Having Fun

BUILDING WEB APPLICATIONS

DAY 1

16 DEC 9AM-5PM

ABOUT ME

Clarence Ngoh

Year 2 SIS

Telegram: @clarencenpy

PREVIOUSLY...

HAVING *Fun*

building

Web Applications

How to draw an Owl.

"A fun and creative guide for beginners"

Fig 1. Draw two circles

Fig 2. Draw the rest of the damn Owl

Jonathan Shariat
@DesignUXUI

Follow

99% of #programming tutorials on the web

1:00 AM - 19 Jul 2014

8,208 5,490

what i hope to teach!

FRONT END?

practice of producing HTML, CSS and Javascript for a website or web application so that a user can see and interact with them directly through the web browser

WHY?

SE and beyond - you need an interface to your app
JS supports a great user experience that static pages can't
Easy to get started with so many libraries to leverage

WEB APPS?

a client-server software application in which the client (or user interface) runs in a web browser.

WHY?

> Write Once Run Anywhere

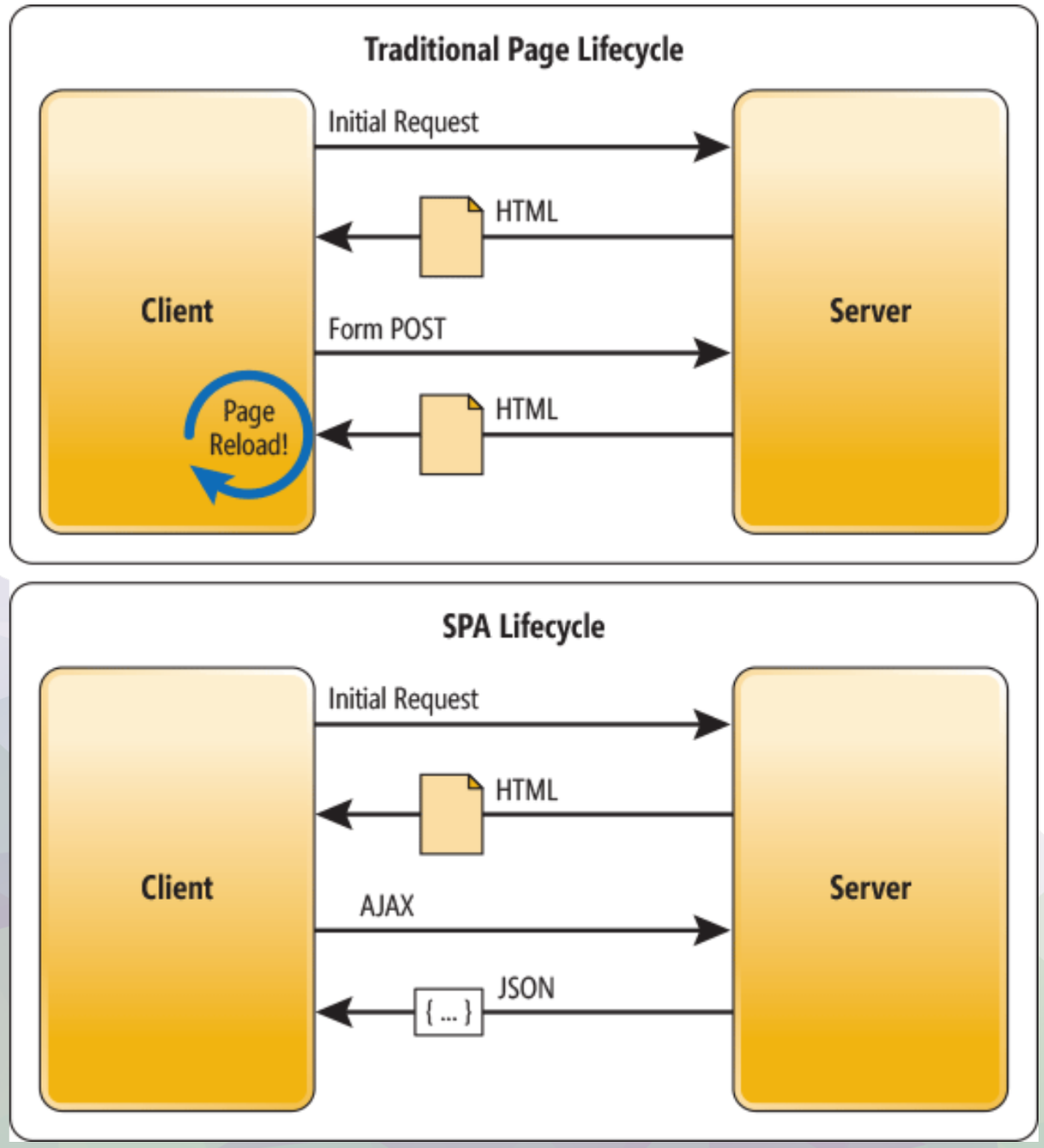
> Easy to reach a huge user base

> Browsers getting better and better

> BUT.. still many things we cant achieve like access hardware and filesystem, a unified payments platform, high performance

> Not better than native, but Good Enough for increasing types of applications

WEB PAGES VS WEBAPPS



html on the wire

data on the wire

- no page refresh
- javascript driven

RESPONSIVE DESIGN

WHAT WE WILL BE BUILDING

Concept / Value proposition

a matchmaking application
for students searching love
that allows the user to match interests with other users
(compatible or not?)
and facilitate their budding love

