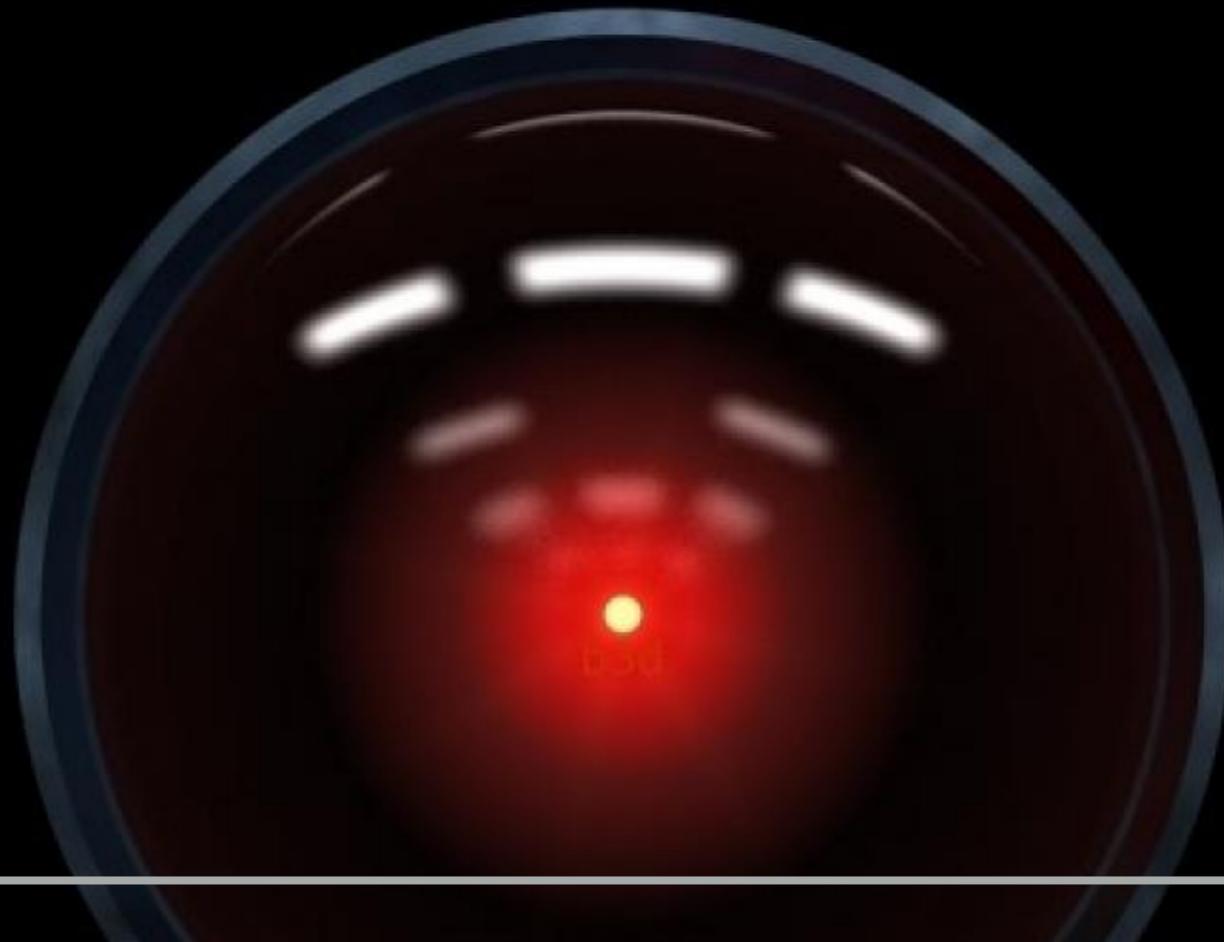


[HTTP://WWW.SIMONWELLS.ORG](http://www.simonwells.org)
[HTTP://ARG.NAPIER.AC.UK](http://arg.napier.ac.uk)



DR SIMON WELLS

THINGS TO DO IN THE BROWSER WHEN YOU'RE BORED - A
TUTORIAL INTRODUCTION TO USING YOUR WEB BROWSER AS A
DEVELOPMENT ENVIRONMENT FOR FUN, 80S STYLE, DAY-TO-
DAY HACKING

WHY ARE WE HERE?

This is not meant to be an existential question

WHAT IS PROGRAMMING?

- ▶ Telling a computer what to do [solving problems]
 - ▶ Identifying parts of the solution [data]
 - ▶ Working out how to handle each part [algorithms]
-
- ▶ (*increasingly*): Giving the machine the ability to find/learn the solution [AI]
-
- ▶ Telling others about your solution [communication/persuasion/teamwork]
 - ▶ Writing (Sorry [not sorry] ;)
 - ▶ Also: precision, accuracy, attention to detail

WHAT IS PROGRAMMING?

IF YOU CAN BAKE A CAKE/PREPARE A POT NOODLE/PUT UP A PICTURE/WIRE A PLUG/FIX A PUNCTURE/LIGHT A FIRE - THEN YOU CAN PROBABLY WRITE A PROGRAM...

HOW DO I BECOME A (GREAT) PROGRAMMER?

- ▶ The programming genius
 - ▶ Just knows how to do it
 - ▶ Doesn't exist, probably
 - ▶ Hollywood has a lot to answer for :(
- ▶ Can read/follow a book/article/tutorial & I'll get it
 - ▶ Only part of the answer

HOW DO I BECOME A PROGRAMMER?

MYTHS

- ▶ Hard work & Effort
 - ▶ (but this can also be a lot of fun)
- ▶ Deliberate Practice (over time):
Thinking -> Doing -> Reflecting
- ▶ There is no magic.

HOW DO I BECOME A (GREAT) PROGRAMMER?

THE TRUTH?

PROGRAMMING IS A
LIFESTYLE CHOICE

Write lots of programmes

BIO

- ▶ First Computer (age 7)
- ▶ Wrote some programmes (often from magazines & books)
- ▶ No real programming experience until university
- ▶ Nobody else in immediate family with a degree
- ▶ Nobody else with a higher degree at all (yet)
- ▶ Interested in ***everything...***



**IN SOME WAYS
MY EXPERIENCE
WAS EASIER**

- Immediacy
- Lower expectations
- Work with less

***** COMMODORE 64 BASIC V2 *****

64K RAM SYSTEM 38911 BASIC BYTES FREE

READY.

