Introduction

Before getting started on anything, the concepts of external stylesheets and the CSS box model require some introduction.

Open your "styles.html" file, or create a new HTML file using the same name.

External CSS

One can link external CSS files to an unlimited number of HTML files. Using external CSS files is usually preferable as this allows one to control an entire website's styling in one place. To better explain this concept, this section presents the same CSS applied in three different ways.

In your "styles.html" experiment with the inline, internal, and external CSS presented below.

inline

```
<body>
  ...

  <p style="color: red;">
    red text
  </p>
```

Inline styling (above) is highly specific and must be edited on a per-tag basis. It's generally considered poor practice. Convert the inline CSS to internal CSS:

internal

```
  <style>
    ...

    .red {
      color: red;
    }
  </style>

  ...

  <p class="red">
```

```
    red text
  </p>
```

Internal CSS (above) is an improvement over inline styling, but should really only be considered for styles specific to the HTML page in which they reside (and not styles that apply across multiple webpages of the site). Now, we will convert your internal stylesheet to an external stylesheet. Firstly, create a new file named "styles.css" and add a `<link />` tag that points to it via an `href` attribute.

external

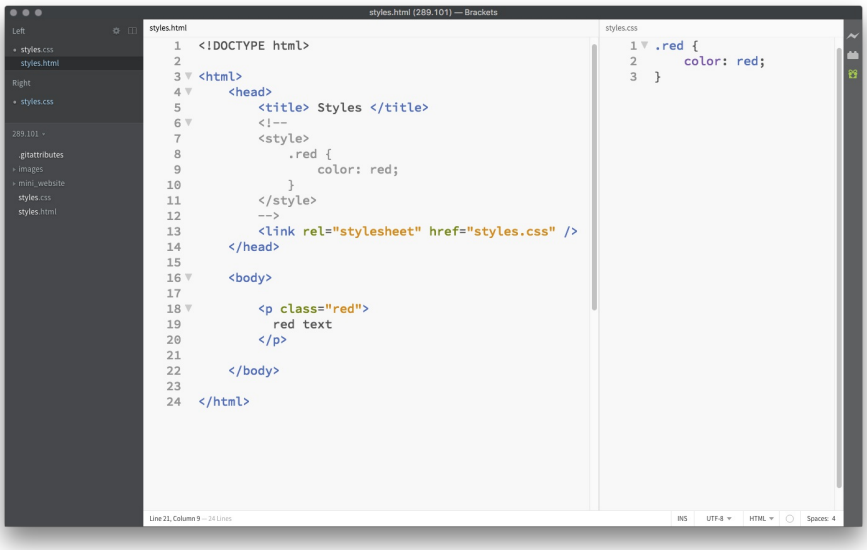
```
  <!-- COMMENT OUT internal stylesheet
  <style>
    .red {
      color: red;
    }
  </style>
  -->

  <!-- LINK TO external stylesheet -->
  <link rel="stylesheet" href="styles.css" />

  ...
```

Then, copy everything within the `<style> ... </style>` tags into this new stylesheet file. Do not copy the style tags themselves. So, the "styles.css" file contains nothing but the CSS that would otherwise be found between the `style` tags, i.e.:

styles.css

```
.red {
  color: red;
}
...
```

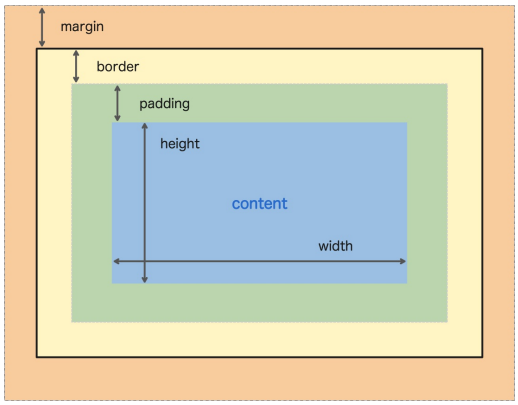
When working with an external CSS file, it's easier if you 'split' your editor into two panes -- one for your HTML and the other for your CSS file. In Brackets, use **View > Vertical Split**



If your webpage styling is working, you can delete the internal stylesheet altogether.

CSS Box Model

The CSS box model is used to control layout in webpages. Elements such as paragraphs, headings, and div's are all examples of *block-level* elements and, therefore, adhere to the CSS box model.



