Learn to code in a day.

Timetable

9.00 setup
9.15 1st session
11.00 break
11.15 2nd session
1.00 lunch
2.00 3rd session
3.30 break
3.45 4th session

Learn to code in a day

is this possible?

3 Aims

Looking at the source code of a web page makes sense.

Appreciate how technologies impact one another.

Foundation in best practices for software development.

actually a 4th aim - to make my job easier!

Also going to show you...

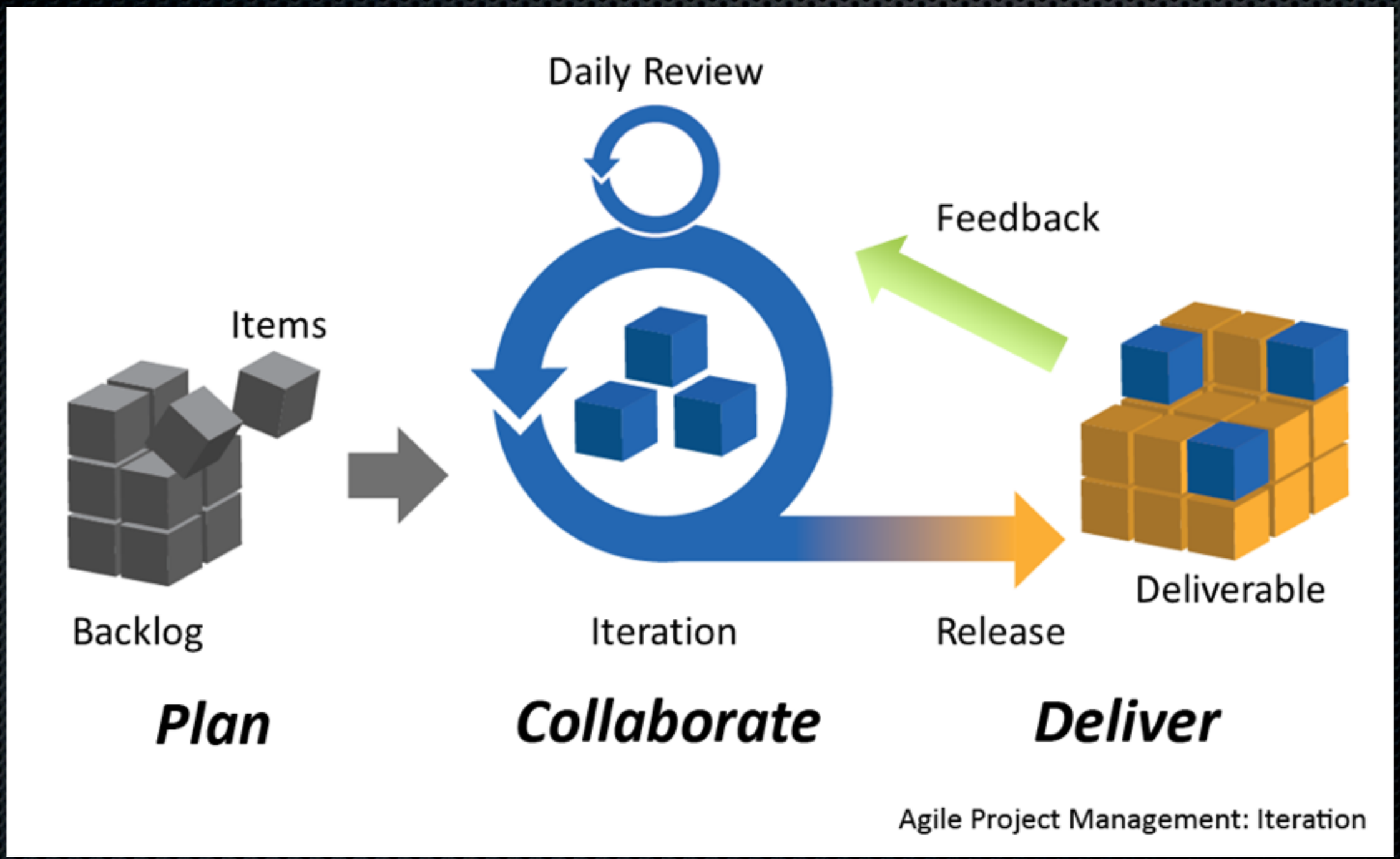
How to make £15 billion!

Course content

Agile / TDD / GIT
HTML / flow / attributes
CSS / responsive design
DOM / dot notation / JSON
jQuery (Javascript)
Retrospective / Q&A

Agile / TDD

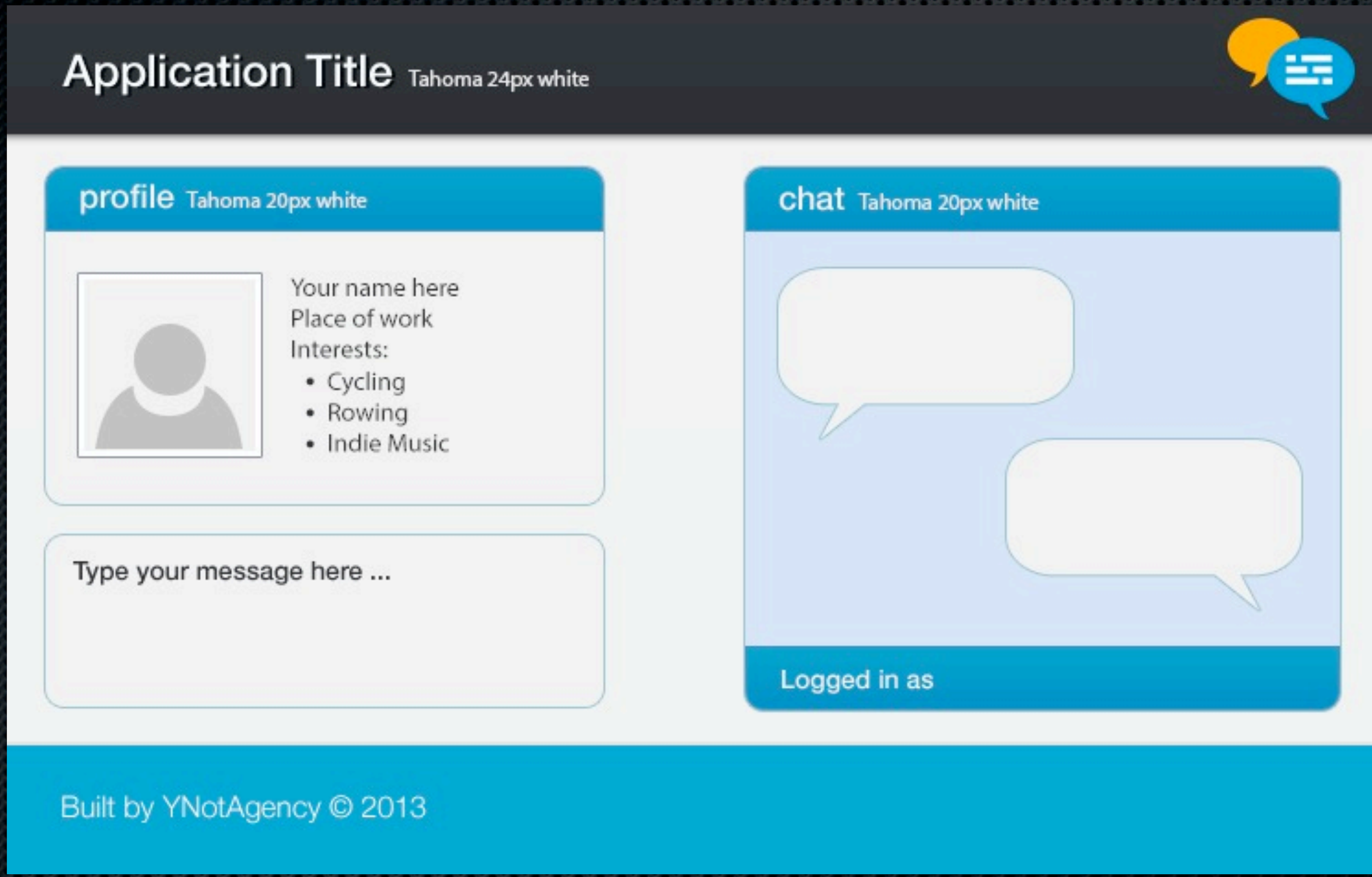
Agile process



Business requirements

App has a header with title in and footer
App has a profile section and a chat section
Do not use times new roman font
The app title should be more prominent
The user is asked to enter a chat name
The user cannot send blank messages

UX design


Application Title Tahoma 24px white
profile Tahoma 20px white
Your name here
Place of work
Interests:
• Cycling
• Rowing
• Indie Music
Type your message here ...
chat Tahoma 20px white
Logged in as
Built by YNotAgency © 2013

