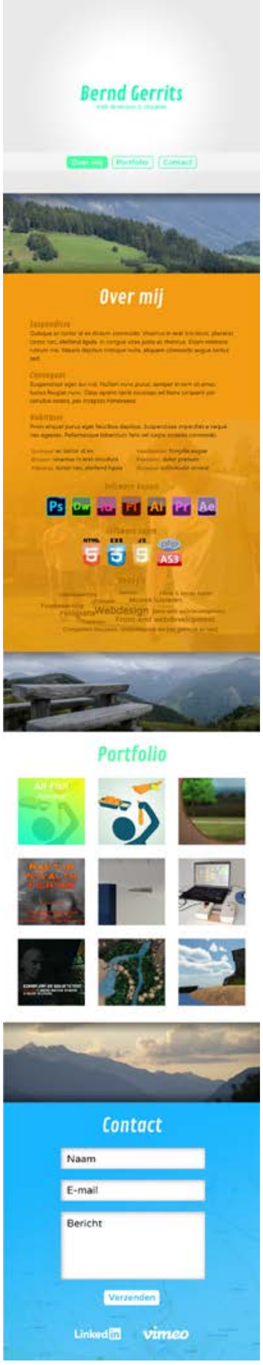
Minor Skilled report

Portfolio website with Wordpress



Bernd Gerrits
350076
MSKLDe

Minor skilled report

Portfolio website with Wordpress

Name:	Bernd Gerrits
Student number:	350076
Class:	MSKLDe
Education:	Kunst & Techniek
Minor:	Skilled
Course leader:	E.H.A. Jannink
Tutor:	W.G.A. Schipper
Study coach:	B.R. Heijnen

Index

1 Plan

1.1 Proposal

The first plan I proposed at the very start of the Minor was building my own content management system. This because it would cover a lot of the facets I wanted to learn about including the inner workings of a CMS.

1.2 The plan

Eventually the plan became creating a portfolio website with WordPress. I was already set on building a new theme in my spare time because my portfolio website at that time was using a third party theme. Which doesn't look very professional if you're selling yourself as a webdeveloper / webdesigner. Combining the Minor and that idea gave me a nice project to work on and learn a lot alongside of.

By selecting a Minor a decided to go with Skilled because of the freedom. I really wanted to learn a lot about if not master HTML5 and CSS3, extend my JavaScript and PHP knowledge, work on my design skills and get to know WordPress development better.
I couldn't find other Minors that were about that or gave the freedom to do so.

This for landing a job after Art & Technology or even help me get an internship now.
I wasn't feeling very confident having witnessed the internet evolve in a rapid rate during my education and visiting company's like Trimm.
I was feeling very out of touch with the whole webdesign / webdevelopment world, having the idea I was miles behind (and not able to recover that).
This also after my education in multimedia design and working at a place not that related to this subject area, reading up on stuff but not gaining experience.

2 Planning

For all the phases I was to keep a logbook and Trello and update them daily.

2.1 Inception phase

During this phase the plan got conceived and I was set on gathering tutorials and generating lists in the following subject area’s:

- HTML5
- CSS3
- JavaScript
- PHP
- Design
- WordPress
- Software (Dreamweaver & Photoshop)

During this phase I was to also start with the software tutorials and look for features that could be implemented in the website.

2.2 Elaboration phase

This phase was about determining the features with a MoSCoW document and having the big picture on all the requirements. Also having a flowchart and a near complete responsive design with mockups for small, medium and big devices.

2.3 Construction phase

At the end if this phase the plan was to have the final design and a rough working version of the website (user unfriendliness being allowed).

2.4 Transition phase

At the end of this phase there should be a website working in multiple browsers and I was to be happy when I understood all the code and when my workflow was efficient. The website should pass HTML5 and CSS3 validation.

3 Software

The first thing I did was getting familiar with the newer CC versions of Adobe software since I got stuck at CS6. It's already been a good 3 years since I updated my Adobe knowledge.

3.1 Dreamweaver

Since the first CC version there is now an inspect feature build in Dreamweaver like all the major browsers have now.
Since CC the live view also started using the WebKit rendering engine. (Williamson, 2013).

3.1.1 Dreamweaver CC

Since 2013 the UI is cleaner without much of the clutter accumulated since 1997. In total 14 dialogs and 62 menu items got removed, often removing the multiple ways access to the same functions.
SASS/SCSS support got added and an insert panel to drag and drop elements including Spry, jQuery and audio/video tags.
The audio and video tags can now be edited using the properties panel.
With the CreativeCloud you've got the ability to sync settings between computers and use TypeKit. (Trani, 2013).

TypeKit is an Adobe service to use more than just the system fonts and has multiple plans, including a free plan.
Next to TypeKit support there is also support for 'Edge web fonts' which is completely free. (Trani, 2013).

The CSS styles panel got replaced with the CSS designer panel, a more visual way to set values without the former dialogue box. A dialogue box is now only needed for adding media queries. Since this version you're also able to configure automatic vendor prefixes for browsers. (Trani, 2013).

In this version support for inserting Edge Animate compositions also got added. (Trani, 2013).

3.1.2 2014 updates

In 2014 Adobe released a complete rewrite of Dreamweaver. The first time a 64-bit version of Dreamweaver got released, resulting in faster loading and a reponsive experience.
Another round of cleanup resulting in even less clutter on the toolbar. (Lowery, 2014).

The new extract panel got introduced that works via the Creative Cloud for slicing and measuring, copying text and extracting CSS from PSD's within Dreamweaver. (Lowery, 2014).

This version also introduced live view editing being able to rearrange elements and theme support, for a much easier way of turning the lights off within Dreamweaver putting less strain on the eyes. For this feature alone I would already upgrade.

Also an ‘email development kick start’ got included providing a basic setup for designing emails. (Lowery, 2014).

3.1.3 2015 updates

With the updates in 2015 Adobe integrated Bootstrap support in Dreamweaver and media query preview to the UI. Also you’re now able to preview on other devices and Emmet got integrated. (Lowery, 2015).

For coding Lint got integrated, for live error reporting with HTML, CSS and JavaScript and also being able to check for things like duplicate CSS properties and capital letters. (Lowery, 2015).

Code folding got also added, you can now had hide a block or a selection of code and for CSS there’s now also a preview when hovering the cursor on code for colors and images. (Lowery, 2015).

3.2 Emmet

Emmet is a free toolkit written in JavaScript that can be incorporated into other software. Emmet provides a shorthand language for HTML, CSS and XSL. (Emmet.io, n.d.).

