

Webrichtlijnen en toegankelijkheid

WCAG, (X)HTML Strict en semantische markup

Bij het maken van webtoepassingen hebben ontwikkelaars onvermijdelijk te maken met webrichtlijnen. En dit is niet beperkt tot websites. Ook voor webapplicaties is het voldoen aan webrichtlijnen belangrijk.

Onder het ruime begrip 'webrichtlijnen' worden veelal de volgende preciezere begrippen, webstandaarden en technologieën geschaard:

- de 'webrichtlijnen' van de Nederlandse overheid
- Web Content Accessibility Guidelines (WCAG) van de W3C
- HTML Strict, XHTML en (gebruik van) CSS
- semantische markup

In dit artikel worden deze toegelicht en aangeven hoe zij gerelateerd zijn.

Webrichtlijnen Nederlandse overheid

Op 'webrichtlijnen.overheid.nl' worden 125 richtlijnen gegeven voor overheidswebsites. Deze hebben betrekking op het aanpassen van bestaande websites en het ontwikkelen van nieuwe websites. De richtlijnen hebben tot doel om overheidswebsites toegankelijk en duurzaam te maken. Onder een toegankelijke website wordt verstaan: "een website waarbij de inhoud van de site toegankelijk is voor zoveel mogelijke bezoekers, de bedoeling is dat er niemand wordt uitgesloten."

"Een echt toegankelijke website sluit geen bezoekers uit:

- Blinden en slechtzienden;
- Kleurenblinden;
- Bezoekers die niet beschikken over "proprietary" of "gesloten" technologieën (denk aan veel plug-ins);
- Bezoekers die niet beschikken over client-side script-ondersteuning of deze uitgeschakeld hebben;
- Bezoekers die gebruik maken van tekstbrowsers of andere alternatieve browsers."

Duurzaamheid heeft betrekking op het uniek adresseerbaar en vindbaar zijn van webpagina's, ook als er later wijzigingen in de structuur van de website gemaakt worden. Dit wordt ook wel

aangeduid met *permalinks*; permanente, liefst goed leesbare (betekenisvolle) links die blijven werken.

De variatie in de richtlijnen is groot. De richtlijn "zorg ervoor dat communicatieve elementen hun betekenis niet uitsluitend door kleur overbrengen" is een te verwachten richtlijn met betrekking tot kleurenblinden. Andere richtlijnen zijn "schrijf heldere, beschrijvende tekst voor links.", "specificeer de karakterset voor webpagina's." en "Indien men elementen in de HTML-hiërarchie manipuleert, maak gebruik van de W3C DOM volgens de specificatie".

Deze variatie geeft ook goed aan dat de verantwoordelijkheid voor het voldoen aan de richtlijnen bij verschillende rollen ligt. De eerst genoemde richtlijn is de verantwoordelijkheid van de webdesigner. De anderen zijn respectievelijk de verantwoordelijkheid van de contentmanager, het gebruikte contentmanagementsysteem of web application development framework en de programmeur. Het onderkennen van deze verantwoordelijkheden is essentieel voor het succes van projecten waarbij deze richtlijnen geïmplementeerd moeten worden.

De webrichtlijnen van de Nederlandse overheid worden door veel andere organisatie en bedrijven overgenomen. Zoals op webrichtlijnen.overheid.nl staat geschreven zijn de richtlijnen gebaseerd "op algemeen bekende webstandaarden en op de toegankelijkheidsrichtlijnen van de W3C". Met algemeen bekende webstandaarden worden met name HTML Strict, XHTML en CSS bedoeld. Deze worden in het vervolg van dit artikel beschreven. De toegankelijkheidsrichtlijnen van de W3C (WCAG) worden eveneens nader toegelicht.