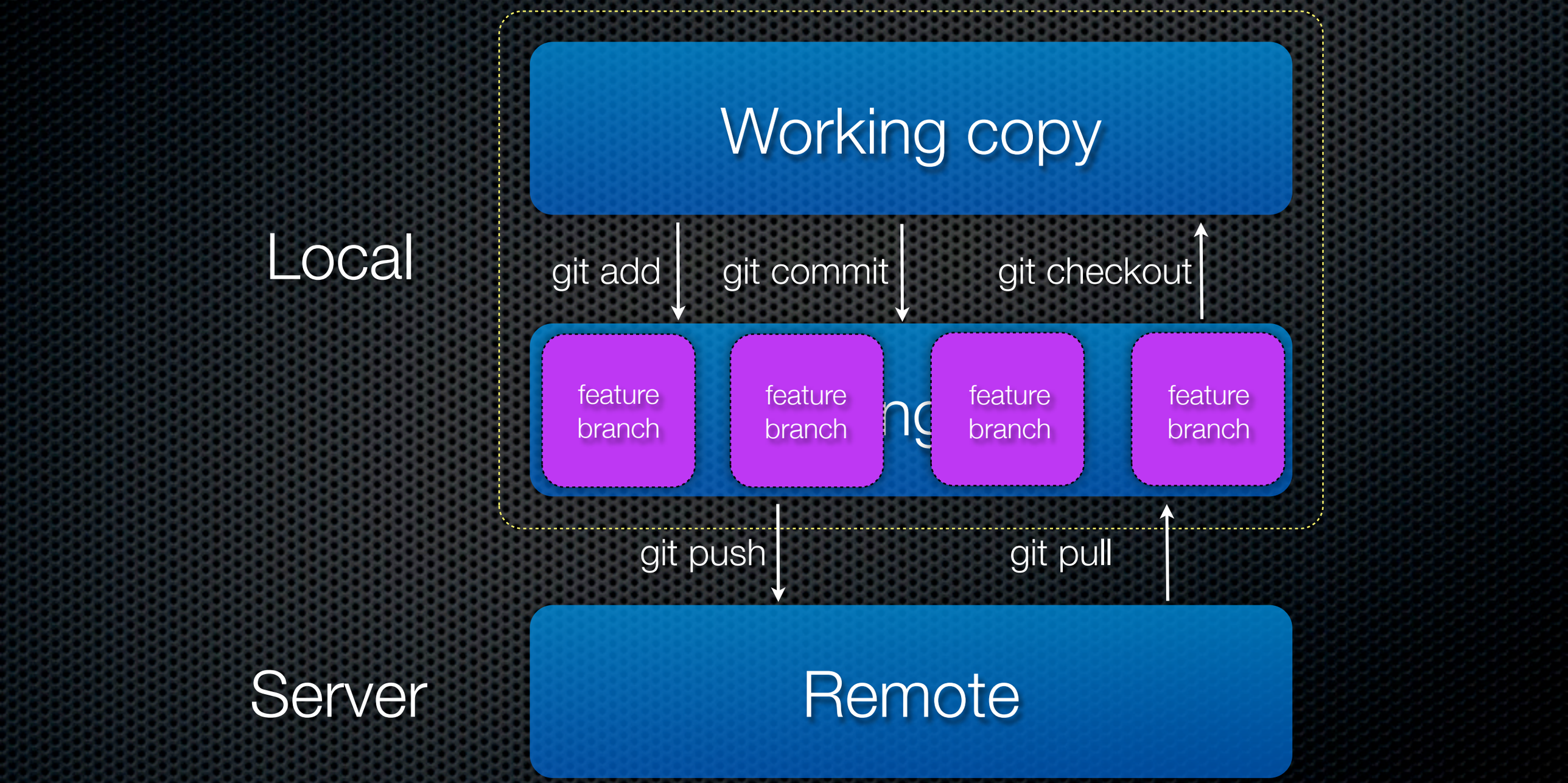
Test driven development TDD

For each feature write a set of failing tests FIRST

These tests can then be automated as a test suite

We have 100% confidence in our application

GIT (version control)

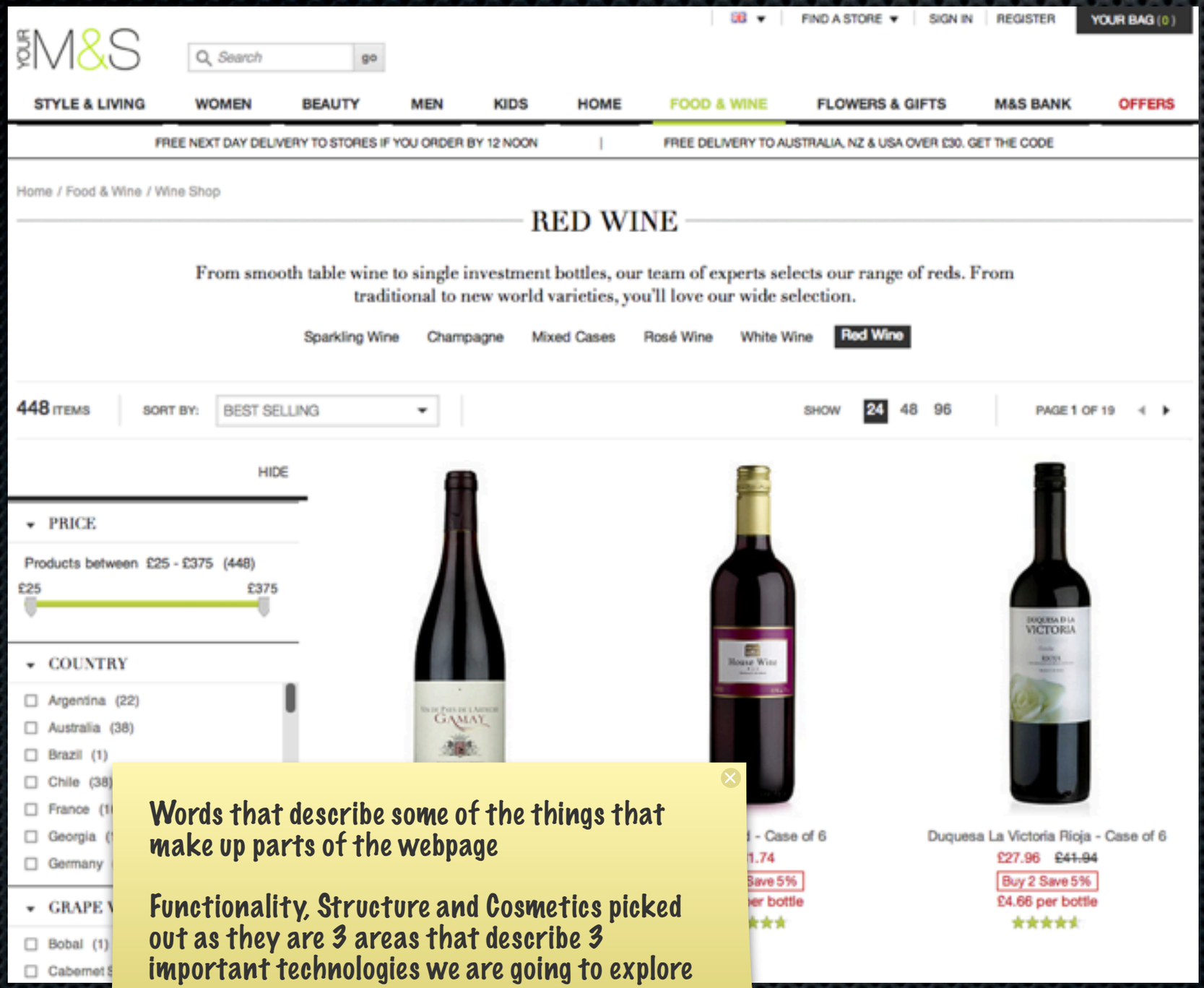


Summary

- Agile is a product development process that works through regular iterations (sprints) to build small shippable features
- TDD works well with agile as it means we can ship our product with confidence
- GIT is a version control system that can take snapshots of a codebase, again helping us ship small features

So what constitutes a web application?

What constitutes a web app?



Words that describe some of the things that make up parts of the webpage

Functionality, Structure and Cosmetics picked out as they are 3 areas that describe 3 important technologies we are going to explore

