WHY ARE WE HERE?

This is not meant to be an existential question.
WHAT IS PROGRAMMING?
WHAT IS PROGRAMMING?

- Telling a computer what to do
- Identifying parts of the solution
- Working out how to handle each part
- (increasingly:) Giving the machine the ability to find the solution
- Writing (Sorry [not sorry] ;)

[solving problems]  [data]  [algorithms]  [AI]
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?
MYTHS

▸ 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?
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
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
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
HELLO NAPIER
`console.log('hello napier!!!')`

`hello napier!!!`

`undefined`
#2
INTERACT WITH THE WEB PAGE/SCREEN
document.body.style.backgroundColor = "lightgreen";
"lightgreen"
#3

USE STANDARD JAVASCRIPT FUNCTIONS
Today's date is Thu Sep 13 2018 16:10:26 GMT+0100 (BST)

```javascript
let d = new Date();
console.log(d);

document.body.innerHTML = "<h1>Today's date is \" + d + \\"</h1>"
```

"<h1>Today's date is Thu Sep 13 2018 16:10:26 GMT+0100 (BST)</h1>"
#4

CONSTRUCT A WEB PAGE
Hello Napier!!!

```javascript
let p = document.createElement("P");
let t = document.createTextNode("Hello Napier!!!");
p.appendChild(t);
document.body.replaceWith(p);

undefined
```
let c = document.createElement("canvas");
var ctx = c.getContext("2d");
ctx.beginPath();
ctx.arc(95,50,40,0,2*Math.PI);
ctx.stroke();
document.body.replaceWith(c);
#6
SOUND - BEEPS
```javascript
var context = new (window.AudioContext || window.webkitAudioContext)();
var oscillator = context.createOscillator();
oscillator.type = 'sine';
oscillator.frequency.value = 440;
oscillator.connect(context.destination);
oscillator.start();
```

```
undefined
```
#7

SOUND - MUSIC (AFTER A FASHION)
```javascript
var context = new (window.AudioContext || window.webkitAudioContext)();
var oscillator = context.createOscillator();
h = window.innerHeight;
oscillator.connect(context.destination);
oscillator.start(0);
document.addEventListener("mousemove", function(e) {
    oscillator.frequency.value = e.clientY / h * 1000 + 300;
});
```

undefined
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.
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.
WHERE DID SIMON START?

- 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
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
RULE 30

<table>
<thead>
<tr>
<th>current pattern</th>
<th>111</th>
<th>110</th>
<th>101</th>
<th>100</th>
<th>011</th>
<th>010</th>
<th>001</th>
<th>000</th>
</tr>
</thead>
<tbody>
<tr>
<td>new state for center cell</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
1D CELLULAR AUTOMATA
```javascript
function draw(generation, population) {
    for (var i = 0; i < population.length; i++) {
        ctx.rect(i * dimension, generation * dimension, dimension, dimension)
        if (population[i] === 1) {
            ctx.fillRect(i * dimension, generation * dimension, dimension, dimension)
        }
        ctx.stroke();
    }
}

function next_gen(old) {
    var old = [0].concat(old, [0]);
    var state = [];
    // The new state.
    for (var i = 1; i < old.length - 1; i++) {
        if (old[i - 1] === 1 && old[i] === 1 && old[i + 1] === 1) { state[i - 1] = 0; }
        else if (old[i - 1] === 1 && old[i] === 1 && old[i + 1] === 0) { state[i - 1] = 0; }
        else if (old[i - 1] === 1 && old[i] === 0 && old[i + 1] === 1) { state[i - 1] = 0; }
        else if (old[i - 1] === 1 && old[i] === 0 && old[i + 1] === 0) { state[i - 1] = 1; }
        else if (old[i - 1] === 0 && old[i] === 1 && old[i + 1] === 1) { state[i - 1] = 1; }
        else if (old[i - 1] === 0 && old[i] === 1 && old[i + 1] === 0) { state[i - 1] = 1; }
        else if (old[i - 1] === 0 && old[i] === 0 && old[i + 1] === 1) { state[i - 1] = 1; }
    }
    return state;
}

var c = document.createElement("canvas");
var ctx = c.getContext("2d");
var dimension = 10;
var current = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0];
var next = [];
var j = 60;
for (var i = 0; i < j; i++) {
    draw(i, current);
    current = next_gen(current);
} document.body.replaceWith(c);
return undefined
```
I DON'T LIKE ANY OF THAT CRAP, WHAT SHOULD I DO?

- 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
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
PROGRAMMING SURGERIES

- Staffed by experienced students
- Can assist in working through bugs, errors, misunderstandings
- Aim is to get you back on track with coursework & lab exercises
- Schedule (Weeks 02-15)
  - Tuesday, Lab MER_C06, 1pm-3pm
  - Wednesday, Lab MER_C06, 12pm-2pm
  - Thursday, Lab MER_C06, 12pm-2pm
- Tutors @ Tuesday session have a specific background in web technologies
- Will also support programming environment setup and virtual machines
WE ARE ALL SMART HERE.
DISTINGUISH YOURSELF BY BEING KIND.
RESOURCES

- 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:
  