Getting familiar with the Console

Debugging and Chrome Dev Tools

Who am I?

- Dannie Håkansson, 27 years old
- Studied backend Java at YHC3L 2013-2015
- Started out working with Frontend in 2015, but slowly moved towards backend
- Since march I work under my own company, currently contracting for Klarna doing mostly React and Node.
- Love working out, meditation, video games, meeting new people
- Prior colleague of Reda ;)

The agenda of today

Before lunch

- Introduction of debugging and the console as concepts
- Walkthrough of debugging from different angles
- Opening the console and going through the “must know” features

After lunch

- Hands-on exercises located in GitHub to get you warm and familiar with the tooling
- I’ll be available of course!
- The slides of today will be shared after the lecture

I’m here to help. Asking is the first step :)

Any questions, anytime!

Let’s start

What is the console?

Also known as the “dev tools”

It is built for you as the developer to have control over your development workflow

You write software, and then you run it. The console is there to help you manage that workflow with ease

It is accessed through ***ALT + CMD + I*** on MacOS.

What power does the console give you?

Debugging your programs

Inspecting applied CSS and HTML elements, visualizing of the “box model”

Viewing website in other viewports

Inspecting all assets downloaded to your website(images, stylesheets, videos, scripts)

Inspecting network requests made(with ***fetch*** for example)

Profiling performance of your program(speed, memory usage)

Inspecting cookies, local storage etc

When is it helpful to open the console?

Why is my website so slow?

Why is this particular image loading way slower compared to all of the others?

Why is my function not behaving the way I’m expecting?

Why doesn’t this layout change when I view this page on my phone?

Why doesn’t my button turn red when I hover it with my mouse?

For live-editing CSS styles to get quick visual feedback

Basically any question related to how your website behaves functionally or visually can be answered by using the tools of the console

As the websites you develop increase in complexity, every tool that can help you verify and troubleshoot is a godsend

On to debugging 🐛

What is debugging?

Wikipedia:

“In computer programming and software development, debugging is the process of finding and resolving bugs (defects or problems that prevent correct operation) within computer programs, software, or systems.”

Thomas Edison, 1878:

“It has been just so in all of my inventions. The first step is an intuition, and comes with a burst, then difficulties arise—this thing gives out and [it is] then that "Bugs"—as such little faults and difficulties are called—show themselves and months of intense watching, study and labor are requisite before commercial success or failure is certainly reached”

When and why is debugging useful?

When you expect something to work but it does NOT

When you expect something to NOT work but it DOES 🤔

To educate yourself on the internals of JavaScript or third party libraries you might be using

Whenever you’re curious. It’s important to stay curious!

What type of errors/bugs are there?

Syntax error - shows red in the console BEFORE the program even starts

Runtime error(Exception and Error) - shows red in the console while the program is executing some functionality

Logical error - Disguised as correct behaviour. According to JavaScript and the browser, everything is working perfectly fine.

How does browser debugging work?

First, a process needs to be running your JavaScript program in the browser

Another process then “attaches” to the running JavaScript process mentioned above. This other process is attached automatically and is what we call the “debugger process”

The first process is aware that it is being “debugged”, and communicates with the other process accordingly