Style

Design

Text

Functionality

Server

Logo

Lists

Layout

Database

Feel

Navigation

API

Images

Content

Data

URL

Elements

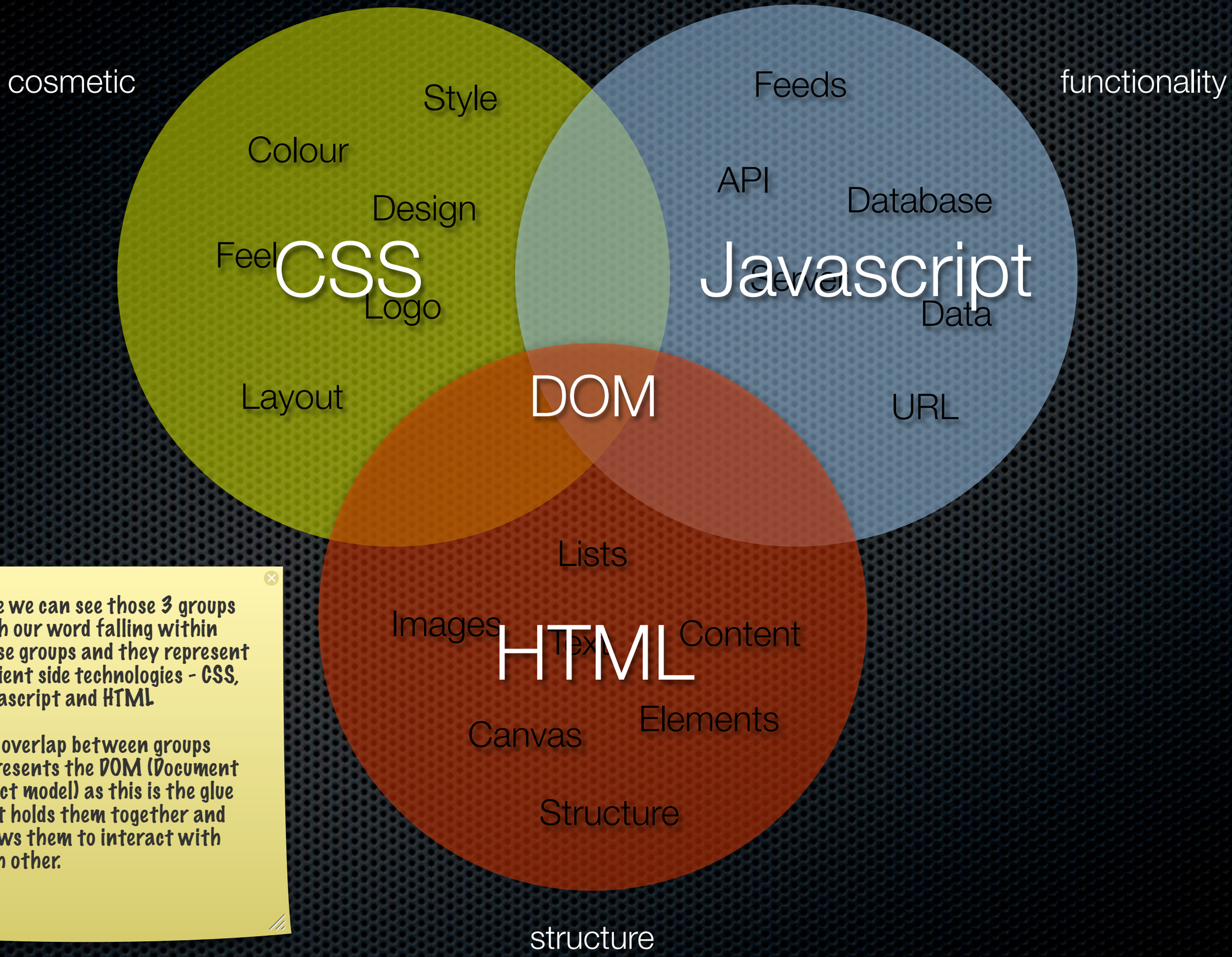
Structure

Canvas

Feeds

Cosmetic

Colour



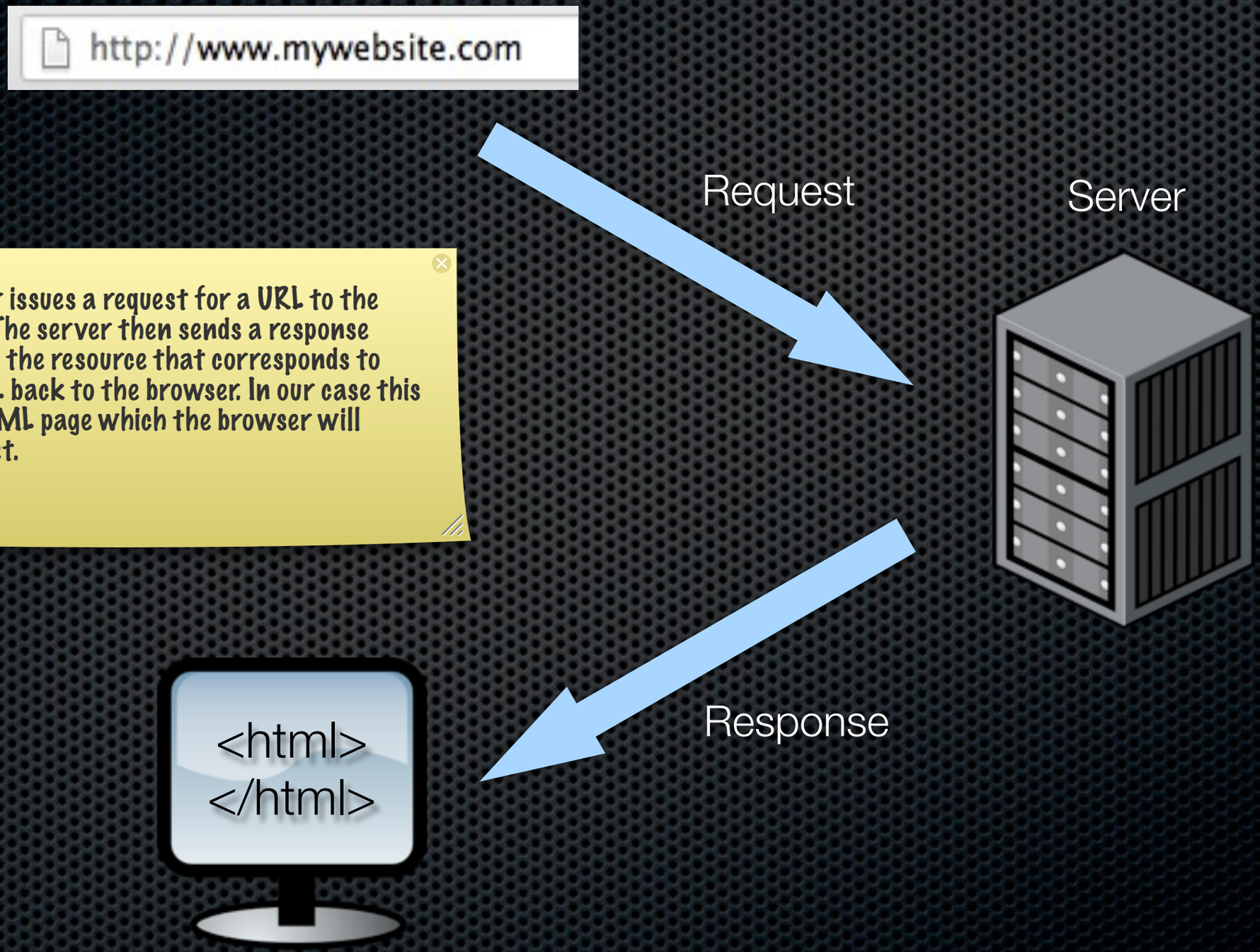
Here we can see those 3 groups with our word falling within those groups and they represent 3 client side technologies - CSS, Javascript and HTML

The overlap between groups represents the DOM (Document object model) as this is the glue that holds them together and allows them to interact with each other.

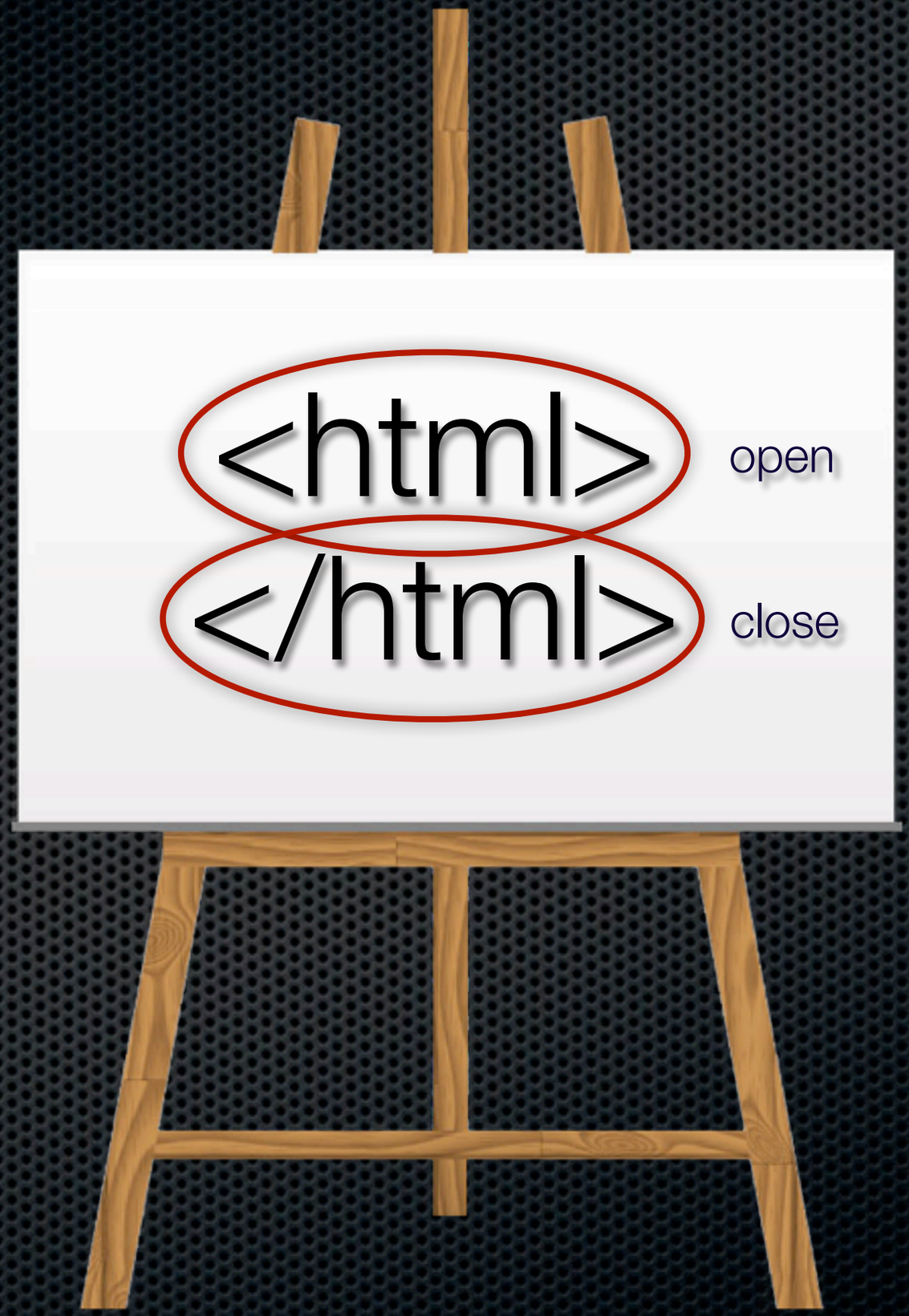
HTML

(hyper text markup language)