Writing the abbreviation `nav#mainNav>ul.menu>li*5>lorem` and pressing the `tab` key creates the following code (, filler text is shortened for the example):

```
<nav id="mainNav">
     <ul class="menu">
          <li>Lorem ipsum dolor sit amet … </li>
          <li>Perspiciatis, nostrum, iusto … </li>
          <li>Delectus, minima praesentium … </li>
          <li>Tenetur, dolores, nesciunt, … </li>
          <li>Sint, accusantium, fugit … </li>
     </ul>
</nav>
```

It matters where the type cursor is placed, text after the cursor gets ignored, a simple undo (ctrl + z) reverses the whole operation back to the abbreviation. (Villalobos, 2013a).

3.2.1 HTML

You can identify elements as siblings to not nest them further with `+` and also group elements within you abbreviation with parenthesis `(` and `)`. Actual content can be included by putting it between brackets `{` and `}` and for wrapping an element around another you can use `^` for example nesting an image in an paragraph. (Villalobos, 2013a).

You can also already add attributes using `[attribute=”value”]` and use `$` for automatic numbering use more dollar signs for padding like `$$$` and use `$@-1` to set the starting value (in this case negative one). (Villalobos, 2013a).

`!` Creates a minimal HTML5 document, `C` creates an empty HTML comment and using the key combination `Ctrl` and `/` converts selected text to a comment (if Emmet is implemented correctly). (Villalobos, 2013a).

3.2.2 CSS

Margins can be described in a very short way, where the standard unit is pixels:
The abbreviation `m20-10e-5%` creates: `margin: 20px 10em -5%;`.

The plus in `bdt+` gives the result the basic settings `border-top: 1px solid #000;`

And for example `lg(top left,red80%,yellow,60%,blue)` creates the CSS for a linear gradient from top left to bottom right (vendor prefixes depending on software):

```
background-image: linear-gradient(to bottom right, red80%,
yellow, 60%, blue);
```

Everything has a shortcut and those can be found in the Emmet cheat sheet. (Villalobos, 2013a).

3.3 Brackets

Brackets is an free open source text editor by Adobe that got recommended by a few courses on Lynda.com that I did. I have switched to Brackets because of incorrect SASS syntax highlighting in Dreamweaver CC 2015. Emmet support does not come with the standard installation but there is a plugin available. (Brackets, n.d.).

In the future Brackets will be the underlying code editor in Dreamweaver. (Stewart, 2015).

3.4 Photoshop

Since I have learned allot of new things about Photoshop I’m only going to mention features I used or that have kind off directly to do with web design in Photoshop. Otherwise I would have to recite nearly entire courses I have followed.

3.4.1 Photoshop CC

Since this version the different versions Photoshop and Photoshop extended are now combined. The UI had an overhaul which is very noticeable when you scale an image. The new image size dialog box can be scaled for a larger preview and you now have options for better image upscaling. (McClelland, 2013).

When you have opened up two images and lined the windows up next to each other, you can now move both images at the same time holding `shift` and `spacebar` and then drag with the mouse. (McClelland, 2013).

When drawing a shape you can now also move your shape holding spacebar when you only have placed one vector. (Live) Shapes can now also have independent rounded corners, being able to reduce a rounded corner rectangle to a square rectangle. Shortcuts for drawing are: `-` to subtract a shape, `/` for intersecting, `*` to exclude and `+` to combine. When double clicking a path outline you now enter 'layer isolation mode', this is also accessible via the selection menu, double click outside of the layer to cancel. (McClelland, 2013).

The color range selection tool now has the ability to work with colors, highlights, mid-tones, shadows and skin-tones. (McClelland, 2013).

Smart objects can now also be linked to external files, not embedding the images, very handy for icons and such. Smart filters now have a slider symbol in the layers panel, double clicking this will open up a dialogue box for blending options. The document size indicator doesn't include smart objects. (McClelland, 2013).

Actions can new have conditional multistep actions in them, found in the fly out menu, this way you can for example check for alpha channels. (McClelland, 2013).

A new feature interesting for web designers is the ability to copy CSS when right clicking on a group or a layer in the layers panel, mimicking the layer effects and font settings with CSS. Also you now have the ability to automatically generate image assets (for web). (McClelland, 2013).

Filenames are derived from layer names (including spaces), the file types (GIF, JPEG, PNG8/24/32) can also be assigned there. Multiple images of the same layer can be exported by comma separating the instructions. Different scales can exported by adding the percentage in front of the filename or typing out the resolution to scale proportionally you can use a question mark like the following: `640x?`. The quality for a JPEG image can be set by typing for example `jpg1` for 1% quality or `jpg100` for 100%, 90% is the default value. This can also be applied to groups of layers. (McClelland, 2013).

For zooming to a 100% you can now use `ctrl` and `1` for the next layer you can now press `tab` and use `shift` and `tab` to go back a layer. You now also filter and search for layers at the top of the layers panel using: kind, name, effect, blend mode, attribute, color, selected. (McClelland, 2013).

Patterns can now be generated with scripts and be customized accessible with the fill tool or dialogue box. You can also generate picture frames and trees. (McClelland, 2013).

Photoshop now also features live 3D painting with 32-bit 3D lightning and 3D printing. (McClelland, 2013).

3.4.2 2014 updates

In 2014 the hue cube got added, is the color panel scalable and a few keyboard shortcuts got added: use `alt` and `backspace` to fill a layer and use `shift` , `alt` and `space` to only fill what is opaque in a layer with the foreground color. (McClelland, 2014).

Color gradients are now allowed to have one color stop for when you use transparency stops only, the eyedropper can be used on a gradient when editing and there now is a reset alignment button for the gradient layer effect. (McClelland, 2014).

The right click menu and panel for brushes now has a section at the top for recently used (unsaved) brushes. (McClelland, 2014).

Right clicking on the ruler now lets you select what distance unit to work with, when holding `alt` and clicking on the ruler the distance will be displayed. (McClelland, 2014).

A new duplicate layer(s) dialogue box can now be summoned by selecting one or multiple layers in the layer panel and then right clicking on one and then selecting ‘duplicate layer(s)…’ or by having the marque tool selected and right clicking on the canvas then selecting ‘duplicate layer(s)…’. (McClelland, 2014).

This version a flame generator got introduced for the hot designs, which can be found in the generate category in the filter menu and can be used to follow a shape. (McClelland, 2014).

Guide creation can now be automated drawing from shape and (new guides from shape) and the new dialogue box ‘new guide layout’ which both can be found under the view menu.
In the new dialogue box you can set columns, gutters and margins and center them. (McClelland, 2014).

A new libraries panel got added accessing the Creative Cloud libraries that can store: text styles, layer styles, colors, graphics (pixel, vector and smart objects) and groups. (McClelland, 2014).

The selection modifiers like border, smooth, expand and contract now work independent from the canvas border taking care of that sticking to the edge behavior. ‘Save’ and ‘Save as…’ commands now get added to history states. (McClelland, 2014).

