

# O.O.P.-Project

# Toegankelijke Webportalen

# aan de K.U.Leuven

Richtlijnen voor het maken van toegankelijke webpagina's

Auteurs: Nadia Diraä & Christophe Strobbe  
email: [nadia.diraa@esat.kuleuven.be](mailto:nadia.diraa@esat.kuleuven.be)

juli 2008

## Inhoudsopgave

1	Inleiding.....	3
1.1	10.O.P.-project Toegankelijke Webportalen.....	3
1.2	Webtoegankelijkheid.....	3
2	Richtlijnen.....	4
2.1	Inhoud.....	4
2.1.1	Scheid structuur van presentatie.....	4
2.2	Presentatie.....	4
2.2.1	Tekstopmaak en vormgeving.....	4
2.2.2	Voorzie alternatieven voor visuele en auditieve inhoud.....	4
2.2.3	Creëer toegankelijke tabellen.....	4
2.2.4	Geef het gebruik van de natuurlijke taal aan.....	4
2.3	Navigatie.....	4
2.3.1	Lever duidelijke navigatiemechanismen.....	4
2.4	Interactie.....	4
2.4.1	Toegankelijke formulieren, Java(Script) en Flash.....	4
3	Technieken.....	5
3.1	Inhoud.....	5
3.1.1	Scheid structuur van presentatie.....	5
3.2	Presentatie.....	6
3.2.1	Tekstopmaak.....	6
3.2.2	Voorzie alternatieven voor visuele en auditieve inhoud.....	8
3.2.3	Creëer toegankelijke tabellen.....	8
3.2.4	Geef het gebruik van de natuurlijke taal aan.....	9
3.3	Navigatie.....	10
3.3.1	Lever duidelijke navigatiemechanismen.....	10
3.4	Interactie.....	11
3.4.1	Formulieren, Java(Script) en Flash.....	11

# O.O.P.-Project Toegankelijke Webportalen aan de K.U.Leuven<sup>1</sup>

## Richtlijnen voor het maken van toegankelijke webpagina's

### 1 Inleiding

#### 1.1 O.O.P.-project Toegankelijke Webportalen

De K.U.Leuven wil zich in het kader van het Onderwijsontwikkelingsplan sterker engageren om de toegankelijkheid voor studenten met een functiebeperking te verhogen. Het project "Toegankelijke Webportalen" spitst zich meer in het bijzonder toe op de toegankelijkheid van ICT-toepassingen voor studenten met een leeshandicap (visuele handicap, motorische handicap en dyslexie).

Na een studie over de verschillende beschikbare toegankelijkheidsrichtlijnen (zoals de WCAG1.0 en WCAG2.0 richtlijnen, de Web Content Accessibility Guidelines van het W3C - zie ook: <http://www.w3c.org/> en Anysurfer) ligt onderstaand voorstel voor over de richtlijn(en) waarop de K.U.Leuven zich best richt m.b.t. de eigen ICT-platformen.

#### 1.2 Webtoegankelijkheid

Webtoegankelijkheid zorgt ervoor dat alle bezoekers de website of webapplicatie gemakkelijk kunnen gebruiken in uiteenlopende omstandigheden en met verschillende configuraties. Webtoegankelijkheid zorgt er dus ook voor dat webpagina's toegankelijk zijn voor mensen met een visuele of andere handicap. Om dit te realiseren, zijn er enkele regels opgesteld waarmee je best rekening houdt. Deze regels vragen geen moeilijke technieken of programmeertalen, enkel dat je rekening houdt met details tijdens het maken van een webpagina.

---

<sup>1</sup>Het OOP project wordt gesponsord door de K.U.Leuven. Het wordt uitgevoerd binnen de SCD-Docarch groep van de Faculteit Ingenieurwetenschappen.  
info: [jan.engelen@esat.kuleuven.be](mailto:jan.engelen@esat.kuleuven.be)

## **2 Richtlijnen**

### **2.1 Inhoud**

#### **2.1.1 Scheid structuur van presentatie**

Maak documenten op met de juiste structuurelementen.  
Zet de inhoud van de pagina in de HTML-code op volgorde van belangrijkheid.  
Zorg dat de structuur en de vormgeving zoveel mogelijk gescheiden zijn. (Gebruik HTML voor de structuur en CSS voor de vormgeving.)  
Zorg dat het taalgebruik is aangepast aan het niveau van de bezoeker van de site.

