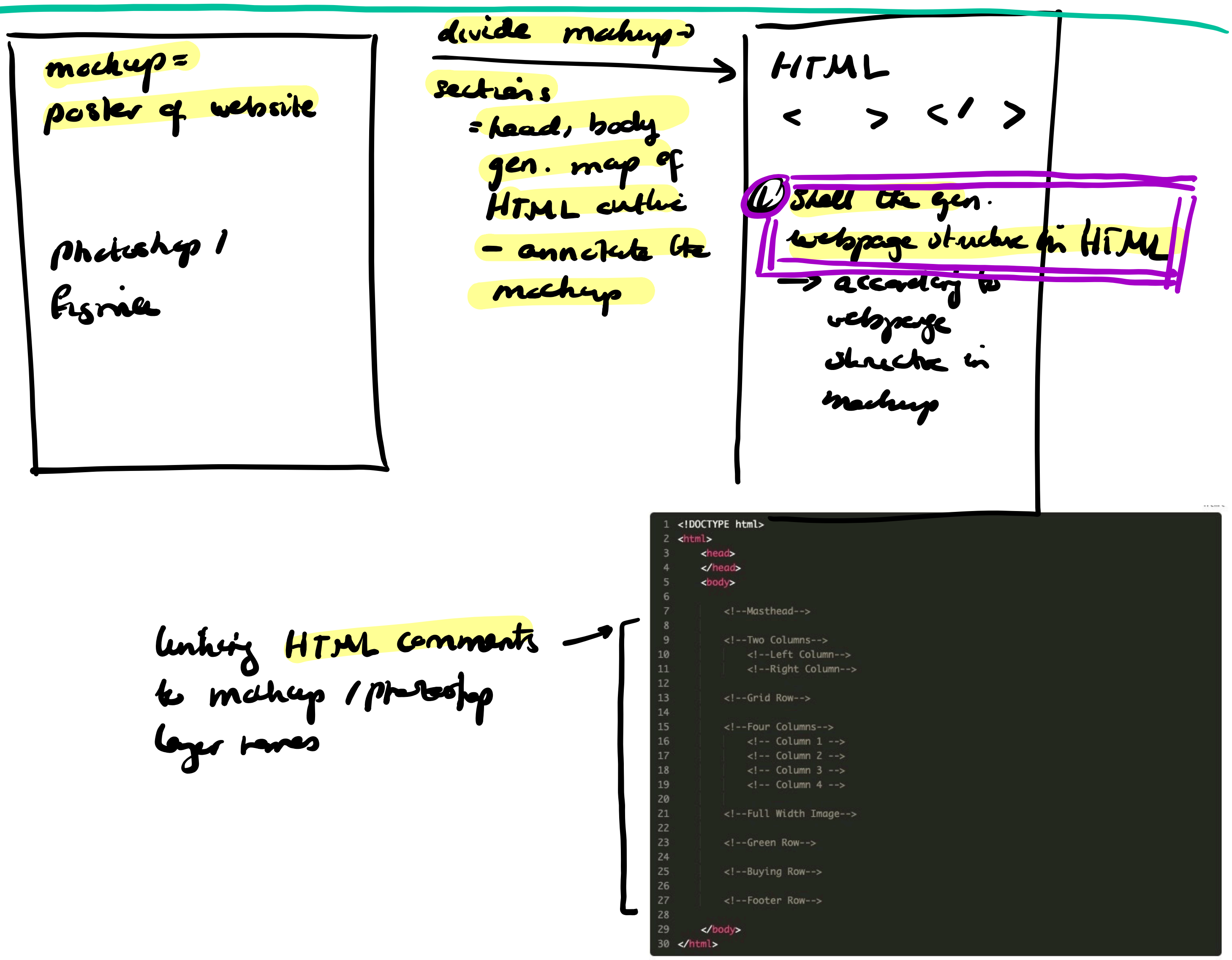
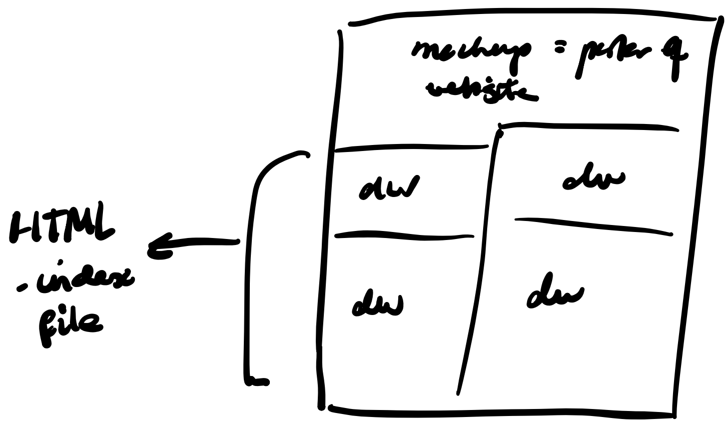
%%
Table of contents
%%

