen, header, footer, kaders enzovoort. Zelfs transformaties, overgangen en animaties zijn mogelijk. CSS heeft dus alles te maken met vormgeving, maar niets met de inhoud van de pagina.

Neem nog even het voorbeeld van het artikel uit de voorgaande paragraaf in gedachten. In HTML wordt aangegeven dat een bepaalde tekst een koptekst moet zijn, maar dat zegt nog niets over het uiterlijk. Welk lettertype en welke kleur moet de koptekst hebben en hoe groot moet die zijn? Dat is de taak van CSS. Onthoud het volgende:

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

Met een paar regels CSS wordt het voorbeeld uit afbeelding 1.5 opgemaakt tot afbeelding 1.6.



Afbeelding 1.6 *Hetzelfde artikel als in vorige afbeelding, opgemaakt met CSS. In de HTML-code is niets gewijzigd, alleen het uiterlijk is aangepast met enkele opmaakregels.*

Het lettertype, het gewicht (de dikte) van de tekst en tekstkleur zijn ingesteld. De tekstgrootte is niet aangepast, maar doordat een ander lettertype is gekozen, is de tekst toch wat groter. Dit zijn eenvoudige aanpassingen en met CSS is heel veel meer mogelijk. De complete CSS-familie bestaat uit enkele honderden eigenschappen die samen alle mogelijke manieren bieden om een webpagina op te maken. Hoe meer CSS-eigenschappen u kunt toepassen, hoe meer mogelijkheden u hebt voor een fraai eindresultaat.



CSS zonder nummer

Net als HTML is CSS sterk in ontwikkeling. Er wordt al jaren gewerkt aan de nieuwste standaard: CSS 3, hoewel die naam eigenlijk niet klopt. Een groot deel van de oorspronkelijke specificatie is verdeeld in modules en er komen ook nieuwe modules bij. Al die modules worden onafhankelijk van elkaar ontwikkeld en daardoor hebben ze ook verschillende versie-nummers (1, 2, 3 maar ook 4). Vandaar dat ‘CSS versie 3’ niet bestaat; het W3C heeft het ook gewoon over CSS.

Browsers

Om webpagina’s te kunnen maken en de uitleg in dit boek te kunnen volgen, hebt u geen bijzondere programma’s nodig. Het maakt ook niet uit of u werkt op een pc met Windows of Linux of op een Mac met macOS (al zullen de gebruikte programma’s natuurlijk iets verschillen). Elk platform gebruikt dezelfde HTML- en CSS-code.

U hebt weinig nodig om aan de slag te kunnen gaan:

- een tekstverwerker (*editor*) om HTML- en CSS-bestanden te schrijven en op te slaan;
- een webbrowsier om te kunnen bekijken wat u maakt.

De browser is natuurlijk een belangrijk programma. Tijdens het coderen van webpagina’s gebruikt u de browser om te bekijken hoe de eindgebruikers uw pagina’s te zien krijgen. Gebruik verschillende browsers, zodat u kunt controleren of uw pagina’s ook in andere browsers goed werken. Bedenk daarbij dat in modern ontwerp verschillen in weergave tussen browsers acceptabel zijn zolang de bruikbaarheid en functionaliteit niet in het geding zijn. Verschillen ontstaan overigens zelden door de HTML-code (HTML gaat over betekenis, niet over uiterlijk), maar meestal door de CSS-code (CSS is voor de opmaak).

In Nederland zijn de meestgebruikte browsers op desktopcomputers Chrome, Safari en Edge. Op telefoon en tablet zijn dat Chrome en Safari. Hoewel het lastig is om exacte marktaandeelen te achterhalen, is de grote lijn duidelijk. Op desktop is het aandeel van Chrome 65% en van Safari en Edge zo’n 15%. Op tablets schommelen Chrome en Safari rond 45% en heeft Android 7%. Op mobiel is de verdeling anders: Safari 55%, Chrome 37% en Samsung Internet 10%. Het aandeel van de andere browsers is op mobiel verwaarloosbaar. Dit zijn cijfers van 2021 (bron: gs.statcounter.com). Europees en wereldwijd gezien zijn er wel verschillen in aandeel, maar in grote lijnen komt de verhouding overeen.