### **2.2 Presentatie**

#### **2.2.1 Tekstopmaak en vormgeving**

Gebruik niet enkel kleur om informatie over te brengen.  
Zorg voor voldoende contrast in kleurgebruik.  
Wees consistent bij het vormgeven van verschillende pagina's van een website.  
Geef de gebruiker voldoende tijd om de pagina te lezen en te gebruiken.

#### **2.2.2 Voorzie alternatieven voor visuele en auditieve inhoud**

Lever inhoud die bij presentatie aan de gebruiker in essentie dezelfde functie of hetzelfde doel uitdrukt als auditieve of visuele inhoud.

#### **2.2.3 Creëer toegankelijke tabellen**

Let erop dat tabellen de noodzakelijke opmaak hebben om door toegankelijke browsers of hulptechnologieën getransformeerd te worden.

#### **2.2.4 Geef het gebruik van de natuurlijke taal aan**

Zorg dat de uitspraak en interpretatie van afgekorte of buitenlandse tekst mogelijk is.

### **2.3 Navigatie**

#### **2.3.1 Lever duidelijke navigatiemechanismen**

Lever duidelijke en consistente navigatiemechanismen - oriëntatie-informatie, navigatiebalken, een site map, etc. - en verhoog de kans dat iemand vindt wat hij zoekt.

### **2.4 Interactie**

#### **2.4.1 Toegankelijke formulieren, Java(JavaScript) en Flash**

Zorg dat de inhoud toegankelijk is bij gebruik van Formulieren, Java(JavaScript) of Flash, of zorg voor een toegankelijk alternatief.

## 3 Technieken

### 3.1 Inhoud

#### 3.1.1 Scheid structuur van presentatie

Houd bij het samenstellen van documenten voor het web al rekening met de manier waarop je dit document wil structureren, voordat je begint aan hoe je je document wil presenteren. Duidelijk gestructureerde webdocumenten zijn gemakkelijk te begrijpen voor iedereen, en dus ook voor mensen met een visuele of cognitieve handicap. Een overzichtelijk document leidt automatisch de aandacht naar de belangrijkste onderdelen van de pagina. Zorg ervoor dat je tekst een duidelijke structuur heeft door het onder te verdelen in hoofdstukken, titels, ondertitels, paragrafen enz.

De structuur van een document in HTML wordt logisch opgebouwd als je gebruik maakt van structurele elementen zoals `<h1>` - `<h6>`, `<p>`, `<blockquote>`. Elk onderdeel van je tekst leid je in door een header element in html (`<h1>` tot `<h6>`). Gebruik deze headers op een duidelijke manier, en respecteer de volgorde (`<h2>` volgt op `<h1>`, `<h3>` op `<h2>`, enz.) Headers invoegen om een bepaald effect te creëren op je lettertype, doe je zeker niet. Hiervoor kan je style sheets gebruiken.

Je verdeelt de inhoud aan de hand van blokken die telkens op een nieuwe lijn beginnen:

- headers (`<h1>`, `<h2>`, `<h3>`, ... `<h6>`)
- divisions (`<div>`)
- paragrafen (`<p>`)
- geordende en ongeordende lijsten (`<ol>` en `<ul>` en de bijhorende `<li>`)
- formulieren (`<form>` en bijhorende elementen zoals `<input>`, `<select>`, ...)
- datatabellen (`<table>` en `<th>`, `<td>`)

Heb je je document gestructureerd aan de hand van bovenstaande lijst van blokken, dan wordt de inhoud op een duidelijke manier weergegeven. Alle bijkomende opmaak plaats je in een style sheet. Dit heeft als voordeel dat je webpagina sneller laadt, en gemakkelijker te onderhouden en te updaten is. Ook is het later mogelijk de stijl van je document te veranderen door enkel je style sheet te wijzigen. Wie gebruik maakt van hulptechnologieën heeft op deze manier vlot toegang tot de informatie uit het document.