Block-level elements can be contrasted with inline-level elements, such as bold and italic tags, that do not affect the flow of the page. One can think of `<div>`'s as generic block-level elements -- kind of like paragraphs (`<p>`) with no gaps above and beneath them. Block-level elements can be filled with other block-level elements to create complex layouts.

You lecturer/tutor will run you through a quick experiment with some simple `<div>`'s. The one thing that catches-out many newcomers is how `padding` is added to the width of a `div`, hence `box1` (below) is square not a rectangle, despite having a `height` of `100px` and `width` of `200px` (although there is a way to change this behaviour using the `box-sizing` property).

CSS

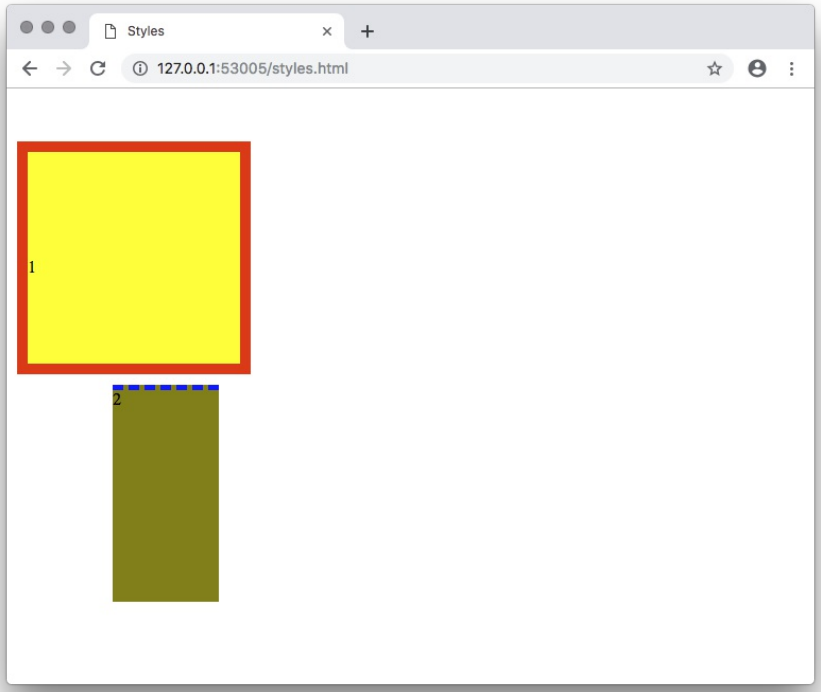
```
body {
```

```
  margin: 0;
}

#box1 {
  background-color: yellow;
  border: 10px solid red;
  height: 100px;
  margin-left: 10px;
  margin-top: 50px;
  padding-top: 100px;
  width: 200px;
}

#box2 {
  background-color: olive;
  border-top: 5px dashed blue;
  height: 200px;
  margin-left: 100px;
  margin-top: 10px;
  width: 100px;
}
```

html

```
  <div id="box1"> 1 </div>
  <div id="box2"> 2 </div>
```

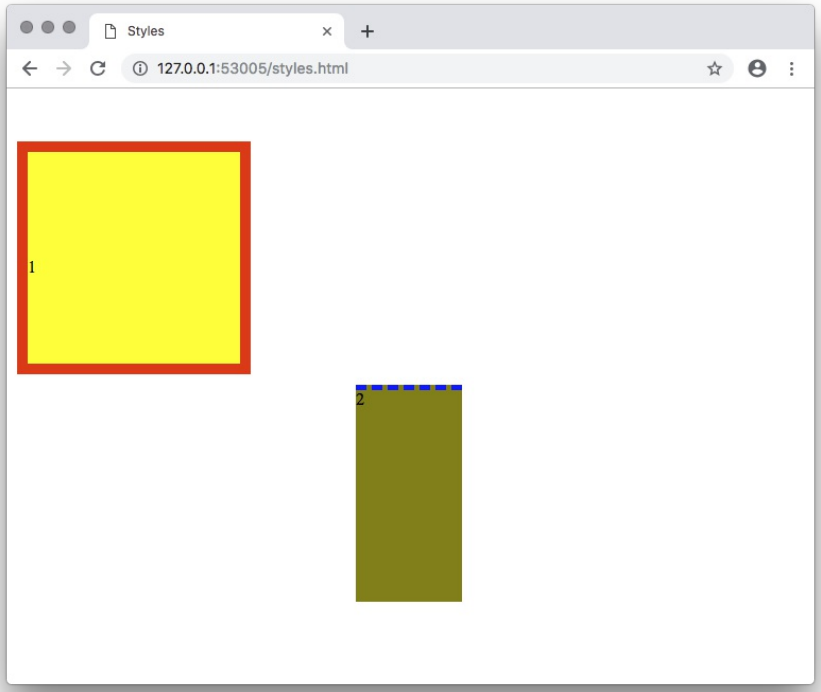
In this instance, the code makes use of ID (as opposed to `class`) selectors. These behave just like a class selector, except rely on an `id` attribute and a `#`. The difference is, elements with an ID should only appear once per webpage. Classes, however, are reusable.