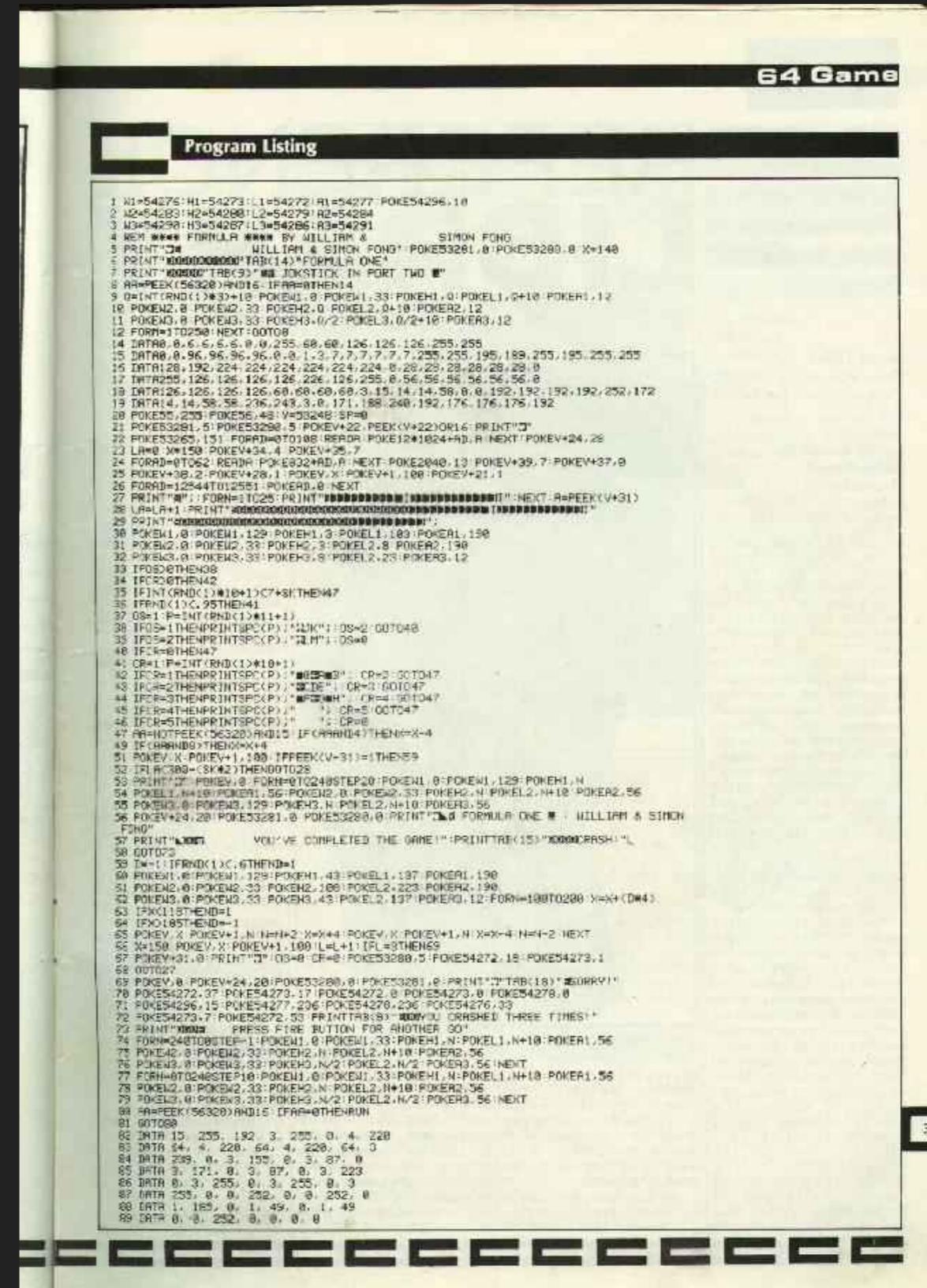
DOING ANYTHING WITH THIS MACHINE
INVOLVED PROGRAMMING:
WE COULD GET STRAIGHT TO THE
PROGRAMMING

:D



WORK WITH LESS

- ▶ A lot fewer programmers around
- ▶ Home computers were untrusted, unreliable, and just not a mainstream consideration (for kids, for games, for the future)
- ▶ No smart phones
- ▶ No Internet/Web (we did have bulletin boards & modems though & Magazines)

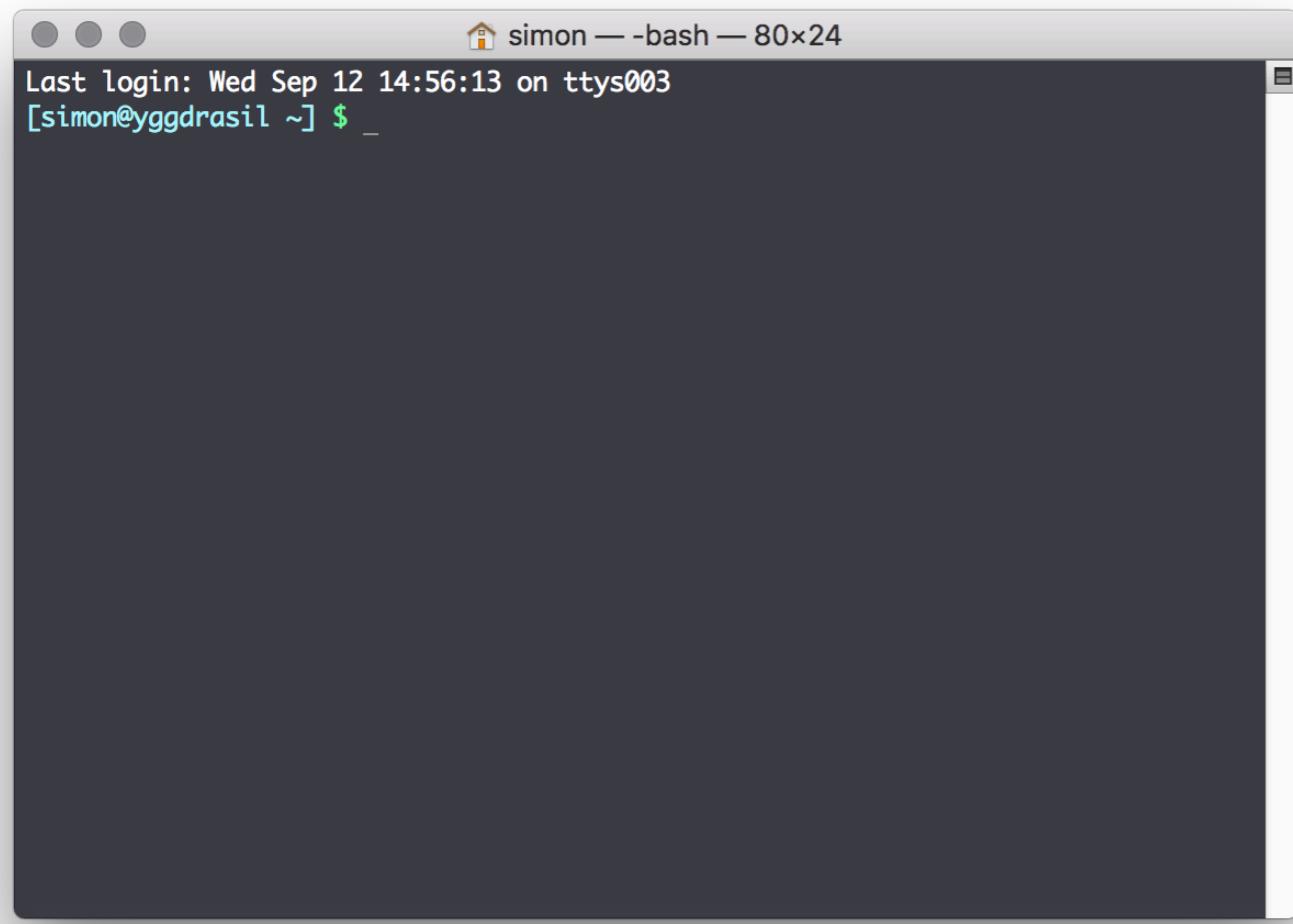


MODERN PROGRAMMING

- ▶ Getting into programming nowadays is easy:
 - ▶ Books, Web pages, Tutorials
 - ▶ Compilers, interpreters, IDEs
 - ▶ Nearly always have at least one computer on our person
- ▶ However:
 - ▶ Most computer experience is now point & click (or swipe)
- ▶ We see lots of really cool stuff but don't know how to get there from here
- ▶ **Bootstrapping is hard:**
 - ▶ there's lots of other stuff to do before you can start hacking away
- ▶ Also:
 - ▶ **What should I programme?**

- ▶ Modern computers aren't really set up to make programming accessible out-of-the-box
- ▶ Some hoop jumping: need to install programming language tools (compiler, interpreter, IDE, editor)
 - ▶ NB. Some computers already have these installed by default (Mac OS & Linux), e.g. python, ruby
- ▶ Not as straightforward as powering up the machine & getting dumped straight into a programming interface