What the browser debugger can see and do

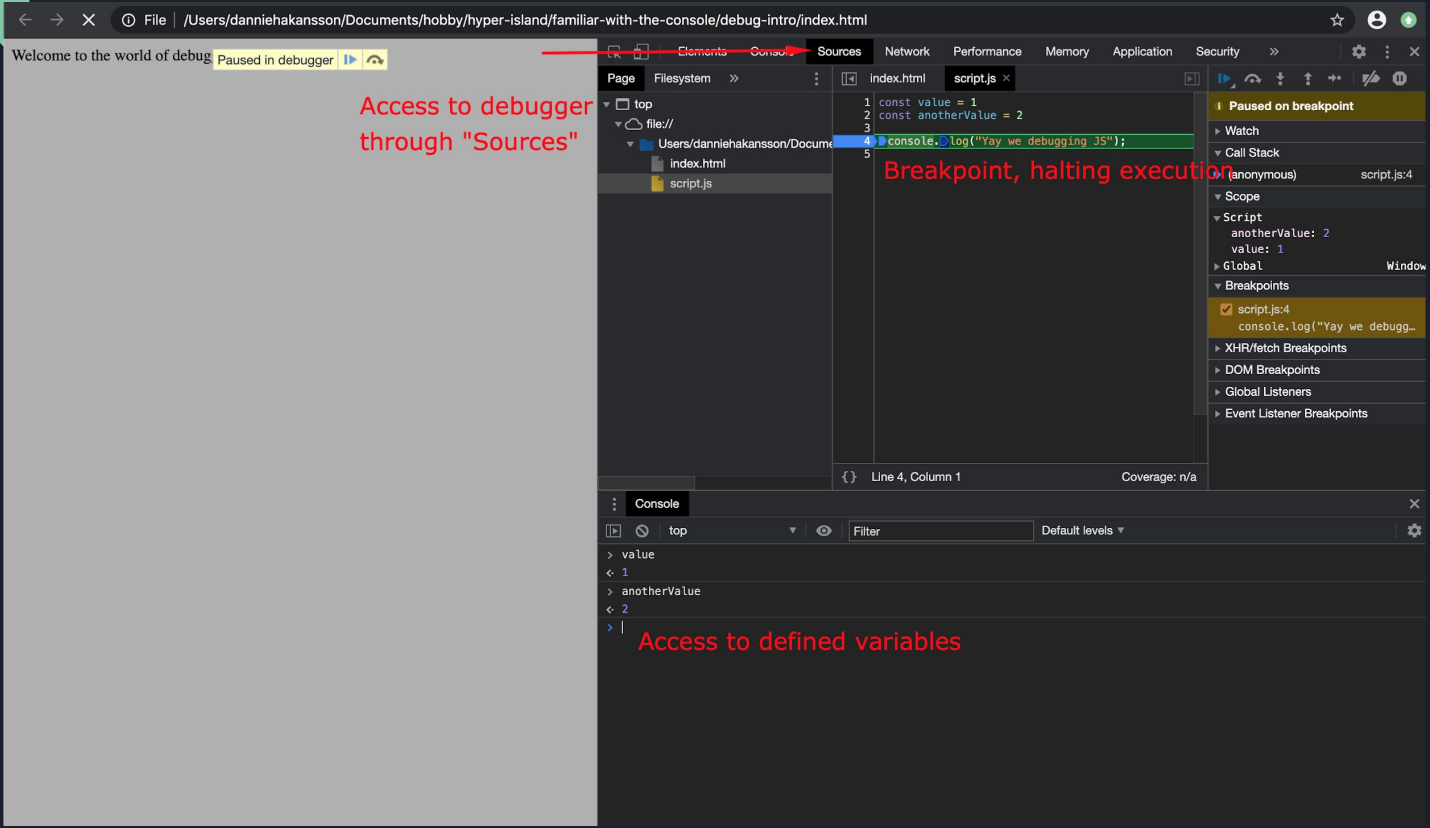
Evaluate variables, functions, expressions

Halt execution at certain points called ***breakpoints***. You as the developer will create those breakpoints. *“At line 10 I want to stop and see if the program is behaving as expected”*.

Moving execution forward only when you as the developer tell it to do so.

Tracing back the control flow that put you in the position you’re at now. *“Function A => Function B => if statement A => Function C”*.

How does browser debugging work?



/Users/danniehakansson/Documents/hobby/hyper-island/familiar-with-the-console/debug-intro/index.html

Welcome to the world of debug

```
const value = 1
const anotherValue = 2

console.log("Yay we debugging JS");
```

```
(anonymous)    script.js:4
Script
  anotherValue: 2
  value: 1
Global    Window
script.js:4
console.log("Yay we debugg…
```

```
> value
<· 1
> anotherValue
<· 2
```

A note on ***console.log()***

You have probably used this function to verify assertions and evaluate expressions.

If you sprinkle ***console.log()*** all over your source code in order to verify that it works properly, then you've done work that could be done with the built in debugger. This is called "print debugging" and is often used by developers not comfortable enough using built in debugging software.

Bugs are introduced to programs because the source code was changed, so naturally "minimizing change = minimizing the number of bugs" is true.

This does not mean that ***console.log()*** is **bad**, but it can be misused.

Once you get familiar with the different techniques of "gathering data from your program", then you'll be comfortable deciding which one is suitable for the given situation

The idea is not to keep everything in your head at all times

Let’s check out some features of the console!

/1-intro-to-debugging

/2-fixing-a-problem-with-debugging

/3-inspecting-elements

/4-network-requests

/5-performance-testing

Questions? 🤔

Lunch break 🌮🌭🍔

Hands on - More debugging

Download or **git clone** the repository from GitHub

Open the repository in your editor/IDE

Let’s start!

Remember to let the tools do their job!

/1-debugging-problem

/2-debugging-problem

/3-debugging-problem

/4-debugging-problem-bonus

Thank you everyone 💃

Want to connect with me?

danniehakansson.com

linkedin.com/in/danniehakansson/

danniehakan@gmail.com