To horizontally-centre a `div`, give it left and right margins of `auto`.

css

```
#box2 {
  background-color: olive;
  border-top: 5px dashed blue;
  height: 200px;
  margin-left: auto;
  margin-right: auto;
  margin-top: 10px;
  width: 100px;
}
```



OpenJam Website

You will now complete the *Open Jam* website using the lesson files provided on Stream. Your lecturer/tutor will get you started. This task will require a mix of the following CSS techniques/properties:

backgrounds
https://developer.mozilla.org/Web/CSS/background-color
https://developer.mozilla.org/Web/CSS/background-image
https://developer.mozilla.org/Web/CSS/background-position
https://developer.mozilla.org/Web/CSS/background-repeat
https://developer.mozilla.org/Web/CSS/background-size

borders
https://developer.mozilla.org/Web/CSS/border
https://developer.mozilla.org/Web/CSS/border-bottom
https://developer.mozilla.org/Web/CSS/border-left
https://developer.mozilla.org/Web/CSS/border-right
https://developer.mozilla.org/Web/CSS/border-top

font / text
https://developer.mozilla.org/Web/CSS/font-family
https://developer.mozilla.org/Web/CSS/font-size
https://developer.mozilla.org/Web/CSS/font-style

https://developer.mozilla.org/Web/CSS/font-weight
https://developer.mozilla.org/Web/CSS/word-spacing
https://developer.mozilla.org/Web/CSS/text-align

padding
https://developer.mozilla.org/Web/CSS/padding
https://developer.mozilla.org/Web/CSS/padding-bottom
https://developer.mozilla.org/Web/CSS/padding-left
https://developer.mozilla.org/Web/CSS/padding-right
https://developer.mozilla.org/Web/CSS/padding-top

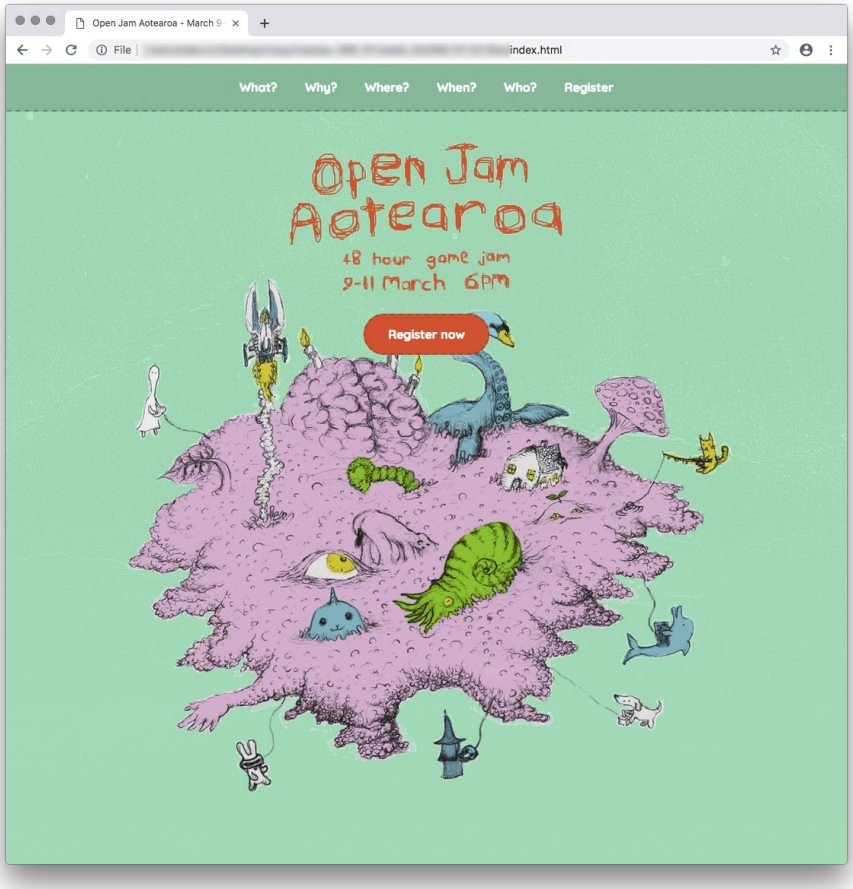
margin
https://developer.mozilla.org/Web/CSS/margin
https://developer.mozilla.org/Web/CSS/margin-bottom
https://developer.mozilla.org/Web/CSS/margin-left
https://developer.mozilla.org/Web/CSS/margin-right
https://developer.mozilla.org/Web/CSS/margin-top