Basic lifecycle of a request



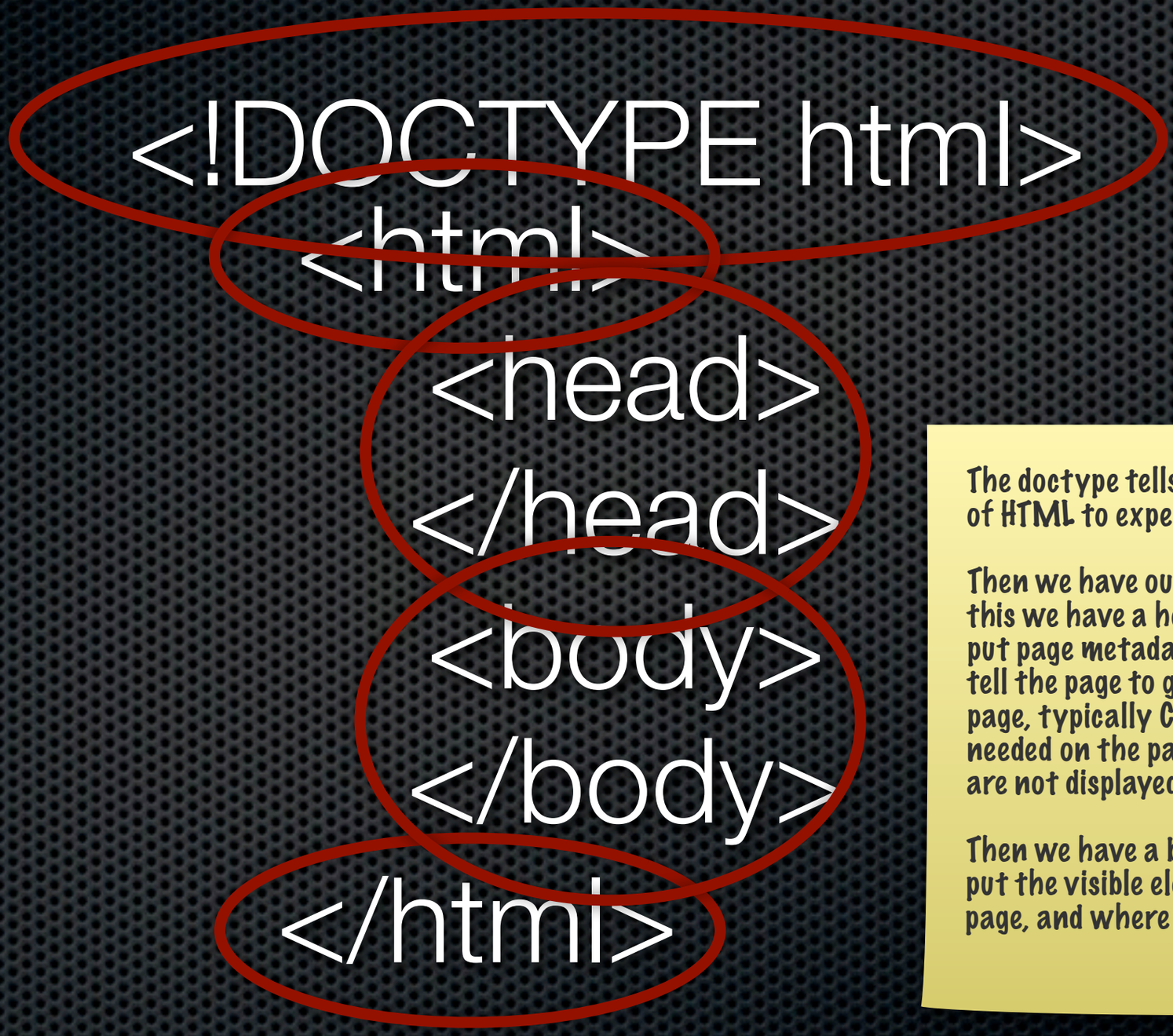
What is an HTML tag?



This is our blank canvas, and most fundamental HTML tag - the HTML tag!

This is the syntax that all tags use, the opened and closed

Minimum HTML Page structure



The doctype tells the browser what sort of HTML to expect.

Then we have our 'base' HTML tag. Within this we have a head tag. This is where we put page metadata. This is also where we tell the page to get other resources for the page, typically CSS and Javascript that is needed on the page. Things put in the head are not displayed in the browser window

Then we have a body tag. This is where we put the visible elements that appear on our page, and where the visible html goes

```
<!DOCTYPE html>
<html>
    <head>
        <title>application title in tab</title>
        <meta charset="utf-8">
        <meta keywords="keywords for search engines">
        <meta description="description sentence for search engines">
    </head>
    <body>
        <div>
            <h1>application title in header</h1>
            <img src="" width="80" height="40" />
        <div>
        <div>
            <h2>Name</h2>
            <p>profile<p>
            <img src="" width="150" height="200" />
            <h2>Interests</h2>
            <ul>
                <ol>x</ol>
                <ol>x</ol>
                <ol>x</ol>
                <ol>x</ol>
            </ul>
        </div>

        <div>
            <div></div>
            <div>
                <span id="status">Connecting...</span>
                <input type="text" disabled="disabled" />
            </div>
        </div>

        <div>
            <ul>
                <li>Built by <a href="#">your agency</a><li>
                <li>&copy 2013</li>
            </ul>
        <div>
    </body>
</html>
```

Here we can see examples of what goes in those sections

HTML tags

```
<h1>1st level header</h1>
<div>container</div>
<ul>
    <li>list item 1</li>
    <li>list item 1</li>
</ul>

<img src='' height='' width='' />
<input type='text' />
```

Html tag examples

top are opening/closing tags. The ul (unordered list) has children - the list items

bottom are self closing

tags as they have no child elements

W3Schools has a comprehensive list and description of HTML tags

http://www.w3schools.com/tags/default.asp

Tips for writing good HTML

HTML family tree

```
<!DOCTYPE html>
<html>
    <head>
        <title>application title in tab</title>
        <meta charset="utf-8">
        <meta keywords="keywords for search engines">
        <meta description="description sentence for search engines">
    </head>
    <body>
        <div>
            <h1>application title in header</h1>
            <img src="" width="80" height="40" />
        </div>
        <div>
            <h2>Name</h2>
            <p>profile<p>
            <img src="" width="150" height="200" />
            <h2>Interests</h2>
            <ul>
                <ol>x</ol>
                <ol>x</ol>
                <ol>x</ol>
                <ol>x</ol>
                <ol>x</ol>
            </ul>
        </div>

        <div>
            <div></div>
            <div>
                <span id="status">Connecting...</span>
                <input type="text" disabled="disabled" />
            </div>
        </div>

        <div>
            <ul>
                <li>Built by <a href="#">your agency</a><li>
                <li>&copy 2013</li>
            </ul>
        </div>
    </body>
</html>
```

Parent/Child

Siblings

Descendant/Ascendant

HTML has a tree structure hence all elements can have either child elements, parent elements or both.

As we saw in the last slide some elements never ha have child elements.

Indentation / formatting

```
    <div><h1>application title in
        <img src="" width="80" he
</div><div>
        <h2>Name</h2>
        <p>profile
        </p>
        <img src="" width="150" height="200" />
        <h2>Interests</h2>
         <ul>
            <ol>x</ol><ol>x</ol>
            <ol>x</ol>
              <ol>x</ol>
         <ol>x</ol>
        </ul>
    </div>
```

no whitespace

1 tag per line

Keep code correctly formatted using tab spacing, no spaces where there should a line break,

The above especially are important when writing any computer code to keep things readable.

Try to keep one tag per line.

for html tags, keep the casing to lowercase

Letter casing

Semantics

Ancient greek term - ‘The study of meaning’

in HTML terms this means use tags for what they were designed for - what they mean

```
<p>
item 1<br />
item 2<br />
item 3<br />
</p>
```

```
<ul>
    <li>item 1</li>
    <li>item 2</li>
    <li>item 3</li>
</ul>
```

This is important to make the page stylable, and for accessibility reasons

World Wide Web Consortium (W3C)

Formed in 1994 as a result of early browser wars.

