The weekly nerd

## Week 1

## Week 2

## Week 3

## Week 4

## Week 5

## Week 6

## Week 7 pt 1 /2

## Week 8 pt 1 /2 /3 reflection

Week 9 1 -2 reflection (q42)

Week 10 reflection

Week 11 1/ 2 reflection (pre css day)

bonus

# Review van de verschillende vakken

## Wafs

## CSS

## Bt

## Hackathon

## API

## HCD

Reflectie algemeen

Drie persoonlijke leerdoelen

WAFS-leerdoelen

Foto’s (bij vakken?)

Andere leuke dingen?

* Sporify aangeraden hoist up the thing (uitleg en vibes) how I feel
* Weather report?
* Printen
* Rubber ducky als logo/ foto
* The spotify list that got me through the day (using the same format as the other one to create substance)

device reflective-> sunny

device loud -> hot

device beeping -> hacked/ notification

device gone -> stolen

spelling wack- > autocorrect

Week 4

Declarative design [title 2]

Jeremy [author]

There are two approaches to design in the work field, declarative and imperative. Neither is wrong, neither is right, but we are going to go over some highlights of both methods.

First, the basis [3]

With imperative design you work with precise instructions.

With declarative design you work with the output you want and let the software think about how to get there.

**How does this apply in the world wide web [3]**

Html – declarative

CSS- declarative

This makes both html and CSS fault tolerant

Js – imperative

“JavaScript should only do what only JavaScript can do” [quote]

**Why does it matter? [3]**

This distinction between mindsets is, in the root, a difference in control. Why do you choose to make your own button, or to trust the browser to make one for you. Do you see the code the browser gives you as features or as bugs? My advice- let the browser work for you.

Assumptions [add text]

“Be the browsers mentor (imperative) not its micromanager (declarative)” [quote]

**Design systems in the real world- thinking about thinking [3]**

Analytic/systems

Communication between the methods might be difficult

Review [review]

This lecture by Jeremy was interesting, and he is a good storyteller. You could tell he is willing to share his experience and knowledge and is not trying to sell us anything. Key points are:

* Find the culture that fits
* Find the way of working that fits for you
* Find yourself

Weekly nerd #1

Stop using JS for that [title, h2]

Killian valkhof [author]

Polyplane, websites testen op grootte [ad/link]

In the world wide web there are an infinite number of ways to solve each problem. These are the most useful tips and tricks and how to apply them.

Rule of the least power [h3]

The rule of the least power entails that you should choose to write in the language with the least power possible for the purpose. In this way, the browser does not have to think too hard and slow down the processes. This makes for a better user experience. In the world wide web are HTML and CSS both lighter than JavaScript.

* Browser hoeft niet na te denken
* Html en CSS zijn beide minder zwaar dan javascript

Html and CSS tricks [h3]

There are many features in standard HTML and CSS that, when used correctly, eliminates the need for JavaScript.

Custom switches [h4]

If you want to make a custom switch, start with appearance: none; This resets the standard filled in by the browser while keeping its functionality intact. When redesigning think about the focus-visible attribute for navigating the page with a keyboard.

In page transitions [h4]

Accordions and models [h4]

The parent selector [h4]

<h4>CSS</h4>

<p>I have learned a lot about style and CSS in this minor, but I think there is more to learn. I wish to

learn CSS in a different way, as for now I have been approaching it as a canvas without limits. I

realise this is probably not how that works in the real world, and in the master assignment i would like

to hone my skills towards them being useful in the real world</p>

<h4>Working with teammates</h4>

<p>A part of the Hackathon that I found most difficult was working together with my teammates. Partly

because I had never collaborated on such a project before. I also found it hard to understand what my

team was doing, and felt I was making things without thinking. I would like to change this and do better

during this assignment. </p>

<h4>Professional presenting</h4>

<p>I prefer a casual style of presenting, in feedback conversations and also as end presentation. This makes

me confident and comfortable in my own skillset, however I do realise this isn’t always a good look. In

this project I would like to experiment with a more professional style of presenting and familiarise

I with it in a way that I feel confident presenting. </p>

weekly nerd #7

Inclusive design and accessibility [h2]

Practical fixes for common wcag problems [title, h2 or h3]

Erik kroes || 16-4-25[author]

There are 4 objects that are so often done wrong and yet so easy to get right. Spending seconds to think about these problems will make the web more accessible. An accessible product is a product that’s usable by everybody, including people with disabilities.