Web Content Accessibility Guidelines

Versie 1.0 van deze toegankelijkheidsrichtlijnen van de W3C dateert al uit 1999. Deze recommendation valt binnen het Web Accessibility Initiative (WAI, zie 'www.w3.org/WAI'). Naast deze richtlijnen voor webcontent zijn er ook richtlijnen voor authoring tools, dat wil zeggen contentmanagementsystemen, (Authoring Tools Accessibility Guidelines, ATAG) en user agents (UJAG). Ook besteedt het WAI aandacht aan rich internet appli-

cations. Inmiddels, of beter gezegd nog steeds, is de W3C bezig met versie 2.0 van deze richtlijnen. De eerste draft dateert al uit 2001 (!). Verwacht wordt dat deze versie een recommendation wordt, dat wil zeggen klaar is, in 2007. De verschillen tussen WCAG 1.0 en WCAG 2.0 worden als volgt toegelicht:

Most Web sites that conform to WCAG 1.0 should not require significant changes in order to conform to WCAG 2.0. The fundamental issues of Web accessibility are the same, though there are some differences in the requirements between WCAG 1.0 and WCAG 2.0.

WCAG 2.0 is being developed to apply to more advanced Web technologies and be more precisely testable than WCAG 1.0. In order to do this, the WCAG 2.0 documents use a slightly different approach, for example, in terminology.

Eén van de bekendste WCAG-richtlijnen is het *niet* gebruiken van tabellen voor lay-out. In plaats daarvan wordt geadviseerd om gebruik te maken van 'div' elementen ('divs') en met behulp van CSS de gewenste lay-out te bepalen. Tabellen mogen wel gebruikt worden voor tabulaire data. Een goed toegankelijke tabel moet onder andere een 'summary' attribuut en 'caption' sub-element bevatten en tabelcellen en tabelheaders expliciet koppelen met de 'headers' en 'id' attributen:

```
<table width="80%" summary="The table lists colors, the metal that is
used to produce the color and an example compound that is
commonly used.">
<caption>Some of the more common color-producing compounds are
tabulated here
</caption>
```

Summary en caption van een tabel.

```
<tr>
<th id="color">Color</th>
<th id="metal">Metal</th>
<th id="compound">Example compounds</th>
</tr>
```

Identificatie van tabelheaders met het 'id' attribuut.

```
<tr>
<td headers="color">Orange</td>
<td headers="metal">Calcium</td>
<td headers="compound">calcium chloride</td>
</tr>
```

Expliciete koppeling van tabelcel aan tabelheader via het 'headers' attribuut.

Webrichtlijnen hebben ook betrekking op webapplicaties. Voor formulieren gelden onder meer de volgende toegankelijkheids-eisen.

- gebruik het 'label' element om een prompt aan een invoerveld te koppelen
- gebruik het 'tabindex' attribuut om de tab-volgorde van invoervelden te specificeren
- gebruik het 'fieldset' element om gerelateerde invoervelden te groeperen
- geef aan of een invoerveld verplicht of optioneel is
- neem geen herstel (reset) knop op voor het formulier

De W3C-website over WCAG biedt veel ondersteuning voor ontwikkelaars om de richtlijnen te implementeren. Veel richtlijnen hebben betrekking op een 'juist', 'net' en *semantisch* gebruik van HTML (mark-up). Hierop wordt in het vervolg van dit artikel verder ingegaan. Belangrijk is nog om op te merken dat in deze toegankelijkheidsrichtlijnen HTML Strict of XHTML niet genoemd worden.

HTML Strict, XHTML en CSS

HTML en CSS zijn de bekendste en belangrijkste webstandaarden (technologieën) voor het bouwen van webtoepassingen, althans voor het allerbovenste stukje van de presentatielaag; dat wat de gebruiker uiteindelijk ziet in zijn of haar browser. Samen met JavaScript bepalen HTML en CSS de *look & feel* van *traditionele* webtoepassingen op het Internet. Wel zijn *Rich Internet Applications* (RIA) sterk in opkomst, waarbij gebruik gemaakt wordt van andere technologieën zoals Flex van Macromedia (Adobe).

HTML mag als bekend verondersteld worden. CSS staat voor Cascading Style Sheets en wordt gebruikt om de 'stijl' (font, kleur, afmetingen, achtergronden et cetera) van een webpagina te definiëren. Een belangrijk onderdeel van de in dit artikel beschreven webrichtlijnen is het vergaand toepassen van CSS om in de HTML-markup geen elementen of attributen te gebruiken voor het definiëren van stijl, maar dit geheel met CSS te doen.