Voor de lezers die een pagina scannen, gebruik je duidelijke, informatieve titels, zodat het niet nodig is alles in detail te lezen om de informatie die je zoekt snel te vinden.

Beperk elke paragraaf tot één hoofdtopic. Bepaal de topic van de alinea of paragraaf in het begin van de zin of paragraaf, hiermee help je mensen die vluchtig doorlezen maar ook mensen die spraaktechnologie gebruiken. Die “scannen” webpagina's door van titel naar titel te springen, of het begin van elke paragraaf te beluisteren om te bepalen welke brok informatie ze juist willen horen. Staat de hoofdidee in het midden of aan het einde van de paragraaf, dan moeten zij bijna heel het document doorluisteren voor ze de informatie vinden die ze zoeken.

Vermijd slang of jargon tenzij het in je document wordt uitgelegd. Gebruik woorden die vaak voorkomen, zoals “proberen” in plaats van “pogen”. Gebruik actieve in plaats van passieve werkwoorden. Vermijd complexe structuren in je zinnen.

Voorzie links naar andere versies van de inhoud, zoals een vertaling, indien beschikbaar en gewenst. Geef de taal van de inhoud aan (zoals via de HTML attributen “type” en “hreflang”).

Mensen die niet goed of helemaal niet kunnen lezen, hebben baat bij een multimedia versie om je document te begrijpen. Let er wel op dat multimedia je tekst begrijpbaar maakt want soms zorgt het voor nog meer verwarring. Voorbeelden zijn: een grafiek van complexe gegevens zoals verkoopcijfers, een vertaling van een tekst naar gebarentaal, ...

## 3.2 Presentatie

### 3.2.1 Tekstopmaak

Zorg ervoor dat informatie niet enkel door kleur wordt overgebracht. Via opmaak en kontekst kan je informatie ook beschikbaar stellen voor iemand die geen of moeilijk kleuren kan onderscheiden. Je kan opmaak zoals <strong> gebruiken of bijkomende tekst zoals een asterisk of verdere uitleg.

Gebruik voor de verschillende paginaonderdelen kleuren die voldoende contrasteren. Een goede helderheidscontrastverhouding is 5 op 1. Dit kan je testen met volgende online tools (zie voor een artikel met een overzicht van verschillende tools ook :

[http://www.456bereastreet.com/archive/200709/10\\_colour\\_contrast\\_checking\\_tools\\_to\\_improve\\_the\\_accessibility\\_of\\_your\\_design/#comments](http://www.456bereastreet.com/archive/200709/10_colour_contrast_checking_tools_to_improve_the_accessibility_of_your_design/#comments))

Kleur contrast controleren/analyseren:

[http://www.snook.ca/technical/colour\\_contrast/colour.html](http://www.snook.ca/technical/colour_contrast/colour.html)

<http://juicystudio.com/services/colourcontrast.php>

<http://juicystudio.com/article/colour-contrast-analyser-firefox-extension.php>

<http://juicystudio.com/services/csstest.php>

<http://juicystudio.com/services/luminositycontrastratio.php>

<http://www.vischeck.com/vischeck/>

<http://www.paciellogroup.com/resources/contrast-analyser.html>

<http://www.graybit.com/main.php>

<http://michelf.com/projects/sim-daltonism/>

Geef kleuren aan met de RGB-waarde (wit is dan color: rgb(100%, 100%, 100%) of color: rgb(255, 255, 255) of de hexadecimale waarde (color: #ffffff of color: #fff)

Om tekst te benadrukken, dienen volgende HTML-elementen: <em> en <strong> in plaats van <b> en <i>. Deze laatsten geven enkel een visueel effect. De HTML-elementen <em> en <strong> geven een structurele nadruk die op verschillende manieren kan weergegeven worden (wijzigingen in het lettertype of stembuigingen).

Ook het gebruik van afbeeldingen om tekst in een bepaalde stijl weer te geven, is af te raden. Door tekst te gebruiken zorg je ervoor dat de informatie beschikbaar is voor een

groter publiek en bied je de mogelijkheid aan de gebruiker om eigen style sheets te gebruiken. Wil je toch een afbeelding gebruiken om een bepaald effect te creëren, zorg er dan voor dat de afbeelding toegankelijk is.

