

Inhoud

1	Webtalen, browsers en editors	1
	Introductie	2
	Een korte terugblik	2
	Kennismaken met HTML en CSS	6
	Browsers	11
	HTML-editors	14
	Aanvullende hulpmiddelen	17
	Over dit boek	22
	Samenvatting	23
	Oefeningen	23
2	De basis van HTML	25
	Kenmerken van HTML-documenten	26
	Elementen, tags en tekst	26
	Attributen	33
	Kenmerken van moderne HTML	40
	De basis van een HTML-document	42
	Paginastructuur	45
	Document Object Model – HTML DOM	46
	Het contentmodel van HTML	47
	Secties markeren	49
	Samenvatting	60
	Oefeningen	61
3	Tekst markeren	63
	Inleiding	64
	Koppen in HTML-documenten	64
	Koppen markeren	66
	Alineatekst markeren	67
	Speciale betekenis aangeven	70

	Inhoud groeperen	75
	Speciale tekens in webpagina's	87
	Samenvatting	89
	Oefeningen	89
4	Koppelingen en interactieve elementen	91
	Inleiding	92
	Verbindingen leggen met <a>	93
	Klik hier voor richtlijnen	101
	Externe bestanden koppelen met <link>	104
	Knoppen en hyperlinks	109
	Disclosure widget: <details> en <summary>	109
	Dialogvenster	110
	Pop-over	113
	Samenvatting	113
	Oefeningen	114
5	Beeld, geluid en andere externe inhoud	115
	Inleiding	116
	URL's en structuur van de website	116
	Soorten afbeeldingen	118
	Bestandsformaten voor bitmaps	121
	De juiste afbeelding voor het scherm	124
	Afbeeldingen plaatsen met 	125
	Responsive images	129
	Externe HTML-inhoud: <iframe>	136
	Insluiten met <embed> of <object>	138
	Video in een webpagina	138
	Het element <video>	139
	Audio op de website gebruiken	144
	Samenvatting	145
	Oefeningen	145
6	Formulieren maken	147
	De rol van formulieren	148
	De basis van een formulier: <form>	150
	Labels bij invoervelden	155
	Het element <button>	156
	De tabvolgorde	157
	Het element <input>	157
	Soorten invoer: het attribuut type	165

Kiezen uit een lijst: <code><select></code> , <code><datalist></code> , <code><option></code>	174
Uitgebreide tekstinput: <code><textarea></code>	178
Groeperen met <code><fieldset></code> en <code><legend></code>	178
Voortgang tonen: <code><progress></code>	181
Schaal: <code><meter></code>	183
Resultaat: <code><output></code>	183
Samenvatting	184
Oefeningen	185
7 Tabellen maken	187
Inleiding	188
De structuur van een HTML-tabel	188
De basis: <code><table></code>	190
Rijen en kolommen: <code><tr></code> en <code><td></code>	190
Koptekst: <code><thead></code>	191
Een bijschrift: <code><caption></code>	192
Overige elementen	193
Voorbeelden van tabellen	195
Samenvatting	198
Oefeningen	198
8 De basis van CSS	199
Wat is CSS?	200
Waarom CSS zo handig is	202
De taal CSS	203
De plaats van de CSS-declaraties	205
Het CSS-bestand	207
De boomstructuur	208
Elementen benaderen met selectors	210
Basisselectors	211
Pseudo-elementen	219
Combinatieselectors: afstamming, kind en sibling	221
Hoe de browser CSS verwerkt: de cascade	222
Er is geen selector maar wel opmaak: overerving	227
Werken met cascade layers	229
De ingebouwde stijlen aanpassen	233
CSS-declaraties schrijven, maar hoe?	234
Waarden en eenheden	237
Samenvatting	245
Oefeningen	246

9	Lay-out: boxmodel en weergavemodel	249
	De opbouw van pagina's	250
	Het boxmodel	251
	Marges	255
	Opvulling	258
	Randen	259
	Overlopende inhoud: overflow	261
	Overloop van de scrol	263
	Weergavemodel	264
	De eigenschap display	265
	Inhoud (niet) weergeven: visibility	268
	Samenvatting	269
	Oefeningen	270
10	Lay-outs maken met CSS	271
	Inleiding	272
	Responsive design	272
	Media queries	273
	Container queries	277
	Weergave op mobiele schermen: de metatag viewport	282
	Lay-out: positionering	284
	Lay-out: multi-column	294
	Lay-out in een raster: gridlay-out	305
	Toepassing: nos.nl	327
	Lay-out: flexbox	338
	De toekomst: box alignment	355
	Samenvatting	356
	Oefeningen	358
11	Tekst en typografie	361
	Inleiding	362
	Lettertype: van systeem of online?	362
	Lettertypen downloaden: @font-face	364
	Eigenschappen voor lettertype: font	369
	Eigenschappen voor tekst	377
	Opmaak van lijsten	387
	Lijst als navigatie	388
	Samenvatting	393
	Oefeningen	393

