

HDE  
FrontEnd Test  
Report – Damien Hervieux

1 – Create the mockup with hardcoded data using HTML and CSS

- find colors schemes (GIMP and color-hex.com)
- initiate a SASS file, import partials (box-sizing, css reset, etc.) and set variables
- use NPM scripts to watch SCSS and Autoprefixer CSS
- make each feature of the page modular using class names
- create the button search with a sprite image
- create git repository

2 – Move the hardcoded data from HTML into Javascript - prevents HTML to be cluttered with repetitive elements and makes the data available in javascript

- create an array to hold samples of emails (as objects)
- loop through all emails in the array to display them correctly in a table row
- store email count and display it in the “Results”
- delete all hardcoded data from HTML
- create an event listener that calls a the sort method when “Date” is clicked
- Call render() to display the new sorted array. This way makes it easy to implement the new feature on mission 2 (content will always be linked to the right row even after being sorted).
- add the new UI feature to display the body of each email:  
use css class to toggle between hide and show on click events

3 – Refactor the Javascript and CSS files – Make it modular

- JS: create a module in Javascript with init(), cacheDom(), bindEvents() and render() methods. Add all others functions as methods to get a single module
- CSS: make class names clearer. Reorganize each feature correctly in modules with the help of class names

4 – Add details

- to show an email is currently displaying its body content: add an arrow to the left when the email is clicked
- to show search list is empty:
  - empty email array and add the logo picture as table content (basic.png)
  - add a “Clear” button to clear the search and display this page
- add some media queries to make the page a little more responsive