**Imperial Visualisations**

****

*Suite Guide*

Welcome to the Imperial Visualisations developer team! It’s expected that a new developer has no previous experience in any web-based language; all we expect is knowledge of code and the enthusiasm to learn. This document will take you through the three main languages with which Imperial Visualisations are developed: HTML, CSS, and JavaScript. Although we have many resources available to

**HTML**

HTML is the backbone of any website. It’s less of a language and more of a collection of elements that define what goes on your page. These elements are defined using opening and closing tags. For example, if I wanted to have an element displaying “hello world”, I would have an open <div> tag, followed by the text, followed by a closing </div> tag. So overall the element looks like:

<div>hello world</div>

These elements can contain anything from text to a slider to a plot, and can be styled to your preference! For more tutorials on HTML, here are a few useful resources:

* W3 Schools: <https://www.w3schools.com/html/html_intro.asp>
* Codeacademy: <https://www.codecademy.com/learn/learn-html>

**Structuring your JavaScript**

A typical Imperial Visualisations JavaScript file should be split into 3 sections: ‘Maths’, ’Interaction’, and ‘Call’.

* Maths – contains all the scientific code behind a visualisation
  + Start by defining maths functions
  + Then define maths functions and tools
* Interaction – contains all the scientific code behind a visualisation
  + Start by defining maths functions
  + Then define maths functions and tools
* Maths – contains all the scientific code behind a visualisation
  + Start by defining maths functions
  + Then define maths functions and tools

Then define jquery variables

Then define interaction functions

Specify which functions are called onclick, etc

Specify which functions are called onload

When dealing with updating/on load, only have one function, that takes True False value for ‘new’ variable. This either carries out plotly.newplot() (if new) and plotly.animate/react (if false).