



Essentials

Auteur: Johan Cleuren

Lectoren: Johan Cleuren
Tim Dupont
Cedriek Vos
Reinaut Krekels
Carine Derkoningen

Nele Custers

Woord vooraf

Opleidingsonderdeel: Web Essentials			
Opleiding:	Prof. Bachelor Toegepaste informatica		
Onderwijstaal:	Nederlands	Kennisoverdracht:	14 u
Studiepunten:	6	Begeleide kennisverwerking:	42 u
Niveau:	Inleidend	Zelfstudie:	112 u
Lesgevers	Johan Cleuren, Nele Custers, Carine Derkoningen, Tim Dupont, Reinaut Krekels	Studiebelastinguren:	168u

Inhoud

Je leert te werken met de gangbare webtechnologieën. Als inleiding wordt kort de geschiedenis en opbouw van de HTML5-webstandaard behandeld, waarna de lay-out van een webpagina behandeld wordt. Hiervoor worden diverse methoden aangeleerd om met CSS-stijlbladen te werken. Die worden later toegepast op teksten, koppelingen, lijsten, afbeeldingen en tabellen. De CSS-technieken om een pagina in te delen worden uitvoerig behandeld. Ook het multimediale aspect van webpagina's wordt niet vergeten.

Je krijgt een ruime introductie tot standaarden om dynamische webpagina's te creëren, waaronder JavaScript, DOM met W3C Event Handling en PHP.

De einddoelstelling van de cursus is het kunnen coderen van browseronafhankelijke en gevalideerde webpagina's die gebruik maken van huidige webstandaarden. Om je de nodige inzichten te geven in de manier waarop dit alles werkt, zal je rechtstreeks met de code werken door middel van een teksteditor en de browser developer tools.

Leermaterialen

- Deze cursus: Cleuren Johan, Web Essentials, 2017
- <http://bb.pxl.be>: de course Web Essentials dient als ondersteuning van dit opleidingsonderdeel en informatie, updates, aanvullingen, getoonde presentaties, animaties en oefeningen.
- <http://www.pluralsight.com>, Pluralsight video tutorials
- <http://www.codeschool.com>, Code School tutorials en oefeningen
- De contacturen practicum, hou nota's bij.

Achtergrondinformatie

- Smashing HTML5, Bill Sanders, ISBN 978-0-470-97727-9, Wiley, 2011
- <http://www.w3c.com>, referentiesite voor webstandaarden, waaronder CSS.
- <https://whatwg.org>, referentiesite voor HTML 5 en DOM.
- <http://www.php.net/manual>, referentiesite voor PHP.

Evaluatiecriteria/leerdoelen

- De student kan een webpagina opbouwen in HyperText Markup Language (HTML5), bestaande uit semantisch correcte elementen.
- De student kan een webpagina opmaken met Cascading Style Sheets (CSS).

- De student kan een webpagina dynamisch maken met JavaScript en het Document Object Model (DOM).
- De student kan een opdracht zelfstandig opdelen in structuur, opmaak, client-side en server-side taken.
- De student kan een webpagina genereren met PHP: Hypertext Preprocessor (PHP).
- De student kan een HTTP GET- en POST-verzoek verwerken met PHP.
- De student kan scripttaalcode modulair opbouwen met functies.
- De student kan de developer tools van browsers gebruiken bij het ontwikkelen van een website.
- De student kan een webpagina volledig coderen volgens de specificaties van de opdrachtgever.
- De student kan zelfstandig documentatie vinden en verwerken om nieuwe technieken toe te passen in een webpagina.

Evaluatievormen

- Eerste examenkans:
 - Praktijk: Schriftelijk open boek examen met laptop (100%)
- Tweede examenkans:
 - Praktijk: Schriftelijk open boek examen met laptop (100%)

Werkvormen

- Opdrachtgestuurd onderwijs: Aan de hand van duidelijk afgebakende opdrachten/taken verwerven en verwerken studenten de leerinhouden. Het gaat hier niet noodzakelijk om realistische of praktijkopdrachten.

1 Inhoudsopgave

1	Inleiding HTML5	1
1.1	Markeertalen en webstandaarden	1
❖	XHTML	1
❖	HTML5	2
1.2	Basis van een webpagina	3
❖	Structuur	3
❖	Verklaring van de opbouw	4
1.3	Elementen, tags en attributen	5
❖	Elementen en hun tags	5
❖	Attributen	6
❖	Commentaar	6
1.4	Verschil tussen de verschillende HTML-versies	7
❖	Verschil tussen HTML 4.01 en XHTML	7
❖	Verschil tussen XHTML en HTML5	7
1.5	Validatie van de code	8
1.6	Browsers	10
❖	De vier browserengines	11
❖	HTML5-ondersteuning	11
2	Lay-out	13
2.1	Blokelementen	13
❖	Paragrafen	13
❖	Blockquote	13
❖	Preformatted	14
❖	Koppen	14
❖	Adres	14
❖	Divisies	15
2.2	De structuurelementen van HTML5	15
❖	Main	15
❖	Secties	15
❖	Artikel	16
❖	Hoofding	16
❖	Voetnoot	16
❖	Navigatieblok	16
❖	Randinformatie	17
2.3	Inline-elementen	18
❖	Line break	18
❖	Word break	18
❖	Tekstopmaaktags	18
❖	Logische elementen	19
❖	Span	20
2.4	Speciale tekens	20
3	Hyperlinks	21
3.1	Verbindingen leggen	21
❖	Opmaak van links	21
3.2	Absolute en relatieve koppelingen	21
3.3	Interne koppelingen	23
3.4	Andere soorten koppelingen	24
❖	Koppeling naar een FTP-site	24

❖	Koppeling naar een e-mailadres	24
❖	Bestanden downloaden	24
3.5	Div en span element	24
❖	Het span-element	24
❖	Het div-element	25
❖	De structuurelementen	25

2 Inleiding HTML5

3 Markeertalen en webstandaarden

Om websites te kunnen bezoeken moeten de computers die aangesloten zijn op het internet bepaalde protocollen hanteren. Deze protocollen zijn beschreven in een aantal standaarden, door o.a. het **World Wide Web Consortium** (W3C), de Internet Engineering Task Force (IETF), Ecma International en de Web Hypertext Application Technology Working Group (WHATWG).

Het protocol dat voor deze cursus belangrijk is het **HyperText Transfer Protocol** (http) dat platformafhankelijk werkt zodat het er niet toe doet welk besturingssysteem aan elke zijde van de communicatie gebruikt wordt.

Voor de ontwikkeling van websites voor het World Wide Web worden **markeertalen** gebruikt. Markeertalen zijn niets meer dan een set van regels die toelaten om de opmaak, het formaat en de structuur van tekst binnen een document te bepalen. De markeertaal **Standard Generalized Markup Language** (SGML) kan als voorloper voor HTML, XML, XHTML en HTML5 beschouwd worden. Bij de opkomst van het internet is een lichtere versie van SGML ontwikkeld, nl. HTML (**Hypertext Markup Language**). De HTML-standaard is telkens uitgebreid tot de huidige HTML-specificatie. Ten tijde van HTML 3.2 voegden vele browserontwikkelaars eigen elementen toe die echter niet langer compatibel zijn met andere browsers. Met de komst van HTML 4 wou het W3C een einde maken aan deze “browseroorlog” door één standaard te bepalen die compatibel is met alle browsers. Vanaf HTML 4 komt er een scheiding tussen de taal die de structuur bepaalt (HTML) en de taal die de presentatie bepaalt (CSS). In 1999 is de HTML 4.01 standaard beschreven met een aantal aanpassingen op de vorige standaard om zo een nog uniformere standaard te bekomen.

De beperkte grootte, het gelimiteerd aantal elementen en het gebruiksgemak zijn de sterke punten van HTML maar beperken eveneens de verdere ontwikkeling. Door de beperkingen van HTML is XHTML 1.0 in 2000 ontwikkeld als opvolger van HTML 4.01. XHTML (**Extensible Hypertext Markup Language**) is het resultaat van de hervorming van HTML 4 in XML 1.0 (**Extensible Markup Language**). XHTML behoudt dus de elementen van HTML 4.01 maar wordt geschreven als XML. Dit zorgt voor een meer rigide taal die “forward compatible” is.

In 2004 werd de WHATWG opgericht, een initiatief van Apple, Mozilla en Opera. Zij ontwikkelde een opvolger voor HTML 4. Gelijktijdig werkte het W3C verder aan de opvolger van XHTML (XHTML 2.0). In 2007 stopt het W3C met deze ontwikkeling en neemt de voorstellen over van de WHATWG in een eerste draft van de nieuwe markeertaal met de officiële benaming HTML5 (zonder spatie).

Hoe bepaalt een browser nu wat voor pagina's worden aangeboden in deze wirwar van standaarden? Om compatibiliteitsredenen zal een browser er vanuit gaan dat er “oude HTML” wordt aangeboden, zodat de voordelen van “nieuwe HTML” eigenlijk verloren gaan. Om dit te voorkomen is het beter de browser mee te delen welke standaard in het document gebruikt wordt, zodat er geen twijfel kan bestaan hoe alles geïnterpreteerd dient te worden.

◆ HTML5

HTML5 is sinds 28 oktober 2014 een W3C-aanbeveling. Een vernieuwde versie, HTML 5.1, is op 1 november 2016 gepromoveerd tot aanbeveling. Naar verwachting zal eind 2017 HTML 5.2 worden gefinaliseerd. Ondertussen houdt WHATWG de HTML-specificatie continu up-to-date, als een ‘levende standaard’. HTML5 heeft de laatste jaren dus nog wel wat wijzigingen ondergaan. Webpagina's kunnen dus binnen deze standaard worden ontwikkeld, maar het blijft raadzaam om de ondersteuning door de verschillende browsers te controleren.

Door de voorgeschiedenis van HTML5 (HTML 4 en XHTML 2) worden er twee verschillende syntaxen ondersteund: de losse HTML-syntax en de strikte XML-syntax. De HTML-syntax wordt veruit het meest gebruikt.