Verschillen tussen browsers

Het is nogal een open deur, maar uw website moet in elke browser goed werken. Ook als die een klein marktaandeel heeft. Dat wil zeggen dat alle tekst leesbaar moet zijn, links en formulieren goed werken enzovoort. Of het bedoelde ontwerp ook in elke browser tot op de pixel perfect moet zijn, hangt vooral af van uw opdrachtgever. Maar verschillen worden steeds vaker geaccepteerd, omdat er zo veel verschillende apparaten, browsers en schermafmetingen bestaan. Bovendien stellen de verschillen van vandaag weinig voor vergeleken bij de verschillen die er waren toen elke browserfabrikant zijn eigen draai aan HTML en CSS gaf.

Google Chrome

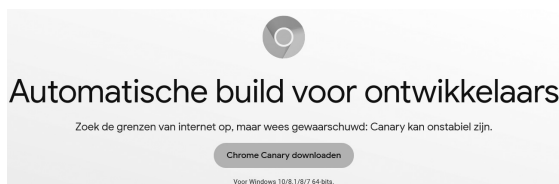
Het lijkt alsof hij er altijd al was, maar toch is Chrome, de browser van Google, pas eind 2008 gelanceerd. Tot die tijd was Internet Explorer de onbetwiste marktleider, al was het maar omdat de browser was vastgeklamd aan Windows. Chrome heeft op de desktop intussen de strijd met Internet Explorer/Edge en Firefox beslist. Vanwege het grote marktaandeel mag deze browser dus niet op uw computer ontbreken. U kunt hem zelf downloaden en installeren vanaf www.google.com/chrome. Chrome volgt de webstandaarden. De browsers werd eerst gebouwd op de engine WebKit, die ook in Safari van Apple wordt gebruikt. Sinds 2013 bevat Chrome een eigen versie van WebKit genaamd Blink.



Browser engine

Een browser engine (ook wel *layout engine* of *rendering engine* genoemd) vertaalt de inhoud van een webpagina (de HTML) en informatie over opmaak (de CSS) naar een weergave op het scherm. De engine bepaalt dus welke onderdelen van HTML en CSS de browser kan gebruiken. Een CSS-eigenschap die niet in de programmacode van de engine zit, kan ook niet goed worden verwerkt. Er wordt dan gezegd dat de eigenschap niet wordt ondersteund. Bekende engines zijn WebKit (Safari), Blink (Chrome) en Gecko (Firefox).

Handig is de testversie van Chrome: Canary. Met deze ‘permanente bètaversie’ kunt u nieuwe functionaliteit testen. Canary wordt elke dag bijgewerkt en kan tegelijk met de stabiele versie van Chrome op uw computer worden uitgevoerd.



Afbeelding 1.7 Google Canary is aan de buitenkant niet anders dan Chrome, maar technisch loopt die versie altijd voor.

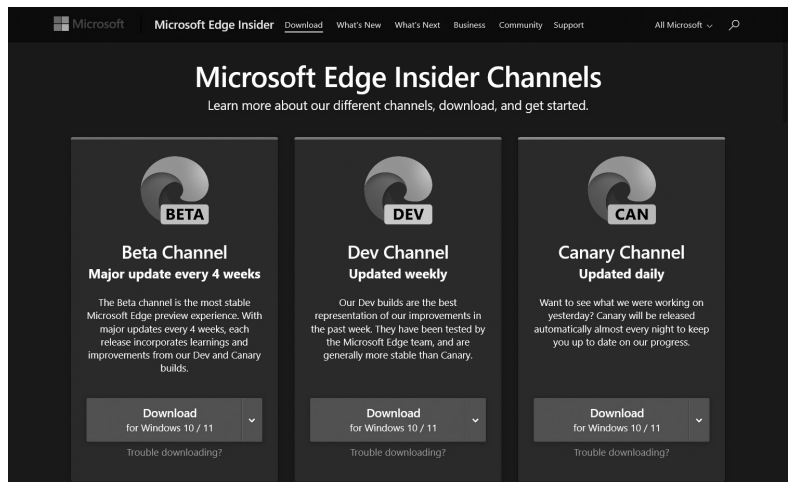
Chrome kan eindeloos worden uitgebreid met extensies. Kijk op de site chrome.google.com/extensions waar u in de categorie Ontwikkelaarstools handige browseruitbreidingen vindt.