If you miss the lesson, here is the CSS that you need to get you started:

```
@import url('https://fonts.googleapis.com/css?family=Quicksand:400,700');

/*
a wildcard selector for outlining all of your elements
you may find this handy for visualising divs
remove it when you are done, or if you don't want it
*/
* { outline: red 1px solid; }

body {
    background-color: #82DDB5;
    background-image: url('img/illustration.gif'), url('img/texture.jpg');
    background-position: center 280px, left top;
    background-repeat: no-repeat, repeat-x;
    background-size: 800px, auto;
    font-family: 'Quicksand', sans-serif;
    margin: 0;
}

#nav {
    background-color: #6B9;
    border-bottom: 2px dashed #487;
    color: white;
    font-weight: bold;
```

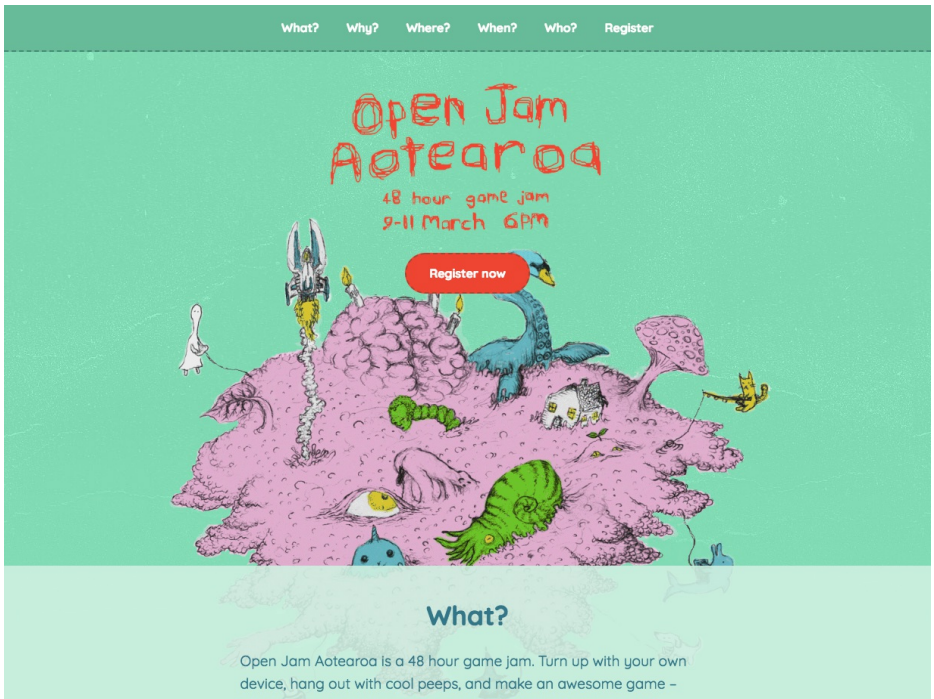
```
    padding: 20px 0;
    text-align: center;
    word-spacing: 2em;
}

#header {
    text-align: center;
    margin-top: 40px;
}

#logo {
    max-width: 360px;
}

#description, #date {
    max-width: 220px;
}

.register_btn {
    background-color: #E43;
    border: 2px dashed #A43;
    border-radius: 2em;
    color: #FFF;
    cursor: pointer;
    display: inline-block;
    font-weight: bold;
    margin-top: 25px;
    padding: 15px 30px;
}
```