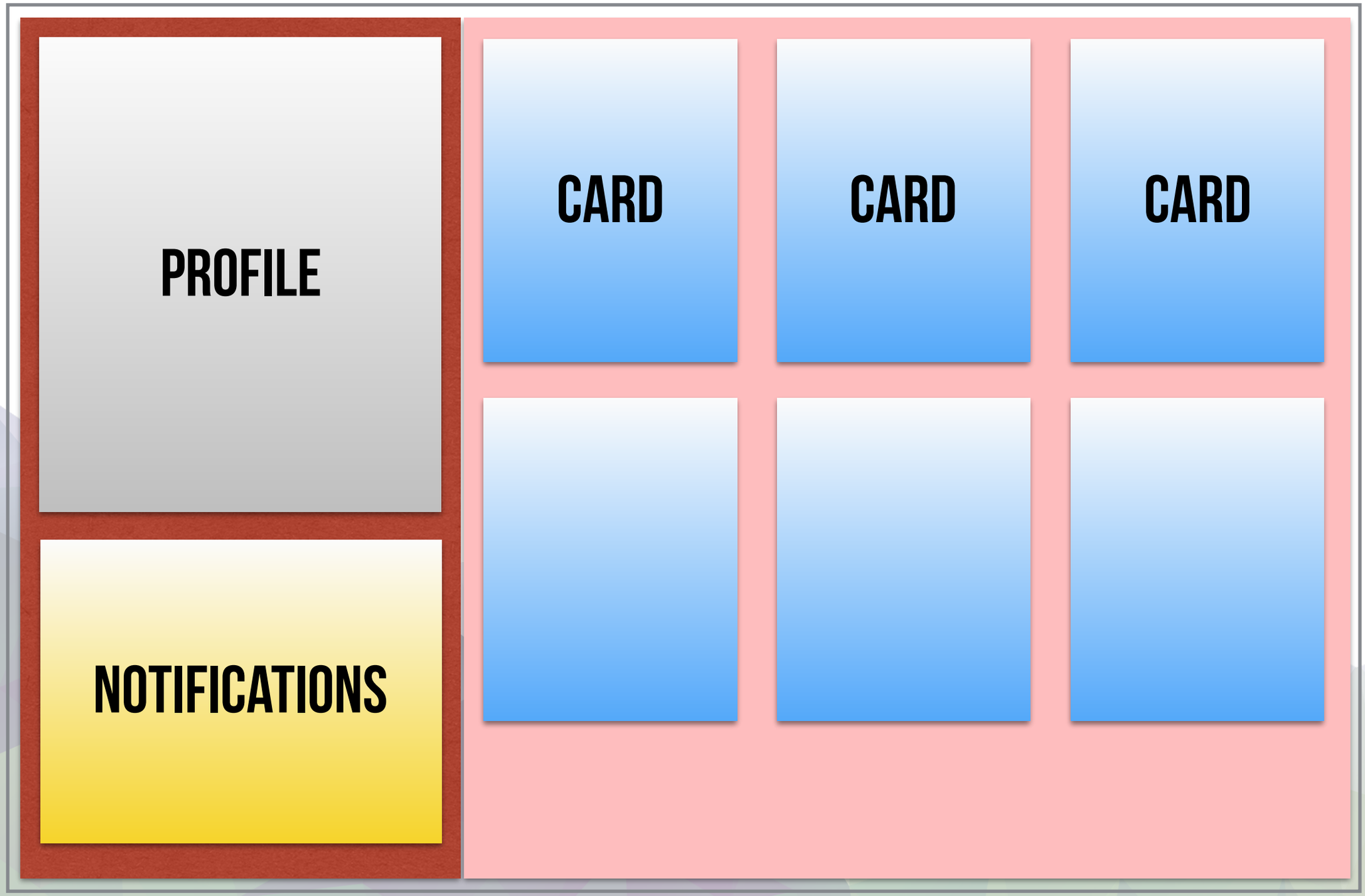
WHAT WE WILL BE BUILDING

Mock Up - Job to be done

- create and update personal profile
- view all other users on platform
- check compatibility with another user
  - matching interests
  - astrology?
- send gifts
- rank by popularity

WHAT WE WILL BE BUILDING

Mock Up - Page Structure



WHAT WE WILL BE BUILDING

Course overview

DAY 1

- ✦ **Pure front end app - no server backend**
- ✦ **Page scaffolding**
- ✦ **No database, we will hardcode data**