Microsoft Edge

Internet Explorer raakte na de komst van Chrome zijn traditionele status als meestgebruikte browser al kwijt en inmiddels is de browser vervangen door Microsoft Edge. Deze is geïntroduceerd bij Windows 10. Edge biedt nagenoeg dezelfde goede ondersteuning van webstandaarden als de concurrenten. Het is een browser in ontwikkeling en er zijn vergeleken met Chrome en Firefox nog niet veel extensies/add-ons voor en dat aantal groeit niet snel. Regelmatig verschijnt een update die de functionaliteit van Edge vergroot. De thuispagina van Microsoft Edge is te vinden op www.microsoft.com/nl-nl/edge.

Aanvankelijk draaide Edge op een door Microsoft zelf ontwikkelde browser-engine: EdgeHTML. Sinds 2020 bevat Edge de engine Blink. Daarmee is de ondersteuning van HTML en CSS vergelijkbaar met die van Chrome.

Om de nieuwste toevoegingen uit te proberen zijn er versies voor ontwikkelaars: Beta, Dev en Canary. U vindt ze op www.microsoftedgeinsider.com/en-us/download.



Afbeelding 1.8 Versies voor ontwikkelaars die vooruitlopen op de publieksversie van de browser.

Mozilla Firefox

Korte tijd leek wereldheerschappij binnen handbereik, maar ook Mozilla Firefox is overvleugeld door Chrome. Er was zelfs een moment waarop Firefox de leidende positie van Internet Explorer leek te gaan overnemen. Vanwege de uitste-

kende implementatie van webstandaarden was het de lieveling van ontwikkelaars. Firefox is ooit ontstaan uit de failliete boedel van de oerbrowser Netscape Navigator. Programmeurs die eerst voor Netscape werkten, hebben in eigen beheer en met samenwerking van de opensourcegemeenschap een nieuwe browser ontwikkeld. Deze browser werd Firefox genoemd en in het najaar van 2004 werd versie 1.0 uitgebracht. In 2006 verscheen Firefox 2.0. Binnen enkele weken na de introductie was het programma miljoenen keren gedownload. Ook Firefox 3.0 (juni 2008) was erg populair en staat zelfs in het Guinness Book of Records met het record 'Meeste downloads in een dag': ruim 8 miljoen. Voorjaar 2022 was de browser bij versie 98 en updates verschijnen met grote regelmaat. Download Firefox vanaf www.mozilla.org/nl/firefox/new/.

Net als bij Chrome zijn er veel extensies/add-ons waarmee de functionaliteit van de browser kan worden uitgebreid. Zoek bijvoorbeeld eens met de trefwoorden HTML en CSS naar extensies op addons.mozilla.org/nl/firefox.

Van Firefox is ook een versie voor ontwikkelaars beschikbaar: Firefox Browser Developer Edition (www.mozilla.org/nl/firefox/developer/). Deze bevat nog meer nuttige hulpmiddelen voor webontwikkelaars dan de gewone editie.

MacOS: Safari

Alle hiervoor genoemde browsers zijn ook beschikbaar voor het besturingssysteem van Apple: macOS. Dat besturingssysteem heeft echter ook een eigen browser: Safari. De ondersteuning van onderdelen van HTML is globaal vergelijkbaar met Chrome. Bij ontwikkelaars die werken op een Mac is vanwege de goede ondersteuning van standaarden en de uitstekende hulpmiddelen voor ontwikkelaars ook hier Google Chrome een populaire browser. Safari was ooit ook beschikbaar voor Windows, maar dat is niet meer zo. Het marktaandeel op de desktop is bescheiden, maar op mobiel (iPhone) en tablet (iPad) is Safari de meestgebruikte browser.