♦ **HTML-syntax**

De browser herkent HTML5 aan de hand van een document type (DOCTYPE) declaratie. Deze DOCTYPE komt helemaal bovenaan in het document. In vroegere HTML-versies bevatte zo'n DOCTYPE een verwijzing naar een document type definition (DTD), die o.a. aangeeft wat de toegelaten elementen zijn. In HTML5 is hiervan afgestapt, omdat browsers deze DTD in feite niet bekijken.

De overblijvende DOCTYPE is zeer eenvoudig:

```
<!DOCTYPE html>
```

Deze DOCTYPE is noodzakelijk! Anders veronderstellen browsers dat de code gemaakt is voor oudere browsers, waardoor ze diverse bugs van toen nabootsen (quirks-mode).

♦ **XML-syntax**

De browser herkent HTML met XML-syntax (XHTML) aan de hand van de Content-Type header. Deze header wordt door de server gebruikt om het type resource, het MIME-type, aan te geven. Bij bijv. een PNG-afbeelding is dit image/png. Voor XHTML is het application/xhtml+xml.

De volgende header moet dus verzonden worden door de server (niet in de code):

```
Content-Type: application/xhtml+xml
```

Deze header is noodzakelijk! Anders veronderstellen browsers dat je code met html-syntax doorstuurt.

Verder is het xmlns-attribuut van het openingselement verplicht voor alle XML-pagina's. De waarde van dit attribuut bepaalt de namespace waartoe de gebruikte elementnamen van het document behoren. Voor XHTML geeft dit:

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

De eerste regel van een XML-document bevat de optionele XML-proloog die beschrijft welke versie van XML in het document gebruikt wordt. Deze declaratie staat in een `<?xml ...?>` processing instruction en kan drie attributen bevatten: version, encoding en standalone.

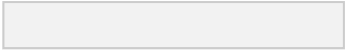
```
<?xml version="..." encoding="..." standalone="..."?>
```

- `version` = staat op "1.0" of "1.1" en geeft de gebruikte versie weer. Begin 2017 ondersteunen browsers versie 1.1 niet. Enkel versie 1.0 is dus beschikbaar.
- `encoding` = dit is optioneel en beschrijft de karakterset. Wanneer dit attribuut wordt weggelaten, geldt UTF-8 als karakterset.
- `standalone` = dit is optioneel en kan op "yes" of "no" (standaard) gezet worden. De standalone geeft aan of het document alle informatie zelf bevat of aangewezen is op externe DTD's voor zijn declaratie.

De DOCTYPE is hetzelfde als bij de HTML-syntax en is optioneel bij HTML5 in XML-syntax.

♦ **Syntax in deze cursus**

Bij de introductie van XHTML hekelden vele webontwikkelaars de strenge benadering terwijl andere de rigide correctheid toejuichten. In deze cursus wordt geopteerd voor de HTML-syntax,



aangevuld met code conventies gebaseerd op de XML-syntax. Hierdoor is de code consistent en goed leesbaar.

♦ **HTML-element**

Het openingselement <html> bevat verder het optionele, aangeraden lang-attribuut. Dit is o.a. belangrijk als de pagina door screenreaders gelezen wordt om de uitspraak te bepalen.

```
<html lang="nl">
```

♦ **Meta-informatie**

Om correct te kunnen valideren en ervoor te zorgen dat de browser weet hoe hij de inhoud moet interpreteren en weergeven, dient bij de HTML-syntax een meta-tag toegevoegd te worden die aangeeft welke karakterset gebruikt wordt, de charset.

```
<meta charset="utf-8">
```

Ontleding van bovenstaande regel:

- meta: de informatie is bedoeld voor de browser
- charset: geeft aan dat er tekst volgt die als HTML dient geïnterpreteerd te worden en die opgebouwd is met de utf-8 karakterencodering. De mogelijke karaktersets zijn ISO-8859 en UTF (Unicode).

De ISO-8859 standaarden zijn:

- ISO-8859-1 of Latin-1 = West-Europese talen (Frans, Spaans, Catalaans, Baskisch, Portugees, Italiaans, Albanees, Nederlands, Duits, Deens, Zweeds, Noors, Fins, Faroese, IJslands, Iers, Schots en Engels).
- ISO-8859-2 of Latin-2 = Centraal en Oost-Europese talen (Tsjechisch, Hongaars, Roemeens, Kroatisch, Slowaaks, Sloveens, Servisch)
- ISO-8859-3 of Latin-3 = Zuid-Europese talen (Esperanto, Maltees, ..)
- ISO-8859-4 of Latin-4 = Baltische staten (Estlands, Litouws, Lets, Groenlands, Laplands).
- ISO-8859-5 of Cyrillic = talen met cyrillisch schrift (Bulgaars, Macedonisch, Russisch).
- ISO-8859- of Arabic = Arabische talen
- ISO-8859-7 of Greek = Grieks
- ISO-8859-8 of Hebrew = Hebreeuws en Yiddisch
- ISO-8859-9 of Latin-5 = Turks
- ISO-8859-11 of Thai = Thaise taal
- ISO-8859-15 of Latin-9 = Update van Latin-1.

Daarnaast kan de gebruikte karakterset ook Unicode zijn, dit wordt aangegeven met UTF-8, UTF-16 of UTF-32. De 16-bit en 32-bit versies zijn enkel nodig wanneer veel Oosterse tekens dienen weergegeven te worden en een kleinere bestandsgrootte wenselijk is.

Vuistregel: gebruik utf-8 maar zorg dat je bestanden effectief geëncodeerd zijn met utf-8.

♦ **Afgekeurde elementen**

Heel wat elementen van de vroegere HTML-versies zijn afgekeurd bij de overgang naar XHTML. De meeste van die elementen blijven afgekeurd of sterk afgeraden in HTML5. Het is de bedoeling om inhoud en opmaak te scheiden. Het zijn dus de elementen en attributen die puur opmaak beschrijven en geen betekenis geven aan de inhoud die zijn verdwenen. Als je deze maatstaf gebruikt is het gemakkelijker te onthouden welke attributen "deprecated" zijn of niet.

Voor de volledigheid is in de bijlagen een lijst opgenomen van verschillende geldige en afgekeurde elementen en de HTML-versie waartoe ze behoren.

4 Basis van een webpagina

❖ Structuur

HTML-webpagina's hebben eenzelfde structuur waarbij ze opgedeeld zijn in een hoofd (head) en een lichaam (body). Het belangrijkste verschil in de structuur situeert zich in de HTML-hoofding.

- HTML5-pagina

UTF-8 code:

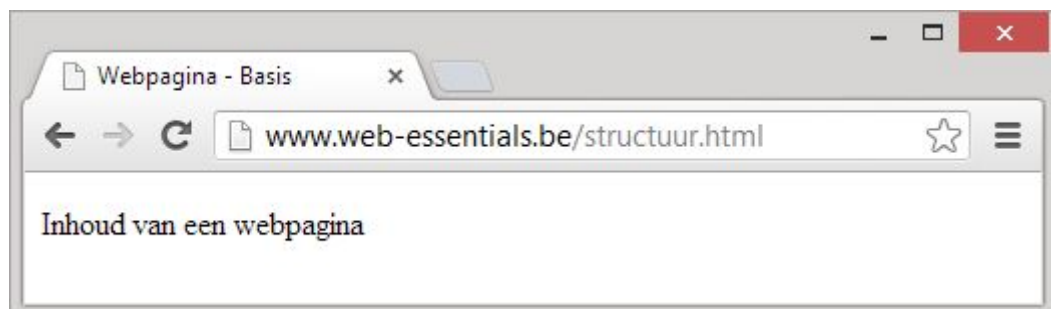
```
<!DOCTYPE html>
<html lang="nl">
<head>
  <meta charset="utf-8">
  <title>Webpagina - Basis</title>
</head>
<body>
  <p>Inhoud van een webpagina</p>
</body>
</html>
```

- HTML5-pagina (XML-syntax)

Header: Content-Type: application/xhtml+xml

UTF-8 code:

```
<html xmlns="http://www.w3.org/1999/xhtml" lang="nl">
<head>
  <title>Webpagina - Basis</title>
</head>
<body>
  <p>Inhoud van een webpagina</p>
</body>
</html>
```



❖ Verklaring van de opbouw

❖ **Het <html>-element**

De <html>...</html> container geeft aan dat het om een HTML-document gaat. De volledige inhoud van het webdocument valt tussen deze tags.

❖ **Het <head>-element**

De hoofding <head>...</head> bevat informatie die betrekking heeft op het document, maar die niet in het documentvenster wordt weergegeven. In de <head> kan je slechts een beperkt aantal elementen gebruiken:

- <title>...</title>: bevat de verplichte titel van een webpagina. Er is slechts één titel per webdocument mogelijk en die dient zo beknopt en verklarend mogelijk te zijn. Je mag ook geen opmaak of speciale tekens gebruiken.
- <base />: specificeert een absoluut basisadres (href-attribuut) en doel (target-attribuut).
- <link />: definieert de relatie van het document met andere documenten en wordt vooral gebruikt om naar externe stijlbestanden te verwijzen (zie CSS).
- <meta />: additionele informatie over het document. Dit bevat instructies voor browsers en zoekmachines en het inhoudstype.
- <style>...</style>: voor de definitie van stylesheets die binnen het document zelf worden toegepast (afgeraden, zie CSS).
- <script>...</script>: bevat scripts of legt een link met scripts (zie JavaScript).

❖ **Het <body>-element**

Al de verdere tekst, afbeeldingen en codes bevinden zich binnen de <body>...</body> container. Alle opmaak-attributen voor het <body>-element zijn afgekeurd en dien je te vervangen door stylesheets.

```
<!DOCTYPE html>
<html lang="nl">
<head>
  <meta charset="utf-8">
  <title>Webpagina - Basis</title>
</head>
<body>
  <p>In het body-element bevindt zich de eigenlijke inhoud.</p>
</body>
</html>
```



5 Elementen, tags en attributen

❖ Elementen en hun tags

HTML-**elementen** zijn de instructies die de opmaak van de tekst, de figuren, de tabellen, de hyperlinks, enzovoort bepalen. Een **tag** is strikt genomen een onderdeel van een HTML-element. Voor de meeste elementen is er een open- en een sluittag, dit noemt men **containertags**. Slechts enkele elementen zijn **open** of leeg en worden dus door slechts één tag aangegeven.