%%
Part #2 - Integrate a mockup 1. Set up basic HTML structure
%%



② You can add divs in HTML

→ <div> </div> and spec. els on mockup - if div up mockup → sections + shell them out in HTML



→ shelling out webpage structure → grps
→ classes can later be added and the div tags → to tgt those el's in CSS (styling) - 1st you need the els / content in HTML

③ Add classes in HTML →

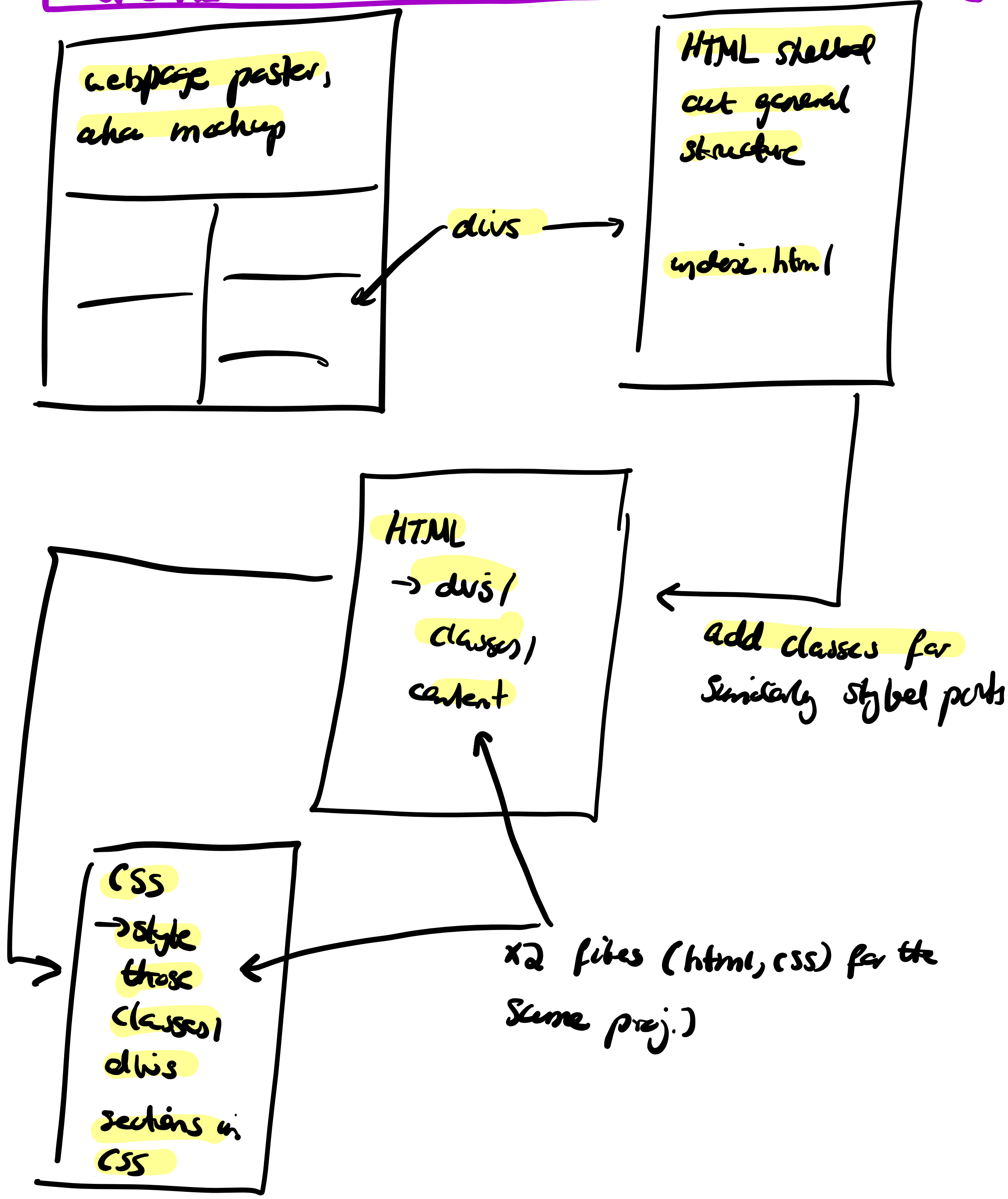
sections of the HTML are made via divs → according to the webpage mockup / poster
→ then those divs are grp'd according to classes for CSS
→ ones which have similar styling on the mockup are grp'd → the same HTML classes

```html
<!DOCTYPE html>
<html>
    <head>
    </head>
<body>

    <!--Masthead-->
    <div class="row">
    </div>

    <!--Two Columns-->
    <div class="row">
        <!--Left Column-->
        <!--Right Column-->
    </div>


    <!--Grid Row-->
    <div class="row">
    </div>

    <!--Four Columns-->
    <div class="row">
        <!-- Column 1 -->
        <!-- Column 2 -->
        <!-- Column 3 -->
        <!-- Column 4 -->
    </div>

    <!--Full Width Image-->
    <div class="row">
    </div>

    <!--Green Row-->
    <div class="row">
    </div>

    <!--Buying Row-->
    <div class="row">
    </div>

    <!--Footer Row-->
    <div class="row">
    </div>

</body>
</html>
```