WHAT WE WILL BE BUILDING

Course overview

DAY 2

- Intro to Meteor.
- Port existing app over
- Database
- User accounts

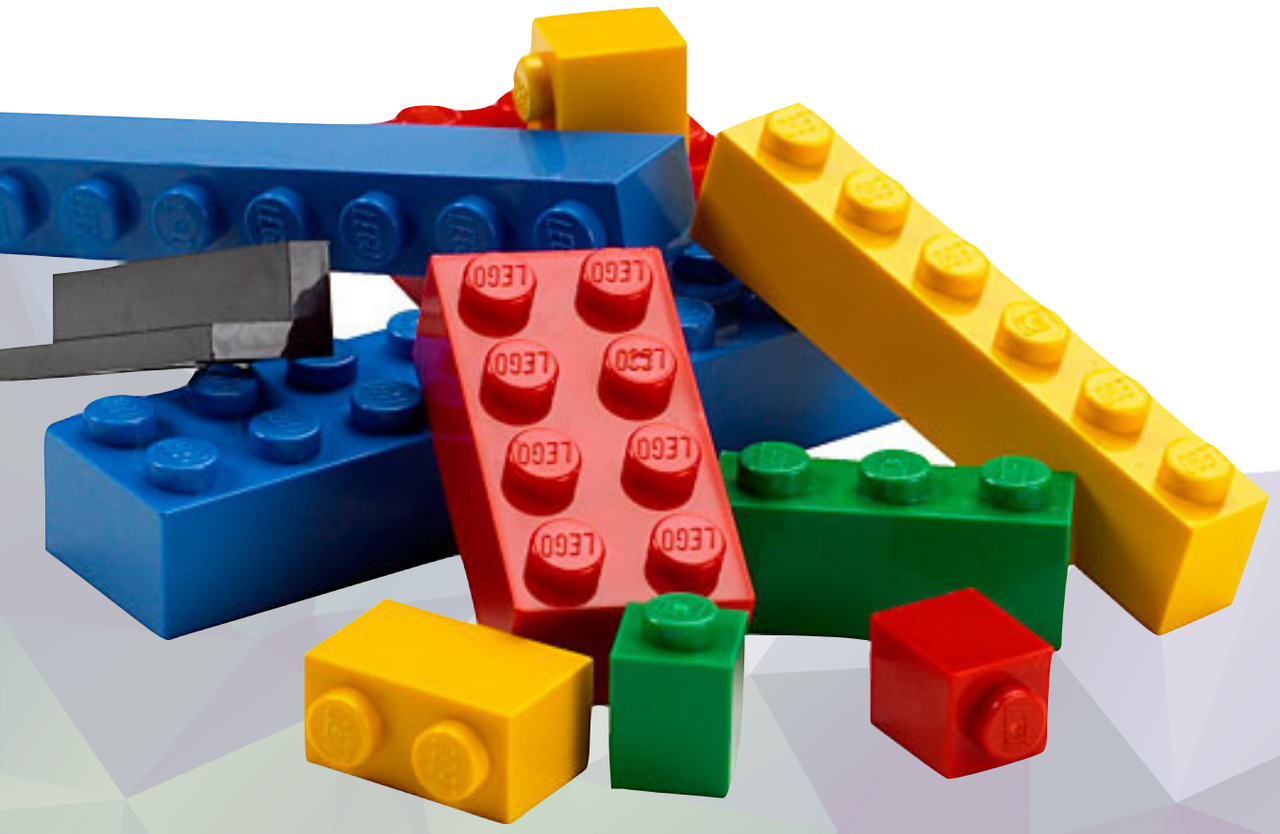
ENVIRONMENT SETUP

- You will need a text editor, use any you like, something you are **familiar** with!
- Suggested: Sublime Text, Notepad++, WebStorm
- Look out for features like:
