**Imperial Visualisations**



*Style Guide*

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Last updated** | **Updated By** | **Reason for Update** |
| 1.5 | 24.11.2019 | Ben R-W | Writing visual ID section |

**Ethos**

At Imperial visualisations, we have several design principles and goals that help guide how visualisations look and feel. As developers, you should always keep these principles in mind; as long as you adhere to our ethos, your specific styling can vary however you want.

* **Playful**: We want to encourage people to get involved in our visualisations. The colours, layout, and animation should be enticing, and invite creativity and curiosity.
* **Intuitive**: Although every ‘learn’ visualisation should be accompanied with guidance, any person should be able to interact with a visualisation immediately. Buttons to be pressed and sliders to be dragged should seem to “pop” out the screen.
* **Light**: In terms of tone, any Imperial Visualisation should never feel dark, heavy, or too artificial. Instead, it should feel light, fun, and natural.
* **Clean and Concise:** As we are dealing with difficult scientific concepts, one of our main priorities is not to overwhelm the user with complexity. As such, each visualisation should look clean, and organised (but not disparate).
* **Part of a journey**: Every visualisation should be taking the user on a journey. The overall design of a page should reflect this; the user’s attention should be drawn to the start of the visualisation journey and move slowly towards the journey’s end.

**Cascading Style Sheet**

Cascading Style Sheet, or CSS, is how we style almost everything on a given web page. CSS is a series of properties and their corresponding values describing all the stylistic features of a HTML element. For example, the following piece of code can set the width, height, and text colour of a div:

width:100%; height: 100%; colour: “#003E74”;

Everything from colour to size to animations is defined using CSS. One way of defining an HTML elements style using CSS is to embed the CSS directly into the element. An example of embedding CSS directly in an HTML element is:

<div class = “textbox” id = “theory” style = “width:100%; height: 100%; colour: “#003E74”;”> Hello world! <\div>

*Note: If you don’t understand the above code, it may be useful if you have a look at the HTML section in the Functionality Guidelines document!*

Alternatively, an elements style can be inherited from a class or ID, with the actual CSS written in a separate standalone file. An example for the HTML in this case is:

<div class = “textbox” id = “theory”> Hello world! <\div>

With the separate stylesheet containing the following CSS:

.textbox{

width: 100%;

height: 100%;

}

#theory{

Colour: “#003E74”;

}

A typical visualisation will have a basic inherited style from one of two global stylesheets: skeleton.css, and style.css. Any further detailed styling by a developer is typically done directly in the HTML file.

For more on CSS, visit:

W3 Schools: <https://www.w3schools.com/css/css_intro.asp>

**Note on Sizes:**

When defining the dimensions of a class in a CSS file, this can be done using absolute or relative units. Note that in the first line of example code in the document, relative units have been used (height and width defined in percent). Another way of using relative units would be to write height in terms of “vh” (view height) and width in terms of “vw” (view width); eg:

width:75vw; height: 85vh;

where 1vw means 1% of the width of the viewport (the user's visible area of a web page), and the same for vh.

Absolute sizes are generally defined in pixels, eg:

width:550px; height: 300px;

The disadvantage with this method of defining sizes is that certain elements will end up being different sizes, depending on the resolution of the user’s screen. For this reason, when defining div sizes, it is strongly recommended you use relative units.

**The ImpVis Visual Identity**

**Colour Schemes:**

When developing the colour scheme for any individual visualisation, special attention should be paid to keeping the visualisations colour-blind friendly. In practice, this means making good use of monochromatic schemes when possible, and using online tools to ensure that, when contrasting colours are chosen (eg. in plots), they are visibly contrasting to everyone. Some useful links are included below:

<http://colorbrewer2.org/#type=sequential&scheme=BuGn&n=3>

<https://venngage.com/blog/color-blind-friendly-palette/>

In general, visualisation colour schemes can be based off the main colour scheme, shown here.

A screenshot of a cell phone

Description automatically generated

The colour scheme is mainly monochromatic, with only one “contrasting” colour. We number the colours from left to right 1 through 5. All blues used in visualisations and across the wider page should be from shades 2 to 5, for consistency, while if only one contrasting colour is required, then it is advised to stick to colour one if opting for a red/pink shade – if another colour is chosen then the choice more open.

**Page layout**

From left to right, 33% of the page is occupied by explanatory text, and derivations, while the remaining 67% is occupied by the accompanying vis. The left panel is scrollable, and it is part of this left panel that is visible on the screen (ie. How far the user has scrolled down the explanatory text) that determines the visualisation shown on the right. There is a nav bar across the whole top of screen – there is still work to be done on determining the dimensions of the nav bar, but a definitive size based in relative units **must** be decided, preferably between 5 and 8vh.

**Text**

Imperial college London’s executive typeface “Meta normal” is used as the font on all visualisations. For more information on this, see:

<https://www.imperial.ac.uk/brand-style-guide/visual-identity/font/>

You can refer to “Notes\_on\_format\_for\_clarity\_and\_comprehension.pdf” in this repository for more information.