
Huishoudelijk Reglement Webvakken

Dit huishoudelijk reglement is van toepassing voor het academiejaar 2011 – 2012. Het geldt voor volgende vakken:

- *Webtechnieken 1*
- *Webtechnieken 2*
- *Webtechnieken 3*
- *Rich Internet Applications deel Javascript*
- *Webprogrammatie (uitzondering: bij conflict krijgen de Visual Studio stijlregels voorrang op de stijlregels uit dit document)*

Voor vakken of individuele opdrachten kunnen eventueel nog afwijkingen of bijkomende regels opgelegd worden.

Bram Van Damme, Frank Haelman, Davy De Winne & Rogier van der Linde
Vakgroep ICT, dept. Gent, KaHo St Lieven

1 Algemene bepalingen

1.1 Plagiaat

De definitie van de KU Leuven een goed uitgangspunt:

“Plagiaat is elke overname van het werk (ideeën, teksten, structuren, beelden, plannen...) van zichzelf of van anderen, op identieke wijze of in licht gewijzigde vorm zonder adequate bronvermelding.”

We zullen nu verder specificeren wat mag en wat niet mag voor code, maar het geldt evengoed voor wireframes, design, afbeeldingen en al het andere materiaal dat in het kader van een opdracht geproduceerd wordt. Actieve plagiaat wordt gepleegd door de student die kopieert, passieve plagiaat wordt gepleegd door de student die laat kopiëren. Beiden worden aanzien als fraude en zijn strafbaar.

Tenzij anders vermeld in de opgave, mag je elkaar helpen bij het uitwerken van opdrachten, met uitzondering van examens en individuele labo's waar elke vorm van samenwerking verboden is. Er zijn wel strenge beperkingen op de manier waarop je elkaar mag helpen; lees daarom het volgende goed, of je riskeert beschuldigd te worden van plagiaat.

Wat mag wel:

- 1.1.1 Je mag een medestudent helpen een lastige bug te vinden en daarvoor zijn code bekijken.
- 1.1.2 Je mag met een medestudent strategieën bespreken voor het oplossen van een beperkt deelprobleem van de opgave. De volledige oplossingsstrategie samen opstellen of delen mag dus niet.
- 1.1.3 Je mag links uitwisselen van online tutorials en andere bronnen die kunnen helpen bij het uitwerken van de oefening.
- 1.1.4 Je mag vrij beschikbare codebibliotheken en –fragmenten gebruiken voor het oplossen van een beperkt deel van de opgave mits duidelijke bronvermelding. Voorbeeld 1: je gebruikt PHPmailer voor het versturen van een mail in een groot PHP project: dit mag. Voorbeeld 2: je moet een Sudoku programmeren en gebruikt daarvoor de oplossingsstrategie van een site waarin dit volledig uitgelegd staat: dit mag niet.

Wat mag nooit:

- 1.1.5 Je mag nooit codefragmenten letterlijk doorgeven of van elkaar overnemen, hoe klein ook. Samen code ontwikkelen is eveneens verboden.
- 1.1.6 Je mag nooit codefragmenten gebruiken zonder toestemming van de auteur of zonder bronvermelding.
- 1.1.7 Je mag nooit grote codefragmenten aan anderen doorgeven of voor anderen beschikbaar maken, via een gedeelte locatie, door online publicatie enz... tenzij expliciet gevraagd wordt je oplossing online te zetten, in welk geval niet meer dan het strikt noodzakelijke gepubliceerd mag worden. Een bijkomende link naar een zip met de oplossing mag dus niet. Een klein

fragment op het studentenforum posten om hulp te krijgen mag wel, maar een hele pagina code posten dan weer niet.

Dit laatste betekent dus ook dat je verantwoordelijk bent voor de bescherming van je bestanden. Als iemand bijvoorbeeld jouw oplossing van je publieke Dropbox folder kopieert ben je in fout, ook al was het zonder jouw medeweten of toestemming. Als ICT-er moet je nu eenmaal gevoelige informatie als een goede huisvader kunnen bewaren. Voor elk fraudegeval stel je je bloot aan sancties; in regel zal een dossier opgesteld en overgemaakt worden aan de betrokkenen (minstens de student zelf, de ombudsdienst, de voorzitter van de examencommissie, het opleidingshoofd en de trajectbegeleider).