Van CSS bestaan verschillende versies: CSS, level 1 (CSS1) dateert uit 1996, level 2 (CSS2) uit 1998 en momenteel wordt gewerkt aan zowel CSS, level 2 revision 1 als CSS, level 3. CSS, level 2 revision 1 bevat een aantal veelgevraagde (zeer gewenste) nieuwe mogelijkheden uit CSS, level 3. Tussen de meest gebruikte browsers (Microsoft Internet Explorer (IE) 6, IE 7, Mozilla Firefox en Safari) zitten helaas behoorlijke verschillen in de ondersteuning van CSS. Dit betreft zowel het al dan niet ondersteunen van bepaalde stijlen, eigen CSS-uitbreidingen als ook verschillen in interpretatie. Dit maakt het ontwikkelen van cascading stylesheets die geschikt zijn voor verschillende browsers een specialisme voor webdesigners. Ook voor HTML is

| Versie | Beschikbaar sinds | Toelichting |
|-----------|----------------------|--|
| HTML 2.0 | 1995 | de historische standaard geschreven door Tim Berners-Lee en Dan Connolly |
| HTML 3.2 | 1997 (14 januari) | |
| HTML 4.0 | 1998 (24 april) | Strict, Transitional en Frameset varianten |
| HTML 4.01 | 1999 (24 december) | Strict, Transitional en Frameset varianten |
| XHTML 1.0 | 2000 (26 januari) | Strict, Transitional en Frameset varianten |
| XHTML 1.1 | 2001 (31 mei) | |
| XHTML 2.0 | nog niet beschikbaar | in ontwikkeling sinds 2002, laatste working draft is van 26 juli 2006 |

het interessant is om te zien wanneer de verschillende versies beschikbaar zijn gekomen en welke er in ontwikkeling is (zie tabel).

XHTML 1.1 is “een op de toekomst gerichte, gemodulariseerde versie van XHTML 1.0 *Strict*” waarmee de “historische ballast” van HTML 4 “afgeschud wordt”. Dit houdt in dat er van XHTML 1.1 geen (Strict, Transitional of Frameset) varianten meer zijn. Alle elementen en attributen die in HTML 4.01 Strict niet waren toegestaan zijn verdwenen in XHTML 1.1. XHTML 2.0 is een controversiële standaard omdat dit in feite een nieuwe markup taal is die niet backwards compatible is met eerdere HTML-versies. IBM karakteriseert XHTML 2.0 als: “*Good-bye backward compatibility, hello structure*”. Dat laatste maakt deze nieuwste versie erg interessant, in theorie *superieur* ten opzichte van eerdere versies, maar in de *praktijk* kansloos doordat het niet backwards compatible is.

Dat XHTML 2.0 ‘mooi’ is, kan al snel geconcludeerd worden uit een aantal voorbeelden. Zo bevat deze versie alleen nog de ‘sup’ (superscript) en ‘sub’ (subscript) elementen als elementen die voor presentatie dienen. Alle andere presentatie-elementen uit eerdere versies zijn verdwenen. De bekende elementen ‘b’ (bold) en ‘i’ (italic) bestaan niet meer. In plaats hiervan moeten ‘strong’ (strong emphasis) en ‘em’ (emphasis) gebruikt worden. Op de ‘sup’ en ‘sub’ elementen na geven alle elementen semantiek of structuur aan.

Een groot gemis aan de huidige versies van (X)HTML is dat er geen *nesting* (hiërarchie) aangegeven kan worden. In XHTML 2.0 kan dit wel middels het te nesten ‘section’ element. De huidige ‘h1’, ‘h2’, ‘h3’, ‘h4’, ‘h5’ en ‘h6’ elementen vervallen en worden vervangen door ‘h’ waarmee de header (titel) van een geneste sectie aangegeven kan worden. Verder vervalt het (*ongestructureerde*) ‘alt’ attribuut van een afbeelding. In plaats daarvan kan binnen het ‘img’-element een alternatief voor de afbeelding *gestructureerd* worden aangegeven (bijvoorbeeld meerdere paragrafen of een lijst). Tenslotte (als

laatste voorbeeld) – kunnen in XHTML 2.0 alle elementen een hyperlink zijn.