Safari is een belangrijke browser. Zelfs als verstokte Windows-gebruiker moet u uw werk ook in Safari testen. Op een apparaat van Apple is de gewone versie altijd geïnstalleerd. Voor de Safari Technology Preview gaat u naar developer.apple.com/safari/download/.

HTML-editors

Om webpagina's te kunnen schrijven, hebt u geen bijzondere of dure software nodig. In theorie is elke tekstverwerker (*editor*) waarmee tekst als kale ASCII-tekst kan worden opgeslagen geschikt. Kladblok (voor Windows) of Tekst-Editor/SimpleText (voor Mac) voldoet al! Maar voor een comfortabeler leven als webdeveloper zijn er ook tal van gespecialiseerde programma's beschikbaar. Veel ervan zijn gratis.

Gespecialiseerde HTML-editors bieden allerlei voordelen: met codekleuring kunt u makkelijk de verschillende onderdelen zoals tags, attributen, eigenschappen en waarden herkennen, automatisch aanvullen van tags bespaart typewerk, spellingcontrole voor uw code scheelt debugtijd, automatisch inspringen maakt code beter leesbaar en zo is er nog veel meer.

Het aanbod van (gratis) editors is enorm en de keus is persoonlijk. Het beste advies is om er een paar uit te proberen. Verschillende websites bieden toplijsten aan van editors in allerlei soorten en smaken. Kijk voor een neutraal overzicht eens naar en.wikipedia.org/wiki/List_of_HTML_editors. Bekende namen zijn Visual Studio Code, Brackets (brackets.io), Atom (atom.io) en Sublime text (www.sublimetext.com). Visual Studio Code wordt hierna besproken.



Microsoft Office is geen codeertool

Schrijf geen HTML, CSS of JavaScript in Microsoft Word. Ook **Bestand, Opslaan als webpagina** is geen goed idee. Hoewel het resultaat in een moderne versie van Office niet zo rampzalig is als het ooit was, bevat de webpagina veel overbodige code. Wilt u de opmaak van een Word-document behouden, plaats dan een pdf-versie op uw website. Of maak er zelf een HTML-pagina van met CSS voor de opmaak.



CSS-editors

U hebt voor het werken met CSS geen ander programma nodig. Alle moderne editors helpen u bij het schrijven van CSS net zo goed als bij HTML, inclusief automatisch aanvullen van code, de juiste opties voor inspringen, het plaatsen van accolades en codekleuring.

Visual Studio Code

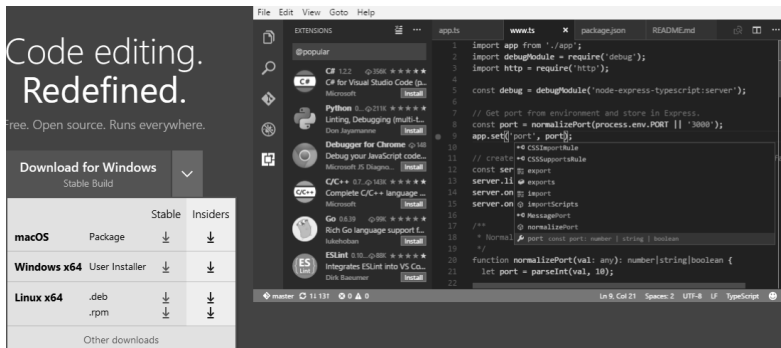
Het overweldigende aanbod van HTML-editors kan het maken een keus nogal bemoeilijken. Stapt u nu in, kijk dan vooral eens naar Visual Studio Code (VS Code), te vinden op code.visualstudio.com. De editor is gratis en beschikbaar voor Linux, macOS en Windows. Aardigheidje: de interface en alle functies zijn geschreven in HTML, CSS en JavaScript. In feite is het programma een web-toepassing die is omgezet naar een oorspronkelijk Linux-, macOS- of Windows-programma. Dat is overigens meer dan een aardigheidje. Die opbouw zorgt er namelijk voor dat de gebruiker de interface en functies eenvoudig kan aanpassen en dat het betrekkelijk eenvoudig is om er extensies voor te maken. Er zijn dan ook veel extensies beschikbaar.