A hole in the wall is technically accessible because people can get inside. [quote]

Keyboard

You should be able to use tab and shift-tab to get to all interactive objects on the page. A link to skip the navigation when necessary, saves minutes of time and repeated information. A controlled focus state makes the website clearer. And remember the biggest sin of all: a div with a role of button.

DO NOT div role is button

This makes the whole thing inaccessible by keyboard.

Contrast

You might think contrast is not as much of a regular problem, but when the sun shines it becomes a problem for everybody.

The contrast ratio should be 3:1 for large texts (24+ px) and bold texts (19+ px). All other text should at least have a contrast of 4.5:1.

[examples?]

Alternatives

Captions, transcripts and text alternatives for images. These don’t have to be in the code, if you add them to your site it works for everybody. Even when there is no audio, it is very useful to know this! Common problems are images in buttons or link, and video content.

Sizing

The internet should be functional in all shapes and sizes, even if your screen is 320px. External content like cookies and popups don’t think about this often, so keep in mind if you really need these services.

Thinking about the accessibility box [title, h2 or h3]

Nienke de Keizer || 16-4-25[author]

In my internship at GVB I noticed the app is not as accessible as it should be, so for my master assignment I decided to research and redesign this, focussed on people with limited with sight.

“Accessibility is made for a small group, but they matter”

A very practical example of the new technology being inaccessible to people with limited sight is the new electric busses. To call on the busses previously, people would listen to the diesel motors to announce the arrival of the bus. Now that those are done, a new system has been implemented in the signs that calls out the arrival of the busses.

Previously, the ReisAssist app was made for the for the visually impaired. In the long term, this app might replace that or the outdated techniques that are in the current app.

Nienke de Keizer has introduced accessibility to GVB.

Weekly nerd #2

Pay the web forward with web monetization

Peter-Paul Koch 26-2-25

The decision for the web to be free was made in the late 90’s. This has become a bad habit that we should alter. The way people currently make money of the web is through the use of ads. This has two effects: The user thinks they don’t have to do anything, and all users only get the results rich people and companies want us to see.

What do we want?

As website developers, we want our work to be appreciated and valued correctly. Currently, a good site costs a lot of work and gains barely anything. If visitors give money from their own initiative developers will be encouraged to make their sites better, so users stay on the site longer or visit more often.

How can we do this?

* We need to set up an online bank account, directly connected to a real bank account. For this we need to be able to verify our customers identity
* We need a browser extension to manage this. We are working with multiple browsers to integrate this into the browsers themselves.
* We need to make a standard amount of money paid and when. The customer must be able to change this.
* We need to comply with international rules. Currently, they EU can only pay for EU-based websites, same as the USA. We need to establish a chain of trust between the organisations.

Support

A link tag in the head of the HTML, being checked by the browser extension. An initial signal is sent to establish connection. Afterwards the money will be sent by either the streaming method (sent the money live for) or post-paid (only sent the money afterwards, with a good internet connection)

Weekly nerd #5

Hackers United

Rosa

What is a Hacker? Somebody with a hoodie and sunglasses that types green numbers on a computer all day? – no! A Hacker is a person skilled in Information Technology who achieves goals by non- standard methods.

What happens if…

You just try stuff? Something wonderful might happen. I myself tried making image magic: photoshop with code. The process is weird and clunky, but the results are satisfying, because I made it together with the machines. I make all my posters using this method now.

[ad?]

(Un)repair café

Want to try it out for yourself but not sure where to start? Come join us at the un-repair café! Where we fuck around and find out, together!

* Rotterdam
* De versterking
* LAG Amsterdam

Weekly nerd #8

About io in general?

The rise of AI powered voice interfaces – Dave bitter 15-5-25

Voice recognition has existed and been developed since 1950. This has been slowly changing the way people find information. From having microphone detection in your search bar to now whole AI’s and applications that are built on this concept, the developments in the last few years have been fast and successful. The newest AI tools all contain 4 key parts.

* Speech recognition - API
* Speech synthesis - API
* Connection with the AI – what is said and what is the response
* Personality and feedback to the user – what does the user see

But it doesn’t feel quite natural yet. Using another AI, we could imitate a human speech pattern with intonation. However, the step to send the information back and forth as a whole creates big delays, making the conversation slow and unnatural.