XHTML 2.0 is dus veel mooier (lees beter) dan voorafgaande versies. Of deze nieuwe versie op een hoge mate van adoptie kan rekenen is echter zeer twijfelachtig.

Even terug naar de voor wat betreft webrichtlijnen belangrijkste versies: HTML 4.01 en XHTML 1.0, met de bijhorende Strict, Transitional en Frameset varianten.

HTML is een *uitgebreide* standaard. Hiervan zijn, mede in reactie op de toegankelijkheidsrichtlijnen van de W3C, drie varianten gekomen: HTML 4.01 Strict, HTML 4.01 Transitional en HTML 4.01 Frameset. De eerste twee zijn hiervan het belangrijkste. De derde heeft betrekking op het gebruik van frames, wat over het algemeen wordt afgeraden. De Strict variant beperkt HTML door bepaalde elementen en attributen niet toe te staan (zie hieronder).

De webrichtlijnen van de Nederlandse overheid stellen het gebruik van HTML 4.01 Transitional verplicht voor het aanpassen van bestaande websites en HTML 4.01 Strict of XHTML 1.0 Strict voor het ontwikkelen van nieuwe websites.

XHTML 1.0 is de XML-variant van HTML 4.01. Aan HTML mark-up worden door browsers, verschillende, maar niet al te strenge eisen gesteld. Zo mag gerust de sluit-tag van een tabelcel (‘td’ element) weggelaten worden, mogen attribuutwaarden ook zonder dubbele quotes opgenomen worden en kunnen elementen elkaar overlappen. In XML, en daarmee in XHTML, worden veel strengere eisen gesteld aan de mark-up: alle elementen moeten netjes afgesloten worden, moeten netjes genest zijn (mogen elkaar niet overlappen) en waarden van attributen moeten tussen dubbele quotes staan. Van XHTML 1.0 bestaan dezelfde drie varianten van HTML 4.01: XHTML 1.0 Strict, XHTML 1.0 Transitional en XHTML 1.0 Frameset.

| Color | Metal | Example compounds |
|--------|--|-------------------------------------|
| Red | Strontium (intense red) | strontium carbonate |
| | Lithium (medium red) | lithium carbonate |
| Orange | Calcium | calcium chloride |
| Yellow | Sodium | sodium nitrate |
| Green | Barium | barium chloride ions |
| Blue | Copper halides | copper chloride, at low temperature |
| Purple | potassium | potassium nitrate |
| Gold | Charcoal, iron, or lampblack | |
| White | Titanium, aluminum, or magnesium powders | |

De beperkingen in de Strict-varianten betreffen het niet mogen gebruiken van bepaalde HTML-elementen en attributen. Namelijk die elementen die stijl of presentatie bepalen zoals de elementen 'font' en 'center' en de attributen 'bgcolor' en 'width' van het 'td' element. Bovendien worden elementen en attributen die uitgefaseerd (*deprecated*) zullen worden uitgesloten in de Strict-varianten. In de Transitional-varianten mogen al deze elementen en attributen wel gebruikt worden. Een goed voorbeeld om de verschillen tussen de beperkte Strict variant en de vrijere Transitional-variant te laten zien, is het opmaken van een tabel.

De meeste mensen zullen gewend zijn om de breedte en uitlijning van een tabelcel aan te geven met respectievelijk de 'width' en 'align' attributen:

```
<th width="15%">Color</th>
<th width="35%">Metal</th>
<th width="50%">Example compounds</th>
<td align="right">copper chloride, at low temperature</td>
```

In de Strict-variant is het 'width'-attribuut niet toegestaan. In plaats hiervan moet van (het ook veel *subtielere*) element 'col' gebruik gemaakt worden om eigenschappen van kolommen aan te geven:

```
<col valign="top" width="15%" />
<col width="35%" />
<col width="50%" align="right" />
```