They write ‘recommendations’ or standards for browser technology.

Standards are written by a wide range of leading web technologies

www.w3.org

Browsers

Gecko

Trident

Webkit

iOS

Can you raed tihs?

Oevr the nxet egiht huors we are giong to fnid out how to cdoe and laern aobut lodas of ralely cool sutff

Nad bceome billoinaiers!

Browsers can interpret incorrect HTML to a certain extent but may hide mistakes that can cause problems later on.

Summary

- Had a look at what constitutes a webpage and what the technologies are behind those
- An HMTL page is made up of a hierachical tree of tags
- Seen it is important that we try to write clean readable semantic HTML

Practical

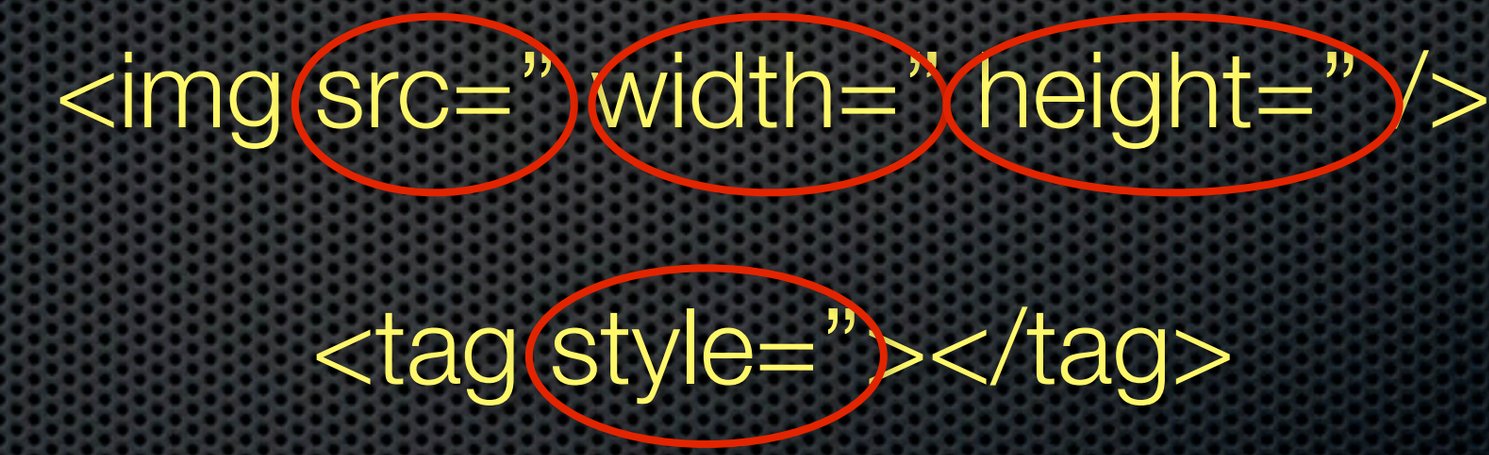
git checkout failHtml

HTML Attributes

HTML attributes

Think of attributes as metadata for that particular tag

They take the format of <tag attribute=”value”></tag>



Class and Id attributes

<tag class=”abc def ghi”></tag>

A tag can have one or more classes

<tag id=”xyz”></tag>

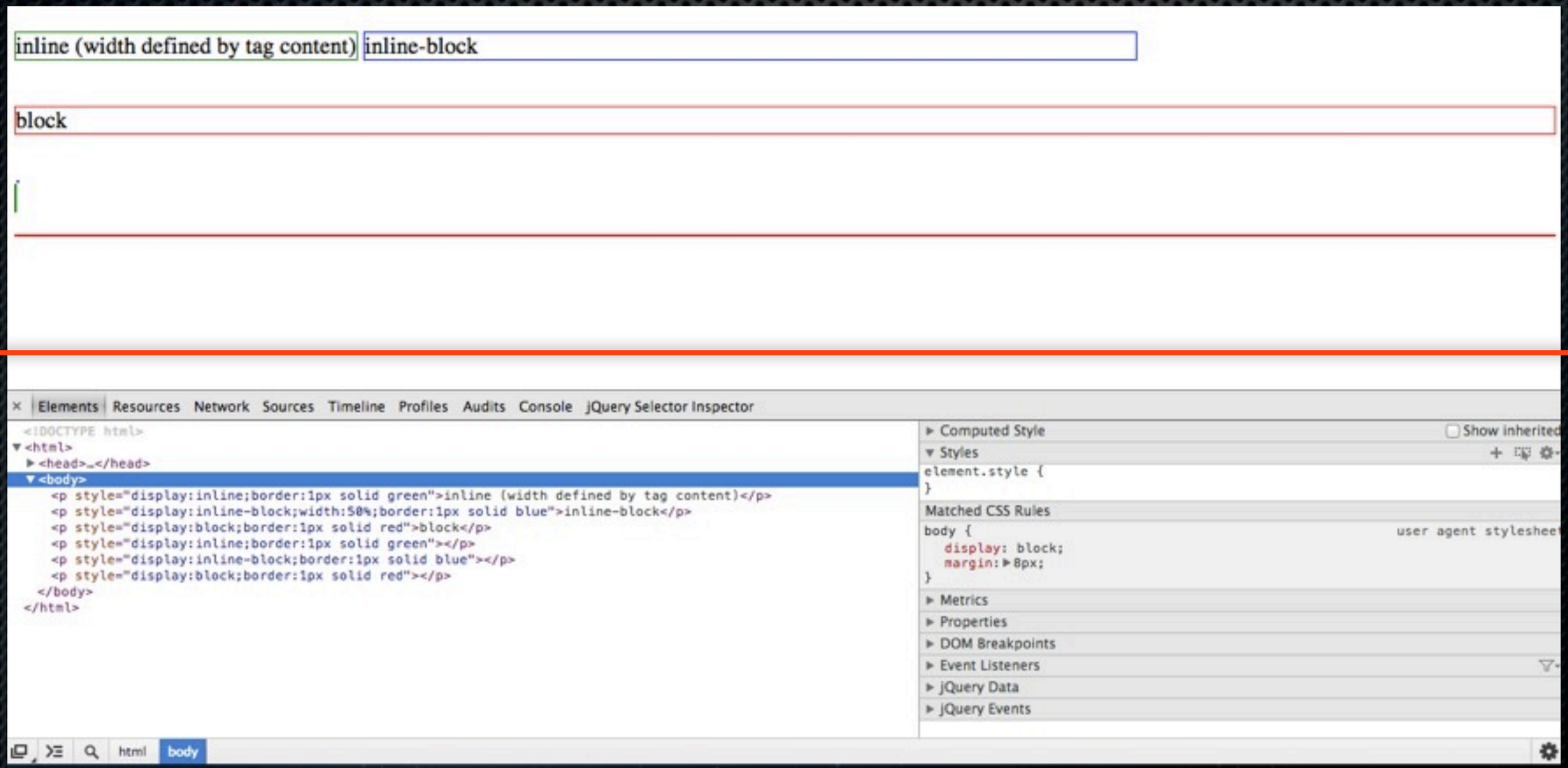
A tag can only have one id

Can only contain characters a-z A-Z 1-9 _-
Must contain at least one letter

Practical

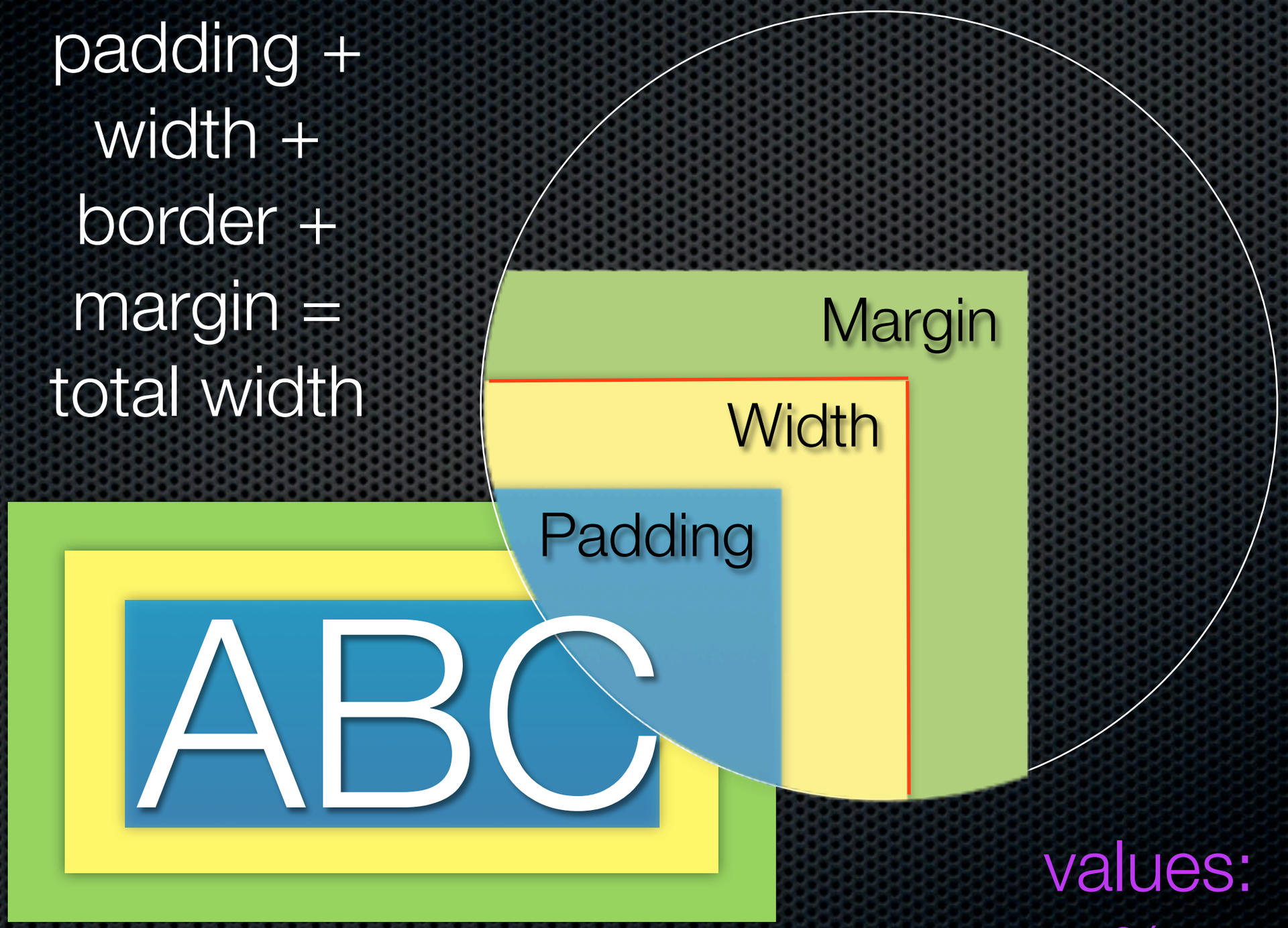
Google chrome inspector

Alt Cmd i (Mac) Shift Ctrl i (Windows)



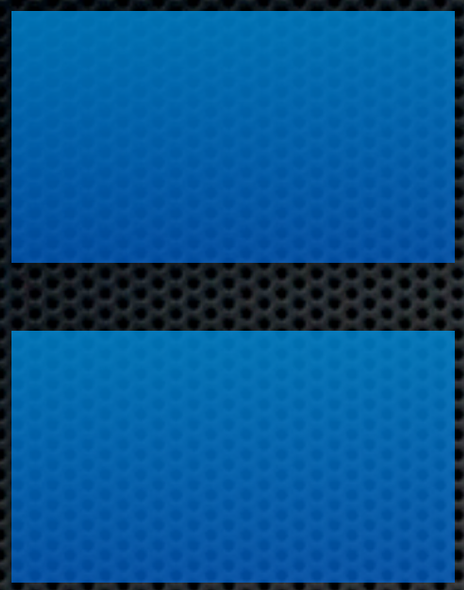
Box Model

padding +
width +
border +
margin =
total width



values:
px, %, em

Page Flow



float:left

Once you start floating, keep floating!

Position property

Relative

Element stays in the flow of the document but any child elements are positioned relative to the bounds of this element.

Absolute

The element is taken out of the flow of the document and is positioned at coordinates relative to the nearest parent with relative positioning.

Practical

Summary

- Seen how we can add metadata to a tag using attributes
- Had a look at how we can inspect html with the browser console
- Seen how we can change the position of HTML elements with some positional style properties

CSS

(cascading style sheets)