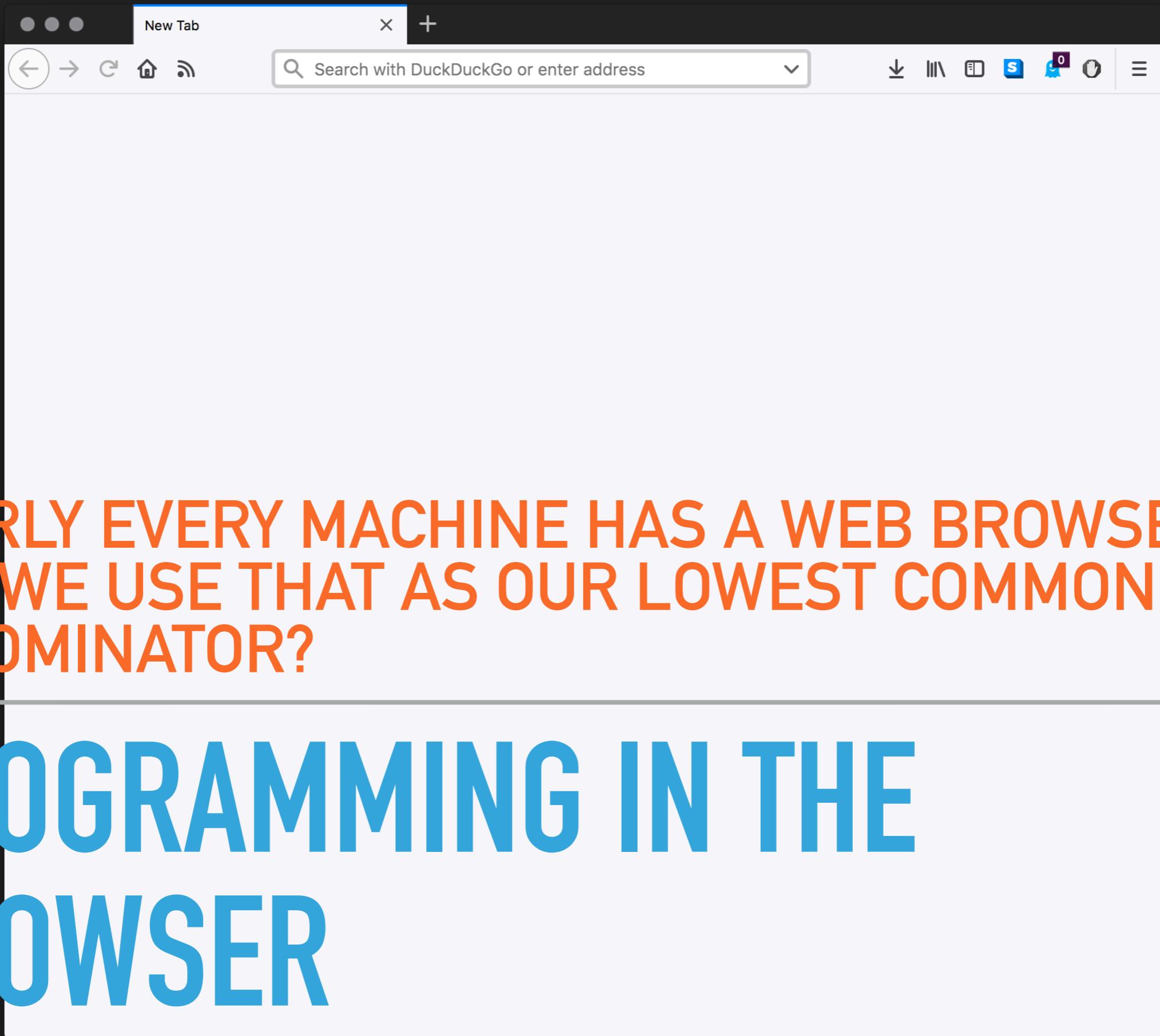
BOOTSTRAPPING IS HARD



```
simon — bash — 80x24
Last login: Wed Sep 12 14:56:13 on ttys003
[simon@yggdrasil ~]$
```

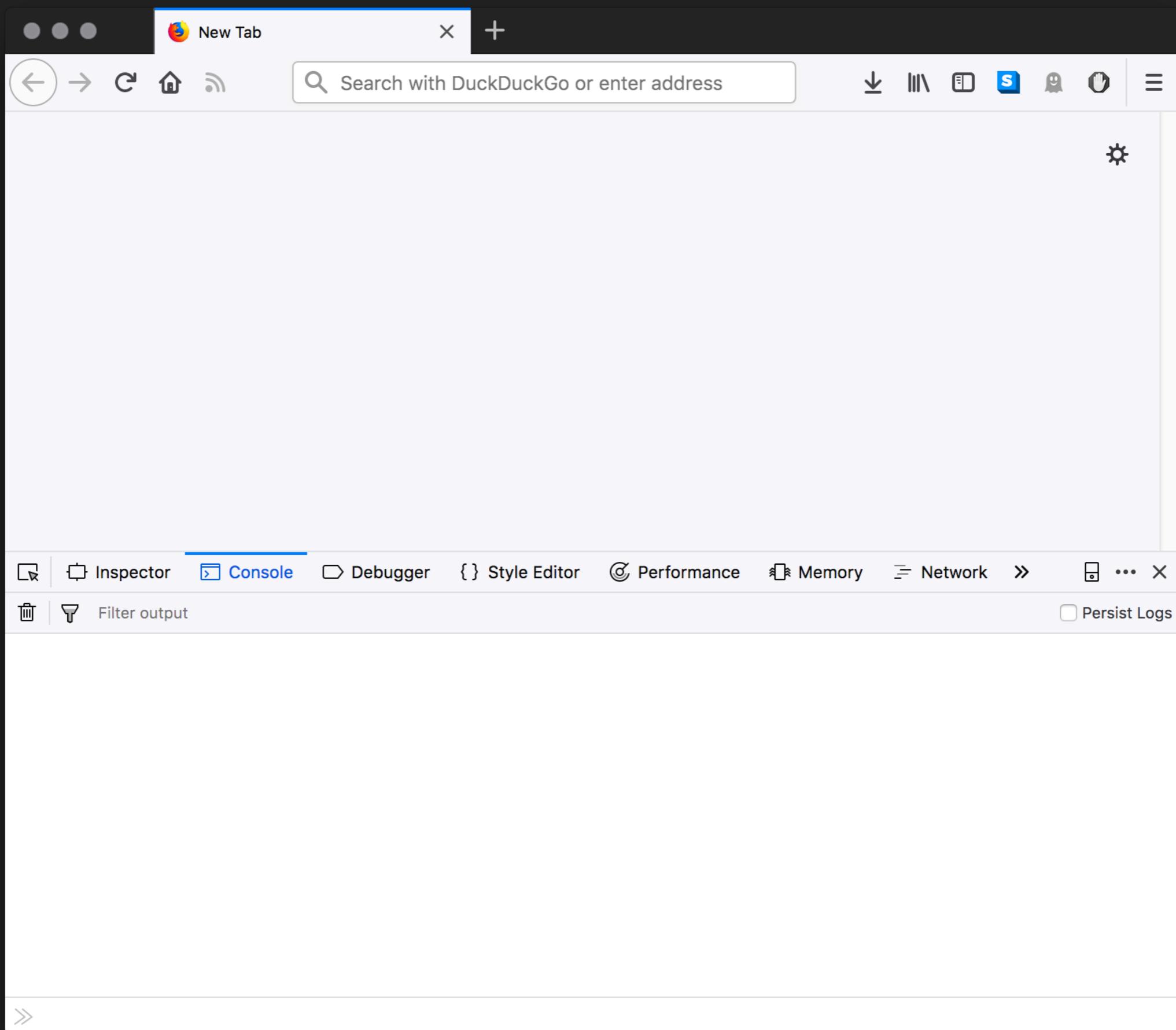
- ▶ Programming is a literate practise
 - ▶ If you only mouse around the GUI then life as a programmer is slightly more difficult
- ▶ CLI gives you the best, most fine-grained control of your computer
- ▶ Neal Stephenson "In the beginning was the command line"