Met het 'col' element wordt niet alleen de width aangegeven maar kunnen ook andere attributen voor alle tabelcellen in de kolom gespecificeerd worden. In het voorbeeld wordt de verticale uitlijning van de eerste kolom op 'top' gezet en de horizontale uitlijning van de derde kolom op 'right'. De uitlijningsattributen mogen, ook in Strict, ook voor de tabelcel zelf gebruikt worden. Maar het is uiteraard veel handiger om dit per kolom te doen. Om voor een bepaalde tabelcel een uitzondering te maken

moet de eigenschap wel op de tabelcel zelf gezet worden. Door de source van een webpagina te bekijken is eenvoudig te zien van welke HTML- of XHTML-variant de site gebruikt maakt, of beter gezegd *claimt* gebruik te maken (zie hieronder). Dit moet namelijk als eerste worden aangegeven in de DOCTYPE processing instruction.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
...
```

XHTML 1.1

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
...
```

XHTML 1.0 Strict

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
...
```

XHTML 1.0 Transitional

```
<!DOCTYPE HTML
PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
...
```

HTML 4.01 Strict

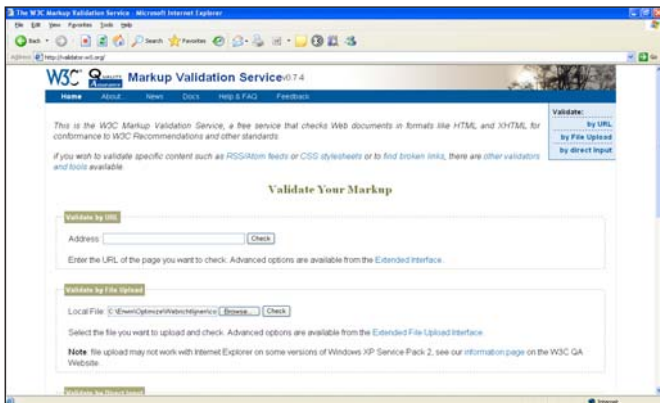
```
<!DOCTYPE HTML
PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
...
```

HTML 4.01 Transitional

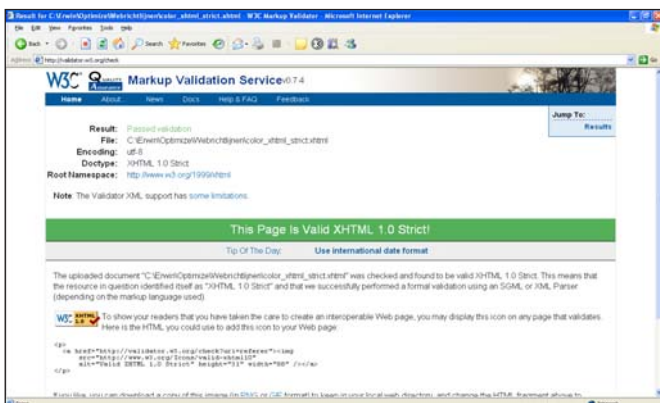
Zo is te zien dat 'nu.nl' (www.nu.nl) gebruik maakt van HTML 4.01 Transitional en de website van het Ministerie van VWS

(www.minvws.nl) van XHTML 1.0 Strict. Met de DOCTYPE processing instruction wordt 'geclaimd' dat de webpagina een bepaalde variant gebruikt. Dit wordt echter niet gecontroleerd (gevalideerd) door de browser. Wel kan de browser anders reageren op basis van deze instructie. Een ontwikkelaar kan wel valideren of de webpagina correct is. Dit kan bijvoorbeeld op validator.w3.org.

Als ontwikkelaar krijg je dan precies te zien welke 'fouten' er eventueel in de HTML zitten. Als de pagina *valid* is wordt een pagina getoond zoals te zien is op afbeelding 2. Opmerkelijk is dat in HTML Strict de 'meta', 'link' en 'col' elementen juist niet zoals in XHTML verplicht is afgesloten mogen worden. Je mag het blijkbaar niet 'te mooi' maken. Om aan te geven dat een