

Evaluatiecriteria

Front End Development, november 2022

De vereisten van de opdracht bij de uitwerking van je website vind je op [Toledo > Opdrachten > Statische Website HTML en CSS](#) (o.a. omvang, inhoud, doelstellingen, ...). Deze tekst staat enkel stil bij technische vereisten.

Hieronder vinden jullie een checklist om zelf stil te staan bij de uitwerking van de website. Wat deed je al goed? Wat kan beter?

1. Validatie

- Ieder HTML bestand heeft een HTML5 declaratie.
 - eerste regel bestand moet zijn: `<!DOCTYPE html>` dit verwijst naar gebruik html5
- Ieder HTML bestand valideert volgens de HTML5-doctype op <https://validator.w3.org/> of <https://html5.validator.nu/>.
 - lees de foutmeldingen (errors, aangeduid in rood) goed na. Die geven aan welke fout je maakte en vaak ook een hint om ze op te lossen.
 - gebruik Google map geeft problemen: zie voor oplossing op [documentatie 'responsive Google Map'](#)
 - bij gebruik van `iframe` voor map moet je frameborder-attribuut volledig verwijderen om te kunnen valideren, o.a. voor Google Map, ...
 - images:
 - attribuut `alt=""` niet vergeten toe te voegen
 - lege images (met `src=""`) valideren niet
 - afbeelding in een lijst: ofwel `img` element omgeven door `li`, ofwel `img` buiten de lijst plaatsen
 - `header` geplaatst in `head` i.p.v. in `body`
 - gebruik onbestaande elementen zoals `p1`, ...
 - warnings (geel) zijn toegestaan, probeer ze tot een minimum te herleiden
- Het CSS-bestand valideert op <https://jigsaw.w3.org/css-validator/>.
 - lees de foutmeldingen goed na. Die geven aan welke fout je maakte en vaak ook een hint om ze op te lossen.
 - let op overbodige of niet toegelaten spaties bvb. bij gebruik van eenheden (`margin: 1.2rem` en niet `margin:1.2 rem`);
 - gebruik correcte 'properties' en 'values', bvb. `flex:right` bestaat niet wel `display:flex` en `align-content:flex-end`;

2. Bestandsstructuur

- Alle (S)CSS-bestanden zijn gegroepeerd in één map (eventueel met submappen)
 - je gebruikt SCSS om CSS te genereren
 - dit resulteert in één *eigen* CSS-bestand, dus geen apart bestand per html-pagina;
- Afbeeldingen zijn gegroepeerd in de map "img", "images" of gelijkaardig.
- Links naar CSS en afbeeldingen vanuit het HTML-bestand zijn relatief.
 - dus bv. niet: file:///Users/krmel/web1/img/logo.png maar wel `img/logo.png`;