12	Kleur, randen en achtergronden	395
	Inleiding	396
	Kleurwaarden	396
	Kleur van tekst	402
	De achtergrond	402
	Lichte en donkere modus	411
	Kleurverlopen	414
	Afgeronde hoeken	417
	Randafbeelding	419
	Schaduw	422
	Afbeeldingen uitsnijden met clip-path	425
	Beeldfilters	429
	Kleuren mengen	429
	Samenvatting	431
	Oefeningen	432
13	Overgangen en animaties, attr() en CSS-variabelen	433
	Beweging in webpagina's	434
	Overgangen (transitions)	434
	Animatie	437
	Transformatie	440
	CSS-variabelen: de functie var()	449
	Attributen uitlezen met attr()	452
	Samenvatting	453
	Oefeningen	453
	Index	455

Webtalen, browsers en editors

HTML is de belangrijkste taal op internet. De taal bestaat al sinds 1990 en er 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.

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 webbrowsers en door browserfabrikanten. In 2014 heeft de beheerder van de webtalen, het World Wide Web Consortium (W3C) de specificatie 'definitief' vastgesteld (in W3C-terminen 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-specificatie 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'.

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 `<blink>` 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 universiteiten, bedrijven uit sectoren als media, transport, uitgeverij, e-commerce, reclame en financiën. In relatie tot dit boek zijn de werkgroepen HTML en CSS belangrijk.



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).



Standards ▾ Groups ▾ Get involved ▾ Resources ▾ News & events ▾ About ▾ Q

Sign in

Making the web work

The World Wide Web Consortium (W3C) develops standards and guidelines to help everyone build a web based on the principles of accessibility, internationalization, privacy and security.

[Read more about W3C](#)



Afbeelding 1.1 In een oogopslag is de reden van bestaan van het W3C duidelijk.



Hyperlinks staan online

Op handboek-html-css.nl vindt u hyperlinks uit het boek, gerangschikt per hoofdstuk.

Het is allerm minst vanzelfsprekend dat er nu een actuele W3C-standaard van HTML 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.



Geen gesloten technieken

Dat gesloten mediatechnieken zoals Flash en Silverlight niet meer bestaan is volledig in lijn met wat Tim Berners-Lee, de bedenker van het World Wide Web, voor ogen had: een netwerk waarmee iedereen met open standaarden eenvoudig kan communiceren en informatie kan vinden en delen. Dat WhatsApp-berichten alleen met de eigen app zijn te versturen en te lezen, is dat duidelijk niet.

De oprichting van WHATWG viel bij de webgemeenschap 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.



Automatische updates

Er worden niet alleen sneller nieuwe versies van browsers uitgebracht, ze worden ook veel sneller op grote schaal gebruikt door het internet-publiek. 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 in elke browser goed werkt. (U mag alle websites vergeten, maar onthoud altijd deze twee: caniuse.com en developer.mozilla.org. U kunt daar vinden of optie X ook werkt in browser Y.)

Sinds 2019 werken W3C en WHATWG weer samen aan HTML. De actuele specificatie is te vinden op 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 1.2 *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-standaard zelf. Ze hebben wel een sterke relatie en het is bijna niet mogelijk om webapplicaties te bouwen zonder

er een of meer te gebruiken. Op www.w3.org/TR/ kunt u zoeken met de term 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 functies waarbij de ontwikkelaar niet 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 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 in een webapplicatie 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 webservern 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 pakket-beheerder 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 hyper-tekst'. 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 hoe u de bijbehorende tag gebruikt. 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 1.3 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, alinea teksten 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 1.3 opgemaakt tot afbeelding 1.4.



Afbeelding 1.4 *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 en verder). 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 webbrowser 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 60%, van Edge zo'n 18%, van Safari 11% en van Firefox 7%. Op tablet zit Chrome op zo'n 58% en Safari rond 35% en heeft Android 4%. Op mobiel is de verdeling: Chrome 53%, Safari 33% en Samsung Internet 9%. Het aandeel van de andere browsers is op mobiel verwaarloosbaar. Dit zijn cijfers van 2025 (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.