webpagina valid HTML of XHTML is kan een logo op de pagina geplaatst worden (zie afbeelding 4). Zoals al eerder opgemerkt wordt het gebruik van de Strict of Transitional varianten of XHTML niet voorgeschreven door WCAG 1.0. Wel helpt het gebruik hiervan om te voldoen aan deze toegankelijkheidseisen. Zoals al eerder opgemerkt wordt het gebruik van de Strict of Transitional varianten of XHTML niet voorgeschreven door WCAG 1.0. Wel helpt het gebruik hiervan om te voldoen aan deze toegankelijkheidseisen.



Afbeelding 1. Een ontwikkelaar kan op deze site valideren of de webpagina correct is.



Afbeelding 2. Als de pagina valid is wordt een pagina als deze getoond.



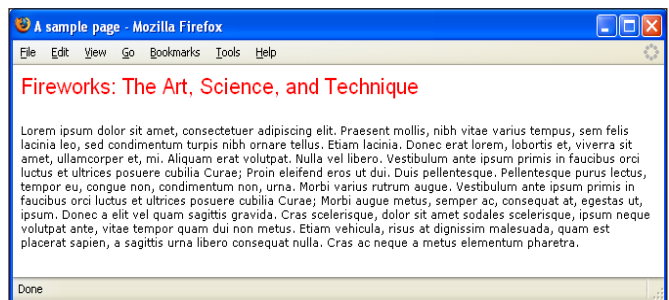
Afbeelding 3. Een foutmelding voor niet-valide code.



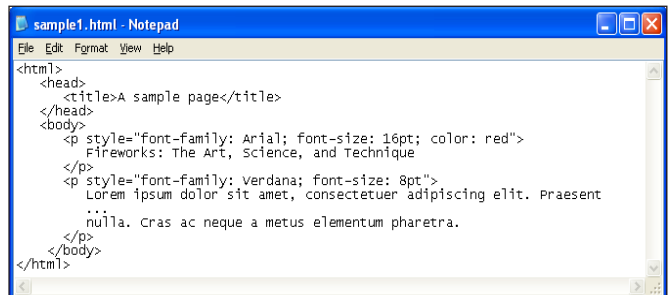
Afbeelding 4. Om aan te geven dat een webpagina valid HTML of XHTML is kan een logo op de pagina geplaatst worden.

Semantische markup

De belangrijkste 'bezoeker' van een website is geen persoon, maar een programma. Namelijk *web crawlers* die webpagina's indexeren voor zoekmachines zoals Google, Yahoo en AltaVista. Deze programma's moeten *betekenis* afleiden uit de HTML mark-up om goed te kunnen bepalen welke woorden belangrijk zijn als *zoektermen*. Hiervoor is het van belang om betekenisvolle, semantische, mark-up te gebruiken in webpagina's. Zo zou de titel van een pagina als een speciale paragraaf in grote rode letters getoond kunnen worden. Voor een (niet-visueel gehandicapte) bezoeker zal het dan duidelijk zijn wat de titel is, en belangrijke zoektermen zijn. Voor de webcrawler is dat niet duidelijk, en de 'titel' zal niet als belangrijker geïndexeerd worden dan woorden in andere paragrafen.



Afbeelding 5. De webcrawler zal de 'titel' niet als belangrijker indexeren dan woorden in andere paragrafen.



Afbeelding 6. Voor indexering door zoekmachines is het van belang om betekenisvolle, semantische mark-up te gebruiken in webpagina's.

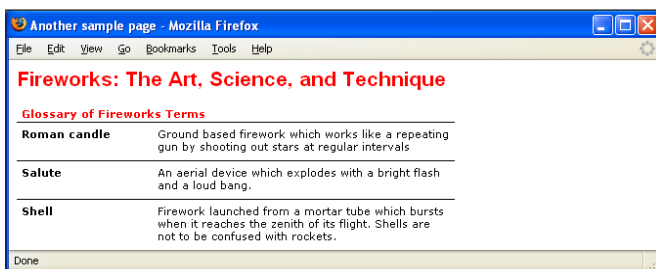
Door gebruik te maken van het voor een titel bedoelde 'h1' element (een header van het hoogste niveau) kan de webcrawler wel betekenis afleiden, de titel herkennen en de woorden in de titel als belangrijke zoektermen indexeren.