Electron

Het is wel vroeg om erover te beginnen omdat u nog nauwelijks uit het startblok bent, maar u leest het goed: webtoepassingen kunnen worden omgezet naar programma's die op verschillende besturingssystemen werken. In dit geval is dat gedaan met Electron, een opensourcebibliotheek die wordt ontwikkeld door GitHub. Kijk als u er klaar voor bent eens op electronjs.org.

VS Code is een uitstekende editor voor HTML en CSS, maar ondersteunt veel meer talen, bijvoorbeeld JavaScript, TypeScript, PHP, Java, Python en C#. De basisondersteuning van HTML en CSS is prima, met codesuggesties (IntelliSense), automatisch sluiten van tags, informatie over tags bij aanwijzen, syntaxiskleuring, voorbeeldkleur, code samenvouwen en meer. Ook handig zijn de vele toetsencombinaties voor het bewerken van code: kopiëren van hele blokken, meerdere cursors tegelijk, alle keren dat een woord voorkomt ineens selecteren en meer. Op de website van VS Code staat een overzicht van toetsencombinaties. Met extensies kunt u nog veel meer handigheidjes toevoegen. De website is marketplace.visualstudio.com/VSCode, maar u kunt net zo makkelijk zoeken vanuit de interface van VS Code.



Afbeelding 1.9 Visual Studio Code, een veelzijdige editor voor HTML en CSS.



Liveserver

Een onmisbare extensie is Live Server van Ritwick Dey. Hiermee wordt op de achtergrond een lokale webserver gestart waarin elke verandering in de code zichtbaar is zodra het document is opgeslagen. En doordat het opslaan kan worden geautomatiseerd, bijvoorbeeld direct als de cursor het codevenster verlaat, werkt dit fantastisch. Zoiets als een liveserver lijkt misschien overdreven als u net begint. Maar zodra u met het schrijven van code aan de slag gaat, zult u merken hoeveel prettiger dat werkt.

Commerciële editors/IDE's

Naast de gratis editors worden er ook editors verkocht. Dergelijke producten vallen in de categorie geïntegreerde ontwikkelomgeving: *integrated development environment* of IDE. Ze bieden veel meer mogelijkheden dan u als beginnende ontwikkelaar nodig hebt en zelfs in de fase daarna kunt u met de genoemde gratis editors prima projecten met HTML, CSS en JavaScript uitvoeren. De meerwaarde van de commerciële producten zit vaak in de ondersteuning en nog betere en meer functies voor ontwikkeling en samenwerken. Voorbeelden zijn:

- WebStorm (www.jetbrains.com/webstorm)
- Dreamweaver (www.adobe.com/nl/products/dreamweaver.html)

Aanvullende hulpmiddelen

Naast browsers en een code-editor zijn er nog een paar hulpmiddelen waarmee u het schrijven en doorgronden van HTML- en CSS-code vereenvoudigt. In de voorgaande paragrafen zijn er al enkele genoemd, maar voor de duidelijkheid staan ze hier bij elkaar.

Hulpmiddelen voor validatie

Eén hulpmiddel is nog geheel niet genoemd: de validator, een hulpmiddel om te controleren of uw code aan de webstandaarden voldoet. Een validator controleert of de code voldoet aan de eisen van de taal, zoals complete tags en verplichte attributen. U kunt een validator vergelijken met de spellingcontrole van een tekstverwerker. U doet uw best om zo zorgvuldig mogelijk te coderen en de validator wijst u op mogelijk achtergebleven fouten.

Fouten in HTML geven lang niet altijd zichtbare problemen in de browser, maar in complexere webapplicaties kunnen ze problemen geven. Met een validator bent u zulke problemen voor. Tijdens het leren van HTML is de waarde van een validator dat die laat zien waar uw code beter kan.