CSS

→ then the sty[...]
these entire classes/[...]
sections are tgt'ed in a[...]
(ca sep css file
linked to the HT[...]
file in its header)

GENERAL THOUGHT PROCESS OF MOCKUP → WEBPAGE



IN THE TOP OF THE INDEX.HTML FILE

to the header of the index.html document

%%%
Part #2 - Integrate a mockup 2. Add page metadata
%%%

METADATA

→ metadata – for SEO / organ'ing diff sites

→ source code → browsers search for it and otherwise diff fill it in

→ "the essential is invisible to the eyes"

→ meta tags → for content of the page
→ sec. networks

[in head of webpage HTML]

<meta ...>

```html
<head>
  <title> Catyler, our new hero</title>
  <meta name="description" content="I need a cat, I'm holding out for a hero 'til the end of the night. He's gotta be strong...">
</head>
```

FACEBOOK: OPEN GRAPH

THERE ARE SPEC. ONES YOU ADD IN FOR SOC. MED. DEP'ING ON THE SOC MED

→ facebook has its own attributes for metadata

→ meta tags look different for some specific social media sites

→ Structured entries

→ open graph → includes image, other prop's

```html
<!-- Open Graph data -->

<meta property="og:title" content="Title Here"/>
<meta property="og:type" content="website"/>
<meta property="og:url" content="http://www.example.com/"/>
<meta property="og:image" content="http://example.com/image.jpg"/>
<meta property="og:description" content="Description Here"/>
```

→ Catyler page – a "boilerplate" is a template for metadata

→ facebook wants to unify → is std. for metadata – so it has the "open graph protocol" – to be t'd → html metadata

TWITTER: TWITTER CARDS

→ Twitter has metatags (Twitter cards)

html

```html
<!-- Place this data between the <head> tags of your website -->
<title>Page Title. Maximum length 60-70 characters</title>
<meta name="description" content="Page description. No longer than 155 characters." />

<!-- Twitter Card data -->
<meta name="twitter:card" value="summary">

<!-- Open Graph data -->
<meta property="og:title" content="Title Here"/>
<meta property="og:type" content="website"/>
<meta property="og:url" content="http://www.example.com/"/>
<meta property="og:image" content="http://example.com/image.jpg"/>
<meta property="og:description" content="Description Here"/>
```

META TAGS SUMMARY

Metatags → go in the header of the index.html doc.
→ tag, aka <meta ... > in html
→ you can have any line e.g for certain soc. med. sites
– Twitter Cards, Facebook Open Graph

→ contain info info abt website for SEO, e.g /browser searching the websites by

TOWARDS A SEMANTIC WEB

HEADER OF WEBPAGE HTML, SEMANTICS RESULT YOU ARE LOOKING FOR IF CAT ANIMAL NOT CAT CONSTRUCTION COMPANY

→ semantic web ← semantic queries
→ cat → animal, also a construction company
→ 2011 → common standards – schema.org
→ structuring data on the internet

So it knows what cat is the [illegible] one

→ there are semantic tags (in index.html header, is < > </ >'s)
- Structured data testing tool
- schema creator ← returns metadata on a site

it's moving forward → the feature

%%
Part #2 - Integrate a mockup 3. Integrate a Bootstrap grid
%%

→ layout of el's according to a grid
→ Bootstrap = HTML, CSS, JS framework from Twitter
- for a) responsiveness
b) grid ← different screen sizes