Je kan tekst opmaken met volgende CSS eigenschappen:

Inspringen doe je met 'text-indent'. Je maakt dus geen gebruik van <blockquote> of een ander structurelement van HTML dat een gelijkaardig visueel effect heeft.

Met 'letter-spacing' en 'word-spacing' kan je bepalen hoeveel ruimte er tussen letters en woorden gelaten wordt boven de standaardruimte. Dit is van belang wanneer tekst wordt voorgelezen door een screen reader.

'white-space' gebruik je als je wilt beïnvloeden hoe de witruimte wordt weergegeven, bijvoorbeeld spaties en de overgang naar een nieuwe regel.

Met :first-letter en :first-line (pseudo-elementen) kan je verwijzen naar de eerste letter of regel van een paragraaf of stukje tekst.

Ook lay out, positionering en uitlijning kan je vastleggen met een style sheet in CSS.

Hiervoor beschik je over volgende eigenschappen: 'text-indent', 'text-align', 'word-spacing', 'font-stretch' waarmee je de ruimte rond tekst kunt bepalen zonder spaties te gebruiken.

Gebruik bijvoorbeeld 'text-align: center' in plaats van het <center> element.

Met 'margin', 'margin-top', 'margin-right', 'margin-bottom', 'margin-left' kan je de vier zijanten van een blok inhoud bepalen zonder een <br> te hoeven gebruiken.

'float', 'position', 'top', 'right', 'bottom', 'left' kan je gebruiken om de visuele positie van bijna om het even welk element te bepalen, onafhankelijk van waar het element in de broncode staat. Hier moet je er wel op letten dat je document nog te begrijpen is zonder het gebruik van een style sheet. Zorg dus dat de inhoud op een logische manier is opgebouwd en pas er daarna een stijl op toe om bepaalde visuele effecten te krijgen. Je kan deze eigenschappen gebruiken om marges te maken, side bars, het effect van frames, eenvoudige titels, voetteksten, enz. ...

Je kan elementen zoals lettergrootte of marge, een waarde geven op 3 verschillende manieren: met de exacte pixels, in percentages of via de "em" eenheid. Wanneer je een exacte waarde geeft, spreekt men over een absolute eenheid; gebruik je de laatste twee, dan hebben we het over relatieve eenheden. Voor de lettergrootte bvb gebruik je best "em", waarbij "em" staat voor de hoogte van het huidige lettertype. Dit zorgt ervoor dat je tekst schaalbaar is. Let op! Tekst in afbeeldingen is op deze manier niet schaalbaar en gebruik je dus best niet.

Voor opsommingen en genummerde lijsten gebruik je best <ul> en <ol> in de documentcode. Deze HTML-elementen toepassen om de opmaakeffecten doe je zeker niet. Blinde of slechtziende gebruikers navigeren met behulp van geordende lijsten en kunnen verloren geraken in geneste lijsten die niet goed zijn opgemaakt. Samengestelde nummering, zoals "1, 1.1, 1.2, 1.2.1, 1.3, 2, 2.1," geeft meer kontekst aan waardoor de gebruiker duidelijk weet waar hij juist zit. Ook bij ongeordende lijsten probeer je een kontekst te voorzien.

Om tekst op te maken kan je volgende CSS-eigenschappen gebruiken:

Hoofd- of kleine letters: 'text-transform' (voor hoofdletters en kleine letters: capitalize, uppercase, lowercase). Schaduw effect: 'text-shadow' - 'text-decoration' : onderstrepen

(underline), lijn boven de tekst (overline), knipperen (blink). Als je de waarde “blink” gebruikt, zorg er dan ook voor dat de gebruiker dit kan afzetten. Via CSS kan dit door de style sheet af te zetten of een gebruikergedefinieerd style sheet te bepalen. HTML-elementen <blink> en <marquee> behoren niet tot de W3C specificatie voor HTML (het zijn dus geen standaardelementen).

Je kan webpagina's automatisch laten bijwerken maar dit kan bij verschillende gebruikers erg storend overkomen. Bied in de plaats de gebruiker de mogelijkheid aan om dit zelf te doen.