3. HTML - Documentstructuur

- HTML5 structuurelementen zijn aanwezig (header, nav, main, footer, article, section, aside, address, ...). Veel voorkomende fouten:
 - je gebruikt geen `main` element
 - je gebruikt geen of amper de elementen `article` en `section`
 - je gebruikt `div` onoordeelkundig (in plaats van `article` of `section`) m.a.w. geen gebruik van `div` als andere elementen toepasbaar zijn, zie [html5doctor sectioning flowchart](#)
- Header bevat `h1` met naam van de site en `nav`.
 - Veel voorkomende fouten:
 - `header` bevat `ul` met links in `header` zonder `nav` element
 - `header` bevat slechts 1 van beide, dit kan als `h1` wél in de `header` zit
 - `nav` zeker gebruiken
- De navigatie is een lijst (`ul`).
 - opsomming van louter links is niet OK omwille van toegankelijkheid
- Ieder sectioning element heeft een hoofding (`h2`, `h3`, ...).
 - hiërarchie volgen: h1=titel/onderwerp gehele website, h2=paginatitel, h3=subtitel van de pagina, ...
 - geen niveau's overslaan bvb. van `h1` naar `h4`
 - als niet elk sectioning element een hoofding heeft, geeft dit een 'warning' géén error. Kijk na of je voldoende en duidelijke hoofdingen gebruikt in verband met 'usability' en 'accessibility'
- `section` en `article` zijn op de juiste manier gebruikt.
 - beiden groeperen inhoud die bij mekaar hoort
 - `article` staat op zichzelf en zou je opnemen in een inhoudstafel, kan ook een `header` en een `footer` bevatten om aan te duiden wanneer en door wie geschreven
 - gebruik `div` enkel om inhoudten te groeperen voor stijldoeleinden en niet omwille van inhoudelijke samenhang
- Structuurelementen worden niet gebruikt voor lay-out
 - als je de stijlen 'uit' zet, bvb. via Web Developer Toolbar - CSS - disable all styles, dan moet de structuur blijken uit de pagina.
 - tabellen
 - gebruik tabellen niet voor layout
 - niet alle inhoud is geschikt voor tabel, bijvoorbeeld als er geen terugkerende titels zijn
 - indien je een tabel gebruikt in een toegelaten context, zorg voor maximale toegankelijkheid ([Web Accessibility Tutorials - tables concepts](#))
 - vraag je af of het een optie is om een lijst te gebruiken i.p.v. een tabel

4. HTML - Scheiding inhoud en structuur

- Er wordt geen `br` gebruikt om witruimte te creëren.
 - gebruik `margin` en `padding`
- Tekst is ondervverdeeld in paragrafen.
 - hangt soms samen met te veel gebruik `br` en te weinig `p`
 - zorg voor voldoende tekst en dus voldoende `p`

5. HTML - Hyperlinks

- Links binnen de site zijn relatief, niet absoluut.
- Alle links werken.
 - Veel voorkomende fouten
 - benaming niet consistent tussen link en eigenlijk bestand (overOns.html versus overons.html)
 - fout pad
 - je laptop is niet hoofdlettergevoelig, maar de webserver is dat wel.

6. HTML - Afbeeldingen

- Indien nodig, zijn afbeeldingen geplaatst in het element figure.
- De laadsnelheid is klein:
 - afbeeldingen worden opgeslagen in de maximale afmeting dat ze gebruikt worden op de website (voor de mobile version én desktop version), bvb. Facebook logo 600px x 400px te groot.
 - afbeeldingen worden voldoende gecomprimeerd.
- Er is een zinvol alt attribuut bij de afbeeldingen gebruikt
 - alt="" bij puur decoratieve foto's/afbeeldingen
 - alt="afbeelding van aantal inschrijvingen opleiding Toegepaste Informatica AJ 2020-2021 (UCLL)" ==> 'afbeelding van' weglaten
 - alt="logo bedrijf" ==> alt="Rode Kruis, helpt helpen"
- Links naar CSS en afbeeldingen vanuit het HTML-bestand zijn relatief.
 - dus bv. niet: file:///Users/krmel/web1/img/logo.png maar wel `img/logo.png`;

7. Layout

- Aangepast kleurgebruik.
 - gebruik een [goede notatie](#), zie bvb. <http://hslpicker.com/>: niet de kleurnaam zoals 'white', wel bvb. #000 (hexadecimaal), of rgb(255, 255, 255) (rgb) of hsl(360, 100%, 100%), en rgba(255, 255, 255, 1) of hsla(360, 100%, 100%, 1) (hsla) om het alfabetaal te gebruiken voor 'opacity'-effect
 - beperk het aantal gebruikte kleuren, stel je kleurenpalet bvb. samen met [Adobe kleurenwiel](#) door het wiel te gebruiken, een bestaand thema te gebruiken of een kleurenpalet samen te stellen aan de hand van een eigen foto/logo, of specifieke kleur
- Aangepaste lettertypes en -groottes en line-height.
 - kies enkele lettertypes die goed samengaan
 - laat je inspireren op [Google Fonts](#), en [FontSquirrel](#)
 - eindig `font-family` steeds met serif of non-serif;
- Voldoende contrast, opletten met teksten op donkere vlakken of op afbeeldingen, gebruik de [Colour Contrast Analyzer](#)
- Correcte allignering: opgepast met centreren (beter te weinig dan te veel!) Zorg voor duidelijke horizontale en verticale lijnen
- Zorg dat inhoud die bij mekaar hoort ook visueel getoond wordt als een geheel dat samenhoort (vormt samen met voorgaande items [CRAP](#) principe).
- Afbeeldingen: [bewerk](#) je afbeeldingen zodat ze niet te groot zijn in pixels noch in kB.