Afbeelding 1.10 HTML-validatieservice van het W3C. Voor CSS is een vergelijkbare interface beschikbaar.

Op de website van het W3C zijn validators beschikbaar voor HTML en CSS, en u vindt er links naar andere validators:

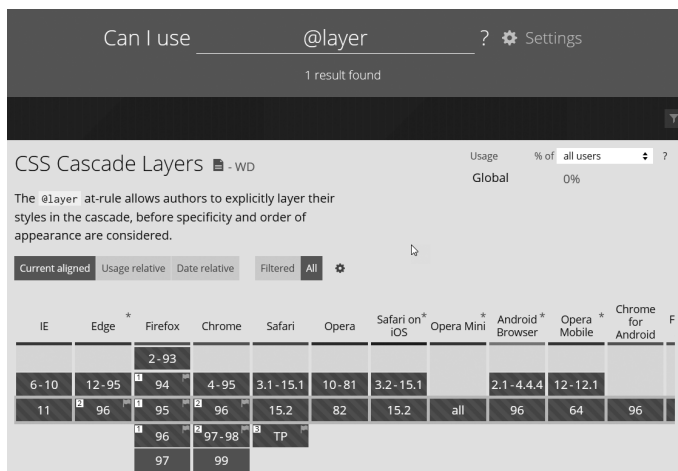
- De HTML-validator staat op validator.w3.org.
- De CSS-validator is te vinden op jigsaw.w3.org/css-validator/.

De werking van de validators wijst zich vanzelf: u kunt een webadres laten controleren, een bestand uploaden of een codefragment typen of plakken.

Hulpmiddelen voor ontwikkeling

Er is een enorme variatie aan hulpmiddelen voor het werken met HTML- en CSS-code. Enkele onmisbare hulpmiddelen zijn:

- Hulpmiddelen voor ontwikkelaars in de diverse webbrowsers, eenvoudig te vinden in het menu van de browser. Deze zijn in de loop van de tijd zo uitgebreid dat u hiermee al een goedgevulde gereedschapskist hebt.
- De website caniuse.com met ondersteuningstabellen voor webtechnologieën (HTML, CSS, HTML API's). Onmisbaar om brekende pagina's te voorkomen.
- De website developer.mozilla.org/en-US/, het Mozilla Developer Network (MDN). De referentiepagina's van HTML en CSS zijn erg handig om terug te zoeken 'hoe het ook alweer zat': had die CSS-eigenschap nou drie of vier waarden nodig, en wat was ook alweer de volgorde? Helaas bezuinigt Mozilla op deze site, maar vrijwilligers doen hun best om de informatie bij te houden.
- Een nieuwe bron is web.dev. Het is een initiatief van Google met bijdragen van professionals die eerder voor Mozilla schreven. Er zijn onder meer online lessen te vinden. De site is nu nog niet zo uitgebreid als MDN, maar geeft goede informatie.

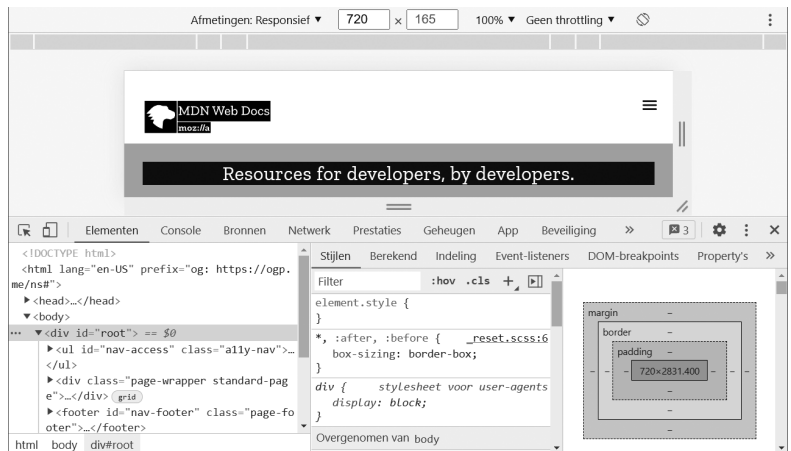


IE	Edge	Firefox	Chrome	Safari	Opera	Safari on iOS	Opera Mini	Android Browser	Opera Mobile	Chrome for Android
		2-93								
6-10	12-95	94	4-95	3.1-15.1	10-81	3.2-15.1		2.1-4.4.4	12-12.1	
11	96	95	96	15.2	82	15.2	all	96	64	96
		96	97-98	TP						
		97	99							