3.4.3 2015 updates

One of the biggest new features is artboards, which is also a new document type in the new document dialog. With artboards you can basically hold multiple documents into one PSD file. With it comes the new artboard tool and with it you edit artboards, rotate them but also create new artboard, which can also be done via the layers panel or via the layer menu and then under new. Artboards are shown in the layers panel, when holding `alt` and clicking on an artboard Photoshop zooms in on the artboard. Because you can't print artboards you will first have to export them to a file in het export section in the file menu. (McClelland, 2015).

I used this feature housing the small, medium and large device designs in one PSD file.

With the 2015 updates you can now also review designs on other devices via your CC account. For typography, like in InDesign and Illustrator, you now also have access to the glyphs panel. If a character has an alternate glyph available a little black square will be shown in its preview. (McClelland, 2015).

It's now possible to double up effects and change their stacking order in the layer effects dialogue box. Undo doesn't work yet in this panel. (McClelland, 2015).

A new, and still experimental feature is 'design space'. It needs to be enabled first in the 'technology preview' section in the preferences window. This adds a button to the bottom of the toolbox which takes you to the design space UI, removing most of the standard UI. Design space is more friendly for quickly arranging and placing elements by just dragging them around, having a vector first approach it's very handy for web design. (McClelland, 2015).

Spectrum is the new dark interface in Photoshop since the November updates. `Shift` and `F1` to lighten the interface and `shift` and `F2` to darken the interface. (McClelland, 2015).

The info panel now also shows width, height and position information about text without the bounding box. You now also have the ability to select similar fonts and filter font types like: serif, sans-serif, black and slab. (McClelland, 2015).

It is now also possible to edit the content from the CC library and extract assets from a document to a library. (McClelland, 2015).

3.4.4 Photoshop CC power shortcuts

When using the brush tool and when holding `shift` and `alt` while right clicking on the canvas the HUD color picker will appear near the cursor consisting out of a hue cube and a rectangle with the entire hue range for quick selection. (Seeley, 2015).

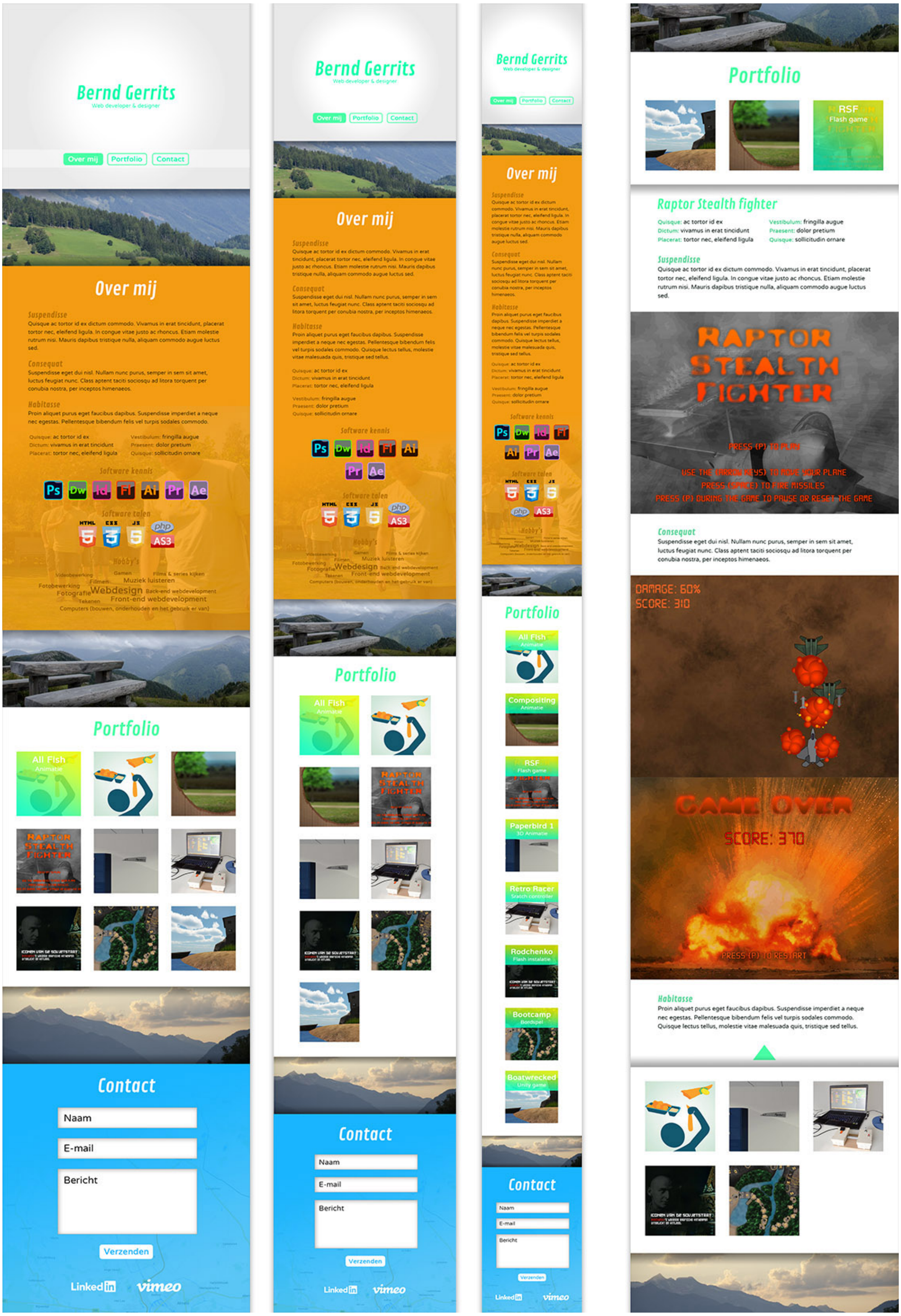
Holding `ctrl` and `alt` while right clicking on the canvas gives you the ability to quickly edit your brush. While holding all the buttons moving the mouse vertically will change the brush hardness, going up for a softer brush and down for a harder brush. Moving the mouse horizontally changes the brush size, going left makes the brush smaller, going right bigger. (Seeley, 2015).

Holding `alt` while clicking on a layers mask in the layers panel shows the layer mask in (grayscale) the canvas. (Seeley, 2015).

While zoomed in hold the `H` key and click and drag to refocus when letting go of `H` key. Transparency can be set with the keyboard using the number keys from `0` to `9`, tap `0` twice to get 0% and tab two numbers quickly to get 25% for example. (Seeley, 2015).

You can also often change values by clicking and dragging horizontally on a label in front of the input field. (Seeley, 2015).