Selectors

class

A tag attribute who's value represents a generic identifier for the element. Elements can have more than one class. A class is prefixed in a selector by a dot(.) before the class name e.g .myClass

id

A tag attribute who's value represents a unique identifier for the element. There must be only one element of an id on a page. An id is prefixed in a selector by a hash(#) before the id name e.g #myId

Examples of selectors

#menu

.blogItem

div#menu

div.blogItem

div#menu li

div.blogItem h2

div#menu li.selected

div.blogItem h2.highlighted

parent - descendant - descendant - target

CSS Rules

selector { property1 : value ; property2 : value ... }

Examples

h1 { font-size:24px;color:#343434 }

div.blogItem h2.highlighted { text-decoration:bold }

Practical

Combining these CSS properties

This can become quite complex and takes years to master. We are going to keep things relatively simple.

A Dark Art...

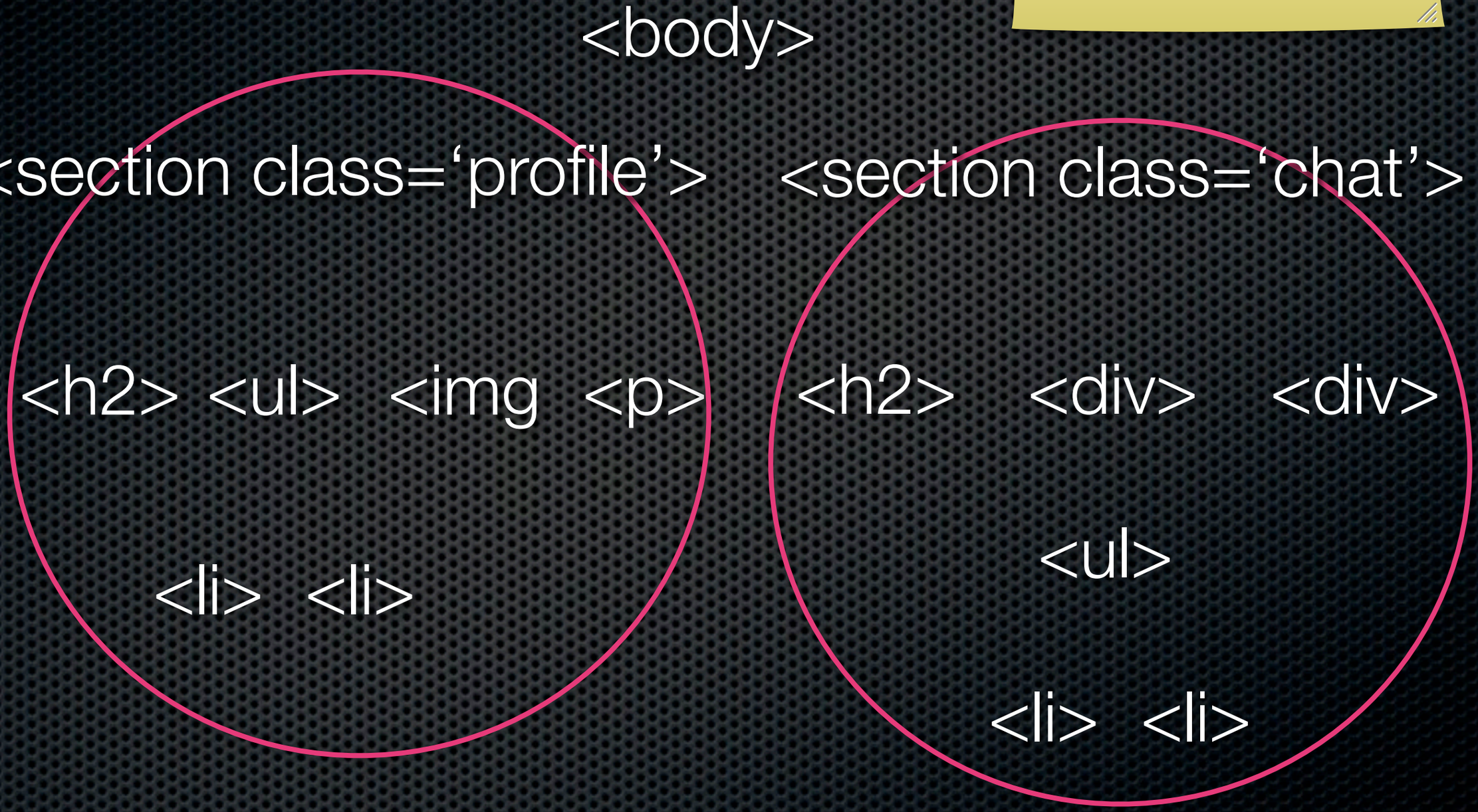
Problem

The selector / CSS rule below would select the li elements and color them red. What happens if we only want to color the ones on the lext hand section?

section ul li { color:red }

Solution

If we give the sections a class different to each other, we can then make our selector more specific to only select the li elements from the 'profile'




```
section.profile ul li { color:red }
```

Separating assets

When writing code it is always a good idea to separate things into logical sections

<link rel="stylesheet" href="" type="text/css" />

Avoid duplication
Promotes reuse
Readability

Practical

```
<link rel="stylesheet" href="css/main.css" type="text/css" />
```

Advanced selectors

> select direct children of	div > a.partner
* select everything	div > *
element[attribute=”value”]	input[type=”password”]
selector, selector { rule }	h1, h2, h3 {color:#cdcdcd}
selector:pseudo-class	a:link a:visited a:hover a:active

Specificity

The rules which define how conflicting styles override one another - this is the cascade.