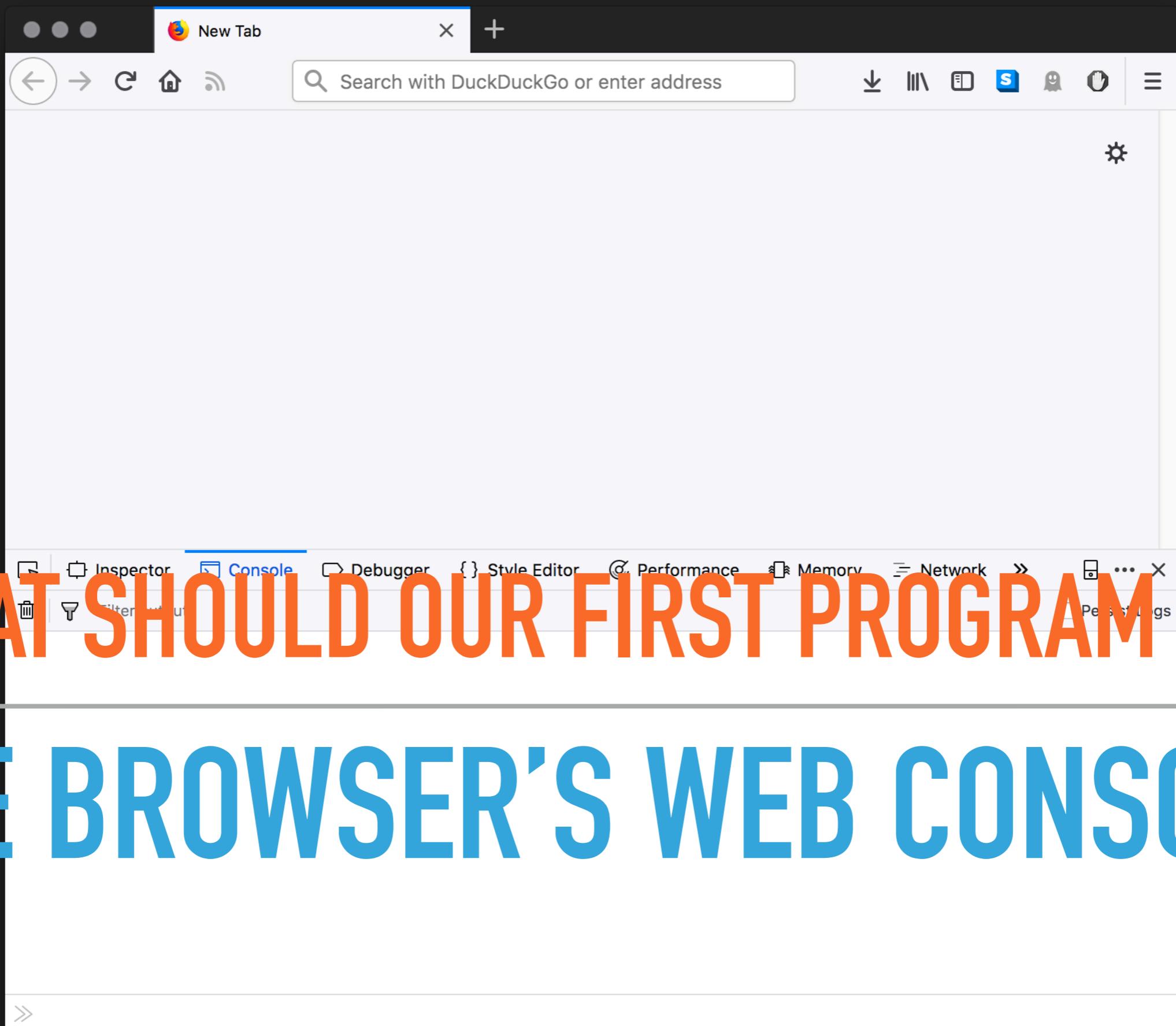
TIP: LEARN TO LOVE THE COMMAND LINE :)



NEARLY EVERY MACHINE HAS A WEB BROWSER -
CAN WE USE THAT AS OUR LOWEST COMMON
DENOMINATOR?

PROGRAMMING IN THE
BROWSER



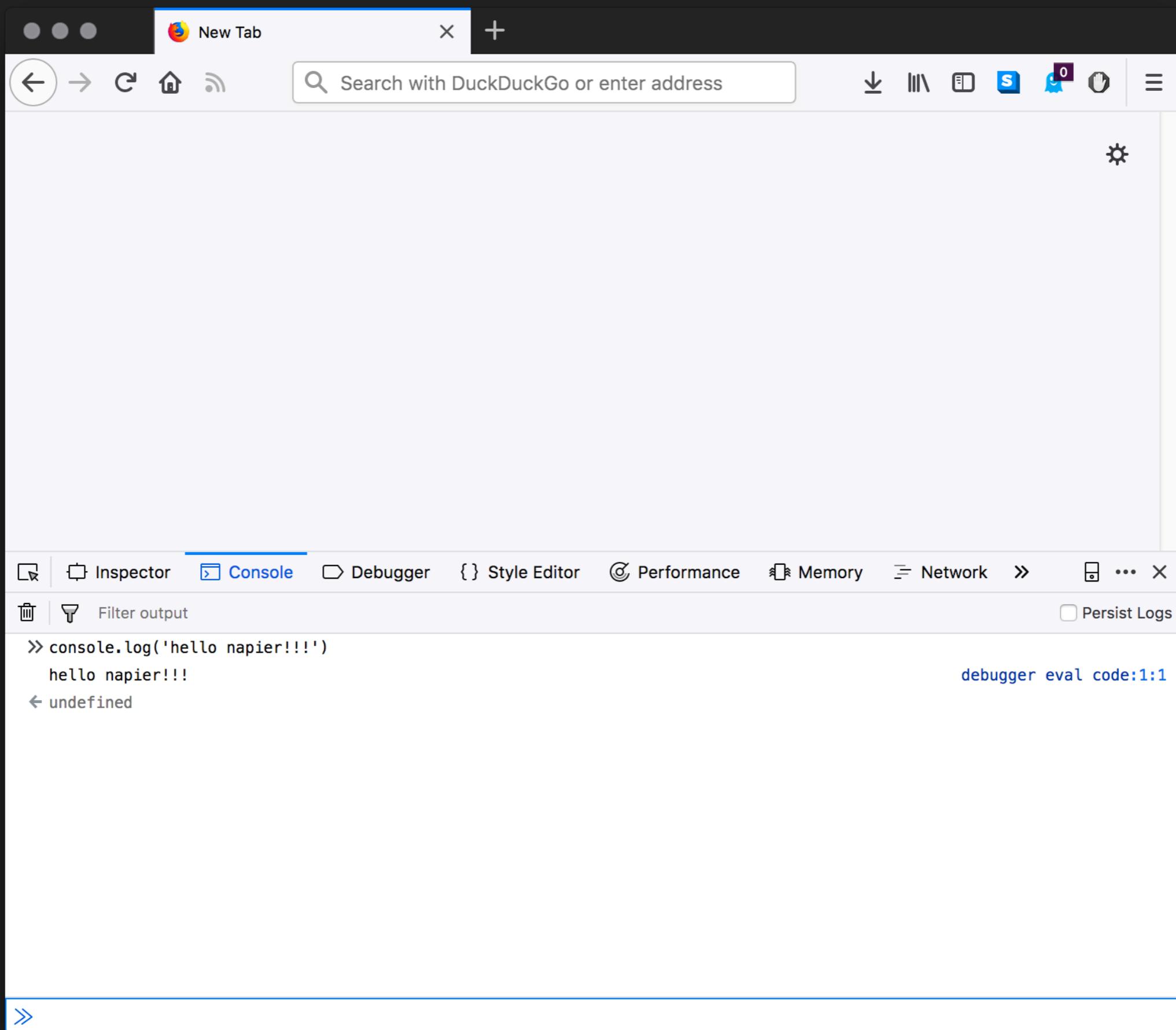


WHAT SHOULD OUR FIRST PROGRAM BE?