Zorg dat de belangrijkste navigatieknoppen op dezelfde plaats staan op elke pagina. Wees consistent met kleuren en tekststijlen zodat gebruikers zich gemakkelijk terugvinden op je site of direct merken dat ze er niet langer zijn. Gebruik hiervoor een minimaal aantal style sheets, vermijd om de stijl per pagina of per onderdeel te bepalen. Zorg dat concepten die terugkomen, dezelfde naam en stijl krijgen in je style sheet.

### **3.2.2 Voorzie alternatieven voor visuele en auditieve inhoud**

Voor volgende elementen neem je een tekstalternatief op: logo's, foto's, submit buttons, applets, bullets in lijsten, ASCII art, alle links in een image map, en ook onzichtbare afbeeldingen die gebruikt worden om een pagina te layouten. Voor deze laatste kan je alt="" gebruiken.

Als je tekst gebruikt, kan dit weergegeven worden op die manier die de gebruiker het best past. Screen readers zetten het om naar spraak, braillelezers naar braille, het kan worden vergroot, gesynchroniseerd worden met een video, ... Zorg dus voor een tekstequivalent voor elke non-tekst informatie. Voor eenvoudige inhoud volstaat een eenvoudige omschrijving, voor meer complexe afbeeldingen (grafieken of diagrammen, ...) kan je een uitgebreidere tekst voorzien.

Beeld- en geluidsporen synchroniseren met ondertitels of een geluidsbeschrijving zorgt ervoor dat bijna iedereen een equivalente ervaring kan hebben.

Met Quicktime en SMIL kan je ondertitels en video-omschrijvingen toevoegen aan een multimediaclip. Indien dit niet mogelijk is, zorg dan dat er verschillende versies beschikbaar zijn, eentje met ondertitels en geluidsbeschrijving en eentje zonder.

### **3.2.3 Creëer toegankelijke tabellen**

Je kan tabellen op verschillende manieren toegankelijk maken. In het <table> element kan je het attribuut summary toevoegen, met een beschrijving van de volledige tabel. Dit valt te vergelijken met de alt-tekst bij afbeeldingen. De informatie uit dit attribuut verschijnt niet op het scherm (daarvoor kan je <caption> gebruiken, dat een bijschrift plaatst bij de tabel) maar dient om mensen die op een niet-visuele manier surfen te helpen bij het begrijpen van de tabel.

Gebruik rij- en kolomkoppen met het <th> element. De browser zal de inhoud van deze cellen in vet weergeven en aldus accentueren. En wie surft met een hulptechnologie krijgt de tabel op een logische manier gelineariseerd. Dit zorgt er immers voor dat elke cel expliciet is geassocieerd met de bijhorende header.



Om groepen rijen die structureel bijeenhoren of herhaald moeten worden (over meerdere bladzijden bij het printen bvb.) gebruik je <thead>, <tfoot> voor ondertitels en <tbody> voor datacellen die bij elkaar horen. Voor kolommen die bij elkaar horen, gebruik je <colgroup> en <col>. De eerste dient om aan te geven wat structureel bij elkaar hoort, de tweede om in één keer een aantal kenmerken van een groep kolommen aan te geven.

Je kan voor niet-visuele gebruikers de tabel nog verder verduidelijken door aan de tabelkoppen het scope attribuut toe te voegen. Hiermee geef je duidelijk aan dat de tabelkop geldt voor elke cel die binnen de scope valt. Geef je de scope waarde col mee, dan geldt dat voor elke cel die in de kolom valt van de tabelkop. Voor de waarde row geldt het voor elke cel die in de rij van de tabelkop valt.

Het <pre> element dat gebruikt wordt om tekst weer te geven zoals het wordt ingetypt (inclusief spaties, e.d.) gebruiken we zeker niet om tabellen te maken. Dit geeft immers een louter visueel effect en maakt ze niet toegankelijk voor wie niet visueel surft.