1.2 Uploaden op Toledo

Meestal zal gevraagd worden je werk op Toledo te uploaden, soms op het einde van een labo, soms vanaf thuis vóór het verstrijken van een opgegeven deadline. Let goed op volgende regels:

- 1.2.1 Deadlines van thuis af te werken opdrachten verlopen om 20u 's avonds
- 1.2.2 Deadlines zijn absoluut; daar wordt onder geen voorwaarde van afgeweken. Te laat geuploade werken worden niet aanvaard.
- 1.2.3 Maak tijdens het werken voldoende backups; een crash van welke aard ook is geen excuus.
- 1.2.4 Uploads gebeuren onder de vorm van één enkele zip; meerdere bestanden per upload, een rar of andere compressieformaten worden niet aanvaard.
- 1.2.5 Controleer na het uploaden nog eens of bijvoorbeeld de zip niet corrupt is en of alle bestanden in de zip aanwezig zijn door het zelf te downloaden en te controleren
- 1.2.6 Mocht er een probleem opduiken bij het uploaden – bijvoorbeeld als Toledo uitvalt of ingeval een corrupte zip – stuur dan een mail naar de betrokken docent vóór het verstrijken van de deadline. De docent zal je dan antwoorden wat je verder moet doen.
- 1.2.7 Upload niet meer dan nodig: steek er niet ongevraagd de opgave of andere versies bij. Je oplossing moet in de root van de zip zitten, dus niet eerst in allerlei submappen.

1.3 Afwezigheden

Voor de theorielessen gelden volgende afspraken:

- 1.3.1 Aanwezigheid op theorielessen is, tenzij anders vermeld, in principe niet verplicht. Het is wel sterk aangeraden aanwezig te zijn, omdat de labo's en opdrachten erop voortbouwen.
- 1.3.2 Hoewel niet verplicht, leg je bij afwezigheid op een theorieles toch best een doktersbriefje voor aan de docent. Er zullen regelmatig afwezigheden opgenomen worden, en indien tijdens het labo de docent oordeelt dat vragen voortkomen uit je ongewettigde afwezigheid tijdens de theorieles(sen), kan je uitleg geweigerd worden.
- 1.3.3 Louter aanwezigheid op een theorieles is niet voldoende: je moet telkens de presentaties, demo's, je eigen nota's enz... uit de theorieles grondig bestudeerd hebben en de belangrijkste begrippen en technieken kennen voor je naar het eerstvolgende labo komt. Het lesmateriaal

uit de theorielessen gebruiken tijdens de labo's mag, het tijdens de labo's bestuderen mag niet.

Voor de labo's gelden volgende afspraken:

- 1.3.4 Aanwezigheid op labo's is verplicht, tenzij anders vermeld.
- 1.3.5 Was je afwezig op een labo, leg dan een doktersbriefje voor aan je labo docent tijdens het eerstvolgende labo. Laattijdige doktersbriefjes worden niet aanvaard.
- 1.3.6 Was je ongewettigd afwezig, dan krijg je een 0 voor dat labo.
- 1.3.7 Was je gewettigd afwezig, dan moet je het gemiste labo eerst inhalen voor je aan de volgende labo opgave mag beginnen, tenzij de labo docent oordeelt dat inhalen niet nodig is en je meteen met de nieuwe opgave mag beginnen. Contacteer spontaan je labo docent om een inhaalmoment af te spreken.

Projecten worden gevolgd door een mondelinge verdediging waarop de punten vastgelegd worden. Daarvoor gelden volgende afspraken:

- 1.3.8 Aanwezigheid op de verdediging van een project is verplicht, en dit tot de docent aangeeft dat het mondeling voorbij is. Zoniet zal een A genoteerd worden en worden geen punten toegekend.