MAKE THE CONTENT INTO A GRID IN HTML WITH DIVS

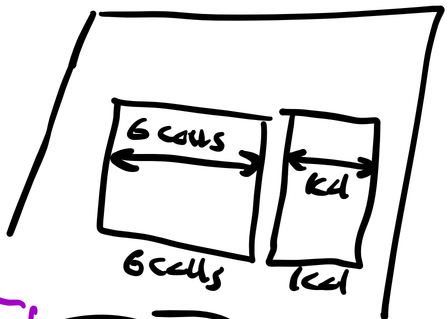
→ 12 cell grid

← bootstrap gives you cells
- you A width in bootstrap

→ a media query is for bootstrap
(breakpts for screen sizes

→ Photoshop also has 12 widths

→ to set a cell width



a cellum is ~ a basic unit of the page → an el can take up x6 e.g.

breakpts

XX = E

{xs, sm, md, lg}

768px

992px

1200px

= # of cells on 12 cell screen which that el takes up

You can measure the # of cells in Photoshop

```
class="col-XX-##"
```

```
<div class="col-md-8 col-sm-12 col-lg-6">
```

```
<!DOCTYPE html>
<html>
    <head>
    </head>
    <body>

        <!--Masthead-->
        <div class="row">
            <div class="col-md-2"></div> <!-- logo -->
        </div>

        <div class="row">
            <div class="col-md-8"></div> <!-- paragraph -->
            <div class="col-md-4"> <!-- phone image -->
        </div>

        <!--Two Columns-->
        <div class="row">
            <!--Left Column-->
            <div class="col-md-6"></div>
            <!--Right Column-->
            <div class="col-md-6"></div>
        </div>

        ...
    </body>
</html>
```

① dissect website poster, aka mockup into sections (divs)

② shell them out on an index.html file

③ add bootstrap classes to the divs, so hn how many cells the els take up

④ content in that div gets put → that cell #'s according to screen size

content

→ we are putting content in the webpage → div's

→ one el on HTML surrounded by a div

→ that div has a class

→ that class is formatted to say how many cells on a 12 cell screen the el will take up for

different screen sizes = bootstrap in Twitter

```html
<!DOCTYPE html>
<html>
    <head>
        <title>Take</title>
        <meta name="description" content="A mockup example" />
        <meta name="twitter:card" value="summary">
        <meta property="og:title" content="Title Here"/>
        <meta property="og:type" content="website"/>
        <meta property="og:url" content="http://www.takeapp.com/"/>
        <meta property="og:image" content="image.jpg"/>
        <meta property="og:description" content="Description"/>

    </head>
    <body>

        <!--Masthead-->
        <div class="row">
            <div class="col-md-2"></div> <!-- logo -->
        </div>

        <div class="row">
        </div>

        <!--Two Columns-->
        <div class="row">
            <!--Left Column-->
            <!--Right Column-->

        </div>

        ...
    </body>
```

metatags

sets up bootstrap cells in different parts of the webpage

- best practices
    - comments w/ the code
    - indent code for organ. → indent nested el's

%%
Part #2 - Integrate a mockup 4. Understand basic CSS principles
%%

→ CSS once have index.html file

→ Cascading style sheets ← for style

→ h1 {

    . . . .
    . . .

}

CSS syntax

DO CSS ONCE HAVE INDEX. HTML w/ CALLS OF ELS FROM BOOTSTRAP

props can ∆ in CSS →

```css
element {
    color:
    opacity:
    background-color:
    font-family:
    text-align:
    font-size:
    border:
    margin:
    padding:
    float:
    height:
    width:
}
```

→ the Mozilla developer network / documentation

CSS priority → rules you set take priority over default + later defined rules take precedent

→ Create a .css file + conn. it to index.html file header

```
h1 { color: white;
   background: orange;
   border: 1px solid bla
   padding: 0 0 0 0;
   font-weight: bold;
}
/* begin: seaside-theme */

body {
   background-color:white;
   color:black;
   font-family:Arial,sans-serif;
   margin: 0 4px 0 0;
   border: 12px solid;
}
```

CSS

CSS example

→ they are styles sheets

```
...
<body>

   <!--Masthead-->
   <div>
       <div class="container">
           <div class="row">
               <div class="col-md-2">
                   <img src="assets/logo.svg" class="logo"/>
               </div>
               <div class="col-md-8"></div>
           </div>

           <div class="row">
               <div class="col-md-8 col-sm-12 col-lg-6">
                   <h1>Picture perfect</h1>
                   <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
eiusmod tempor incididunt ut labore et dolore magna aliqua. </p>
                   <a href="#">Read More →</a>
               </div>
               <div class="col-md-4 hidden-sm-down">
                   <img src="assets/iPhone.png/">
               </div>
           </div>
       </div>
   </div>
   ...
```

make a .css file

CLASSES - LOOK @ WHAT WANT THE FORMATS TO BE

bootstrap cell

dummy text

nested divs go out

→ fonts → google fonts (Open Sans, can also choose character sets)