THE BROWSER'S WEB CONSOLE

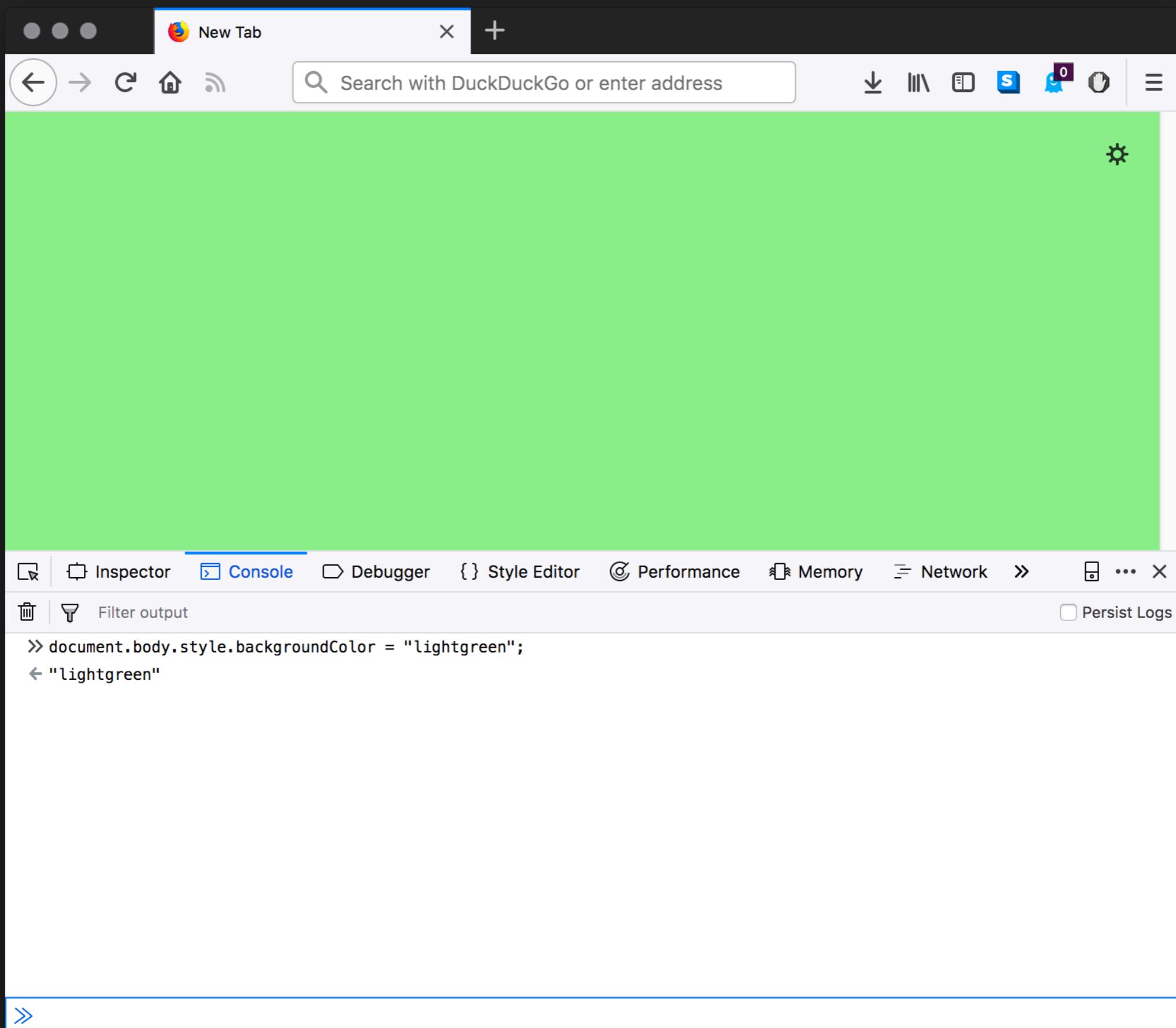
#1

HELLO NAPIER



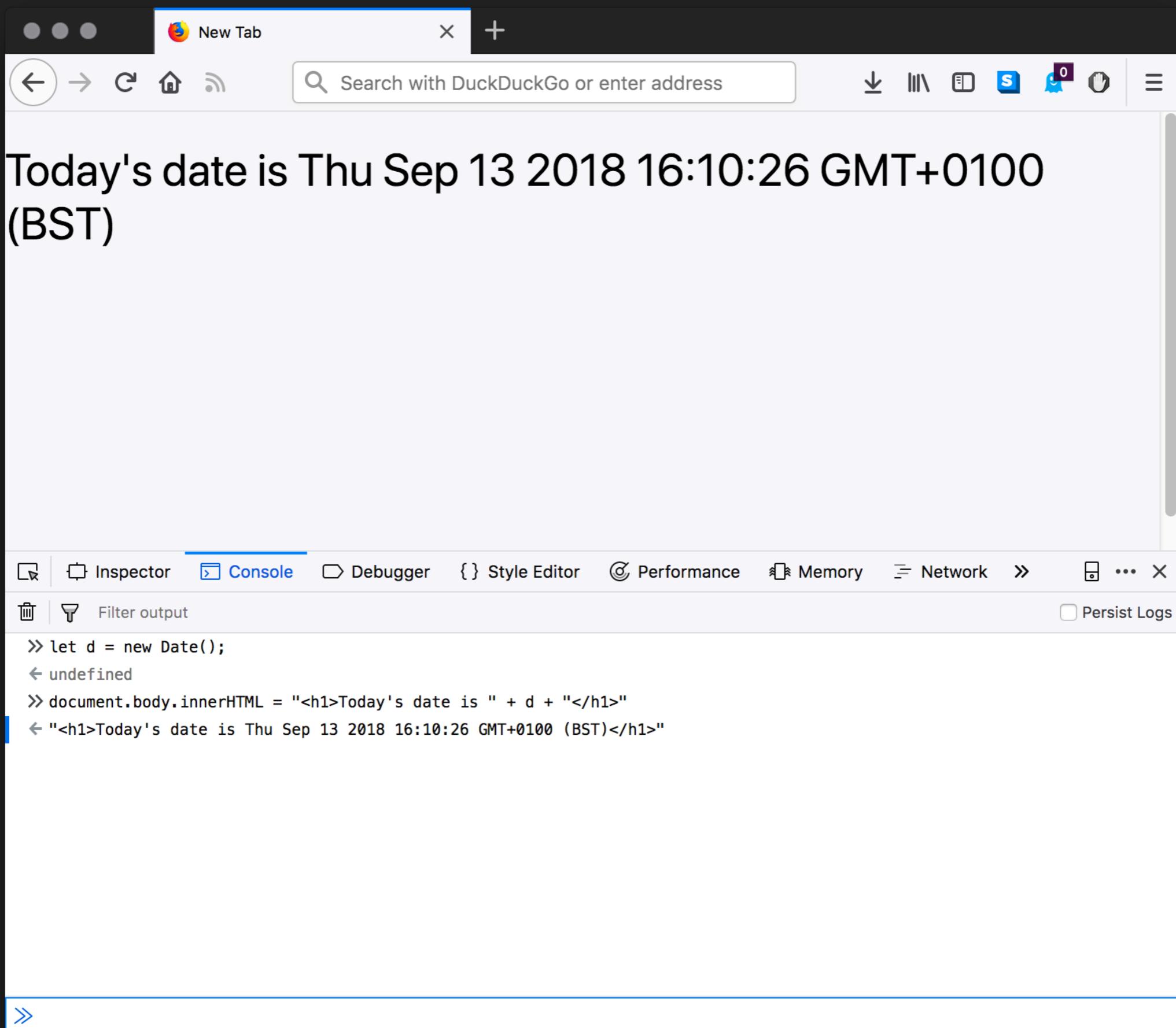
#2

INTERACT WITH THE WEB PAGE/SCREEN



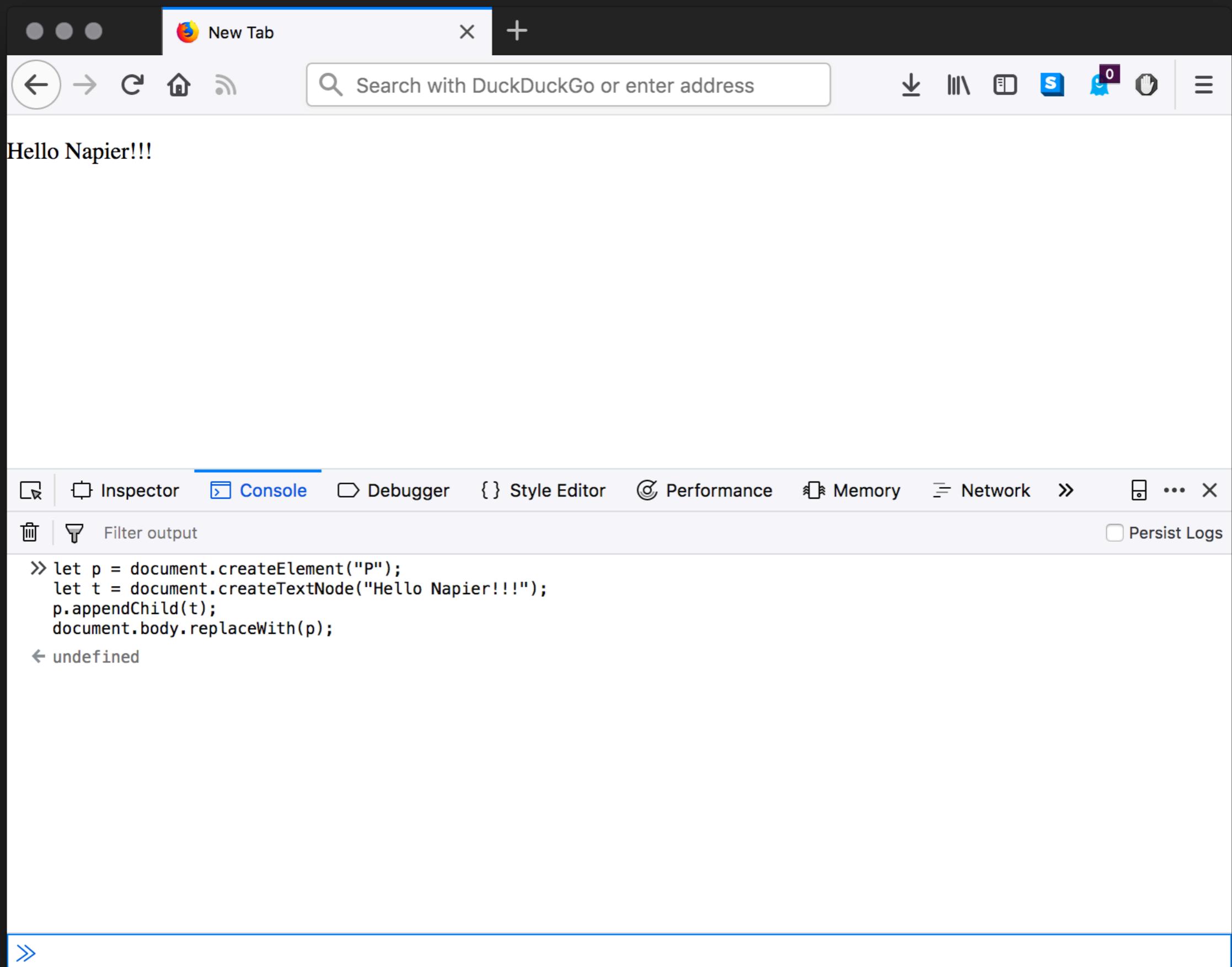
#3

USE STANDARD JAVASCRIPT FUNCTIONS



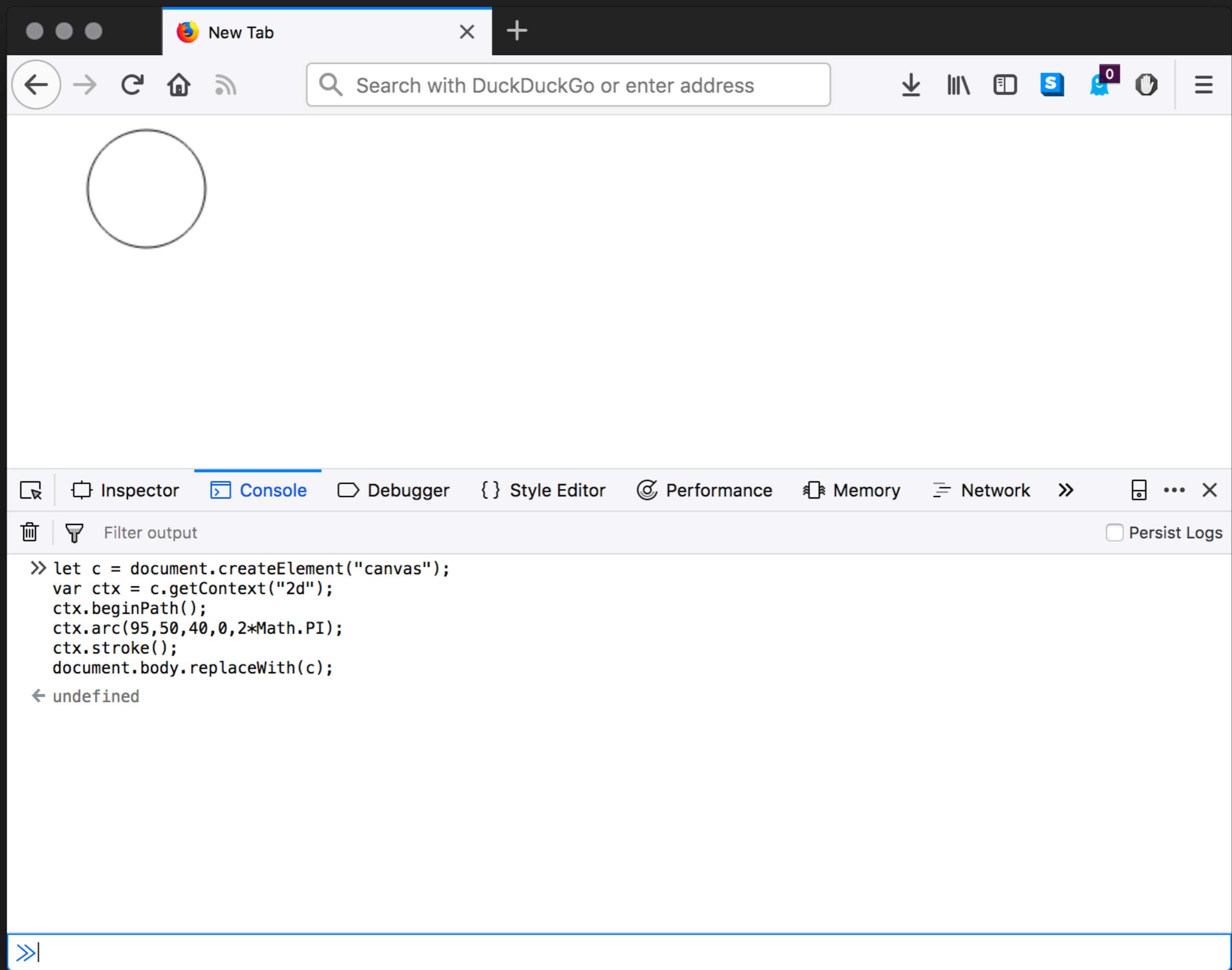
#4

CONSTRUCT A WEB PAGE



#5

GRAPHICS



#6

SOUND - BEEPS, BOOPS, CHIPTUNES

A screenshot of a browser developer tools window, specifically the Console tab, demonstrating a piece of JavaScript code that generates musical notes using the Web Audio API.