2 Huisstijl

In een team of in een bedrijf is het belangrijk dat iedereen dezelfde programmeerstijl hanteert om de code leesbaar te houden. Voor pas afgestudeerden, die veelal gewend zijn alleen te werken, vraagt dit vaak een hele aanpassing. Daarom dat we best nu al oefenen in het lezen van en leren werken met een huisstijl.

De stijlregels waar wij voor kiezen zijn niet universeel; in andere vakken kunnen andere regels gelden. Deze regels zijn ook niet perse 'de beste'; het belangrijkste is *dat* er keuzes gemaakt worden die voor iedereen gelijk zijn, en dat je leert je daar aan te houden.

Op het eerste zicht lijkt dit document zeer lijvig. Laat je hier niet door afschrikken, de meeste regels zijn gezond verstand en zullen na verloop van tijd een reflex worden.

De voorbeelden komen uit alle vakken waarvoor dit huishoudelijk reglement geldt; enkel waar verwarring kan bestaan wordt de taal expliciet vermeld (HTML, Javascript, CSS of PHP). Voor het vak webprogrammatie gelden deze regels als aanvulling op de Visual Studio stijlregels; bij conflicten: behoud je de Visual Studio stijlregels.

Bemerkt dat de gegeven codevoorbeelden bij de presentaties/labo's soms (nog) niet aan deze regels voldoen, ze worden nog volop omgevormd om de stijl te volgen. In de codevoorbeelden in de presentaties zelf wordt soms bewust tegen de regels gezondigd om plaats te besparen of om educatieve redenen. Bij copy-pasten van de code uit de slides zal je vaak de codelay-out moeten herzien.

2.1 Bestanden en mappen

2.1.1 Om bestanden te comprimeren is alleen .zip toegelaten, dus geen .rar, .7z of andere

2.1.2 Alle namen van bestanden en mappen moeten

- in het Engels
- met underscore _notatie, kleine letters en cijfers, beginnen met een letter

```
goed
```

```
home.html, shopping_step2.html, mycollections_add.html, includes
```

2.1.3 Structuur HTML & Javascript projecten:

```
.
|-- css
|   |-- reset.css
|   |-- common.css
|-- img
|   |-- icons
|   |   |-- edit.png
|   |   |-- help.gif
|   |-- logo.jpg
|-- js
|   |-- common.js
|   |-- scripts.js
|-- about.html
|-- index.html
```

2.1.4 Structuur PHP projecten:

```
.
|-- dev
|   |-- logo.psd
|-- www
|   |-- core
|       |-- css
|           |-- styles.css
|       |-- img
|           |-- background.gif
|           |-- logo.jpg
|       |-- includes
|           |-- classes
|               |-- myclass.php
|           |-- functions.php
|           |-- includes.php
|       |-- js
|       |-- layout
|           |-- about.tpl
|           |-- index.tpl
|   |-- files
|       |-- avatar.jpg
|       |-- document.pdf
|   |-- library
|       |-- plonk
|       |-- phpmailer
|   |-- modules // elke module in eigen submap met css, img, js en layout
|   |-- about.php
|   |-- index.php
|-- db.sql
|-- readme.txt
```

2.2 Naamgevingen

2.2.1 Alle namen van functies, variabelen, classes, (HTML) id- en class-attributen e.d. in het Engels

2.2.2 Gebruik camelCase notatie

2.2.3 Classes beginnen met een hoofdletter, al de rest begint met een kleine letter

2.2.4 Gebruik voldoende beschrijvende namen; uitzondering: tellers in loops *i*, *j* (PHP) *\$x*, *\$y*

goed

MyMailerClass, rightColumn, btnOk, \$maxNumberOfStudents, fetchArticlesForDate()