To fix this problem I cut up the response of the AI in separate sentences, and ‘humanised’ the information while it was speaking, thus removing some of the delay and making the conversation more fluent.

“AI is just another data source”

Twitter follow bot and follower scraper – Steve Jonk 14-5-25

I once got asked by a marketing bureau to make a twitter bot that follows all followers of another account. Let me show you how I did it. Disclaimer: this is not entirely legal.

Tools

* N8n – automiser of a workflow
* Puppeteer – browser automation
* Twitter API
* Postgreas – database
* Docker – containerisation
* Ubuntu – server

Obstacles [ make (longer) text of this]

Making and using a Bot is against twitters terms of service. So I had to make my bot seem as it was not a machine, and to act like a human. I will follow, pause and after a randomised amount of time unfollow or follow the next person.

The next obstacle I encountered was how to get the data. Twitter has an API of themselves, but to use it to the extend that I needed it to the cost would be in the five figures. So, I used a scraper to get the publicly accessible follower information of the internet. However, this information is only available on different pages. So again, randomise and pause.

* No bots on twitter (How to act like a human, randomising and pausing)
* Twitter API be $$$$$ (use a scraper to find followers)
* Twitter VS the scraper (to find the way around)

Final product

In the end I made a working product that took longer than expected, is visually supported by n8n and is borderline illegal.

Conclusion: it can be very lucrative to work for yourself in this field.

How do we automate testing your components like a real user- Clarke Verdel 14-5-25

Weekly nerd #6

Geld verdienen met project FUGU

Niels leenheer || 9-4-25

Project FUGU Makes Native web applications, things that work in the browser and on the Internet. How do you do this for a register?

Analog appliances have different end points for different functionalities. Which was originally probably a great idea, but now we need to figure out which endpoint to send the information to, and in which language.

ASCII

A lot of the languages are build on a ASCII language system. This means letters numbers and symbols are placed on a grid, with the y axes being first and the x axes being second. In the below example this would entail that A has the sign 41.

[image]

Then, you have the issue of which type letter to use, and how to tell the machine this. For this, we use commando (esc) language.

[code blurp] Esc E 1

In this, we start the command with the esc sign, which tells the machine a code is coming. The second sigil is what we want to interact with, in this case E for emphasis, bold text. The 1 means we put in on, a zero in this would end the emphasized line.

What about images?

Well, if you take an image, convert it to black and white, then place the pixels in rows of three, then translate it though some codes it might just about work. If you need to do this, I wrote a library for it. Saves a lot of headache.

“If somebody says – you don’t need another receipt printer- Ignore Them!”

The restrictions of the 70’s make for some interesting choices, but lucky for you there exist libraries now! You don’t need to do this the way I did. Please don’t. Save yourself the headache.

Week 8 pt 3

How do you automate testing your components like a real user

Clarke verdel || 14/5/25

What if… we can ship software that just works

[testing trophy image]

Static testing

* Use Lint to check your best practices
* JS type checking
* Type script -> compatibility in output

Unit testing/ Component testing

* Use code to test code
* Input fields and smaller parts

Integration testing

* There are lots of parts, so lots of tests that must be done

E2E tests (end to end)

* Visual environments
* Testing everything together

Confidence in your code

There is only one valid way to gain confidence in your code: lots and lots of testing. I recommend these:

* Node runtime
* Synthetic DOM
* Browser testing (my favorite ☺ )

Using these methods, your work becomes more accessible, and write with a higher confidence

Utilizing these tests ensures accessibility of your code,.

“a good developer is a developer that can test”

Week 8 review

I really liked the tree talks we had at IO. It really shows the advancement of technology and web development, and the growing number of ways we can utilize it. The first two talks gave insight on how a web developer works, their thinking process and actions. The last one gave insight about the importance of testing during the process of creation, with tips and best practices on how to do so.

Week 9 review

Week 10 review

In the previous weeks, we heard so much about the new rules, but barely anything about what they actually say or how they will be handled. Having concrete examples from someone who knows and works with it directly brings much more clarity than hearing it second-hand.