overrides

h2 { color:white }
h2.highlight { color:red }
div h2.highlight { color:green }
div h2.highlight { color:blue }

overrides

user style sheet
general style sheet
inline styles
rule !important

CSS3

Started in 1998?! ‘Web 2.0’
Exact same syntax
vendor prefixes
webkit - (chrome and safari)

border-radius
transition
transform
shadow
opacity
fonts

Keyframes / Mediaqueries

Practical

Summary

- Learnt how we can ‘target’ html elements using selectors
- We can then use these selectors to form CSS rules to apply cosmetic styling to HTML tags
- Seen a first example of separating assets with the use of a cascading style sheet

HTML CSS Quiz

Q. What does semantic HTML mean?

A. Use HTML tags for what they were designed for

HTML CSS Quiz

Q. What is a self closing HTML tag?

A. An HTML tag that can have no child elements
eg

HTML CSS Quiz

Q. What is the difference between a class and id?

A. An id is unique to the page, there must be only one

HTML CSS Quiz

Q. In a CSS selector how are classes and ids represented?

A. A class is prefixed with a . ids with a #

HTML CSS Quiz

Q. What does specificity mean in CSS

A. A style rule with a more specific selector will override a conflicting rule

HTML CSS Quiz

Q. What makes up the width of an element

A. padding + width + border + margin = total width

DOM / Javascript

Javascript Basics

Executes line by line from the top of the script

Each line of code ends with a semicolon ;

We can create variables to store values using var

Variable values can be anything:

e.g var x = 3; or var x = ‘text’;

D.O.M

Document Object Model

Provides a structural representation of the document, enabling the developer to modify its content and visual presentation.

Window is an object (see next slide) and document / location / navigator are objects nested within it

window.document

window.location

window.navigator

Dot notation

JSON

JavaScript Object Notation

```
{
    "key" : "value",
    "key" : "value"
}
```

An object

```
{
    "key" : "value",
    "key" : {
        "key" : "value",
        "key" : "value"
    }
}
```

Objects can be nested

And of course variables can be JSON objects too!

Last weeks homework

```
{
    "name"      : "Alex Bowen",
    "profile"   : "...",
    "image"     : "url/to/photo.png",
    "interest1" : "eating",
    "interest2" : "sleeping",
    "interest3" : "running"
}
```

Javascript arrays

Similar to a JSON object but only has values

Array syntax is [value0, value1, value2]

To access an array, syntax is array[x]

variables can also be arrays var x = [a, b, c]

```
{
    "name" : "Alex Bowen",
    "profile" : "...",
    "image" : "url/to/photo.png",
    "interests" : ['eating', 'sleeping', 'running']
}
```

Improving the homework task using an array

Javascript functions

Think of a function as a verb - do something

Do something with x and y and optionally tell me about what you've done

```
function (x, y) {
    Do something
    optionally return something
}
```

variables can also be functions! `var x = function() {}`

JSON

Combining JSON and functions is very powerful as Values can also be functions:

```
var application = {
    "doThis" : function() {},
    "doThat" : function() {}
}
```

```
application.doThis();
```

Guess what else our application can do!

So we can build a JSON object full of functions, one famous one being....

jQuery

A library of functions

Uses selectors to find HTML with which to use the functions

Is designed to work cross browser

jQuery is usually represented by a $

```
$('selector').lotsoffunctions();
```

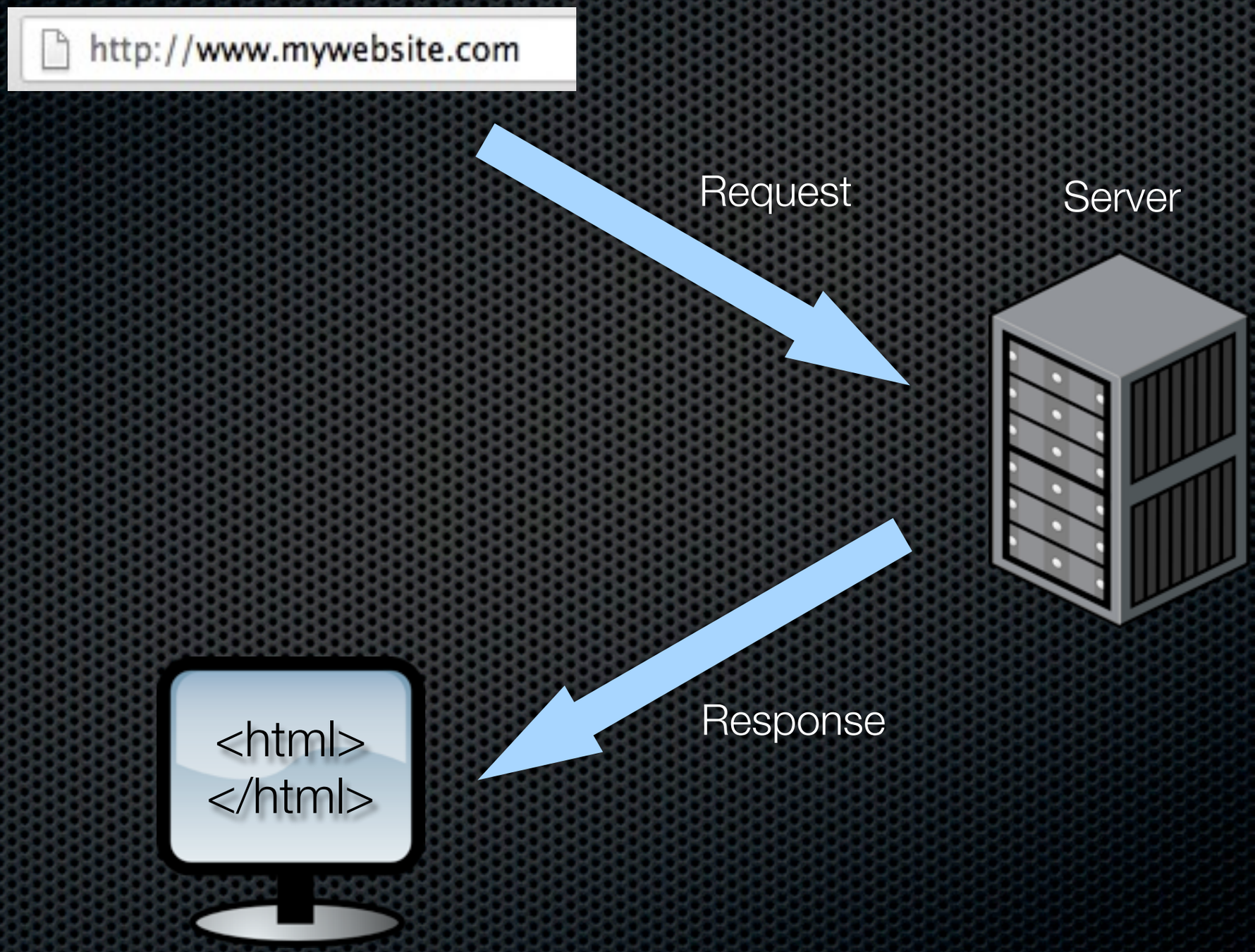
Practical

```
<script type="text/javascript" src="js/app.js"></script>
```

Summary

- The DOM is a JSON represenation of everything that is going on inside the browser
- Had a look at some key Javascript principles and constructs eg functions
- JSON and Javascript functions can be used together to build potentially powerful functionality

Http Request Cycle



This isn’t ideal for realtime messaging, why?