The code defines a frequency mapping object `freq` and iterates over a list of notes to create oscillators and play them. The notes are mapped to specific frequencies:

Note	Frequency (Hz)
'a4'	440.00
'a4#'	466.16
'b4'	493.92
'c5'	523.28
'c5#'	554.40
'd5'	587.36
'd5#'	622.24
'e5'	659.28
'f5'	698.48
'f5#'	740.00
'g5'	784.00
'g5#'	830.64
'a5'	880.0000

```
>> var context = new (window.AudioContext || window.webkit.AudioContext)();  
var freq = {  
    'a4' : 440.00,  
    'a4#' : 466.16,  
    'b4' : 493.92,  
    'c5' : 523.28,  
    'c5#' : 554.40,  
    'd5' : 587.36,  
    'd5#' : 622.24,  
    'e5' : 659.28,  
    'f5' : 698.48,  
    'f5#' : 740.00,  
    'g5' : 784.00,  
    'g5#' : 830.64,  
    'a5' : 880.0000  
}  
  
for (i in D = ['e5',, 'e5',, 'e5',, 'c5',, 'e5',, 'g5',, 'g5']) {  
    console.log(i);  
    var oscillator = context.createOscillator();  
    if (D[i]) {  
  
        onended = function() { console.log('Note has stopped playing'); }  
  
        oscillator.connect(context.destination);  
        note = D[i]  
        oscillator.frequency.setValueAtTime(freq[note], context.currentTime);  
        oscillator.type = 'square';  
        oscillator.start(i * .1);  
        oscillator.stop(i * .1 + .1);  
    }  
}
```

#7

SOUND - MUSIC (AFTER A FASHION)

A screenshot of a browser's developer tools interface, specifically the 'Console' tab. The top navigation bar includes tabs for Inspector, Console (which is selected), Debugger, Network, Style Editor, Performance, Memory, Storage, Accessibility, and more. Below the tabs is a toolbar with icons for download, copy, search, and other developer functions. A search bar at the top says 'Search with DuckDuckGo or enter address'. The main area shows a multi-line code editor with syntax highlighting for JavaScript. The code creates an audio context, sets up an oscillator and gain node, and adds event listeners for mousedown, mousemove, and mouseup events to control the oscillator's frequency and volume based on the mouse position.

```
>> var context = new (window.AudioContext || window.webkitAudioContext)();  
var oscillator = context.createOscillator();  
var volume = context.createGain();  
var active = false;  
  
volume.gain.setValueAtTime(0, context.currentTime);  
  
oscillator.connect(volume).connect(context.destination);  
oscillator.start(0);  
  
document.addEventListener('mousedown',function(e){on(e)})  
document.addEventListener('mousemove',function(e){play(e)})  
document.addEventListener('mouseup',function(e){off(e)})  
  
function on(e){  
    active = true;  
    e.preventDefault()  
    oscillator.frequency.setValueAtTime(~(1000*(1-((e.clientY)/window.innerHeight))), context.currentTime);  
    volume.gain.setValueAtTime(~(e.clientX/window.innerWidth*100)/100, context.currentTime);  
}  
|  
function play(e){  
    e.preventDefault()  
    if(active){  
        oscillator.frequency.setValueAtTime(~(1000*(1-((e.clientY)/window.innerHeight))), context.currentTime);  
        volume.gain.setValueAtTime(~(e.clientX/window.innerWidth*100)/100, context.currentTime);  
    }  
}  
  
function off(e){  
    active = false;  
    volume.gain.setValueAtTime(0, context.currentTime);  
}
```

- ▶ Nearly every computer has a browser so we can programme “old school” style almost anywhere at any time
- ▶ More likely to run against our own limitations right now than those of the browser/JS
- ▶ Can build simple hackery into our daily programming habits

WHERE ARE WE?

WHAT SHOULD I PROGRAMME?

- ▶ Good Question!
 - ▶ I've shown some simple things to get started
 - ▶ What are you interested in?
 - ▶ Key is to start small (remember the limitations & lower expectations I mentioned earlier)
 - ▶ We want to make small increments without biting off more than we can chew.
-

WHAT SHOULD I PROGRAMME?

- ▶ Codes & Ciphers
 - ▶ This is actually an assignment in my second year web tech class (so I won't spoil it here)
- ▶ Chaos, Fractals, Artificial Life, & Cellular Automata
- ▶ Procedural Generation
- ▶ Simple Games (text-based dungeon crawlers)

WHERE DID SIMON START?

- ▶ A grid of cells that can be on or off

Take a starting generation

Some cells on & the rest off

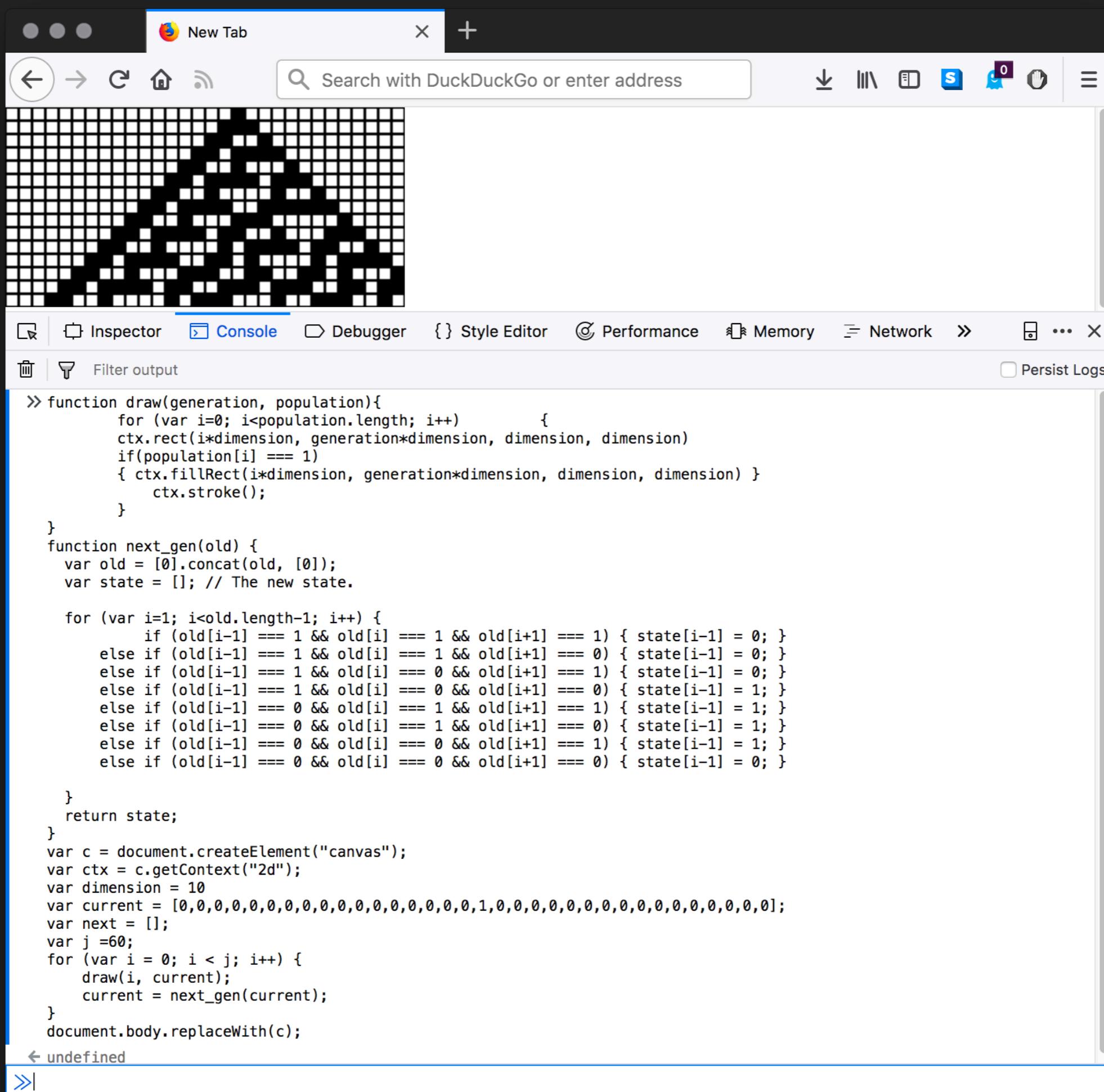
Calculate the next *generation* according to some simple rules & repeat