♦ **Containertag:**

Deze tags beïnvloeden de tekst die tussen de open- en de sluittag staat. De combinatie van de opentag, de inhoud en sluittag vormt het HTML-element. Een voorbeeld is `<body>...</body>`.

♦ **Lege tag:**

Deze tags gebieden de browser eenmalig actie te ondernemen om een bepaald effect te bekomen. Aangezien in XHTML alle elementen dienen afgesloten te worden met een sluittag worden de open HTML-elementen geschreven met een "/" op het einde. Binnen HTML5 is het niet langer noodzakelijk om de tag af te sluiten. Voorbeelden zijn `
` en ``.

❖ Attributen

Haast alle HTML-elementen kan je een aantal extra kenmerken meegeven door middel van attributen. Ze vergroten de functionaliteit van de elementen. Het attribuut staat steeds in de open tag. Indien meerdere attributen worden opgegeven speelt de volgorde geen rol.

Aan het correct gebruik van attributen zijn niet langer voorwaarden verbonden. In deze cursus hanteren we echter de volgende regels:

- Alle namen staan in kleine letters;
- De waarde die meegegeven wordt aan de attributen staat tussen dubbele aanhalingstekens;

```

<a href="page2.html">...</a>
<style type="text/css">...</style>
```

De HTML5-standaard heeft enkele kernattributen die je in haast elk element kan gebruiken.

kernattribuut	waarde
accesskey	snelkoppeling om een element te benaderen.
class	klasse waartoe het element behoort.
contenteditable	maakt de inhoud aanpasbaar door de gebruiker.
dir	de tekstrichting.
draggable	maakt het element versleepbaar.
dropzone	beschrijft de actie als een item in een element gedropt wordt.
hidden	maakt het element verborgen.
id	unieke identificatie die in het hele document geldt.
lang	bepaalt de taal van de elementinhoud.
spellcheck	bepaalt of de spellingscontrole moet toegepast worden.
style	bijhorende informatie over de opmaakstijl.
tabindex	bepaalt de tabvolgorde van het element.
title	titel van het element.

De elementen waarbij deze kernattributen niet toegepast kunnen worden, zijn <base>, <head>, <html>(uitgezonderd het lang-attribuut), <meta>, <param>, <script>, <style> en <title>.

◆ Commentaar

Tussen commentaartags kan informatie geplaatst worden die genegeerd wordt door de browser. Net zoals in programmeertalen biedt het plaatsen van commentaar samen met inspringen een verhoogde leesbaarheid. De commentaar kan enkel in de broncode bekeken worden.

```
<!-- Commentaar -->
```

6 Verschil tussen de verschillende HTML5-syntaxen

Ondanks de bijna volledige overlap van elementen en attributen in de HTML- en de XML-syntax zijn er toch belangrijke verschillen die hier nogmaals worden opgesomd.

De XML-syntax:

- In XML-documenten is optioneel een XML-hoofding aanwezig;
- XML heeft een strikter formaat; elke fout wordt getoond aan de eindgebruiker;
- De strikte XML-syntax is eenvoudig te verwerken door niet-HTML tools;
- Alle elementen en attributen moeten in kleine letters ingegeven worden. In tegenstelling tot HTML is XHTML hoofdlettergevoelig (bv. <table> en niet <Table> of <TABLE>);
- Alle elementen moeten een eindtag bezitten (bv.
,);
- Alle attribuutwaarden staan tussen apostrofs of aanhalingstekens. Je hebt de vrije keuze tussen dubbele of enkele, maar je mag ze niet onderling mengen bij een waarde (bv. width="50%" of width='50%', maar niet width='50'" of width="50%');
- Verkleinde attributen zijn niet langer toegelaten, maar moeten als attribuut="attribuutwaarde" geschreven worden (bv. selected="selected" i.p.v. selected);
- Alle inhoud moet binnen een tag opgenomen worden. Losse tekst of andere inline elementen rechtstreeks ingeven in de body is niet toegelaten, maar moet je in een paragraaf of een ander containerelement steken;
- Bij geneste elementen mag er geen overlapping zijn tussen de elementen (bv. <i>Cursief en vet</i> en niet <i>Cursief en vet</i>).

De HTML-syntax is veel flexibeler. De regels waar je je nog moet aan houden zijn:

- Plaatsing van de eenvoudige DOCTYPE-declaratie;
- Het strikte formaat van XML mag nog, maar hoeft niet;
- HTML5 is niet hoofdlettergevoelig (bv. <table> of <TABLE>);
- Open elementen mogen maar moeten geen eindtag bezitten (of);
- Het plaatsen van attributen tussen apostrofs of aanhalingstekens wordt optioneel, zolang er geen witruimte (spaties, ...) in voorkomt. (bv. width="50%" of width='50%' of width=50%);
- De verkleinde attributen zijn toegelaten (selected);
- Alle inhoud moet binnen een tag opgenomen worden;
- Het nesten van elementen blijft strikt.

Zoals eerder aangehaald, wordt in deze cursus geopteerd voor de HTML-syntax, aangevuld met code conventies gebaseerd op de XML-syntax. In de voorbeeldcodes en oefeningen worden altijd de volgende syntaxregels nagevolgd, ook al zijn de meesten niet noodzakelijk om geldige HTML5-code te schrijven.

- UTF-8 tekst;
- <DOCTYPE html> en <meta charset="utf-8"> aanwezig;
- Elementen en attributen altijd in kleine letters;
- Attributen worden tussen aanhalingstekens geplaatst;
- Eén indentatieniveau per open tag aan het begin van de regel, behalve <html>;
- Geen opmaak-elementen, tenzij specifiek voor het voorbeeld;

De volgende HTML-code moet dus aangepast worden om een geldig HTML5-document te krijgen.

```
<HTML>
<Head>
  <title>Web Technology</title>
</head>
<BODY>
<h1>Vak: <i>Web Technology</i></h1></i>
<img SRC=astridrm.gif Alt="Astrid Ramakers">
<p>De volgende onderdelen worden behandeld:
<UL>
<li>HTML5
<li>CSS
  <li>DOM
<li>JavaScript
  </UL>
</BODY>
</html>
```

De correcte HTML5-code is:

```
<!DOCTYPE html>
<html lang="nl">
<head>
  <meta charset="utf-8">
  <title>Web Technology</title>
</head>
<body>
  <h1>Vak: <span class="vak">Web Technology</span></h1>
  <p></p>
  <p>De volgende onderdelen worden behandeld:</p>
  <ul>
    <li>HTML5</li>
    <li>CSS</li>
    <li>DOM</li>
    <li>JavaScript</li>
  </ul>
</body>
</html>
```

Browsers zijn zeer vergevingsgezind. Hierdoor leveren inbreuken tegen het gebruik van kleine letters en het plaatsen van aanhalingstekens niet onmiddellijk problemen op bij de weergave in de browser. Bovenstaande codes geven eenzelfde resultaat in een browser, ook al zit de pagina vol fouten.

7 Validatie van de code

Ondanks deze grote vergevingsgezindheid van de meeste browsers, zorg je er toch best voor dat je webpagina's voldoen aan de standaard om zo het meest kans te hebben dat ze in alle browsers identiek worden weergegeven. Deze controle ten opzichte van de standaard heet **validatie**. Bij de validatie wordt een HTML5-document gecheckt ten opzichte van de vereisten in de specificatie.

Niet enkel zal er gecontroleerd worden of de elementen en attributen bestaan, maar eveneens zullen alle andere fouten aangeduid worden. De andere veel voorkomende fouten zijn:

- Containerelementen niet afsluiten;
- Foutief nesten van elementen;
- Het opnemen van losse tekst (niet in een container);
- Schrijffouten (bijv. scr i.p.v. src);
- Het vergeten afsluiten van apostroffen en aanhalingstekens;
- Het gebruiken van afgekeurde elementen en attributen;
- En nog vele andere

Om te valideren kan je gebruik maken van ingebouwde validators in een HTML-editor of van webbased validators. De bekendste hiervan is <http://validator.w3.org>. Bij deze validator van het World Wide Web Consortium (W3C) kan je zowel code als online en offline documenten controleren.

Het controleren van een onderstaande code geeft 7 fouten weer in het validatorvenster. Dit aantal fouten komt niet altijd overeen met het echte aantal aangezien het vergeten van een sluittag of een aanhalingsteken zich kan doorrekenen in de rest van het document en zo meer fouten genereert. Het is dus belangrijk om de fouten van het begin naar het einde van het document te verbeteren.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Validatiepagina</title>
</head>
<body>
  <h1>Vak: <span class="vak">Web Technology</h1></span>
  <hr width="30%">
  De volgende onderdelen worden behandeld:
  <ul>
    <li>HTML</li>
    <li>CSS</li>
    <li>JavaScript</li>
  <body>
</body>
</html>
```

Na correctie van de code krijg je wel een gevalideerd document.