4 Design



Over mij

Suspendisse
Quisque ac tortor id ex dictum commodo. Vivamus in erat tincidunt, placerat tortor nec, eleifend ligula. In congue vitae justo ac rhoncus. Etiam molestie rutrum nisi. Mauris dapibus tristique nulla, aliquam commodo augue luctus sed.

Consequat
Suspendisse eget dui nisl. Nullam nunc purus, semper in sem sit amet, luctus feugiat nunc. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos.

Habitasse
Proin aliquet purus eget faucibus dapibus. Suspendisse imperdiet a neque nec egestas. Pellentesque bibendum felis vel turpis sodales commodo.

Quisque: ac tortor id ex
Dictum: vivamus in erat tincidunt
Placerat: tortor nec, eleifend ligula
Vestibulum: fringilla augue
Praesent: dolor pretium
Quisque: sollicitudin ornare

Raptor Stealth fighter

Quisque: ac tortor id ex
Dictum: vivamus in erat tincidunt
Placerat: tortor nec, eleifend ligula
Vestibulum: fringilla augue
Praesent: dolor pretium
Quisque: sollicitudin ornare

Suspendisse
Quisque ac tortor id ex dictum commodo. Vivamus in erat tincidunt, placerat tortor nec, eleifend ligula. In congue vitae justo ac rhoncus. Etiam molestie rutrum nisi. Mauris dapibus tristique nulla, aliquam commodo augue luctus sed.

RAPTOR STEALTH FIGHTER
PRESS (P) TO PLAY
USE THE (ARROW KEYS) TO MOVE YOUR PLANE
PRESS (SPACE) TO FIRE MISSILES
PRESS (P) DURING THE GAME TO PAUSE OR RESET THE GAME

Consequat
Suspendisse eget dui nisl. Nullam nunc purus, semper in sem sit amet, luctus feugiat nunc. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos.

Habitasse
Proin aliquet purus eget faucibus dapibus. Suspendisse imperdiet a neque nec egestas. Pellentesque bibendum felis vel turpis sodales commodo. Quisque lectus tellus, molestie vitae malesuada quis, tristique sed tellus.

Figure 1. Large, medium and small devices and portfolio item mockup overview

4.1 Web design

At first I assessed what content the website would contain and thought about the importance and in what order it should be presented. I first sketched out the ideas I had, comparing the minor differences and went with the first one. Designing mobile first I have chosen to use mainly a single column layout, also because there is no need for more with my content. (Williamson, 2014b).

4.2 User experience design

4.2.1 Keep users from leaving

To make sense immediately when entering the site I have designed my 'intro' to show what my site is about: it's about me, what I want to do for work and my portfolio. I'm not hiding information other than the portfolio items in the portfolio item grid, otherwise the page would also become too tall. I have no extra visual design stuff other than necessary what would distract. Making my contact form as painless as possible I will also won't be asking for a subject. (Nodder, 2012).

4.2.2 Simple design

The text I have used in the design and I am going to use in the content is being kept to a minimum. In my former websites I would just type and fill to look 'professional' but this has been studied and results were that a more complex text means lower intelligence.
Also web text should be half the amount of what print text is, people read 25% slower on screens and I also don't like 'creating' more text just to fill. Because of that I will also won't be having a home page this time. (Nodder, 2012).

Headings and sub headings should be used for dividing content and I'm planning to do this as much as possible as people scan for them. So use words of interests, the words people would be searching for. Never use more than three heading depths and for body text it's a good idea to use the inverted pyramid for paragraphs, facts first. This is what journalist use for print ranking from: who, what, when, where, how, conclusion, important details, general info. (Nodder, 2012).

4.2.3 Consistent design

This is easy for a single page design, you don't have to deal with consistency on multiple pages (but have to mind the sections); navigation sticks to the top of the viewport and the interactive elements keep the same working. (Nodder, 2012).

4.2.4 Standard design

On my website I use known elements also presented on other (bigger) website so users will be familiar with their working. The scrolling (instead of teleportation) when clicking on a button in the navigation, the expanding 'accordion' in the portfolio item grid and the validation in the contact form. (Nodder, 2012).

4.3 Aesthetics

4.3.1 Design methods

The main design methods I borrowed from are flat design and mechanical design. Flat design is a trend at the moment therefore quite modern and I wanted to design something ‘flat’. Mechanical design is noticeable in the top and bottom margin of the main headings, both being the same height as the heading itself. Both design styles used for a modern look.

4.3.2 Backgrounds

The main background colors are borrowed from flat design. Orange for the about me section and blue for the contact section. I don’t have to keep in check with company colors and since I design this site for myself I don’t think it’s bad thing picking colors because I like them. Which is very fitting for the about me section.

The other sections, the intro and the portfolio section, are white alternating between white and colored across the sections creating a clear distinction when scrolling across them.

4.3.3 Text

The headings on the white backgrounds I have colored cyan green because it’s a fresh color and I don’t come across this color a lot online, standing out a bit.
Headings on the colored background are white for a negative effect.

Body texts aren’t fully black but a very dark variant of the background color, or for the white backgrounds a very dark cyan green. This makes the text a little bit softer.

The links are bright very much like the headings for the white background and for colored backgrounds a bit darker then the background color but not blue as bigger sites do. (Nodder, 2012).

4.3.4 Patterns

At first the navigation buttons appear as ghost buttons, another trend. When the navigation bar sticks to the top the buttons themselves turn white with the cyan green as text color, while the navigation bar turns transparent. For the hover effect the colors get swapped around Other buttons get the background color of their section using the background image for contrasting.

Between sections there is an dividing fixed background images with shadows at the top and bottom for separation and creating some depth.

The portfolio grid uses big whitespace to give it a modern look and when an item expands after clicking it also gets a shadow at the top and bottom sectioning of a new section within the white of the portfolio. When an item gets expanded between others it has the same top and bottom margin as the images have in the portfolio item grid.

4.4 Interaction design

Every interactive element has some hover effect. The progress images in the about me section only have simple title tooltip but on hover they get slightly transparent. With this I want to display in text how much off a certain topic I think I know. The images themselves already do this in a way, but just to be sure. (Hogue, 2012).

The navigation is always present by sticking to the top so users don't have to scroll up. Because I don't have a very elaborate menu and don't need submenu's I don't have to deal with delay's for the appearance, fly outs or disappearance. I have build in that when hovering on the navigation the navigation buttons turn opaque. Normally they are 50% transparent to not distract. (Hogue, 2013).

4.5 Typography

As a font service I chose Google Fonts. I was looking for fresh, clean and modern looking sans-serifs and ended up downloading around twenty fonts.