Afbeelding 1.11 Op caniuse.com ziet u direct welke browsers een eigenschap ondersteunen. Voor `@layer` was het nog niet zover, maar dat kan vandaag al anders zijn.

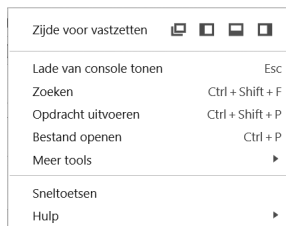
Met de hulpmiddelen voor ontwikkelaars kunt u bijvoorbeeld HTML- en CSS-code bekijken en live bewerken. Daarmee wordt experimenteren met code eenvoudiger en leuker. Van elke webpagina, dus niet alleen die u zelf maakt, maar van elke pagina op internet, kan de code worden aangepast om te bekijken welk effect een verandering heeft (uw aanpassingen verdwijnen zodra de pagina opnieuw wordt geladen). Vooral tijdens het leerproces is het onderzoeken van door professionals gebouwde pagina's een goede tijdbesteding. Ook de ervaren webontwikkelaar profiteert van deze hulpmiddelen, omdat ze foutopsporing vereenvoudigen.

Vanwege de grote overeenkomsten worden niet de afzonderlijke hulpmiddelen van de verschillende browsers besproken. Ter illustratie kijken we naar **Hulpprogramma's voor ontwikkelaars** van Chrome. Ze worden ook wel kortweg de Chrome DevTools genoemd. Ze zijn in het hoofdmenu van Chrome te vinden onder **Meer hulpprogramma's** en kunnen direct worden geactiveerd met de sneltoets Ctrl+Shift+I (Option+Command+I). Een andere manier om ze te activeren is rechtsklikken op een onderdeel van een webpagina; kies in het snelmenu **Inspecteren**. Op deze manier ziet u direct alle informatie over het gekozen element.

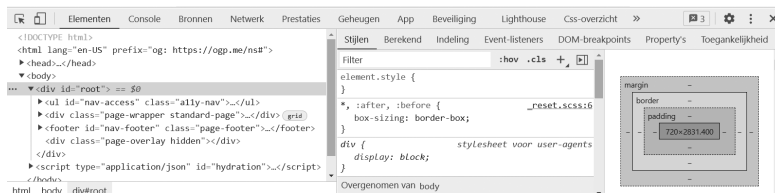


Afbeelding 1.12 Hulpprogramma's voor ontwikkelaars in Google Chrome.

Afbeelding 1.12 toont het venster van de DevTools. Het verschijnt standaard onder in het browservenster. Het kan ook links of rechts staan of als los venster worden getoond. De optie **Zijde voor vastzetten** vindt u in het menu (afbeelding 1.13). U ziet nu direct wat in het kader van dit boek de belangrijkste informatie is: het deelvenster links toont de HTML-elementen en rechts ziet u de CSS-regels die op het geselecteerde element van toepassing zijn. Ziet u heel andere dingen, kijk dan of in de werkbalk de tab **Elementen** actief is en in het rechterdeelvenster de tab **Stijlen** (afbeelding 1.14).



Afbeelding 1.13 Het menu van Chrome DevTools.



Afbeelding 1.14 Het tabblad Elementen toont de HTML-elementen en Stijlen toont de CSS-opmaakregels. Rechts staat het boxmodel, met van binnen naar buiten de content, opvulling, randdikte en marge.