2.3 Indentatie

2.3.1 Spring in met één tab per keer, geen spaties; controleer dit door ¶ aan te zetten

2.3.2 HTML: begin- en eindtag moeten met inhoud ofwel op één regel staan, ofwel onder elkaar op gelijke afstand ingesprongen

fout

```
<div><p>Lorem ipsum dolor sid amet...<p><p>sed do eiusmod...</p>
</div>
```

```
<div>
  <p>Lorem ipsum dolor sid amet...<p><p>sed do eiusmod...</p></div>
```

goed

```
<div><p>Lorem ipsum dolor sid amet...<p><p>sed do eiusmod...</p></div>

<div>
  <p>Lorem ipsum dolor sid amet...<p>
  <p>sed do eiusmod...</p>
</div>

<div>
  <p>
    Lorem ipsum dolor sid amet...
  </p>
  <p>sed do eiusmod...</p>
</div>
```

2.3.3 Accolades: open deze op het einde van de regel, en sluit af verticaal eronder

```
fout

p {
  color: red;
  font-size: 12px;
  border: 1px solid gray;
}

var myFunc = new function() { alert('hello'); }

var myFunc = new function()
{
  alert('hello');
}

for ($x = 0; $x < 10; $x++)
{
  // ...
}
```

goed

```
p {
  color: red;
  font-size: 12px;
  border: 1px solid gray;
}

var myFunc = new function() {
  alert('hello');
}

for ($x = 0; $x < 10; $x++) {
  // ...
}
```

Uitzondering: indien leeg mogen de accolades op één regel, en bij CSS mag alles ook eventueel op één regel

```
goed

var myFunc = new function() {}

try {
  // ...
} catch (Exception e) {}

p { color: red; font-size: 12px; border: 1px solid gray; }
```

2.3.4 Accolades moeten overal geplaatst worden waar ze optioneel zijn

```
fout

for ($x = 0; $x < 10; $x++)
    // ...

if ($loggedIn === true)
    // ...
else
    // ...
```

```
goed

for ($x = 0; $x < 10; $x++) {
    // ...
}

if ($loggedIn === true) {
    // ...
} else {
    // ...
}
```

Uitzondering: een `if` die kort genoeg is mag op één regel

```
goed

if (x % 2 === 0) return 'even';
```

2.4 Spaties en open regels

2.4.1 Wees consequent met open regels; gebruik ze enkel om blokken code van elkaar te scheiden

2.4.2 Zet nooit twee spaties, of een spatie en een tab naast elkaar

```
fout

<input type="hidden"      value="test" />

p.err {
  font-style: italic;
  color:      red;
}
```

```
goed

<input type="hidden" value="test" />

p.err {
  font-style: italic;
  color: red;
}
```

2.4.3 Zet spaties

- na leestekens
- voor en na toewijzingen en operatoren (behalve `++` en `--`)
- na keywords als `if`, `for`, `while`
- na het casten
- na het openen van commentaar

- voor en na de ? en : van de ternaire operator
- voor en na accolades

```
fout

p+a{
  font-size: large;
  color: red;
}

p + a{font-size: large; color: red;}

$username= 'johndoe';
for (var x=0; x<10; x++) {
  // ...
}

if((string)$username === 'johndoe') {
  // ...
}

var hasItems = arrItems.length>0?true:false;

$name = $firstname.' '.$lastname;
```

```
goed

p + a {
  font-size: large;
  color: red;
}

p + a { font-size: large; color: red; }

$username = 'johndoe';
for (var x = 0; x < 10; x++) {
  // ...
}

if ((string) $username === 'johndoe') {
  // ...
}

var hasItems = arrItems.length > 0 ? true : false;

$name = $firstname . ' ' . $lastname;
```

2.4.4 Zet geen spaties

- voor leestekens
- voor of na = in HTML attributen
- onnodig aan het begin en het einde van de inhoud van een HTML element

```
fout

<input type = "hidden" value = "test" />

p , a {
  color : red;
}

<span class="trefwoord"> lorem ipsum </span>
```

```
goed

<input type="hidden" value="test" />

p, a {
    color: red;
}

<span class="trefwoord">lorem ipsum</span>
```

2.5 Commentaar

2.5.1 Alle commentaar staat in het Engels

2.5.2 Voorzie elke pagina bovenaan van een korte beschrijving en auteur, en eventueel bijkomende informatie als versie enz...; gebruik de PHPDoc stijl van commentaar

```
goed

/**
 *
 * Generic stylesheet for forms
 *
 * @author John Doe <john.doe@company.com>
 * @version 2.1
 *
 */
```