```
<!DOCTYPE html>
<html>
<head>
```

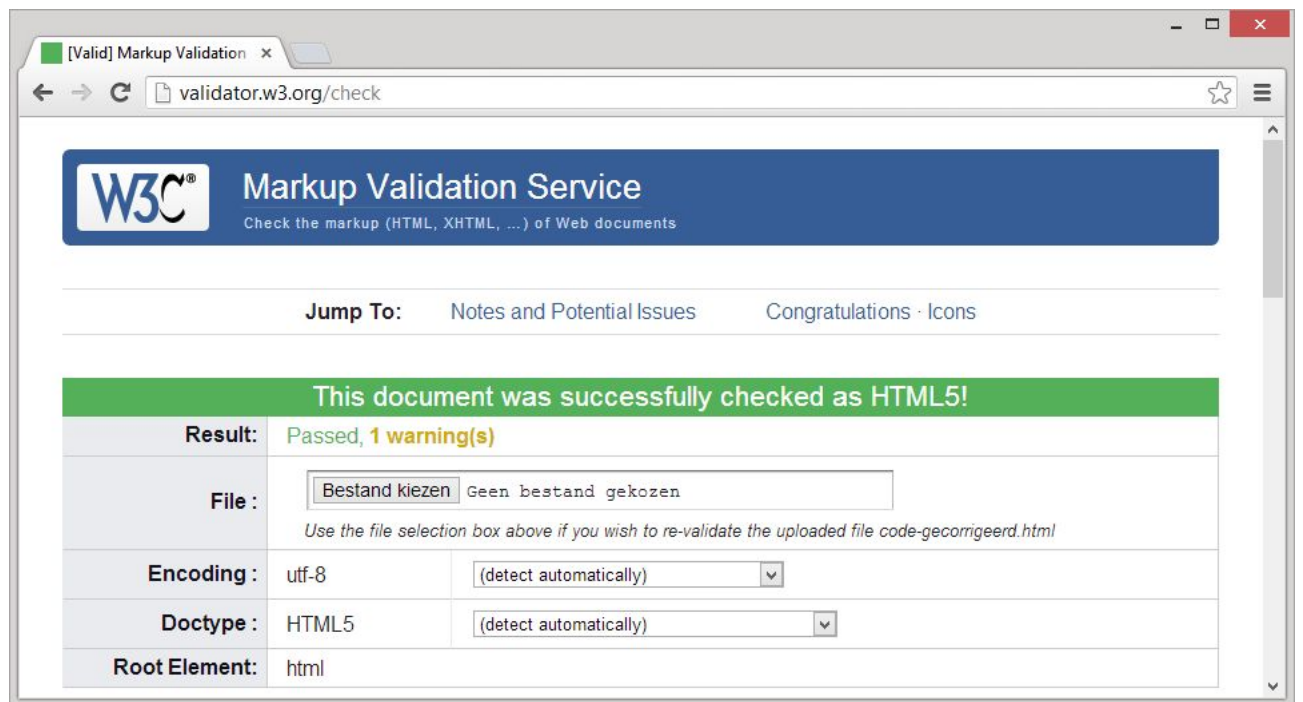
```

<meta charset="utf-8">
<title>Validatiepagina</title>
<link rel="stylesheet" href="style.css">
</head>
<body>
  <h1>Vak: <span class="vak">Web Technology</span></h1>
  <hr>
  <p>De volgende onderdelen worden behandeld:</p>
  <ul>
    <li>HTML5</li>
    <li>CSS</li>
    <li>JavaScript</li>
  </ul>
</body>
</html>

```

The screenshot shows the W3C Markup Validation Service interface. The browser tab is titled "[Invalid] Markup Validation". The address bar shows "validator.w3.org/check". The page header includes the W3C logo and the text "Markup Validation Service" and "Check the markup (HTML, XHTML, ...) of Web documents". Below the header, there are links for "Jump To: Notes and Potential Issues" and "Validation Output". A red banner indicates "Errors found while checking this document as HTML5!". Below this, a table shows the validation results:

Result:	7 Errors, 1 warning(s)
File :	<input type="text" value="Bestand kiezen"/> Geen bestand gekozen <small>Use the file selection box above if you wish to re-validate the uploaded file code-fout.html</small>
Encoding :	utf-8 <input type="button" value="(detect automatically)"/>
Doctype :	HTML5 <input type="button" value="(detect automatically)"/>
Root Element:	html



8 Browsers

Er zijn zeer veel browsers beschikbaar om HTML-pagina's mee te bekijken. Onderhuids kunnen ze opgedeeld worden in vier types van lay-outengine (Trident, EdgeHTML, Gecko en Blink/WebKit). Al deze engines en de op hun gebaseerde browsers hebben andere eigenschappen en verschillen ook in de ondersteuning van de verschillende webstandaarden.

❖ De vier browserengines

♦ *Trident*

Microsoft beheert de Trident-engine en gebruikt die enkel voor Windows-toepassingen. De meest gekende browsers en toepassingen zijn Internet Explorer (IE4-IE11) en Microsoft Outlook. Het is een gesloten standaard, dus met licentiekosten.

In 2002 had Microsoft bijna een monopolie met een marktverdeling van ongeveer 85%. De enige concurrenten waren toen AOL en Netscape. Momenteel dreigt IE onder de 10% te komen. De late ondersteuning van de verschillende webstandaarden en de vele pluginproblemen zouden daarvan de oorzaak kunnen zijn.

♦ *EdgeHTML*

De EdgeHTML engine is gemaakt door Microsoft voor de Edge browser, als antwoord op de problemen die Trident meesleurt uit het verleden. De engine ondersteunt bijv. oude programma's die voor een deel oude html-code willen tonen niet. Dit vermindert de complexiteit, waardoor Microsoft sneller nieuwe standaarden kan implementeren en de engine sneller, stabiel en veiliger is dan Trident. Internet Explorer en Trident worden dan ook niet verder ontwikkeld. Edge heeft momenteel een marktaandeel van ongeveer 5%.

♦ *Gecko*

De populaire op de Gecko-engine gebaseerde toepassingen zijn Firefox, Thunderbird en vroeger Netscape. Dit zijn allemaal open-source pakketten die beschikbaar zijn voor Windows, Mac OS X, en Linux/BSD. De ontwikkeling is in 1997 gestart met Netscape, maar is nu overgenomen door de Mozilla Corporation. Firefox heeft de laatste jaren veel marktaandeel verloren aan Chrome.

♦ *Blink/WebKit*

De open-source webkit-engine wordt gebruikt door de WebKit Foundation, Apple, Adobe, en het KDteam en is beschikbaar voor alle grote besturingssystemen, inclusief iOS voor de iPhone en iPod Touch. Blink is een fork van deze engine door Google, maar beiden blijven grotendeels dezelfde ondersteuning bieden. De bekende toepassingen zijn Safari, Chrome, Epiphany (Gnome) en Konqueror (KDE). Binnen Android is een Chrome-browser voorzien. De huidige marktaandelen van Safari en Chrome zijn respectievelijk rond de 20% en 45% in België. Sinds Google pc-bouwers betaalt om Chrome voorgeïnstalleerd mee te leveren bij nieuwe pc's, is het voor concurrenten veel moeilijker geworden om de gewone gebruiker te overtuigen hun browser(s) te gebruiken.

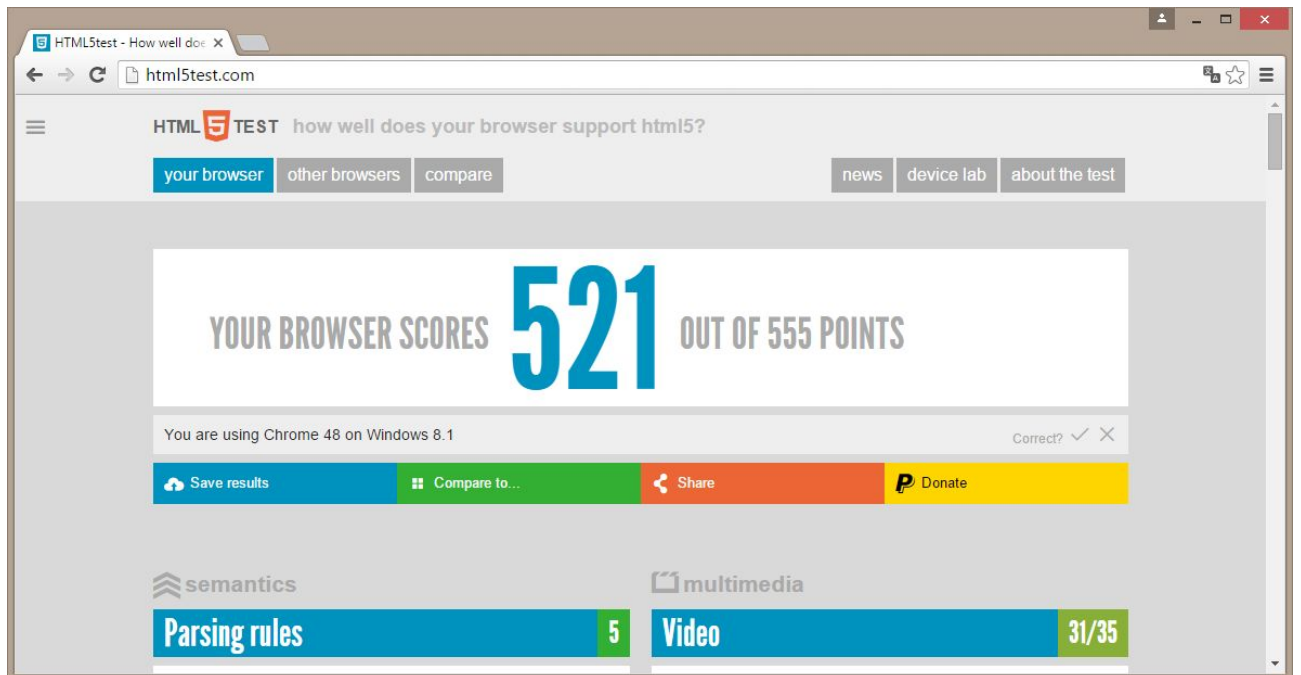
Alle browserontwikkelaars hebben de laatste jaren ook versies voor mobiele telefoons en tablets toegevoegd aan hun aanbod.

❖ HTML5-ondersteuning

De HTML5-standaard is pas sinds eind 2014 een aanbevolen W3C-standaard. Voor 2010 was er een tijdlang weinig vernieuwing qua webtechnologie, maar alle browsers zijn aan een heuse inhaalbeweging bezig. De versies van de browsers volgen elkaar steeds sneller op.

Vooraleer nieuwe HTML5-elementen te gebruiken in je webpagina, test je dus best de ondersteuning door de meest gebruikte browsers en meer bepaald in de versies die door jouw eindgebruikers het meest gebruikt worden. Bovendien gebruiken steeds meer mensen mobiele browsers, die soms andere zaken ondersteunen dan hun gelijknamige pc-aanverwanten.

Je kan een gedetailleerde score per browser opvragen om <http://www.html5test.com>. Hou er echter rekening mee dat sommige browsers bewust zaken niet ondersteunen, o.a. omwille van privacy (bijv. ping-attribuut van <a>). Onderstaande screenshot geeft de score voor Chrome versie 48.



Op de tab "other browsers" kan je de score opvragen van de browsers voor desktops, tablets en mobiele telefoons.

	Chrome	Firefox	Internet Explorer	Opera	Safari
Upcoming			Edge 13 453		
Current			Edge 402		9.0 400
Older	44 526	40 467	11 336	31 525	8.0 396
	42 523	37 449	10 297	29 519	7.0 352
	40 511	35 449	9 113	26 497	6.0 326
	36 486	28 416	8 33	12.10 338	5.1 250
	28 433	18 371			

Andere nuttige websites om de ondersteuning van HTML5 te controleren zijn:

- <http://www.html5doctor.com>
- <http://www.caniuse.com>

Op de volgende website kan je meer informatie vinden over een template waarbij de ondersteuning voor oudere browsers wordt opgevangen.

- <http://www.html5boilerplate.com>

9 Structuur

Alle gewone tekst, geplaatst binnen het <body>-element, wordt door de browser rechtstreeks op het scherm geplaatst, maar alle in een tekstverwerker toegevoegde tekstopmaak zoals returns, meervoudige spaties en tab's worden door de browser genegeerd. Tekst die als tabel opgemaakt is in een editor wordt dus volledig op een hoop gegooid. Om structuur en semantische betekenis aan een tekst mee te geven kan gebruik gemaakt worden van een aantal structuurelementen.

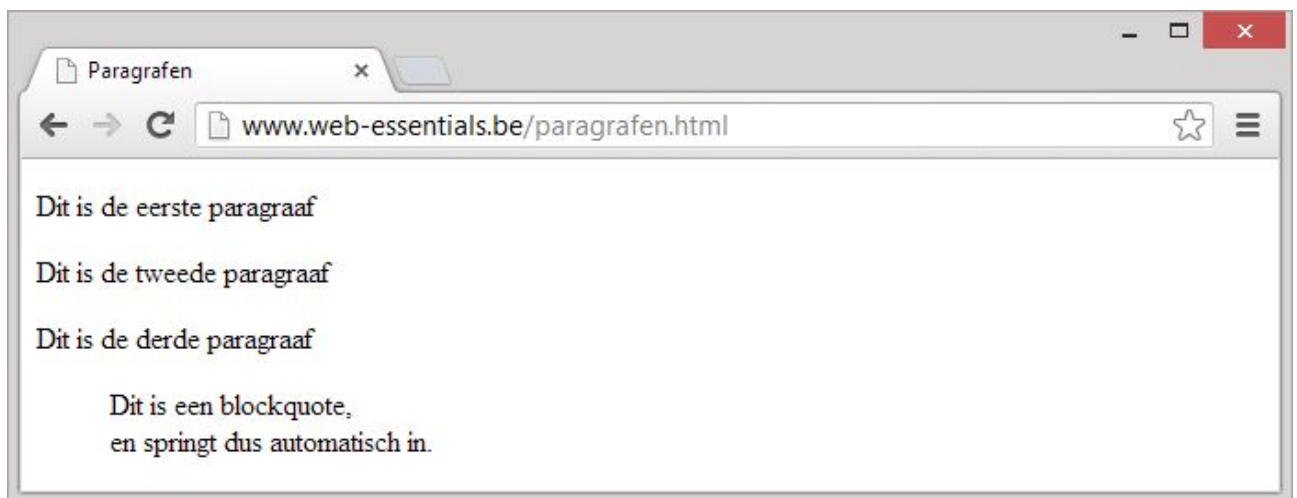
10 Blokelementen

Sommige HTML-elementen zijn **blokelementen** en worden altijd gescheiden van de andere elementen door de overgang naar een nieuwe regel. Blokelementen worden dus standaard onder elkaar getoond en niet naast elkaar. Dit geldt voor de paragraaf <p>, de <blockquote>, de koppen <h1> tot en met <h6>, de dividers <div>, de <pre>, alle elementen van lijsten (ul, ol, li, dl, dt, dd) en tabellen (table, tr, td, th), ... Met CSS kan daarna elk blokelement een opmaak meekrijgen die geldt voor de volledige inhoud, bijvoorbeeld de horizontale uitlijning (left, right, center en justify).

❖ Paragrafen

De paragraaf <p>...</p> dient om een alinea aan te geven. De browser voegt in de standaard lay-out een lege regel toe zodat afgescheiden alinea's of paragrafen bekomen worden. In auditieve browsers wordt een pauze gelaten tussen het voorlezen van twee paragrafen. In een paragraaf kan nooit een ander blokelement opgenomen worden (zoals koppen, andere paragrafen, enzovoort), maar mogen wel inline-elementen genest worden (zoals , ,).