```
<h1>Fireworks: The Art, Science, and Technique</h1>
```

Naast de bekende en voor de hand liggende h1, h2, h3, h4, h5 en h6 elementen zijn er nog een groot aantal minder bekende HTML-elementen die voor semantische mark-up gebruikt kunnen worden, bijvoorbeeld voor het markeren van termen die voor het eerst op een pagina gebruikt worden. Hiervoor kan het element 'dfn' (definition) gebruikt worden, en voor verwijzingen naar boeken, rapporten of websites het 'cite' (citation) element. Voor een lijst van definities is er het 'dl' (definition list) element, met daarbinnen de 'lh' (list header), 'dt' (definition term) en 'dd' (definition description) sub-elementen, zoals in afbeelding 7 te zien is. De definition list kan uiteraard met behulp van CSS gestyled worden.

```
semantics.html - Notepad
File Edit Format View Help
<html>
<head>
<title>Another sample page</title>
<link rel="stylesheet" type="text/css" href="semantics.css"/>
</head>
<body>
<h1>Fireworks: The Art, Science, and Technique</h1>
<dl>
<lh>Glossary of Fireworks Terms</lh>
<dt>Roman candle</dt><dd>Ground based firework which works like a repeating
gun by shooting out stars at regular intervals
<dt>Salute</dt><dd>An aerial device which explodes with a bright flash and
a loud bang.
<dt>Shell</dt><dd>Firework launched from a mortar tube which bursts when
it reaches the zenith of its flight. Shells are
not to be confused with rockets.
</body>
</html>
```

Afbeelding 7. Voor een lijst van definities is er het 'dl' (definition list) element, met daarbinnen verschillende sub-elementen.



Afbeelding 8. De definition list kan met behulp van CSS gestyled worden.

Voor semantische mark-up kunnen in plaats van de veelgebruikte, maar betekenisloze 'i' (italic) en 'b' (bold) elementen beter de betekenisvolle elementen 'em' (emphasis) en 'strong' (strong emphasis) gebruikt worden. Ook voor tabellen en afbeeldingen bestaan (sub-)elementen en attributen die waardevolle semantiek toevoegen. Semantische mark-up heeft dus belangrijke waarde voor het vindbaar maken van webpagina's. Maar het is ook belangrijk voor screen readers voor visueel

gehandicapten. Een groot aantal aspecten van semantische mark-up zijn daarom ook terug te vinden in de WCAG-richtlijnen van W3C en de webrichtlijnen van de Nederlandse overheid.

Conclusie

Het is van eminent belang dat de opdrachtgever (bewust) het *ambitieniveau* bepaalt, dat hij een besluit neemt tot hoe ver webrichtlijnen, toegankelijkheid, (X)HTML Strict en semantische mark-up geïmplementeerd moeten worden. Hoewel het volgen van webrichtlijnen vooral de verantwoordelijkheid is van de webdesigner, of beter gezegd de HTML-ontwikkelaar, is zullen ook programmeurs en consultants goed op de hoogte moeten zijn van webrichtlijnen. Met name om te kunnen adviseren en ondersteunen bij de afstemming tussen klant en webdesigner en om de deliverables van de webdesigner te kunnen beoordelen. Ook de content manager heeft een belangrijke rol bij het naleven van webrichtlijnen. De richtlijnen betreffen namelijk niet alleen 'harde' technische aspecten, maar hebben ook een 'zachte kant' met betrekking tot het schrijven van teksten. Het onderkennen van de verschillende verantwoordelijkheden is essentieel voor het succes van projecten waarbij webrichtlijnen geïmplementeerd moeten worden.

Erwin Groenendal is Senior Solutions Architect bij Cumquat Information Technology. Cumquat richt zich op de het ontwikkelen van enterprise Java-oplossingen op het Oracle-platform. Voor vragen en opmerkingen over dit artikel is Erwin te bereiken via erwin@cumquat.nl.