2.5.3 Bij HTML pagina's is het formaat iets anders, en moet de doctype bovendien op de eerste regel staan, dus moet de commentaar er net onder:

```
goed

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
    "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<!--

    Solution to exercise 3, lab session 1, Webtechnology 1 (2011)

    @author Rogier van der Linde <rogier.vanderlinde@kahosl.be>

-->
```

2.5.4 Voorzie inline commentaar boven of naast de relevante code

2.5.5 Gebruik blokken commentaar om grote stukken logische code af te scheiden

2.5.6 Indien je externe codefragmenten gebruikt, moeten ze voorzien zijn van een weblink of referentie

```
goed

/**
 *
 * Eric Meyer's famous CSS reset v.2,
 *
 * @see http://meyerweb.com/eric/tools/css/reset/
 *
 */

html, body, div, span, applet, object, iframe, h1, h2, h3, h4, h5, h6, p,
blockquote, pre, ...
```

2.6 Code organisatie

- 2.6.1 Sorteer groepen functies/methodes, variabelen enz... binnen bijvoorbeeld classes alfabetisch oplopend op functienaam; zo vind je ze makkelijker terug

2.7 HTML

- 2.7.1 Gebruik XHTML1.1 of, indien toegelaten, HTML5 (zie verder)

- 2.7.2 Syntaxregels:

- kleine letters voor tagnamen
- kleine letters voor attribuutnamen
- lege elementen (area, br, hr, img, input...) moeten afgesloten zijn; voor de / moet een spatie staan

```
fout  
<br>
```

```
goed  
<br />
```

- rond attribuutwaarden moeten dubbele quotes staan

```
fout  
<textarea cols=40 rows=8>
```

```
goed  
<textarea cols="40" rows="8">
```

- 2.7.3 Basisdocument:

- doctype aanwezig
- overall relevante <title> (dus niet in de hele site identiek)
- metatag met karakterset aanwezig
- CSS en HTML moeten valideren bij het W3C
- moet werken in IE8+ en de laatste versies van FF, Chrome, Safari, Opera

- 2.7.4 Elementen:

- geen <frame> of <iframe>
- geen tabellen voor layout
- geen presentationele elementen als , <center> enz... (voor de volledige lijst zie appendices)
- liefst geen image maps
- geen twee
's na elkaar
- geen twee 's na elkaar
- gebruik of <dl> voor menu's, ook voor kleine menu's zoals taalkeuze
- alle speciale karakters moeten gecodeerd worden
- enkel voor inhoudelijke afbeeldingen (reclame, foto's bij artikels enz...)
- links die nog niet gekend zijn, verwijst je naar "#"
- correct gebruik alt bij en for attribuut bij <label>

- inline elementen mogen niet rechtstreeks in `<div />`'s gebruikt voor layout, `<form>` of `<body>` zitten, maar moeten in blocklevel elementen gestructureerd zijn, bv

```
fout

<form><input /></form>
goed

<form><p><input /></p></form>

<form><fieldset><input /></fieldset></form>
```

- gebruik `<h1>` voor de titel van de site, `<h2>` voor de titel van de pagina, `<h3>` voor de onderverdeling van de pagina
- ergonomieregels: zie appendix I uit Webtechnieken 1 en handboek 'Webdesign & Usability' van F. Haelman

2.7.5 Gebruik karaktercodering en de juiste typografie

```
fout

<p>Geschiedenis van België in 1914-1918</p>
```

```
goed

<p>Geschiedenis van België&uum; in 1914&ndash;1918</p>
```

2.7.6 Geen attribuut minimalisatie

```
fout

<input checked />
```

```
goed

<input checked="checked" />
```

2.8 HTML5

Deze regels zijn aanpassingen en aanvullingen van de regels voor HTML.