```
<p>Dit is de eerste paragraaf</p>
<p>Dit is de tweede paragraaf</p>
<p>Dit is de derde paragraaf</p>
```



❖ Blockquote

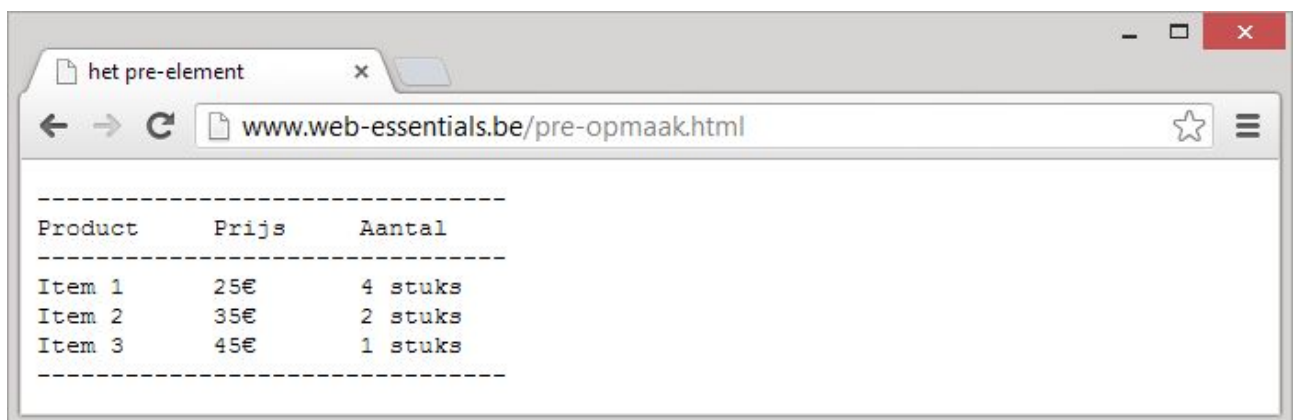
De blockquote <blockquote>...</blockquote> was oorspronkelijk bedoeld om te refereren naar beroemde uitspraken. Nu stelt het een citaat uit een andere bron voor. Kenmerkend is het automatisch inspringen en de mogelijkheid om andere blokelementen te kunnen nesten.

```
<blockquote>
  <p>Dit is een blockquote,<br>
    en springt dus automatisch in.</p>
</blockquote>
```

❖ Preformatted

Met `<pre>...</pre>` wordt een voorgevormde (preformatted) uitlijning overgenomen door de browser. De tekst wordt dus overgenomen, inclusief tab's en meervoudige spaties. Het nadeel van dit element is dat de betekenis die de tekststructuur geeft aan de inhoud zonder extra elementen niet kan begrepen worden door o.a. zoekmachines en niet-visuele browsers.

```
<pre>
-----
Product      Prijs      Aantal
-----
Item 1       25 EURO   4 stuks
Item 2       35 EURO   2 stuks
Item 3       45 EURO   1 stuks
-----
</pre>
```



❖ Koppen

Binnen HTML zijn zes koppen (headings) gedefinieerd. Koppen zijn blokelementen en nemen dus telkens een nieuwe alinea in. Startend met `<h1>...</h1>` tot en met `<h6>...</h6>` worden telkens kleinere koppen gegenereerd. De normale tekstgrootte situeert zich meestal tussen `<h3>` en `<h4>`.

```
<body>
<h1>Koptitel 1</h1>
<h2>Koptitel 2</h2>
<h3>Koptitel 3</h3>
<h4>Koptitel 4</h4>
<h5>Koptitel 5</h5>
<h6>Koptitel 6</h6>
<p>gewone tekst</p>
</body>
```

❖ Adres

De tag `<address>...</address>` is ook een blokelement en wordt gebruikt om contactinformatie te tonen die verband houdt met het dichtstbijzijnde bovenliggende `<article>`- of `<body>`-element. Het address-element dient niet gebruikt te worden voor postadressen, tenzij die adressen in feite de relevante contactinformatie zijn. Je vindt dit vaak terug samen met andere informatie in een footer-element.