I put them all next to each other and crossed out the ones that were to thin or to thick (also checking the bold variant), then by how nice the shape was compare to the others, readability, nitpicking, which got hard to do in the end.

I settled with 'Contrail One' ,which is slanted by default, for the headings and 'NTR' for the body text.

Font sizes of 10 to 14 pixels used to be the norm for paragraph text, nowadays it's 14 to 16 pixels. (Franz, 2011).

I ended up designing 16 pixels for the paragraph text, 37 pixels for the first heading level, 23 pixels for the second level and for the third level I used the normal font weight, not bold.

In this stage I was already planning to actually scale with em's (a typographic unit) in CSS.
With em's in CSS you scale what font size the parent element has set.
By setting `font-size: 62.5%` the actual font size would be 10 pixels for easy scaling in em's later, 2 em font size on an element being 20 pixels.
Setting a font-size with pixels would disable scaling by the user (when zooming).
Setting a font-size at the lower level like the body tag disables font scaling. (Franz, 2011).

5 Static

5.1 HTML

Below a stripped overview the sectioning that ended up between the body tags:

```
    <header>
        <nav>
            <ul></ul>
        </nav>
    </header>

    <div class="divider"></div>

    <main>
        <article id="about"></article>

        <div class="divider"></div>

        <section id="portfolio">
            <div id="accordion">
               <!-- Portfolio item start -->
                <div class="portfolio-item-header">
                    <img>
                    <p>Title
                        <span>Category</span>
                    </p>
                </div>
                <article class="portfolio-item">
                    <h1>Portfolio item 1</h1>
                </article>
                <!-- Portfolio item end -->
            </div>
        </section>

        <div class="divider"></div>

        <article id="contact">
            <h1>Contact</h1>
            <form action=""></form>
            <ul class="links"></ul>
        </article>
    </main>
```

5.1.1 Semantics

An `article` element contains content that can stand alone, removed from the rest it would still make sense. A `section` element is for content that needs to be listed in the document outline, this element can also be used for sectioning in an `article` element.
The `aside` element I didn't use can be placed at the same level as previous mentioned elements with relevant content to the document or when placed in an `article` or `section` element, contain relevant content to the element it is placed inside of. (Williamson, 2014a).

When you are just grouping content you should use an `DIV` (division) element just because of divitis (overusing div's for markup) in the former HTML versions doesn't mean you should use the new HTML5 elements for everything now. The `DIV` element has no special meaning. (Williamson, 2014a).

The `main` element should be used once per page and doesn't influence the page outline. The outline is an index generated by the page content. (Williamson, 2014a).

The `header` element can be used multiple times, not only to be used as the header for the document but can be used as the header for nearly everything going with the parent element. (Williamson, 2014a).

I sectioned the normally individual pages like 'about me' and 'contact' into articles, and contained the portfolio 'pages' in a section because they all belong together.

The whole 'intro' of the site is contained in the header as it's the first thing you see, the site navigation is also housed there.

The portfolio headers are contained in a div because of no special meaning other than the presentation of text on an image, a paragraph for the title and a span for the category.

The navigation buttons and links at the bottom of the contact section are placed in `UL` (unordered list) elements and show up different because of CSS.

The progress images in the 'about me' section are contained in a div that centers them and stacks them when needed.

5.1.2 Living standard

`<!doctype html>` HTML5, because of the rapid development of the web, is a living standard, that's why unlike the former versions there is no version number in the doctype. New additions now get added on the fly, you just have to wait for browsers to support them. (Williamson, 2015).

With HTML5 some things switched around again `B` (bold) and `I` (italic) tags are visual again and `strong` and `em` (emphasize) tags now have semantic meaning. (Williamson, 2015).

List elements like `UL` and `OL` now have the `reverse` attribute. `Blockquote`s can now contain the author in the `footer` element. (Williamson, 2015).

And the `small` element is now there for small print (like in contracts) and a new `time` element got introduced to represent displayed and machine (in the attributes) time or duration. (Williamson, 2015).

`Video` and `audio` elements got also introduced for standard playback in the browser with its own controls. (Williamson, 2015).

5.2 CSS

Very little CSS I have used on this website was new for me. SASS was really the new thing CSS thing in this including Suzy, Compass and Breakpoint. Instead of reciting every tutorial CSS tutorial I did I will explain some very weird bugs (and fixes).

5.2.1 Chrome placeholder fix

The placeholder text in the name and email input fields of the contact form got cut of being placed too high. The normal text when typing was fine. This was fixed adding 0.55em padding to the top using the `input::-webkit-input-placeholder` selector.

5.2.2 Firefox fix

The top and bottom padding on again, the name and email input fields wasn't working in Firefox resulting in having to add `min-height: 2em;` 1em for the text plus two times 0.5em for the padding.

5.3 jQuery

5.3.1 jQuery UI accordion

Still in the static phase I tested the jQuery UI accordion and it worked as I wanted. I only needed to set a few settings: the accordion should be as high as the content, the user should be able to collapse it and it should start collapsed.

With configuration I also changed the header element it uses from `H3` (heading level 3) to `DIV` so that without special meaning it could contain the thumbnail image, a paragraph for the portfolio title and a `span` element for the portfolio category.

What normally follows the accordion header is a `DIV` element what gets expanded and where the actual content is located, that could be easily changed to an `article` element without configuration to make the accordion more semantic.

5.3.2 Delayed image loading

For the static version of the website I used lazyload.js to see if ‘lazy’ loading images would actually work with the accordion.

Which is quite important, for crawlers from search engines I should have all the content loaded immediately. Not loading it on the fly with ajax that doesn’t work for crawlers.

But loading all the images at once for every portfolio item would take quite some time and not everybody is going to look at all of them. Luckily it worked.

5.3.3 Scrolling

There was a quite complicated script included with the tutorial for creating a one-page WordPress website. It recognizes every in site link, every anchor element that leads to an ID within the page, and when clicked slowly scrolls to that point. Not immediately teleporting the viewport.

There is one fixed value in the script you can edit to offset the scrolling to take into account the sticky navigation (to not cover where you scroll to). But when your navigation changes in size based on the width of the viewport that’s not so handy.

In short I changed the following, 80 being the fixed number:

```
scrollTop: (target.offset().top - 80 )
```

To the following:

```
scrollTop: (target.offset().top - $('header nav').outerHeight() )
```

Now every time an in site link gets clicked it calculates the height of the navigation bar including the padding an the border.

5.3.4 Sticky navigation

Also with the tutorial came the script that fixes the navigation to the top when scrolling past it.

It works by adding a class to the navigation when the top of the viewport is at the same location vertically as the navigation element.

Normally it also adds a class to another element (often the header) to counteract the fact that the navigation is taken out of the layout, what would move the site upward. This class would normally fix this with an extra margin but I didn’t need this so I edited it out of the code.