- 2.8.1 Niet alle nieuwe mogelijkheden van HTML5 zullen in alle browsers werken; in alle browsers moeten toegankelijkheid, layout en gebruiksgemak bewaard blijven, maar op het vlak van presentatie (b.v. tabel of grafiek), design (b.v. rechte of afgeronde hoeken) en gebruikservaring (b.v. animaties of niet) mogen kleine te rechtvaardigen verschillen bestaan
- 2.8.2 Hoewel HTML5 ook een lossere syntax toestaat, beperken wij ons tot de XHTML syntax, met één uitzondering: attribuut minimalisatie is toegestaan

```
goed (HTML5)

<input required />
```

- 2.8.3 Basisdocument:
 - HTML5 doctype aanwezig
 - geen xmlns aanwezig

- IE shiv aanwezig
- vereenvoudigde versies voor declaratie content type, style en script blokken

2.8.4 Elementen:

- gebruik `<header>` voor inleidende inhoud of navigatie van de eigenlijke content
- gebruik `<footer>` voor meta-informatie over de eigenlijke content: copyright, auteur, datum...
- gebruik `<aside>` voor niet-essentiële inhoud zijlings gerelateerd aan de content: interessante links, weetjes, bijkomende uitleg...
- gebruik `<nav>` enkel voor het hoofdmenu, niet voor andere menu's
- gebruik `<article>` voor content die op zichzelf moet kunnen staan, bijvoorbeeld in RSS feeds of als gastcontent in andere sites
- gebruik `<section>` om gerelateerde content te groeperen (alle vorige elementen zijn speciale gevallen van dit element)
- gebruik `<div>` als technisch hulpmiddel: site wrapper, Javascript dropzone...
- herstart de nummering van headers (`<h1>`, `<h2>` enz...) enkel binnen een `<article>` en enkel als dit element ook effectief in een andere toepassing (RSS, andere site...) gebruikt wordt

2.9 CSS

- 2.9.1 Inline styles zijn altijd verboden
- 2.9.2 Interne stylesheets zijn enkel voor pagina-specifieke styles
- 2.9.3 Kleine letters voor tag selectors en properties
- 2.9.4 Vermijd het gebruik van hacks. Gebruik indien nodig conditionals
- 2.9.5 Gebruik de Phark methode voor image replacement
- 2.9.6 Gebruik de shorthand notatie van properties indien je hiermee plaats bespaart

2.10 Javascript

- 2.10.1 Inline scripts zijn altijd verboden
- 2.10.2 Interne scripts zijn enkel voor pagina-specifieke scripts
- 2.10.3 Gebruik de objectnotatie om functies aan te maken:

```
fout

function doSomething() {
    // ...
}
```

```
goed

var doSomething = function() {
    // ...
}
```

2.10.4 Sluit elke statement af met een puntkomma

2.11 PHP

2.11.1 Ontwikkel met `E_ALL` als `error_reporting` level en `display_errors` op "on" ingesteld

2.11.2 Gebruik steeds de volledige php open tag `<?php`

2.11.3 De sluittag `?>` mag je vervangen door `// EOF` indien je PHP bestand enkel PHP omvat

2.11.4 Strings bakenen we af tussen enkele quotes

```
fout
$username = "johndoe";
```

```
goed
$username = 'johndoe';
```

2.11.5 Gebruik identical operators bij vergelijkingen

```
fout
if ($numItems == 0)
```

```
goed
if ($numItems === 0)
```

2.11.6 Cast variabelen om domme fouten te vermijden

```
fout
$pricePerItem = $totalPrice / $quantity;
```

```
goed
$pricePerItem = (double) $totalPrice / (int) $quantity;
```

2.11.7 Gebruik nooit `$_REQUEST`

2.11.8 Gebruik nooit het `global` keyword maar voorzie constanten, defines of extra parameters

2.11.9 Indien de variabelen voor het ganse project geldig zijn, plaats dan de `define` in `config.php`

2.12 Functies/methodes

2.12.1 Sorteer functies/methodes alfabetisch op functienaam, bijvoorbeeld binnen de klasse waartoe ze behoren