```
<address>Ken Barbie <a href="mailto:ken.barbie@x.com">mail me</a></address>
```

◆ Divisies

De tag `<div>...</div>` heeft geen betekenis. Ze kan echter elementen bevatten en groepeert zo deze kinderelementen. Met dit element kunnen groepen worden gedefinieerd om op te maken met CSS, zonder ze een eigen betekenis te moeten geven. Dit element kan enkel gebruikt worden als de groep effectief niets betekent, anders is een ander element op zijn plaats.

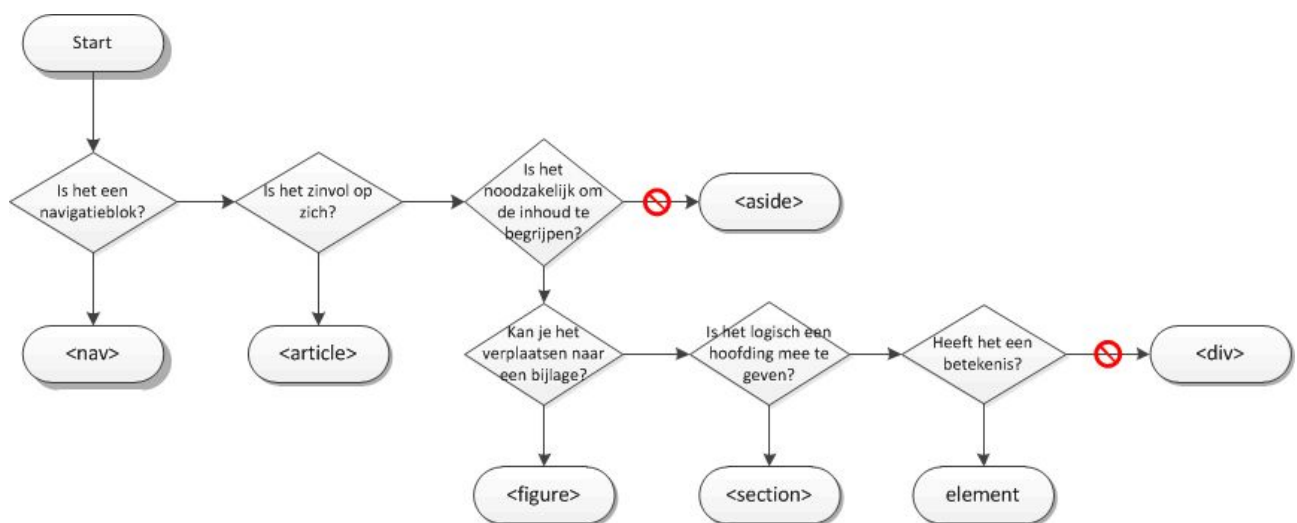
Dit element is onmisbaar in moderne webpagina's die uitvoerig gebruik maken van CSS. Deze divisies worden meestal samen gebruikt met stijldefinities om zo een lagenstructuur te bekomen (zie hoofdstuk over positioneren met CSS).

11 De structuurelementen van HTML5

In de voorlopers van HTML5 werd een nietsbetekenend blokelement toegevoegd, het `div`-element. Deze `div`'s zeggen echter niets over de inhoud waardoor het interpreteren van een pagina met veel `div`-elementen erg moeilijk is. In HTML5 zijn nieuwe structuurelementen toegevoegd om deze tekortkoming op te lossen. De zeven semantische blokelementen om de structuur van een webpagina te bepalen zijn `<section>`, `<main>`, `<article>`, `<header>`, `<footer>`, `<nav>` en `<aside>`.

Dit wil niet zeggen dat het `div`-element in onbruik is geraakt of fout is. Soms blijft het nuttig om bepaalde elementen te groeperen zonder daar een bepaalde betekenis aan te geven. In het geval dat de blok een structuur van de pagina beschrijft is het echter nodig gebruik te maken van de structuur elementen.

Het is niet altijd eenvoudig om te bepalen wanneer je moet kiezen voor welk structuurelement om een bepaalde inhoudsblok te definiëren. Vooral de keuze tussen `section`, `article` en `aside` is voor interpretatie vatbaar. Hiervoor bestaat geen perfecte regel, enkel een leidraad. Het is vooral belangrijk om consequent te zijn waar mogelijk en de voorbeelden in de standaarden te begrijpen. Een handige manier om consequent te blijven is de redenering volgen van de website HTML5 doctor (<http://html5doctor.com/downloads/h5d-sectioning-flowchart.pdf>). De onderstaande figuur is een vertaling van hun flowchart.



❖ Main

Het main-element kan gebruikt worden als groep voor de dominante inhoud van een ander element. Als het bovenliggende element veel kinderelementen telt, kan hiermee worden aangegeven wat de essentiële inhoud is. Tegenwoordig kan een pagina meerdere main-elementen bevatten, voor bijv. de belangrijkste inhoud van verschillende article-elementen.

❖ Secties

Het section-element wordt gebruikt om verschillende secties aan te duiden van een geheel. Een sectie is een onderverdeling die op zichzelf onvolledig is, bijv. een hoofdstuk in een boek of de eerste pagina van een blogpost met meerdere pagina's. Dit element is soms moeilijk exact te definiëren element en komt eigenlijk het dichtst tegen het div-element. Het verschil met een article is dat dat wel op zichzelf zou kunnen bestaan (zie Artikel). Het verschil met aside is dat een aside kan worden weggelaten, het is 'extra' (zie Aside). Een section kan andere sections en articles bevatten, die op hun beurt weer deelsections of -articles bevatten, ...

❖ Artikel

Het article-element geeft een groep elementen aan met samenhangende inhoud binnen de huidige groep. Een leidraad om te kiezen voor een artikel is de vraag of de inhoud op zichzelf gelezen of gedeeld kan worden.

Soms wordt aangeraden het article-element vooral te gebruiken voor inhoud die van eventueel van buitenaf zou kunnen worden ingevoegd. In geval van een krantenartikel, een forumpost, een nieuwsitem, enz. is dit dus een goede keuze.

```
<section class="agenda">
  <article class="eventement">...</article>
  <article class="eventement">...</article>
</section>
<section class="terugblik">
  <article class="eventement verslag">...</article>
  <article class="eventement verslag">
    <section class="fotoboek">...</section>
    <section class="interviews">
      <article>...</article>
      <article>
        <p>...</p>
        <section class="comments">...</section>
      </article>
    </section>
  </article>
</section>
```

❖ Hoofding

Het header-element is veel gemakkelijker betekenisvol te plaatsen. Het wordt gebruikt voor hulpmiddelen met een inleidend of verwijzend karakter (niet te verwarren met het <head>-element). Bij uitgebreide webpagina's zal het header-element dus meerdere malen aanwezig zijn. Het eerste kinderelement binnen de header is meestal een kopelement (h1, h2, h3, ...). Merk op dat header-elementen niet altijd bovenaan staan.

```
<header>
  <h1>Pagina titel</h1>
  <h2>Subtitel</h2>
```

</header>

❖ Voettekst

Het footer-element geeft voettekst aan, met meta- of extra detailinformatie. Ook dit element kan per pagina dus meerdere keren voorkomen. Binnen de footer van een webpagina wordt vaak de copyrightinformatie, links naar gerelateerde documenten en contactgegevens opgenomen. Bij artikels wordt in de footer-tag vaak de auteur, de aanmaakdatum, enzovoort opgenomen.

```
<article>
  <h2>Artikel</h2>
  <footer>
    Auteur: <address><a href="mailto:naam@mail.yy">Naam</a></address>
  </footer>
  <p>De eigenlijke inhoud.</p>
</article>
```

❖ Navigatieblok

De <nav>..</nav>-tags omvatten het hoofdmenu met navigatielinks van de webpagina, maar kan verderop ook gebruikt worden voor submenu's, inhoudstafels, ... Het nav-element kan dus eveneens meerdere keren op een webpagina voorkomen.

```
<nav>
  <ul>
    <li><a href="#">Hyperlink 1</a></li>
    <li><a href="#">Hyperlink 2</a></li>
    <li><a href="#">Hyperlink 3</a></li>
  </ul>
</nav>
```

❖ Randinformatie

Het aside-element kan gebruikt worden om een paginafragment aan te duiden dat zich in de marge van de hoofdpagina bevindt. Het betreft dan meestal een extra kolom met bijkomende informatie die niet rechtstreeks gerelateerd is aan de eigenlijke inhoud. Een voorbeeld is een banner, reclame, een figuur, een wist-je-datje, enzovoort.

```
<aside>
  <h2>Ook dat nog...</h2>
  <p>Wist je dat aside niet hetzelfde is als offside?</p>
</aside>
```

Een mogelijke indeling van een webpagina met bovenstaande semantische elementen kan er dus als volgt uitzien. Dit voorbeeld is slechts één van de vele mogelijkheden met de nieuwe elementen.

```
<!DOCTYPE html>
<html lang="nl">
<head>
  <meta charset="utf-8">
  <title>Structuurelementen</title>
</head>

<body>
  <header>
    <h1>Paginatitel</h1>
    <p>Tekst</p>
  </header>
```

```
<nav>
  <ul>
    <li><a href="link1.html">Link 1</a></li>
    <li><a href="link2.html">Link 2</a></li>
    <li><a href="link3.html">Link 3</a></li>
  </ul>
</nav>

<section>
  <header>
    <h2>Titel</h2>
  </header>
  <article>
    <header>
      <h3>Artikeltitel</h3>
    </header>
    <p>Artikelinhoud 1</p>
    <footer><p>Artikelvoetnoot<p></footer>
  </article>
  <article>
    <p>Artikelinhoud 2</p>
  </article>
  <article>
    <p>Artikelinhoud 3</p>
  </article>
</section>

<aside>
  <p>Bijkomende inhoud</p>
</aside>

<footer>
  <p>Paginavoetnoot</p>
</footer>
</body>
</html>
```

12 Inline-elementen

Zoals de naam het al aangeeft worden **inline-elementen** allemaal binnen dezelfde regel geplaatst. Ze zijn niet gescheiden door een overgang naar een nieuwe regel, maar staan naast elkaar.

❖ Line break

Indien je inline elementen te midden van een paragraaf toch wil scheiden, kan je dit doen door een line break te plaatsen. De line break dient om doelbewust een regeleinde te creëren zonder de bij de <p>-tag horende blanco regel. Dit element kan geen inhoud bevatten, dus in de HTML-syntax is de sluittag niet nodig.