The script normally also uses classes to target elements which I have changed to use elements.

6 Workflow

6.1 SASS

SASS stands for Syntactically Awesome Style Sheets and is a CSS preprocessor.
All though I am using the newer SCSS (Sassy CSS) syntax which is a superset to CSS which is a lot like CSS itself unlike the original SASS syntax. The file type used is also .SCSS. (Villalobos, 2015).

A SCSS file doesn't run in the browser for that you process it to a CSS file. You can export with three profiles: compact (condensed) file, nested (default) or expanded which is the most human friendly leaving comments to what file the selector is from. (Villalobos, 2015).

6.1.1 Variables

With SASS you get the power of variables writing your styles so you can now reference the same value over and over and when it needs change you only need to change one value. (Villalobos, 2015).

I have used this to store the colors, font-family's, complete font short hand notations and breakpoints (media queries) I used but you can also store: strings, lists and other variables. (Villalobos, 2015).

Array's or rather lists start at one instead of zero to select for example:
`$var: 1px solid #000` here `($var,1)` would result in `1px` . (Villalobos, 2015).

6.1.2 Functions

In SASS there are functions to change colors on processing for example:
`lighten($blue,10)` this would mix the color with 10% white, but you can also: darken, saturate, desaturate, fade, change the hue, saturation or lightness, change the alpha level, turn color to grayscale, complement colors, invert colors, transparentize colors and mix 2 colors together using `mix(color1, color2)`. (Villalobos, 2015).

There are also math functions like rounding numbers (floor and ceiling) for when the included operators (like in CSS) aren't cutting it. (Villalobos, 2015).

6.1.3 Nesting

Writing selectors inside of another like so:

```
p {
    a {
    color: #000;

        &:hover {
        color: #CCC;
        }
    }
}
```

The ampersand being a parent selector, outputs like this

```
p a{
color: #000;
}

p a:hover{
color: #CCC;
}
```

6.1.4 Mixin

For extending selector with the same set of properties you can use mixin’s which can also handle parameters, I have created and used the following mxin:

```
@mixin sectionBackground($color,$image){
    background: $color url("images/#{$image}.jpg");
}
```

Calling this would look like this:

`@include sectionBackground($aboutBright,about-bg);`

6.1.5 Import

Normally the files that are imported with `@import` get combined into one single file, using http in the URL or using a CSS file takes care of importing it the normal way, not combining it. If you put an underscore in front of the filename you can import files in the same folder or subdirectory without the file type. (Villalobos, 2015).

This makes it easy to breakup your code into parts. I have separated files for: variables, mixins, base styling, layout, intro, navigation, form, accordion (portfolio), and WordPress standard styling.

6.1.6 Booleans and loops

With SASS you can also write if else statements like:

`@if length($args) == 1 { css:styles; } @else{ css: style; }`

And also for, each and while loops like so:

`@for $i from 1 through 3 { .item-#{$i} { css: style;} }`. (Villalobos, 2015).

6.2 Gulp.js

Gulp.js is a task runner to get it you first have to install Node.js. (Villalobos, 2014b).

6.2.1 Node.js

When Node.js is installed and you have a command tool like BASH running you can install Gulp.js with the command `NPM install –g gulp` NPM standing for node package manager. (Villalobos, 2014b).

6.2.2 Browserify

Browserify handles the downloading of script/frameworks like jQuery and Mustache for you. Browserify is install with `npm install –g browserify` . (Villalobos, 2014b).

6.2.3 Ruby

For SASS, Compass and Breakpoint you need to have Ruby installed. Ruby is the environment that runs ‘gems’, a gem being a plugin in Ruby. (Villalobos, 2014b).

6.2.4 SASS & Compass

When Ruby is installed and you have a command line running type `gem install sass` to install SASS itself and `gem install compass` to install the Compass framework. Compass is like an extension to SASS solving common problems like vender prefixing for you. (Villalobos, 2013b).

6.2.5 Breakpoint

Breakpoint is great tool in sass to handle creating media queries for you. Breakpoint is installed with `gem install breakpoint` . (Villalobos, 2014a).

To put it to use you include in you SASS like this:

```
@include breakpoint($small) {
    font-size: 2.7em;
} // breakpoint small
```

This creates a media query for the selector it is used in with the predetermined variable: `$small: max-width  480px;`

6.2.6 Susy

Susy is a responsive grid framework that does everything in the CSS itself. No extra HTML markup for columns. Suzy gets installed by running `gem install susy`. I myself eventually only used Susy for its border-box-sizing. (Villalobos, 2013b).

6.2.7 Package.json

This file gets created by running `npm init` while being navigated to the folder you want to use. This file also comes in handy multi user projects, dependencies for the project are installed using `npm install` command. (Villalobos, 2013b).

The `npm install --save-dev gulp` command where Gulp is an example saves Gulp as a dependency to edit the project.
The `npm install --save gulp` command where, again Gulp is an example saves Gulp as a dependency that is needed to run the project. My package.json file looks like this:

```
{
  "name": "PortfolioTheme",
  "version": "0.0.1",
  "description": "Wordpress theme",
  "repository": {
    "type": "git",
    "url": "?.git"
  },

  "author": "Bernd Gerrits",
  "devDependencies": {
    "gulp": "^3.8.7",
    "gulp-util": "^2.2.20",
    "gulp-concat": "^2.3.4",
    "gulp-browserify": "^0.5.0",
    "jquery": "^2.1.1",
    "gulp-compass": "^1.2.0",
    "gulp-connect": "^2.0.6",
    "gulp-if": "^1.2.4",
    "gulp-uglify": "^0.3.1",
    "gulp-minify-html": "^0.1.4"
  }
}
```

When you would work on this project you would have the same version of every plugin installed so the result would be the same conflicts because of version won't arise. (Villalobos, 2014b).

6.2.8 Gulpfile.js

This file contains the actual tasks you can run from the command line. Using the following command `gulp html` would run the html task, which depending on the environment 'development' would do nothing or when the environment would be set to 'production' minify and save it to the production folder. (Villalobos, 2014b).

The environment variable can be edited in the gulpfile or passed with the `NODE_ENV=production gulp` command. (Villalobos, 2014b).

The actual HTML task looks like:

```
gulp.task('html', function() {
  gulp.src('builds/development/*.html')
    .pipe(gulpif(env === 'production', minifyHTML()))
    .pipe(gulpif(env === 'production', gulp.dest(outputDir)))
    .pipe(connect.reload())
});
```

You can also set a default task, like so:

```
gulp.task('default', ['watch', 'html', 'js', 'compass', 'move',
'connect']);
```

Now when issuing the gulp command this would run all tasks and because of the watch task every time you save file the task responsible will run. Making writing SASS much easier by not having to compile yourself each time. A nice addition to that is the connect task, by having your browser set to localhost:8080, every time when you save a file it gets live refreshed. (Villalobos, 2014b).