Om de layout en positionering van een pagina op te maken, dien je gebruik te maken van style sheets. Kan je echt niet anders, zorg er dan voor dat je tabel logisch in elkaar steekt als ze wordt gelineariseerd. Gelineariseerd wordt je tabel een reeks paragrafen achter elkaar. Daarom is het belangrijk dat er ook structurele elementen (paragrafen, titels, lijsten, ...) worden gebruikt. Ook hier creëren we geen visuele opmaak door het gebruik van structurele opmaak (bvb. <th> zorgt ervoor dat de tekst gecentreerd en vet wordt weergegeven). Om een bepaalde cel een bepaalde opmaak te geven gebruik je best een style sheet.

### 3.2.4 Geef het gebruik van de natuurlijke taal aan

De natuurlijke taal van een document is de taal waarin de tekst is opgesteld: Nederlands, Engels, ... Indien in de tekst deze taal verandert, kunnen gebruikers die bijvoorbeeld gebruik maken van spraaktechnologie, tekst te horen krijgen met een vreemde uitspraak of een raar accent. Als het is aangegeven kunnen meertalige gebruikers de software automatisch laten switchen naar de juiste taal.

Staat er op je pagina bijvoorbeeld een citaat in het Frans, dan kan je dit in de code aangeven: <blockquote lang="fr">Ceci n'est pas Français.</blockquote>.

Gebruik <abbr> voor afkortingen, zeker als er onduidelijkheid kan bestaan over de betekenis ervan. Voor letterwoorden kan je <acronym title="Young Urban Professional" lang="en">Yuppie</acronym> gebruiken.

De belangrijkste taal van een webpagina wordt best aangegeven in het <html> element. Dit gebeurt als volgt: <html lang="nl">. De meest voorkomende talen en hun afkorting:

- Engels: en
- Frans: fr
- Duits: de
- Spaans: es
- Nederlands: nl

Voor een volledige lijst, zie [http://www.loc.gov/standards/iso639-2/php/code\\_list.php](http://www.loc.gov/standards/iso639-2/php/code_list.php).

## 3.3 Navigatie

### 3.3.1 Lever duidelijke navigatiemechanismen

De links op een pagina maak je toegankelijk door heldere, beschrijvende tekst te kiezen en deze beknopt te houden. Zorg ervoor dat ze duidelijk te onderscheiden zijn van andere tekst. Gebruik zeker geen tekst als “Klik hier” of “Meer info hier”. Buiten de context geven deze links immers geen enkele aanduiding waarnaar het verwijst. Je kan de linktekst ook nog bijkomende informatie geven door het title attribuut te gebruiken. Hiermee kan je de bestemming van de link duidelijk en accuraat beschrijven.

Zijn er meerdere links op een pagina die naar dezelfde webpagina leiden, geef ze dan allemaal dezelfde naam. Omgekeerd, zijn er links met dezelfde naam, zorg er dan voor dat ze ook naar dezelfde externe pagina verwijzen.

Maak navigatie doorheen een website consistent. Gebruik een navigatiemechanisme dat op elke pagina terugkeert en een gemeenschappelijke stijl heeft. Als het navigeren doorheen een website vlot leidt tot de informatie die de gebruiker zoekt, hebben alle gebruikers en dus ook mensen met lees- of leerproblemen hier baat bij. Je kan hierbij gebruik maken van visuele en tekstuele opmaakeffecten die aangeven en de gebruiker helpen onthouden waar hij juist zit op de website.

Een ander voordeel van een consistent navigatiemechanisme is de voorspelbaarheid. Dit maakt het voor de gebruiker mogelijk om (dezelfde set van) links over te slaan en dadelijk naar de inhoud van de pagina te springen.

Gerelateerde links, zoals een navigatiebalk die op elke pagina in een website voorkomt, worden best opgemaakt als een eenheid. Vaak zijn navigatiebalken de eerste delen van een pagina die ingeladen worden. Gebruikers die surfen met een screenreader, horen dan telkens dezelfde reeks links voor ze bij de inhoud van een pagina komen. Je kan daarom bovenaan elke pagina een link voorzien waarmee de navigatie kan worden overgeslaan en de gebruiker dadelijk bij de inhoud van de pagina komt.