In deze weergave kunt u de opbouw van elke pagina uitvlooien en zien hoe elk onderdeel aan zijn opmaak komt. Op dit moment zult u nog niet alles begrijpen; we staan immers nog aan het begin. Naarmate uw kennis van HTML en CSS groeit, zult u deze informatie steeds meer waarderen.

De andere tabs (**Console** t/m **CSS-overzicht**) bekijken we wanneer dat van toepassing is. Meer informatie is ook te vinden op developer.chrome.com/docs/devtools/.

Toch zijn nog twee functies ook op dit moment al de moeite van het onderzoeken waard. Links in de werkbalk (links van de tab **Elementen**) staat een knop met een pijlpictogram. Deze doet hetzelfde als de optie **Inspecteren** in het snelmenu. Rechts daarvan ziet u de knop **Apparaatwerkbalk schakelen**. Hiermee schakelt u een functie in die de weergave van uw pagina op andere (mobiele) apparaten simuleert, bijvoorbeeld een Pixel 2 of iPhone X, een iPad of een laptop met HiDPI-scherm.

- 1 Open een webpagina en activeer de DevTools (Ctrl+Shift+I of Option+Command+I).
- 2 Klik op de knop **Apparaatwerkbalk schakelen** (links van de tab **Elementen**). De weergave verandert en u ziet de webpagina alsof deze wordt weergegeven op het geselecteerde apparaat. Als de pagina er vreemd uitziet, drukt u op F5 (Ctrl+R/Command+R) om hem te vernieuwen.
- 3 In de keuzelijst links van de afmetingen kiest u een ander apparaat, bijvoorbeeld **Pixel 2**.

Merk op dat de pagina er anders uitziet dan op uw desktop. U ziet hem zoals u hem op een echte smartphone zou zien. De standaardweergave is portret. Met de knop **Draaien** draait u de weergave naar landschap.

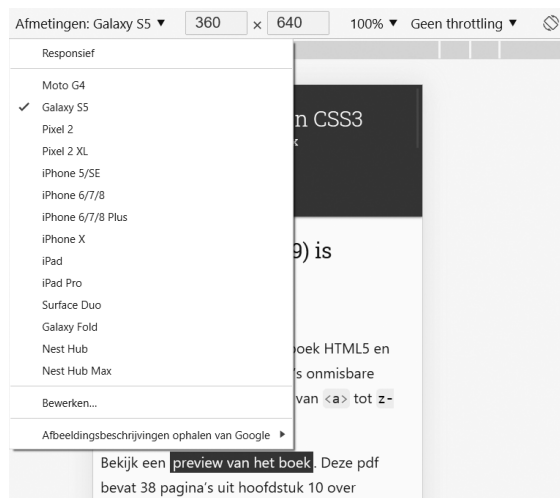
- De cursor is veranderd in een schaduwronde dat uw vingertop representeert. U kunt de pagina ermee verschuiven of op een koppeling ‘tikken’. Om een element te kunnen onderzoeken klikt u op de knop **Selecteer een element...** (het pijltje links van **Apparaatwerk balk schakelen**) en klikt u op een onderdeel in de pagina. De code van het element wordt geselecteerd in het codevenster.



Geen aanwijseffect op mobiel

In DevTools ziet u bij apparaten die een aanraakscherm simuleren geen aanwijseffecten, net als op uw eigen telefoon. Kiest u een gewone laptop dan ziet u ze wel. Dit is een realistische simulatie en iets om rekening mee te houden: laat de gebruikerservaring op aanraakschermen niet afhangen van aanwijseffecten met een muis.

Steeds vaker worden websites bekeken op mobiele apparaten. Het is daardoor noodzakelijk dat websites het ook op die apparaten goed doen (of zelfs in de eerste plaats op die apparaten) en dat maakt deze optie van de DevTools bijzonder nuttig. Ontwerpen voor mobiel vraagt van de ontwikkelaar een wat andere manier van werken en stelt aanvullende eisen aan de code, maar de basis blijft hetzelfde: HTML en CSS.



Afbeelding 1.15 Simulatie van mobiele weergave.