7 Wordpress

7.1 Theme

7.1.1 Images

The images folder contains the background images, divider images and the portfolio item shadow pattern images for the top and bottom shadows, but also the progress icon images and the link images which are used in the 'about me' and 'contact' me pages referenced by using absolute URL's.

Not in the images folder but in the root folder of the theme is the screenshot which needs to be included with every theme.

7.1.2 Scripts

The 'js' folder houses only one file; scriptcol.js (because scripts.js wasn't loading). This file holds the three scripts and the configuration part for the jQuery UI accordion.

The difference between the scripts in this file and that of the static version is that the `$` sign at the start of every jQuery script now has been replaced with `jQuery` and the scripts are held in one contained function starting with the following:

```
jQuery(document).ready(function($)
```

This to not conflict with jQuery scripts used in WordPress.

7.1.3 Header and footer

The `header.php` and `footer.php` are files that I could have left out, this theme being a single page theme. But for plugins that might hook into these files I put them in.

7.1.4 404 handling

Because there are no other pages every other 'page' that is tried to access should send the user to the one page that does exists. In the `404.php` file is PHP header that redirects the browser to the front page using the WordPress `.get_bloginfo` function to get the URL.

7.1.5 Index.php

This file isn't normally used within my theme but should be there for the theme to work as it is one of the minimal required files. In this file there's a short message with a link to the root of the site, just in case it might show up.

7.1.6 Front page

This is the file where everything happens and because of it existing in the theme folder gets used as the front page automatically.

I didn't think it would be this simple but basically this file consists out of three loops.
The first loop query's the 'about me' page by using the page slug `over-mij` acquiring it.

The second loop is in the accordion area and takes care of all the portfolio items which are regular posts. In the portfolio item header part it outputs the `post title` in the paragraph element and the `post category` in the span element located in the paragraph element.
In the image `src` attribute the `post thumbnail` is outputted using the image size of 250 pixels by 250 pixels with hard crop, preconfigured in the `functions.php` file.

The heading in the actual portfolio item is put in the content itself and is not acquired from the post title, this is done to be able to work with different lengths of text or to have the title on the image be different from the level one heading in the item.

The third loop does the same as the first loop the only difference being using the ‘contact’ slug to get the contact page. The contact form itself is created by a plugin and the link images are put into the page content.

The only thing you have to add to the loop when you’re working with multiple loops is the calling of the `wp_reset_postdata()` function before starting a new query.

7.1.7 Styles

This also one of the minimal required files, in this file the theme description is stored.
The generated CSS file from SASS when expanded is 31kB in size, the compact version is 19kB in size and the resulting `style.css` turned about to 20kB in size.

7.1.8 Functions.php

In the file the registration the navigation gets registered which is looked for in the header file.

The thumbnail image size of 250 pixels by 250 pixels is set in here also enabling the generating of the image (for when uploading images), hard cropping enabled for a true square image.

Because of the document title ending up being empty I also added a filter that checks if the actual viewed page is the front or home page and if the title is empty, when this is true ‘BerndGerrits.nl – Portfolio’ is used as the standard document title.

The enqueueing of the ‘scripts collection’ file is also done in the functions file. At first this broke the admin area of WordPress because of the script also being run in that area. The enqueue is now wrapped in an if statement checking for when being not in the admin area.

WordPress comes included with jQuery but also with jQuery UI. Loading the required accordion component can be done after enqueueing jQuery UI with `add_action( 'wp_enqueue_scripts', 'load_jquery_ui' )`, requesting the accordion with the `load_jquery_ui` function using `wp_enqueue_script('jquery-ui-accordion')`.

Also because WordPress likes to wrap images located in posts and pages in a paragraph element I added a filter to undo this adding of these unnecessary elements.

7.2 Plugins

7.2.1 A3 lazy load

This is the plugin I eventually used for lazy image loading. Really easy plug and play plugin.

Lazy loading plugins disable the `src` attribute on the `img` elements which is where the path to the image and the image name itself is referenced.
Because of the `noscript` element fallback in this plugin it won't affect SEO.

7.2.2 ContactForm 7

Not as plug and play as the previous plugin. I had to add the following:

```
define('WPCF7_AUTOP', false);
```

to the `wp-config.php` file to remove unneeded element usage also messing with semantics, introducing paragraph elements where the actual input elements would reside.

What's not that nice is that the configuration file is not part of the theme files but of the WordPress installation itself. That would cause everyone that wants to use this theme to have an extra step with installing the theme correctly.

8 Looking back

8.1 Inception phase

A little bit of a slow start because my original plan didn't get agreed upon by my study coach which was a good thing. Because of this the plan got to be me working on my new portfolio website. As planned updated myself on Dreamweaver and Photoshop (my knowledge kind of stopped at version CS6) and learned about Emmet.

This phase I also learned about imposter syndrome, knowing I wasn't the only one that felt like I didn't/wouldn't belong in this industry not knowing enough of subjects or having enough experience got me motivated again to learn more.

My plan was to also get a more steady sleep schedule and to work on the Minor like work days five days in the week keeping the evening and the weekend to myself, unlike what happened in nearly all the (school) projects before (the opposite, working continuously), which worked out pretty good.

8.2 Elaboration phase

Brushing up on design knowledge in the topics: webdesign, aesthetics, user experience, interaction and typography. I'm happy I learned a lot, more than I could apply to the design, but I lost two days (that I even scheduled) because of nothing happening. Only after the two days when I had an inspiration quest work got done, but only to work late at night again.

8.3 Construction phase

After all the CSS tutorials I did the last one introduced me to a new workflow. Involving Git, Github, Node.js, Ruby, Gulp, Browserify, Breakpoint and Susy. Because of seeing magic happening in the tutorial course I decided to postpone the construction of the theme and learning PHP and JavaScript.

Because of this I got experience with an advanced styling techniques, Node.js and Ruby which I have had postponed learning for myself before.

8.4 Transition phase

Right after the construction phase motivation problems started to arise worsening over time. Losing the drive to work and missing the sense of pressure of the deadlines, basically not caring was a first one for me. Effectively losing two weeks, the notion I had to finish with time to spare (this time) got destroyed.

Getting an internship was overshadowing my Minor and caused it to not be that much fun. The imposter syndrome dread came back, everything was interesting but I dreaded starting actual work.

I doubted if I should have dropped SASS as it was a time wasting problem at the very end of the construction phase trying to produce the really rough version and instead just go vanilla CSS but I am glad I didn’t go that route and actually got some experience.