8. CSS - SCSS

- Je gebruikt SCSS om CSS te genereren.
- Je SCSS bestanden staan op de server in dezelfde map als je CSS bestand(en).
- SCSS is opgebouwd uit componenten die een logische structuur hebben.
- De bijhorende CSS staat in partials of deelbestanden (bijv. `_header.scss`) die georganiseerd zijn in mappen (directories).
- Je gebruikt nesting.
- Je gebruikt variabelen.

9. CSS - Positionering

- Je gebruikt de juiste techniek op de juiste plaats.
- `Flex` wordt correct gebruikt.
 - 1D (ééndimensionaal), dus gebruik voor navigatie, maar bvb. ook voor `footer`
 - (eerder) niet gebruiken voor volledige pagina layout
- `Grid` wordt correct gebruikt.
 - 2D (tweedimensionaal), dus gebruiken voor volledige pagina layout (hoofdcomponenten) én 1 niveau lager
 - gebruik de mogelijkheden van grid, werk waar mogelijk met gap i.p.v. margin/padding
- Tekstblokken hebben geen vaste hoogte.
- `Position:absolute` wordt beperkt gebruikt of helemaal niet.
- `Position:absolute` wordt gebruikt in combinatie met position:relative van een ouder-element (op die manier wordt er absoluut gepositioneerd t.o.v. die ouder)
- Je gebruikt een normalize.css (optioneel).
 - bijvoorbeeld [normalize.css](#)
 - specificiteit: eerst de normalize.css invoegen, pas daarna eigen stijlen!

10. CSS - Selectoren

- CSS code wordt zo weinig mogelijk herhaald (DRY).
 - bvb. door default styles te definiëren voor `body`, door selectoren te groeperen, ...
 - CSS-regels worden zo algemeen mogelijk opgesteld, bijv. `padding` voor de omhullende div i.p.v. `margin` voor elke afzonderlijke `p`.
- Goed leesbare naamgeving van id en klasse-attributen.
 - gebruik `id` en `class` met mate
 - naamgeving verwijst best naar het beoogde effect van de klasse, (bvb. `class="right"` voor alle afbeeldingen die rechts komen te staan), dus niet `class="foto1"`
- Als er een groep van elementen is die dezelfde CSS-eigenschappen hebben, wordt een groepselector gebruikt (bijv. `h1,h2 { ...}`).

11. RWD

- In het HTML-bestand staat in de head het element `<meta name="viewport" content="width=device-width, initial-scale=1">`.
- Het aantal breekpunten is beperkt.
 - 3 tot 5 breekpunten uitwerken, het hoeven er géén 10 te zijn
- De breekpunten zijn goed gekozen.
- Werk bij voorkeur mobile first, dus `@media ...` and (`min-width:...`)
- Test in Firefox met Developer toolbar>Resize-View responsive layouts (of gelijkaardige tools) geeft goede resultaten.
- Test met eigen smartphone/tablet geeft goede resultaten.
- Indien nodig, verdwijnt inhoud. Er verdwijnt niet te veel inhoud.
- Breedtes van blokken zijn (meestal) relatief gedefinieerd (indien nodig).
 - gebruik `rem`, % of `fr` (grid) voor breedtes van sections, articles, paragrafen;
 - voor margins en paddings kan gebruik je ook relatieve eenheden bvb. waarden uitgedrukt in `rem` (of in %);
- Breedtes van afbeeldingen zijn relatief gedefinieerd (indien nodig).
 - zet bvb. de breedte van je afbeelding t.o.v. de container met `max-width:100%`; en/of `width:100%`; en zorg voor proportionele afbeeldingen met `height:auto`;
- Na het opslaan in de juiste afmetingen werden de afbeeldingen gecomprimeerd en geoptimaliseerd, bvb. via [Cloudinary](#)