  - Syntax/error highlighting
  - HTML tag expansion (Emmet)
  - Auto-reformatting
  - Auto-completion
- Chrome Web Browser

LETS START

USING 3RD PARTY CODE

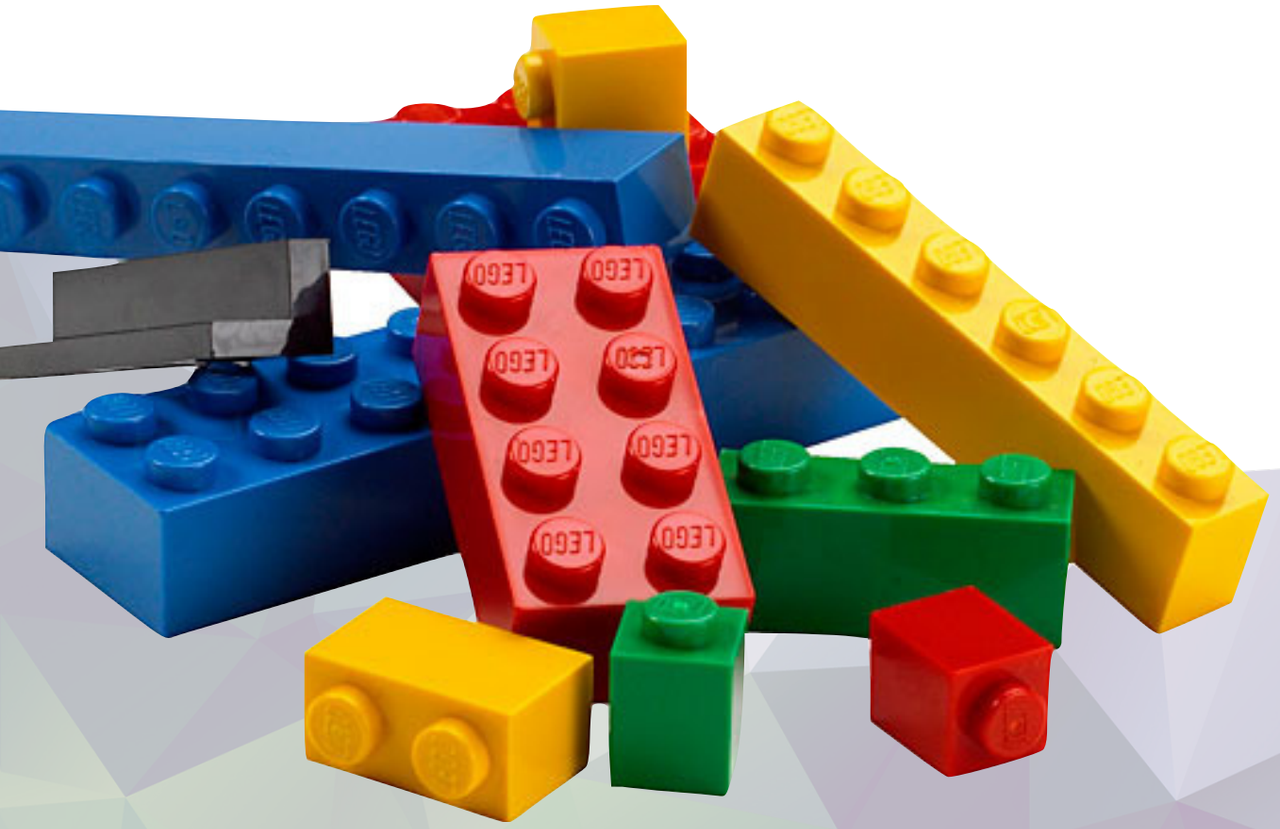
we code some, we kope some



- Web development about composing building blocks to create something
- Good programmers **reuse** what is already available