I also got to do a little bit of jQuery which I didn’t thought I would get to do at first which made the learning experience much better.

8.5 Overall

Even for losing that time I ended up with something I’m proud of and think is just right for what I need as my portfolio website.

I am also happy with design working (for me) and not disliking it already.

9 Sources

Brackets,. *A modern, open source code editor that understands web design*. Retrieved 12 January 2016, from http://brackets.io/

Emmet.io,. *Emmet — the essential toolkit for web-developers*. Retrieved 12 January 2016, from http://emmet.io

Franz, L. (2011). *Typography for Web Designers | Lynda.com Training*. *Lynda.com - A LinkedIn Company*. Retrieved 5 October 2015, from http://www.lynda.com/Web-Fonts-tutorials/Typography-for-Web-Designers/79411-2.html

Hogue, D. (2012). *Interaction Design Fundamentals | Lynda.com Training*. *Lynda.com - A LinkedIn Company*. Retrieved 29 September 2015, from http://www.lynda.com/Dreamweaver-CS5-training/Interaction-Design-Fundamentals/82822-2.html

Hogue, D. (2013). *Applied Interaction Design | Lynda.com Training*. *Lynda.com - A LinkedIn Company*. Retrieved 30 September 2015, from http://www.lynda.com/Web-Interaction-Design-tutorials/Applied-Interaction-Design/110712-2.html

Lowery, J. (2014). *Dreamweaver: 2014 Creative Cloud Updates | Lynda.com Training*. *Lynda.com - A LinkedIn Company*. Retrieved 4 January 2016, from http://www.lynda.com/Dreamweaver-tutorials/Dreamweaver-2014-Creative-Cloud-Updates/182375-2.html

Lowery, J. (2015). *Dreamweaver: 2015 Creative Cloud Updates | Lynda.com Training*. *Lynda.com - A LinkedIn Company*. Retrieved 4 January 2016, from http://www.lynda.com/Dreamweaver-tutorials/Dreamweaver-2015-Creative-Cloud-Updates/373789-2.html

McClelland, D. (2013). *Photoshop: 2013 Creative Cloud Updates | Lynda.com Training*. *Lynda.com - A LinkedIn Company*. Retrieved 8 October 2013, from http://www.lynda.com/Photoshop-tutorials/Photoshop-2013-Creative-Cloud-Updates/122468-2.html

McClelland, D. (2014). *Photoshop: 2014 Creative Cloud Updates | Lynda.com*. *Lynda.com - A LinkedIn Company*. Retrieved 12 October 2015, from http://www.lynda.com/Photoshop-tutorials/Photoshop-2014-Creative-Cloud-Updates/171056-2.html

McClelland, D. (2015). *Photoshop: 2015 Creative Cloud Updates | Lynda.com*. *Lynda.com - A LinkedIn Company*. Retrieved 2 December 2015, from http://www.lynda.com/Photoshop-tutorials/Photoshop-2015-Creative-Cloud-Updates/369899-2.html

Nodder, C. (2012). *User Experience Fundamentals for Web Design (2012) | Lynda.com Training.Lynda.com - A LinkedIn Company*. Retrieved 25 September 2015, from http://www.lynda.com/Web-User-Experience-tutorials/User-Experience-Fundamentals-Web-Design/103677-2.html

Seeley, J. (2015). *Photoshop CC Power Shortcuts | Lynda.com Training*. *Lynda.com - A LinkedIn Company*. Retrieved 16 October 2015, from http://www.lynda.com/Photoshop-tutorials/Photoshop-CC-Power-Shortcuts/378054-2.html

Stewart, R. (2015). *Brackets Blog - The Free, Open Source Code Editor for the Web*. *Blog.brackets.io*. Retrieved 12 January 2016, from http://blog.brackets.io/2015/10/05/update-from-the-adobe-brackets-team/

Trani, P. (2013). *Dreamweaver: Creative Cloud Updates | Lynda.com Training*. *Lynda.com - A LinkedIn Company*. Retrieved 4 January 2016, from http://www.lynda.com/Dreamweaver-tutorials/Dreamweaver-Creative-Cloud-Updates/124066-2.html

Villalobos, R. (2013a). *Coding Faster with Emmet | Lynda.com Training*. *Lynda.com - A LinkedIn Company*. Retrieved 3 December 2015, from http://www.lynda.com/Emmet-tutorials/Coding-Faster-Emmet/133353-2.html

Villalobos, R. (2013b). *Responsive CSS with Sass and Compass | Lynda.com Training*. *Lynda.com - A LinkedIn Company*. Retrieved 19 November 2015, from http://www.lynda.com/CSS-tutorials/Responsive-CSS-Sass-Compass/140777-2.html

Villalobos, R. (2014a). *Building a Responsive Single-Page Design with Sass | Lynda.com Training*.*Lynda.com - A LinkedIn Company*. Retrieved 3 December 2015, from http://www.lynda.com/CSS-tutorials/Building-Responsive-Single-Page-Design/182174-2.html

Villalobos, R. (2014b). *Web Project Workflows with Gulp.js, Git, and Browserify | Lynda.com Training*.*Lynda.com - A LinkedIn Company*. Retrieved 1 December 2015, from http://www.lynda.com/Web-Design-tutorials/Web-Project-Workflows-Gulp-js-Git-Browserify/154416-2.html

Villalobos, R. (2015). *SASS Essential Training | Lynda.com Training*. *Lynda.com - A LinkedIn Company*. Retrieved 17 November 2015, from http://www.lynda.com/Sass-tutorials/Sass-Essential-Training/375925-2.html

Williamson, J. (2013). *Dreamweaver CC Essential Training | Lynda.com Training*. *Lynda.com - A LinkedIn Company*. Retrieved 4 January 2016, from http://www.lynda.com/Dreamweaver-tutorials/Dreamweaver-CC-Essential-Training/124084-2.html

Williamson, J. (2014a). *lynda.com Training | HTML Essential Training*. *Lynda.com - A LinkedIn Company*. Retrieved 14 September 2015, from http://www.lynda.com/HTML-tutorials/HTML-Essential-Training/170427-2.html

Williamson, J. (2014b). *Web Design Fundamentals | Lynda.com Training*. *Lynda.com - A LinkedIn Company*. Retrieved 25 September 2015, from http://www.lynda.com/Web-Design-tutorials/Web-Design-Fundamentals/177837-2.html

Williamson, J. (2015). *HTML5: Structure, Syntax, and Semantics | Lynda.com Training*. *Lynda.com - A LinkedIn Company*. Retrieved 14 September 2015, from http://www.lynda.com/HTML-tutorials/HTML5-Structure-Syntax-Semantics/182177-2.html