Maak geen gebruik van pop ups of het openen van documenten in een nieuw venster, zonder de gebruiker hiervan op de hoogte te stellen. Indien je gebruik maakt van een pop up, vermeld dit dan bij de linktekst. Zorg er ook voor dat het pop up venster schaalbaar is, zodat de gebruiker het indien gewenst kan vergroten.

Indien je linkt naar andere documenten dan webpagina's, vermeld dan steeds het formaat van het document (zoals Word of PDF) en eventueel de grootte van het document. Zorg er ook voor dat indien er een plugin nodig is (zoals QuickTime of Acrobat Reader) er ook wordt verwezen naar waar die plugin kan gedownload worden. Niet iedereen heeft al deze plugins standaard op zijn pc geïnstalleerd of wenst die te installeren.

Met event handler bedoelt men een script dat wordt uitgevoerd wanneer een bepaalde event of handeling gebeurt (zoals wanneer de muis beweegt, een toets wordt ingedrukt, een document geladen, ...). Je hebt event handlers die, eenmaal uitgevoerd, enkel de opmaak veranderen of een ander decoratief effect hebben, deze vormen geen probleem. Maar daarnaast zijn er ook die meer substantiële effecten hebben zoals extra informatie aan de gebruiker geven in de vorm van een subnavigatie, of een formulier doorsturen. Zorg in deze

gevallen dat er ook een alternatieve manier is om deze events of informatie te bekomen.

Zorg dat elke interactie op een webpagina kan gebeuren, onafhankelijk van de manier waarop de gebruiker een site raadpleegt. Niet iedereen bekijkt websites op een visuele manier of kan/wil gebruik maken van een muis.

Met het gebruik van frames verdeel je een webpagina onder in meerdere subframes of paginaonderdelen. In elke frame kan een pagina of document worden geladen en de verschillende frames werken los van elkaar. Voor niet-visuele gebruikers is het onderlinge verband tussen frames niet evident. Screenreaders kunnen de frames enkel apart van elkaar weergeven of een lijst opsommen van de links naar elke frame. Daarom is het belangrijk dat elk frame een betekenisvolle en duidelijke naam en titel heeft.

## 3.4 Interactie

### 3.4.1 Formulieren, Java(JavaScript) en Flash

Plaats in een formulier een label bij elke control (zoals de radiobutton of checkbox), zo is het voor iedereen duidelijk welke control bij welk label hoort. Je kan dan ook op de control klikken of op het label om de control te activeren. Deze extra informatie over hoe het formulier te besturen, helpt gebruikers bij het navigeren en begrijpen van complexe formulieren. Let erop dat het label ofwel onmiddellijk na de control, op dezelfde regel wordt geplaatst, ofwel onmiddellijk erna op de volgende regel, zonder verwarring te scheppen over welk label bij welke control hoort.

Gebruik `<fieldset>` om een aantal (bvb. thematisch) gerelateerde controls van een formulier te groeperen. Hierdoor wordt rond deze fieldset een kader geplaatst. Gebruik het `<legend>` element om een bijschrift bij deze groep te plaatsen.

Het `<optgroup>` element gebruik je in een formulier om keuzemogelijkheden in een keuzelijst op een logische wijze te groeperen. Vooral bij lange lijsten biedt dat de gebruiker meer gemak. Met het "label" attribuut, dat aan elke optgroup element moet worden toegevoegd, krijgt elke groep een naam.

Met het attribuut `tabindex` kan je bepalen in welke volgorde links en formulierelden elkaar opvolgen in een webpagina. Indien geen volgorde is bepaald, houdt de browser zich aan de volgorde in de HTML-code.

Wanneer je een object of applet gebruikt op je website, zorg er dan voor dat de besturing ervan toegankelijk is.

Via scripts kan men onderdelen van een pagina veranderen. Zorg er in dat geval voor dat het toegankelijk blijft. Als het dynamische onderdeel bvb. een steeds wisselende afbeelding is, zorg er dan voor dat de alt-tekst mee verandert. Indien mogelijk, gebruik hiervoor dan server side scripts (zoals ASP of PHP). Als het niet mogelijk is de dynamische webpagina toegankelijk te maken, zorg dan voor een vlot toegankelijke alternatieve pagina.