Week 11 review (pre css day)

Pt 1

We already had this talk with Nils at the start of the minor, but hearing it again with more knowledge was definitely beneficial. You notice new details and topics that interest you more now. I already used some of the padding structures displayed in a project.

PT 2

Miriam's talk was quite interesting and very fast-paced. She shared many examples, jumping rapidly between topics — so many that her main point became somewhat lost. I do think it’s valuable to reflect on where we started and the project's original intention. Comparing it to today's standards shows that there’s still much we can learn from the past.

Reflection Master test

Mening

Zwaar en intensief

Niet weten of wat je doet goed is

Opdrachten team

Learning goals

CSS layout and structure

During this project, I challenged myself to use and research new grid and flexbox techniques. It was in this project that I finally managed to get grid-area working. I had tried to implement this in projects before but had never succeeded.

I based my grid on Liam's design. In that design, the title of the card was part of the relation box, and the idea was to have it separate from the other items in the HTML. However, due to the way grid works, these items would stack and not order themselves in a box. Due to this, I decided to alter the design slightly so the title would be at the top, separate from other relations.

During this time, Matthijs was working on a component similar to mine. When he wanted to try his hand at learning grid and making a grid for the component, I was able to help him get set up with an explanation, some useful sites, and a few sketches.

For the mobile-scaled version, I debated using container queries vs media queries. However, since only the card itself needed to scale, and not its contents, I opted for using media queries.

During my discovery, I also came across the columns property, which automatically sorts items in columns, with scalable widths, lines in between, and a whole lot of other solutions. I didn’t get to use this in the Master Assessment, but I did use it in the Weekly Nerd (see the wide articles).

Team Collaboration

During this second team project of the minor, I wanted to be more of an active member of the group. I do think I managed to do this, though I found it difficult to share my ideas and point out flaws I noticed. To overcome this social obstacle, I started writing things down, either fixing them myself or bringing them up later. This worked for me, opening communication and not causing excess stress.

Communication within the team during the project was clear. In the first weeks of the project, people started working on components, and if there were changes for that part later, the original person fixed it. This way, no one would overwrite each other's work or cause conflicts.

I also made an effort to check in with my teammates more often, to see what they were working on and to stay up to date on the latest decisions. I also informed teammates of any decisions I made that could affect their work, to prevent unexpected issues.

Proffessional Documentation

To complete my goal of improving my documentation, I made sure to update my product biography weekly. To do this accurately, I made small updates throughout the week, writing bullet points for different tasks and topics that I knew I would expand on later. At the end of the week, or at the start of the next, I reviewed all the bullet points and rephrased them, incorporating everything I had learned during the week.

Because I kept my biography up to date, many people — teammates and others — copied my layout. I told them this was fine, as it demonstrated that I had created a professional and well-structured biography.

I also made sure to document our talks with the client. I took notes during the presentations and later added them to the wiki, so the whole team could review and access them. This ensured we wouldn’t forget any important points and allowed us to acknowledge items that were discussed but not included in the project.

Design rationale

Accesibility

When I started this project, I wanted to focus on accessibility, but I found it difficult. The first weeks were mostly spent on design and planning. We chose to focus on visual impairments, specifically blindness and partial sight. However, due to the nature of the project, we couldn’t exclusively target people with those disabilities or create a fully HCD-level solution tailored to them.

Even so, while making our design, I wanted to make it as semantically correct as possible. This worked initially, but with multiple people working on designs, the complexity of the Liquid file structure, and time pressure, we missed some obvious issues—like inputs without labels.

While writing the structure of the HTML, I wanted it to be as semantic as possible and even go the extra mile. We ensured each page had a manageable number of links, placed the title first in the HTML order (even though the initial design called for it elsewhere), made the entire relationship card clickable instead of just the title, and used progressive enhancement to hide extra relations.

I tested a lot during the creation of the site. I used different tests, like a screen reader and colorblind extensions, to simulate different conditions. I discovered several issues this way, such as duplicate buttons and inconsistent icon titles.

I also paid special attention to the color palette. I checked contrast levels and the distinguishability of colors for different types of colorblindness. The current official FF colors mostly passed the contrast check but failed the color-differentiation test. I decided to use a new palette based on a professionally made set for colorblind users. One of the colors didn’t have enough contrast with black lettering, so I replaced it to meet both criteria. I also added a legend to make the color coding clear for users.