12. Toegankelijkheid

- De lay-out is goed in Edge, Firefox, Safari en Chrome.
- De fontsize is relatief gedefinieerd.
 - geen `cm`, `mm` of `pt` gebruiken, maar liever ook geen `px`
- Je website doormaakt minimaal de 15 criteria van de [Quickscan van Anysurfer](#) (onder de kop "Hoe scoorden de websites?"), of beter nog de [checklist van WebAIM](#)

13. Bruikbaarheid

- Elke pagina heeft een specifieke `title`, `h1`, en `h2`
 - `title`: verwijzing naar onderwerp EN/OF specifieke pagina ontbreekt
 - `h1`: geeft onderwerp website weer
 - `h2`: geeft paginatitel weer, link met huidige menu-item
- De laadsnelheid is in orde.
- Optimaal schermgebruik bij alle resoluties (zorg o.a. voor mediaqueries die inhoudsblokken anders plaatsen naargelang de scherm breedte)
- Duidelijke navigatie
 - De huidige pagina wordt gemarkeerd in het menu. Voeg bvb. een class="current" toe om de huidige pagina aan te duiden en overeenkomstig te stylen in CSS;
 - Menu-items: duidelijke benaming en goedgekozen volgorde van de menu-items
- Duidelijke links:
 - duidelijke linkbeschrijving (dus niet "klik hier")
 - duidelijk te onderscheiden vormgeving
- Degelijke inhoud:
 - relevante inhoud
 - schrijven voor het web: Keep It Short and Simple (KISS), denk aan je doelpubliek, ...
 - scanbare weergave: voldoende karakters op paragrafen, lees met lijsten of met benadrukken van woorden via `strong` en `em`, max. 60-70 karakters op een leesregel, ...
 - belangrijkste bovenaan
- Maak gebruik van conventies: logo, ID, navigatie, ...
- Zorg voor consistentie doorheen je website

14. Formulier

- Is zinvol en relevant voor doelpubliek, houdt rekening met aspecten van 'usability'
- Heeft voldoende complexiteit: minstens 3 verschillende `input` types, naast type 'submit'
- Heeft aangepaste layout (via css) met speciale aandacht voor 'mobiele' weergave
- Client-side validatie
 - gebruik attribuut `required`
 - gebruik correct `input type`
- Is toegankelijk, zie [WAI Tutorial - forms](#), en ook:
 - [labels en formulievelden zijn verbonden](#)
 - verplichte velden zijn aangekondigd binnen label, en attribuut `required` wordt gebruikt

15. En verder

- Consistente bestandsnamen volgens afspraak
 - geen spaties of speciale tekens in de bestandsnamen:
 - niet 'over_ons.html', wel overOns.html, over_ons.html of over-ons.html
 - niet q&a.html, wel q&A.html, q_and_a.html of q-and-a.html
 - steeds kleine letters voor bestanden: logo.png en niet Logo.png of logo.PNG of DSC1234.png, ...
 - idem voor mappen: map images en niet Images
 - index.html (niet home.html, Index.html, indexWasserij.html, ...)
- Webpagina is performant op vlak van snelheid, toegankelijkheid, ...: [Lighthouse op Google Chrome](#) (Element inspecteren > Audit) geeft onmiddellijk informatie over de volledige website, ook accessibility