```
<br>
```

❖ Word break

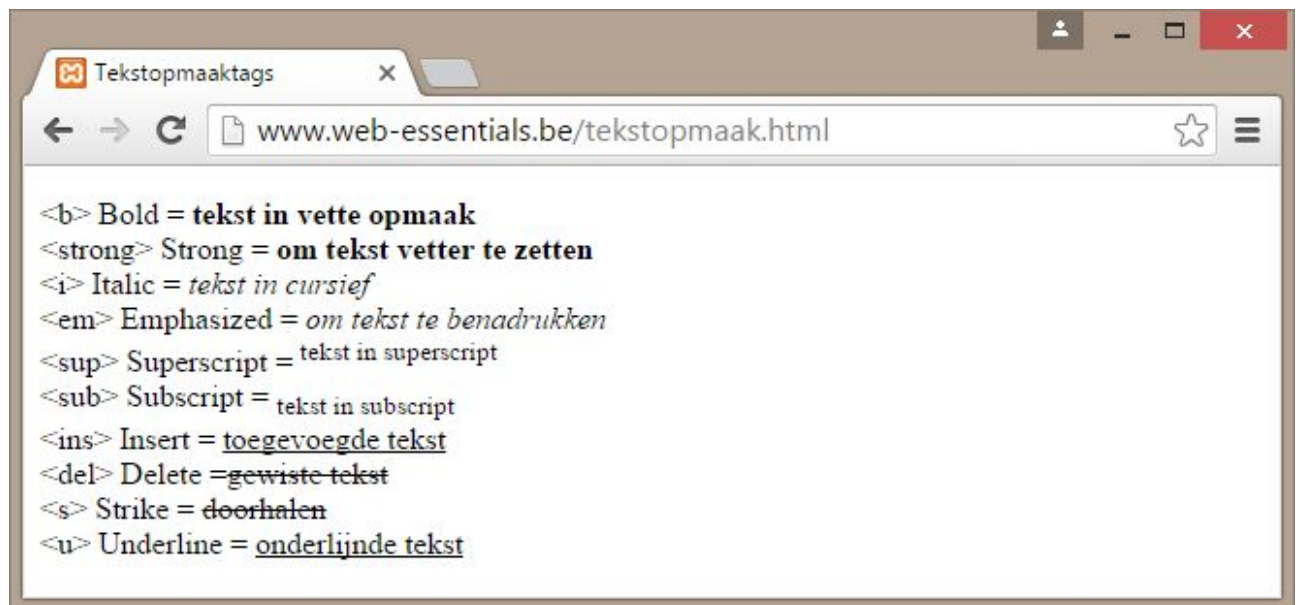
Met het wbr-element (word break) kan je mogelijke plaatsen aangeven om een woord af te breken indien het woord te lang wordt om op de regel te passen. Dit geeft echter geen koppeltekens. Gebruik daarvoor het bijhorende Unicode teken U+00AD 'soft hyphen'.

```
<p>Als je hippopoto<wbr />monstro<wbr />sesquipedalio<wbr />fobie hebt,  
dan heb je angst voor lange woorden.</p>
```

❖ Teksttags

Alle teksttags zijn inline-elementen en geven de auteur zeggenschap over de betekenis van die tekst. Deze betekenis kan soms erg subtiel zijn. De tags hebben ook een standaardopmaak, te zien in de onderstaande figuur.

...	Belangrijk, urgent
...	Beklemtoond
^{...}	Superscript
_{...}	Subscript
<ins>...</ins>	Ingevoegd
...	Gewist
<s>...</s>	Niet langer relevant
...	Uitgelicht, maar zonder extra betekenis
<i>...</i>	Andere stemming
<u>...</u>	Met een onuitgesproken niet-tekstuele annotatie



Het ``- en ``-element resulteren in dezelfde vette opmaak. Het geniet echter altijd de voorkeur om betekenis te geven aan de inhoud, dus indien je iets belangrijk wil maken is `` de betere keuze. Hetzelfde geldt voor `<i>` en `` die beiden een cursieve opmaak geven, maar waar de tekst een andere betekenis krijgt door de klemtoon op een deel te leggen, gebruik je het ``-element. Waar de tekst echter een andere stemming weergeeft (bijv. een droom) dan de omgevende tekst, is het `<i>`-element gepast.

De `<big>` en `<u>`-elementen zijn afgekeurd in HTML5. Het is trouwens af te raden om onderlijnde tekst te gebruiken aangezien dit verwarring geeft met hyperlinks die standaard onderlijnd zijn. Als je louter een visueel effect wilt bekomen gebruik je het element met de meest passende betekenis of het ``-element samen met de bijhorende CSS-code.

Om revisies aan te geven kunnen de elementen `<ins>` en `` dienen. Gewiste tekst wordt aangeduid met `` (deleted) en toegevoegde tekst met `<ins>` (inserted). Enkel als je tekst wil aangeven die niet langer geldig of relevant is, is het `<s>`-element aangewezen.

◆ Logische elementen

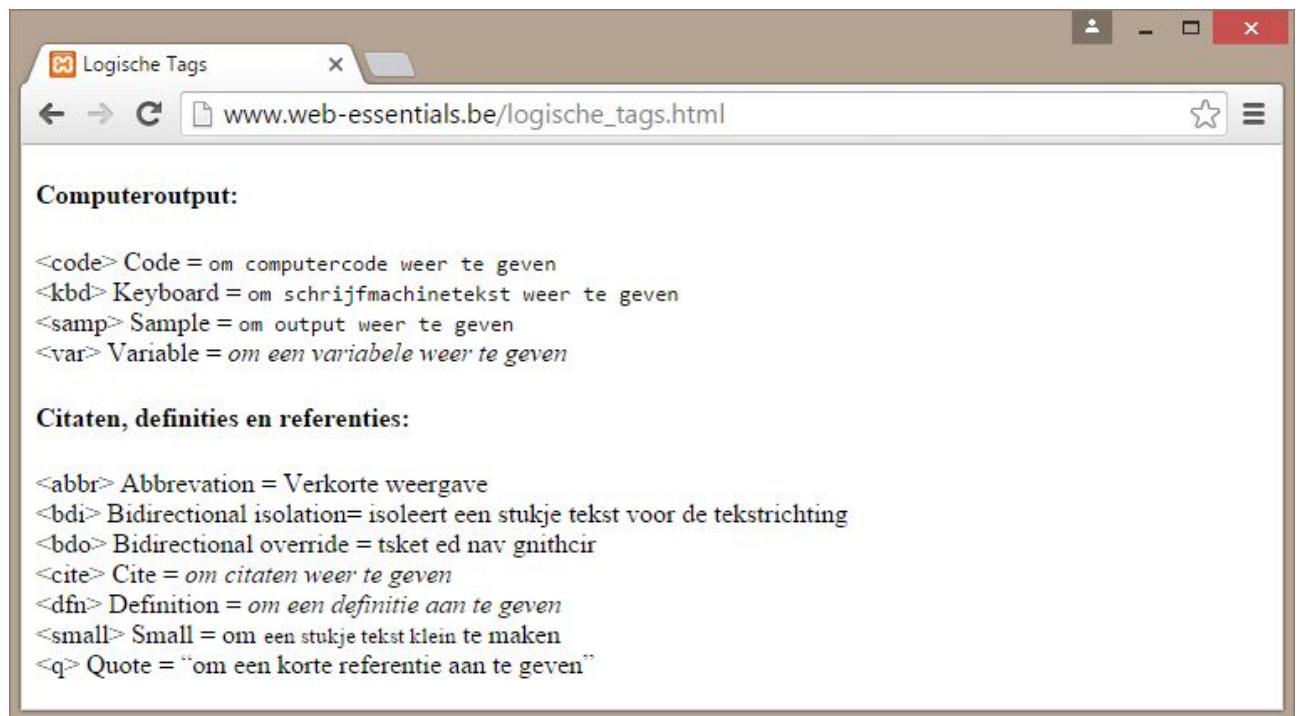
Met deze elementen wordt een logische betekenis gegeven aan flarden tekst. De verdere uitwerking ervan hangt af van de gebruikte browser.

De betekenis kan computeruitvoer gebaseerd zijn:

<code><code>...</code></code>	om computercode weer te geven
<code><kbd>...</kbd></code>	om gebruikersinvoer weer te geven
<code><samp>...</samp></code>	om output van een programma weer te geven
<code><var>...</var></code>	om variabelen weer te geven

Of kan verwijzen naar citaten, definities en referenties:

<code><abbr>...</abbr></code>	verkorte weergave (afkorting, acroniem, ...)
<code><bdi>...</bdi></code>	isoleert een stukje tekst voor de tekstrichting (bidirectional isolation)
<code><bdo>...</bdo></code>	richting van de tekst (bidirectional override)
<code><cite>...</cite></code>	voor citeren van de titel van een werk (lied, blogpost, programma, ...)
<code><dfn>...</dfn></code>	voor definities.
<code><small>...</small></code>	Korte nevenbemerkingen
<code><q>...</q></code>	Quotes, te midden van andere tekst



Zoals in bovenstaande figuur blijkt resulteren de elementen `<code>...</code>`, `<kbd>...</kbd>` en `<samp>...</samp>` in een lettertype met vaste letterbreedte (niet-proportioneel).

❖ Span

Indien je een stukje tekst wil opmaken met CSS zonder specifieke betekenis en zonder dat er al enige opmaak wordt meegegeven, kan je dit doen door gebruik te maken van het span-element.

`<p>` Met span kan je een bepaald ``stukje tekst`` een andere opmaak meegeven `</p>`

13 Speciale tekens

Voor het gebruik van speciale symbolen zijn er codes ontworpen, de **entiteitsnamen**. Een entiteitsnaam begint met een ampersand, bevat een afgekorte benaming en eindigt op een puntkomma. In plaats van de naam kan ook de decimale of hexadecimale ASCII- of Unicodenummer gebruikt worden.

<	<	<	<	kleiner dan
>	>	>	>	groter dan
&	&	&	&	ampersand
een spatie	 	 	 	non breaking space
"	"	"	"	dubbel aanhalingsteken

Er zijn veel meer entiteiten, een volledige lijst van de mogelijkheden kan je terugvinden op <https://html.spec.whatwg.org/multipage/syntax.html#named-character-references>.

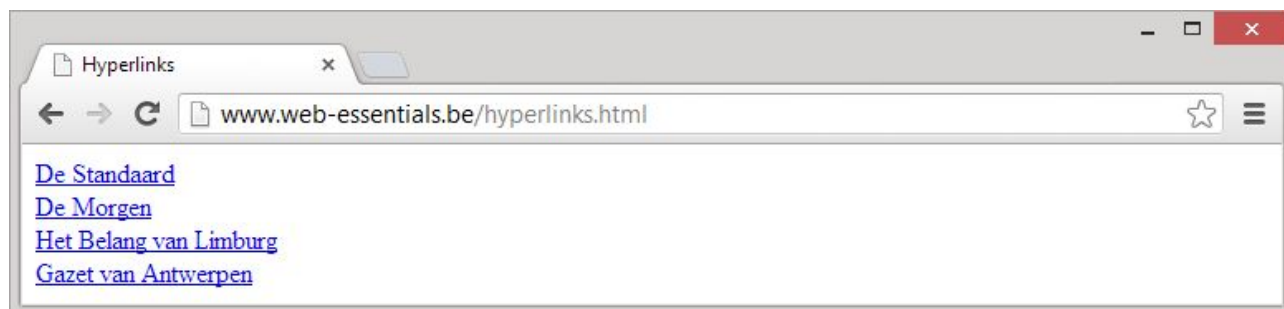
Voor andere karakters dan diegenen in de tabel hierboven is het makkelijker om de html-code met karakterset UTF-8 te maken en de karakters gewoon letterlijk in te geven, eventueel met hulp van een virtueel toetsenbord, kopiëren-plakken of <http://www.unicode.org/charts/charindex.html>

14 Hyperlinks

15 Verbindingen leggen

Om een webpagina te laten verwijzen naar een ander hypertext item gebruik je het **anchor** element `<a>...`. Het meest gebruikte attribuut van `<a>` is `href`, de **hypertext reference**.