→ you inc. font in header in index.html

```
<link href='https://fonts.googleapis.com/css?family=Open+Sans'
rel='stylesheet' type='text/css'>
```

→ then in css → font-family: 'open-sans' e.g. ← name of font from google fonts

(name of font from google font)

So we → set-up index.html bootstrap cells w/ divs/class/cells w/ bootstrap from mockup, aka poster of webpage

→ get in a css sheet → link it to index.html in its header + add in links to fonts with it (↗ .css file)

→ format styling of webpage in CSS (... { }) and beware that priority rules (css rules go in a chronological order)

→ you then get in fonts → pick them on google fonts + add to html header section → then we the

content HTML < > < / > fwd to future

vs CSS =

s

Styling

Corresponding font name in CSS for styling, also bootstrap [illegible]

each HTML new line is understood in [illegible]

in CSS it's font-family

%%%

Part #2 - Integrate a mockup 5. Integrate specific CSS rules

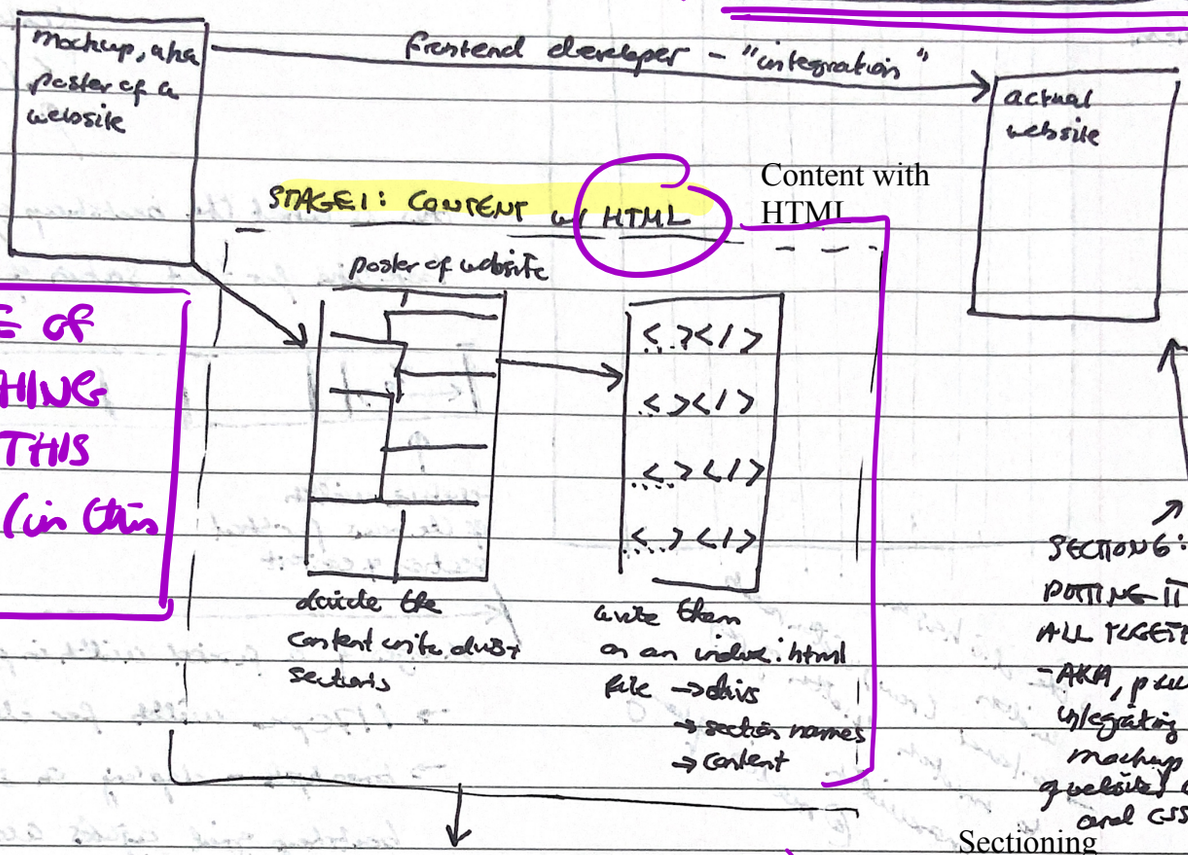
%%%

Integrating specific CSS rules

INTEGRATE SPECIFIC CSS RULES

In the context of everything else

OUTLINE OF EVERYTHING UP TO THIS POINT (in this note)



Colours / styling with CSS