USING 3RD PARTY CODE

we code some, we kope some



- Use HTML <script> tag to import
- src = use local copy or Content Delivery Network (CDN)
- Benefits of CDN include:
  - Distributed data centers
  - Browser caching
  - Faster page load

EXERCISE 1

Import jQuery

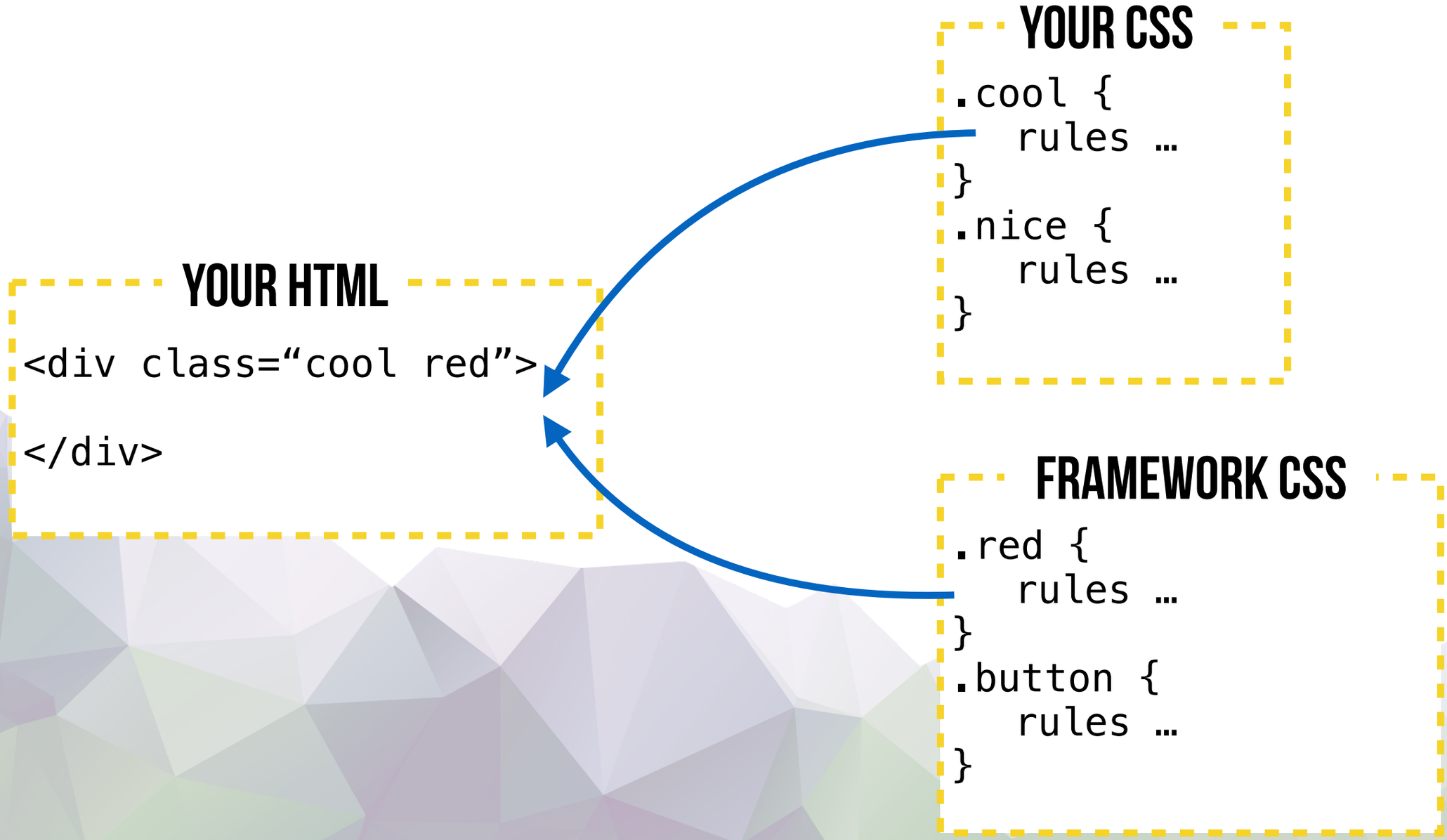
Open your web app in Chrome (no errors)