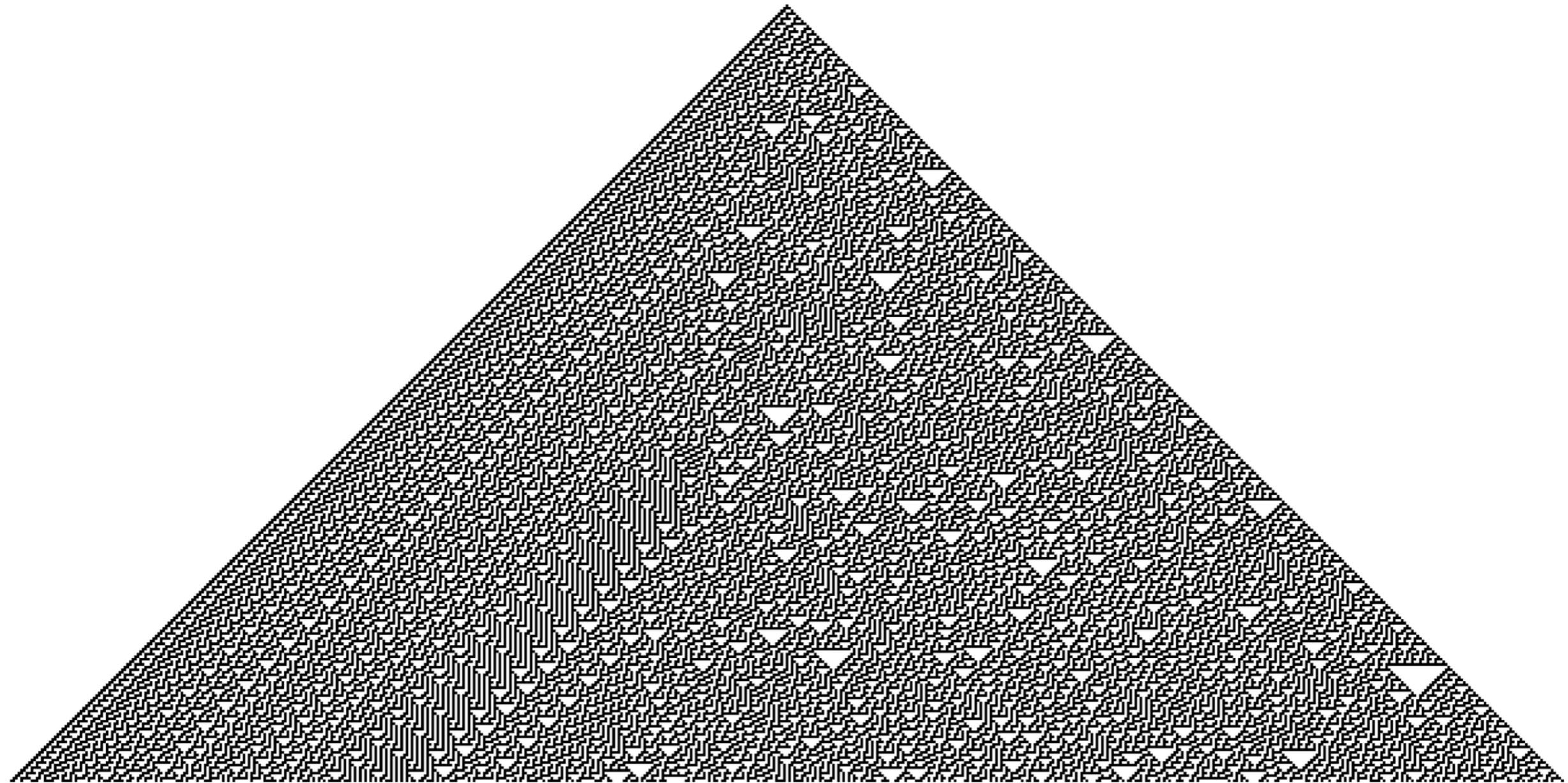
- ▶ Can lead to very complex, sometime chaotic, behaviours
- ▶ The CompSci bit: Some CA have been proven to be able to calculate anything that a regular computer can calculate

CELLULAR AUTOMATA

#8

1D CELLULAR AUTOMATA





RULE 30

current pattern	111	110	101	100	011	010	001	000
new state for center cell	0	0	0	1	1	1	1	0



SIMULATING THE
WORLD.....



- ▶ There are some places that collect programming problems & issue challenges:
 - ▶ [Project Euler](#)
 - ▶ [Stack Exchange Code Golf](#)
 - ▶ [Code kata](#)
- ▶ [Reddit Daily Programmer](#)
- ▶ [Programming Praxis](#)
- ▶ [Rosetta Code](#)
- ▶ [International Collegiate Programming Contest Problems Index](#)
- ▶ [Algorithmist](#)

I DON'T LIKE ANY OF THAT CRAP, WHAT SHOULD I DO?



**WHAT ELSE ARE YOU INTO?
YOU CAN FIND
COMPUTATION/
PROGRAMMING PROBLEMS
IN LOTS OF PLACES...**

IN SUMMARY

- ▶ Think small (until it's time to think big)
- ▶ Follow your interests
- ▶ If you don't have any interests then:
 - ▶ look around you | read more | steal from others
- ▶ Become a daily programmer
- ▶ Write LOTS of code
- ▶ Have fun

WE ARE ALL SMART HERE.
DISTINGUISH YOURSELF BY
BEING KIND.

RESOURCES

- ▶ Website for this talk:

<https://siwells.github.io/offer-holder-day/>

- ▶ Related website for this talk:

<https://siwells.github.io/READY/>

- ▶ Code for all of the examples (& more) is available here:

<https://github.com/siwells/READY/tree/master>

- ▶ If you want to find out more, these books are a good starting place for learning JavaScript:

- ▶ "JavaScript: The Good Parts" by Douglas Crockford
- ▶ "Eloquent JavaScript" by Marijn Haverbeke
- ▶ "The "You don't know JS" series by Kyle Simpson

- ▶ The MDN web docs site:

<https://developer.mozilla.org/en-US/>