```
<a href="http://www.destandaard.be">De Standaard</a><br>
<a href="http://www.demorgen.be">De Morgen</a><br>
<a href="http://www.hbvl.be">Het Belang van Limburg</a><br>
<a href="http://www.gva.be">Gazet van Antwerpen</a><br>
```



De andere attributen van `<a>` kunnen alleen samen met het `href`-attribuut aanwezig zijn.

HTML	waarde
<code>href</code>	De hyperlink die geopend wordt.
<code>hreflang</code>	De taal van de hyperlinkreferentie.
<code>media</code>	Geeft het medium/apparaat aan (vb. screen, pda, ...).
<code>rel</code>	De relatie tussen het document en de doel-url: alternate author bookmark help license next nofollow norereferrer prefetch prev search tag.
<code>target</code>	Doel waar de link geopend wordt: _blank _parent _self _top.
<code>type</code>	MIME-type van de doel-url.

❖ Opmaak van links

De opmaak van hyperlinks dient met pseudoklassen gedefinieerd te worden in CSS. Standaard zijn links altijd onderlijnd. De kleur geeft de status aan, niet-bezochte links zijn blauw, bezochte links paars en actieve links rood.

```
a {text-decoration:none;}
a:link {color:#0000ff;}
a:visited {color:#000099;}
a:hover {color:#ffffff; background-color:#0000ff;}
a:active {color:#000099;}
```

16 Absolute en relatieve koppelingen

Een **absolute koppeling** verwijst naar een bepaald document op een webserver en gebruikt hiervoor een URL-adres. Als van het bestand de naam wordt gewijzigd, als het naar een andere

directory of op een andere server wordt geplaatst is de betovering verbroken en volgt er bij het aanklikken een foutmelding.

```
<a href="http://www.voorbeeld.be/index.html">...</a>
```

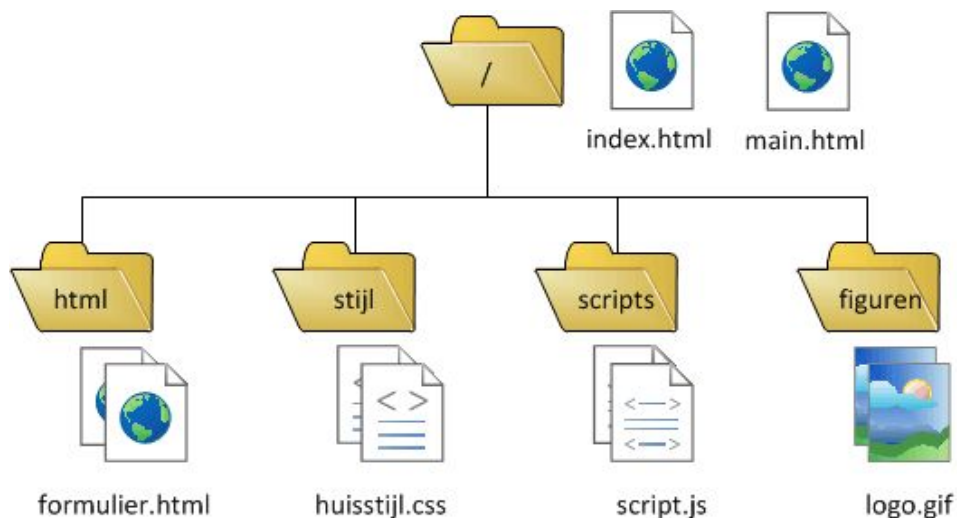
Een **documentrelatieve koppeling** verwijst naar een ander bestand, maar doet dit relatief ten opzichte van het bestand van waaruit de link wordt aangeklikt. Het voordeel hiervan is dat de directories in hun geheel, met al de onderliggende directories naar een andere computer overgebracht kunnen worden waarbij de structuur en de koppeling behouden blijft.

Indien vanuit het bestand index.html dient gelinkt te worden naar het bestand main.html dat zich in dezelfde directory bevindt, dan geeft dit volgende syntax.

```
<a href="main.html">...</a>
```

Indien vanuit het bestand index.html dient gelinkt te worden naar formulier.html dat zich in de directory html bevindt, dan dien je de foldernaam mee op te nemen. Let erop dat voorwaartse slashes gebruikt worden om foldernamen aan te geven. Moeten er meerdere niveaus gedaald worden, dan worden alle foldernamen gescheiden door een slash (dus **geen backslash**).

```
<a href="html/formulier.html"> ...</a>
```



Indien er in de folderhiërarchie dient gestegen te worden dan maak je gebruik van ../. Zo kan je van formulier.html linken naar index.html met de volgende syntax.

```
<a href="../../index.html">...</a>
```

Het benaderen van de figuur logo.gif in het huisstijl.css bestand in de stijlmap ziet er dus zo uit.

```
div {
    background-image: url(../figuren/logo.gif);
}
```

Een andere methode om relatieve links te leggen is met een **site-rootrelatieve koppeling**. Hierbij wordt de rootdirectory van de site aangeduid door een slash, en vertrek je altijd van deze locatie.

Enkele voorbeelden van site-rootrelatieve koppelingen:

```
<a href="/html/formulier.html">...</a>

<a href="/index.html">...</a>
<link rel="stylesheet" href="/stijl/huisstijl.css">
```

De site-rootrelatieve links werken enkel goed indien de bestanden reeds op een server geplaatst zijn en kunnen dus slecht getest worden wanneer ze op de lokale harde schijf staan.

17 Interne koppelingen

Een **interne koppeling** wordt gebruikt om een sprong te maken binnen een document. Hiermee worden dus ankers bedoeld die je naar een andere positie op dezelfde of een andere pagina brengen.

Een interne koppeling bestaat uit twee delen, een anker met naam `...` en een hyperlink naar die naam toe `...`.

```
<h2>Inhoudsopgave</h2>
<ul>
  <li><a href="#deel1">deel 1</a></li>
  <li><a href="#deel2">deel 2</a></li>
  ...
</ul>
<article>
  <h3><a id="deel1">Deel 1: Inleiding</a></h3>
  ...
</article>
<article>
  <h3><a id="deel2">Deel 2: Structuur</a></h3>
  ...
</article>
```

De ankerkoppeling met de hypertext reference wordt door de browser onderstreept weergegeven, die met het id-attribuut wordt echter niet onderstreept. Beiden worden afgesloten met ``.

In de oudere HTML-standaarden wordt gebruik gemaakt van het name-attribuut, tegenwoordig geniet het id-attribuut de voorkeur. In HTML5 hoeft er zelfs geen anker meer aanwezig te zijn en kan je het vermelde id ook aan een ander element toekennen. Onderstaande code werkt dus hetzelfde als de bovenstaande.

```
<h2>Inhoudsopgave</h2>
<ul>
  <li><a href="#deel1">deel 1</a></li>
  <li><a href="#deel2">deel 2</a></li>
  ...
</ul>
<article id="deel1">
  <h3>Deel 1: Inleiding</h3>
  ...
</article>
<article id="deel2">
  <h3>Deel 2: Structuur</h3>
  ...
</article>
```

Er kan eveneens vanuit een ander bestand gelinkt worden naar een interne link van een bestand.

```
<a href="index.html#deel1">deel 1</a><br>
```

Een veelgebruikte toepassing is de top-link in webpagina's met veel inhoud waarbij een verticale scrollbar nodig is om alle inhoud te kunnen zien. Hierbij wordt er bovenaan de pagina een `id="top"` geplaatst en onderaan een link naar die toplocatie.

```
<h1><a id="top">Hoofdstuk 1</a></h1>
...
<a href="#top">Top</a>
```

Het plaatsen van het topanker is strikt genomen niet noodzakelijk omdat `href="#"` sowieso naar het begin van de pagina verwijst.

```
<h1>Hoofdstuk 1</h1>
...
<a href="#">Top</a>
```

18 Andere soorten koppelingen

❖ Koppeling naar een e-mailadres

Hiervoor wordt het protocol `mailto` gebruikt met naam@systeem.be om op te geven wie de e-mail dient te ontvangen.

```
<a href="mailto:naam@systeem.be">...</a>
```

De `mailto`-opdracht kan uitgebreid worden met een "cc" (carbon copy), een "bcc" (blind carbon copy) en/of een "subject". Dit wordt ingegeven door achter de geadresseerde een vraagteken in te geven. Daarna komt de "cc", de "bcc" en/of de "subject" gescheiden door een ampersand.

```
<a href="mailto:naam@systeem.be?cc=iemandanders@bedrijf.be&
bcc=nogiemand@bedrijf2.be&subject=Dringende%20boodschap!">...</a>
```

Merk op dat `&` geëncodeerd is als entiteit `&`, omdat `&` op zich niet mag voorkomen in HTML. Verder is de spatie in het onderwerp geëncodeerd als `%20`, omdat een spatie niet mag voorkomen in een URL.

❖ Bestanden downloaden

Voor veel bestanden is het niet nodig een server met een apart protocol op te zetten. Door middel van een normale hyperlink kan je zip-bestanden, mp3-muziek, pdf-bestanden, videofragmenten (*.wmv, *.mov), Word-documenten, enzovoort downloaden.

Als de browser het bestandstype zelf ondersteunt kan het bestand geopend worden in een aparte webpagina of tab. In het andere geval krijg je niet altijd duidelijk de keuze het bestand op te slaan. Met pagina opslaan (Ctrl+S) lukt dit meestal wel. Rechtsklikken op de hyperlink biedt vaak ook deze mogelijkheid.

```
<a href="figuren.zip">download de figuren</a>
<a href="cursus.pdf">download de cursus</a>
```