CSS FRAMEWORKS

- Get pretty components out of the box
- Help us to resolve browser differences
- Someone else do the hard work (eg browser prefixes)
- Don’t have to write your own CSS - only if you want to customise
- Usually come with good documentation
- Good way to learn CSS best practices
- Take your pick: Bootstrap, Foundation, Semantic

HOW CSS FRAMEWORKS WORK

for styling



HOW CSS FRAMEWORKS WORK

for behaviour

YOUR HTML

```
<div class="cool-
component">

</div>
```

FRAMEWORK CSS

```
.cool-component {
   rules…
}
```

YOUR JS

```
$('.cool-component)
     .startPlugin()
```

FRAMEWORK JS

```
lots of code…
```

EXERCISE 2

Play with SemanticUI components
Play with Grid System

EXERCISE 3

Create a 2 panel layout
Add card for user profile

LUNCH

1130 - 1200pm

EXERCISE 4

Scaffold UI to display cards of other people

JQUERY

a must have library for working on the browser

- Help us to resolve browser differences
- Great documentation
- Super easy way to manipulate page (DOM elements)
- http://www.sitepoint.com/jquery-vs-raw-javascript-3-events-ajax/
- Extensive plugin system

JQUERY

we need to know how to...

Select Elements

`$(…)`

Add Behaviour

`.on()` `.click()`

Change state

`.class()` `.html()`

```
$(‘.donebutton’).class(‘hidden’)
```

https://api.jquery.com

EXERCISE 5

Create global object with application data
Use this to generate n number of cards
for other users

GET ORGANISED

some tips

- Encapsulate everything into an object (don't pollute the global namespace)
- Group related code into functions, so we know where to find/add our code
  - init()
  - bindEvents()
  - loadSomething()
  - refresh()
- Initialise app when *Document Ready* *$(function)*

TEMPLATING ENGINE

creating html strings manually is painful

so how?

https://mustache.github.io

TEMPLATING ENGINE

how it works

Template + **Data** = **HTML**

```
<div class="entry">
  <h1>{{name}}</h1>
  <div class="body">
    {{message}}
  </div>
</div>
```

```
{
  name: “John”
  message: “hi”
}
```

```
<div class="entry">
  <h1>John</h1>
  <div class="body">
    Hi
  </div>
</div>
```

Mustache.render(template, data)

TEMPLATING ENGINE

template helpers

Loops

{{# array }}

html that is displayed for every element in array

{{/ array }}

Conditionals

{{# boolean }}

html that is displayed only if boolean is true

{{/ boolean }}

EXERCISE 6

Use Mustache templating!

EVENT HANDLING

we use the .on() jQuery method to attach a function to a DOM element

DOM element rendered

Attach callback function to element

DOM element not yet rendered

Attach callback to parent element that is already rendered, passing in additional filter argument

PASSING DATA

when we attach the same event handler for multiple DOM elements, we need a way to uniquely identify each one

HTML5 Data Attributes

can be attached to these elements using `data-<name>`

and retrieved with jQuery using `.data(name)`

EXERCISE 7

Attach event handlers to *Check Compatibility* button, and print the person id of the clicked element to console

JAVASCRIPT UTILITIES

sugary syntax for common algos

- Perform common algorithms with lesser pain
- Find item in array, looping, sorting, mapping etc.
- We are using Underscore.js

EXERCISE 8

Implement check compatibility function
- Display modal on click
- Calculate number of common interests
- Display progress bars

APIS, OUR BEST FRIEND

Application programming interface

- API is an **agreement/contract** between two people, saying that if you give me this instruction, I will execute this and do something for you.

- This something could be **anything**:
  sending an email, drawing a chart, fetching data etc

- We need an API because we are relying on some code someone else did

- Look at the API documentation to see what data they take in, and what is the corresponding result

APIS, OUR BEST FRIEND

Application programming interface

General Steps

- Download the library (if required)
- Get the API keys (if required)
- Transform your data to match their expected input
- Call the API!

EXERCISE 9

Follow the API documentation of Vedic API to display Horoscope matching report!

EXERCISE 10

Every time the *Check Compatibility* button is clicked, increment the number of views of the person by 1 and refresh UI

END OF DAY 1

https://clarencengoh.typeform.com/to/kqWcZb