Google Chrome

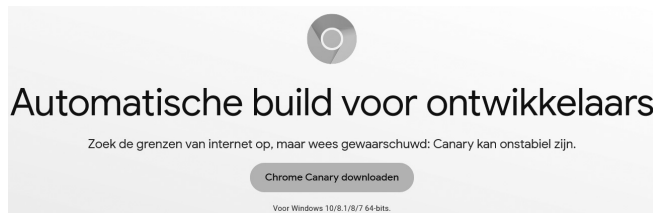
Het lijkt alsof hij er altijd al was, maar toch is Chrome, de browser van Google, pas eind 2008 gelanceerd. Tot die tijd was Internet Explorer de onbetwiste marktleider, al was het maar omdat de browser was vastgeklamd aan Windows. Chrome heeft op de desktop intussen de strijd met Internet Explorer/Edge, Safari 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 Edge) 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 1.5 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.

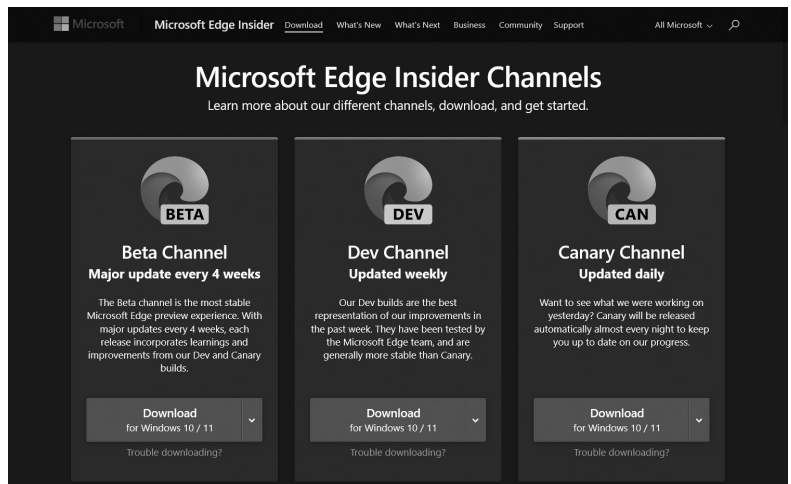
Microsoft Edge

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 1.6 Versies voor ontwikkelaars die vooruitlopen op de publieksversie van de browser. Deze pagina wordt alleen getoond als u dit adres met Edge bezoekt.

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 uitstekende 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. Voor-

jaar 2025 was de browser bij versie 136 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 besturings-systeem 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 is niet beschikbaar voor Windows of Linux. Het marktaandeel op de desktop is bescheiden, en ook op mobiel (iPhone) en tablet (iPad) wordt Safari steeds meer verdrongen door Chrome.

Safari blijft 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 TekstEditor/SimpleText (voor Mac) voldoet al! Maar voor een comfortabeler leven als web-developer 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.



Word is geen codeertool

Schrijf geen HTML, CSS of JavaScript in Microsoft Word. Ook **Bestand, Opslaan als webpagina** is geen goed idee. Gewoon niet doen.



CSS-editors

U hebt voor het werken met CSS geen ander programma nodig dan voor HTML. 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 webtoepassing 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.

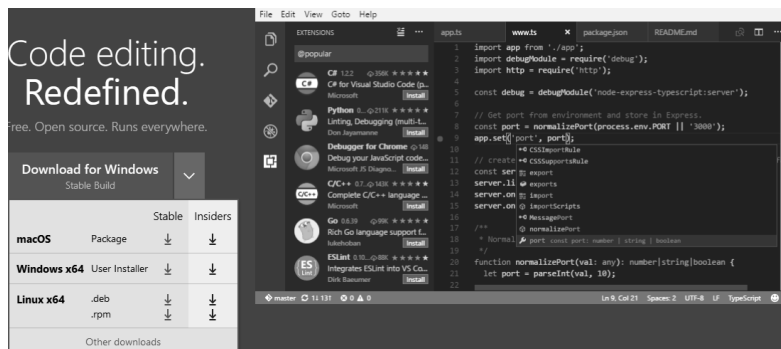


Electron

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, syntaxkleuring, 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 toetsen-

combinaties. 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 1.7 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 live-server 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 vaak direct uit de doos meer mogelijkheden dan gratis editors (maar met invoegtoepassingen komt u vaak net zo ver). 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)



Gratis Webstorm

Webstorm is gratis voor persoonlijk (niet-commercieel) gebruik. Daarmee haalt u een uitstekend alternatief voor Visual Studio Code in huis.