We could use long polling

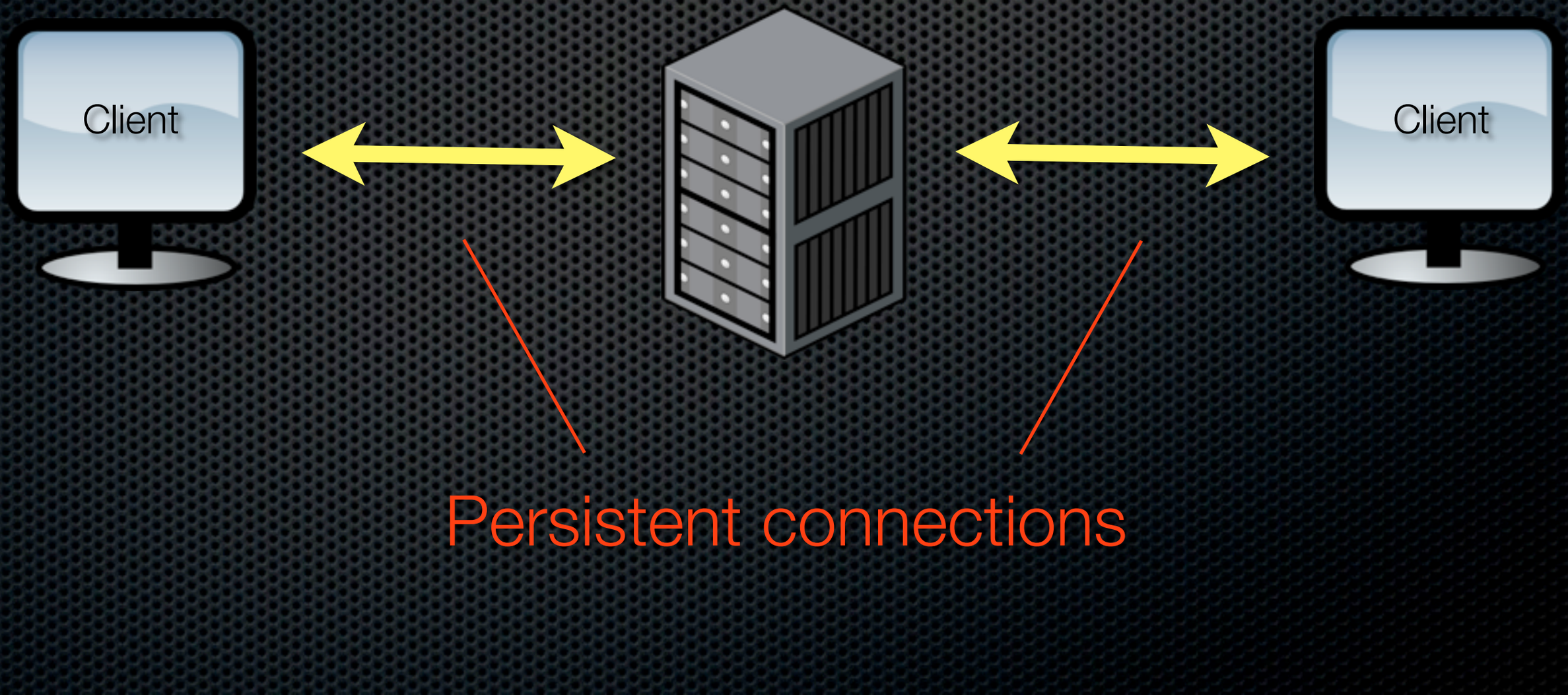
Always going to be a ‘lag’ time

Many connections are going to cripple the server

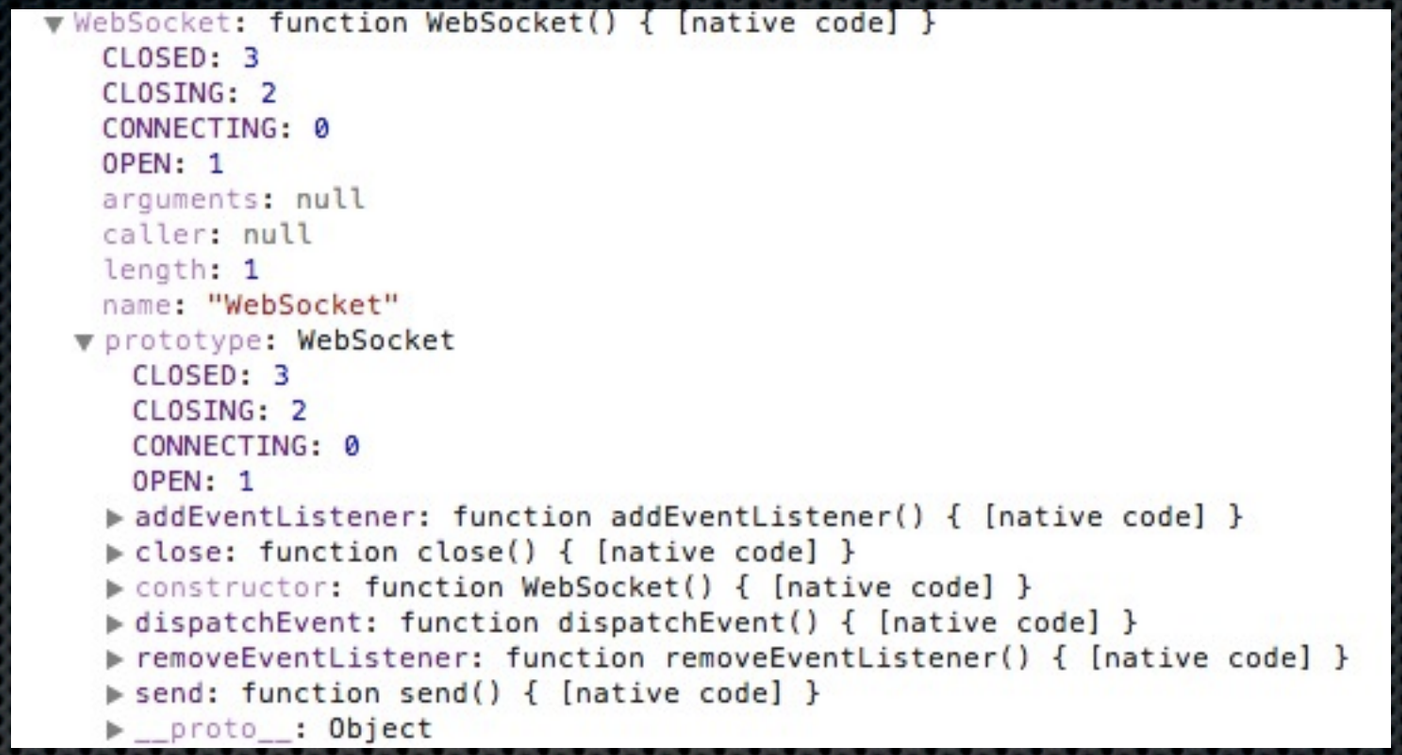
lots of metadata (headers) with each request

Websockets

Are a different protocol ws://

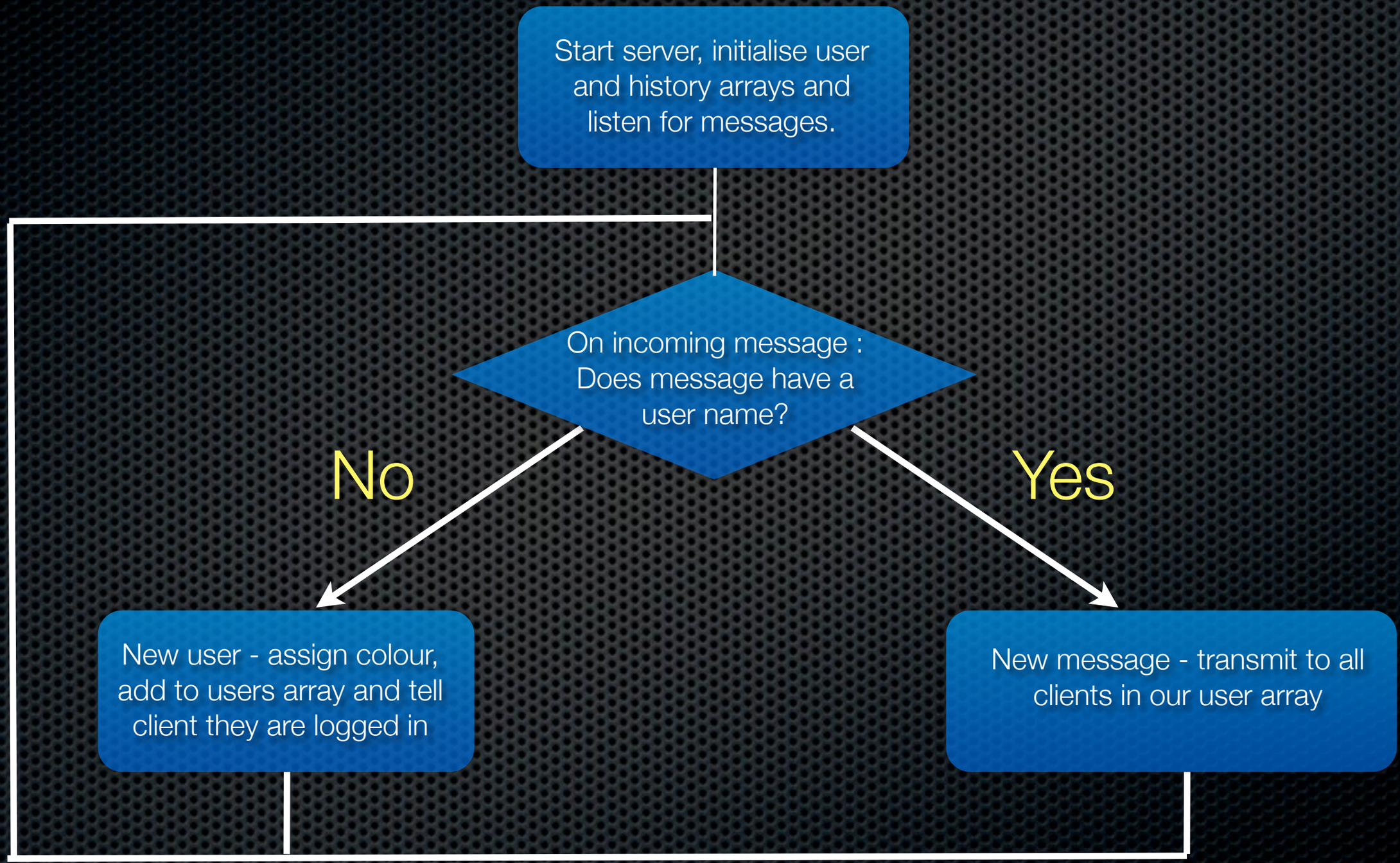


WebSocket in the DOM




```
{
    onopen : function () { do something...},
    onerror : function () { do something...},
    onmessage : function () { do something...}
}
```

Websocket server



Practical

If we had thought of this 5 years ago we could have sold it for $15billion...

This is the exact same technology that Whatsapp is built with...

Retrospective / Q & A

http://www.justgiving.com/Alex-Bowen1