Of course, this is just the start, so it will get you this far only:



Note that the default margins on the `<h1>` elements may cause odd gaps to appear between the sections. To avoid this, ensure that all of your section divs (`#what`, `#why`, `#where`, `#when`, `#who`) have sufficient top- and bottom padding.

The completed task looks like this (there is a *reference.png* provided with the workshop files):



even if you have never made a game before but love the idea!

There will be enough power points and WiFi available and we will provide snacks (and soup!) each day. We will have a networked harddrive on-site for participants to openly share assets created during the jam. We'd love to see some weird cross-pollination of game art/code/audio/etc. between games.

It will be $20 cash on the Friday to participate in the jam (the money will all go to food and paying the poster designer).

Sponsors

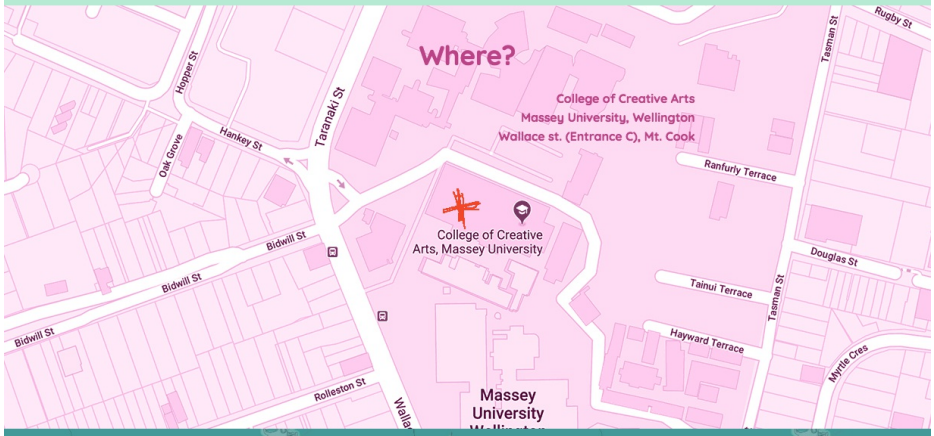
MASSEY UNIVERSITY College of Creative Arts Toi Rauwhārangi | ORACLE

Why?

A game jam is a hackathon for video games. It is a gathering of people for the purpose of planning, designing, and creating one or more games within a short span of time, in our case 48 hours. Open Jam Aotearoa aims to stretch the classical definition of video game development towards more artistic and experimental outputs. Addressing a prevalent lack of diversity in the game industry we plan to create an inclusive environment, inviting participants like artists, designers, musicians, storytellers that don't necessarily consider themselves programmers, developers or gamers. On Friday evening we will announce the theme of the jam and participants can spend as much time as they want during the opening hours to work on their game(s), either in a group or by themselves, if they prefer.

Where?

College of Creative Arts
Massey University, Wellington
Wallace st. (Entrance C), Mt. Cook



When?

Fri Mar 9, 6 PM – Sun Mar 11, 8 PM

Friday

- **6.00 PM** – Doors open
- **6.30 PM** – Theme announcement & Jam Kickoff
- **1.00 AM** (Saturday) – Vacating the premises and have a good night's sleep

Saturday

- **8.00 AM** – Doors re-open
- **1.00 AM** (Sunday) – Vacating the premises and have a good night's sleep

Sunday

- **8.00 AM** – Doors re-open
- **6.30 PM** – Jam ends
- **7.00 PM** – Gameplay
- **8.00 PM** – Prizes & Wrap-Up

Who?

Open Jam Aotearoa aims to be an open event for participants from all backgrounds and skill levels who are keen to spend a weekend together with like-minded creative people to develop games. We are especially inviting people who do not consider themselves traditional game developers and who can bring new perspectives, stories and skills to the jam.

Participants do not need to be part of a team to be involved, they can attend and contribute their skills to any other project at the jam freely.

end