Task description

You are going to create a **mobile first** responsive web-page about **making green energy affordable for people with average budget**. The web-page will have **nav-bar**, **masthead**, **main-content** and **footer**. Nav-bar has 4 items: **home**, **about**, **contact** and **blog**. Masthead has to have a background image and a tag-line.

For main-content, you will have 3 buttons and 3 contents (your 3 solutions about making a clean energy affordable for majority of people) stored in an object. On the page load, the first content will be hard-coded HTML. This content will be replaced by another content from JavaScript object once a button is clicked. The first button (because it corresponds to the first content displayed on the page-load) will have a different style than other two (indicating active button). As user clicks any button, id "active-button" will move to the clicked button and corresponding content will be displayed. Size of images and fonts change with the screen-width. Break-point for responsive design will be at: **30rem.** **Max-width of content is 60rem**. This means that content has to be centered in the view-port once 60rem of content-width is exceeded.

Footer needs to have links to at least 3 social media networks (use icon-font).

Order of Operations

1. Create directories and files
2. Create wire-frame and mockup
3. Complete HTML
4. Complete CSS
5. Complete JavaScript

Directories - Structure

Directory tree (including names) has to look like this:

/green-energy/

/css/

/img/

/wireframe-mockup/

/js/

/index.html

Fonts

You need to use serif/sans-serif Google font pair. Serif font is for headings and sans-serif font is for body text. Fall-back font is going to be default serif/sans-serif font.

References:

* [http://fontpair.co (Links to an external site.)Links to an external site.](http://fontpair.co/)
* [https://fonts.google.com (Links to an external site.)Links to an external site.](https://fonts.google.com/)

Line height ratio has to be between 1.4 and 1.6. You need to use following font-sizes:

* Mobile View: 0.75rem
* Tablet/Desktop View: 1rem

Colors

Color scheme needs to be analogous. List the hexadecimal values of colors (as CSS comment) inside CSS style-sheet .

HTML elements with background color will expand across the screen while content is centered in the middle of view-port and has limited width.

Reference: [color.adobe.com (Links to an external site.)Links to an external site.](https://color.adobe.com/create/color-wheel/?base=2&rule=Analogous&selected=2&name=My%20Color%20Theme&mode=rgb&rgbvalues=0.48828182115766083,0.013612449956360129,1,0.19953300165134202,0.012387329460287717,0.91,0.036387550043639916,0.09611296892434906,1,0.012387329460287717,0.3042358934196418,0.91,0.013612449956360129,0.5939008286677046,1&swatchOrder=0,1,2,3,4)

Images

You could use **your own images** or choose images from **http://www.freeimages.com/** or **http://www.morguefile.com/**.

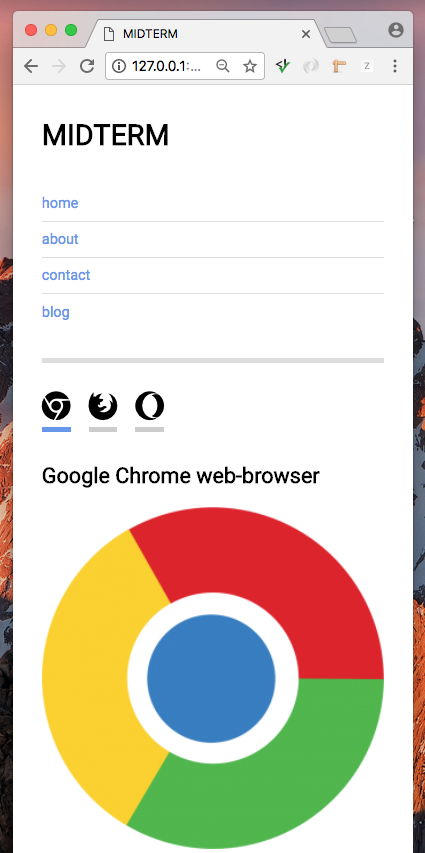
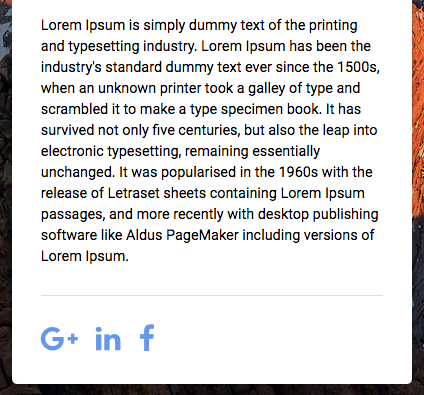
All images have to be OPTIMIZED for web.

Set the background image for masthead.

The screen-shots below are not to be replicated in design nor content - **they only represent the guidelines for required layout and alignment**. The topic is already given in the task description. You will choose **your** color-scheme, **your** images and **your** font-family (font-stack), BUT the content **has to be responsive in the way that screen-shots indicate** (including order of elements, alignments, horizontal and vertical gutters of white space).

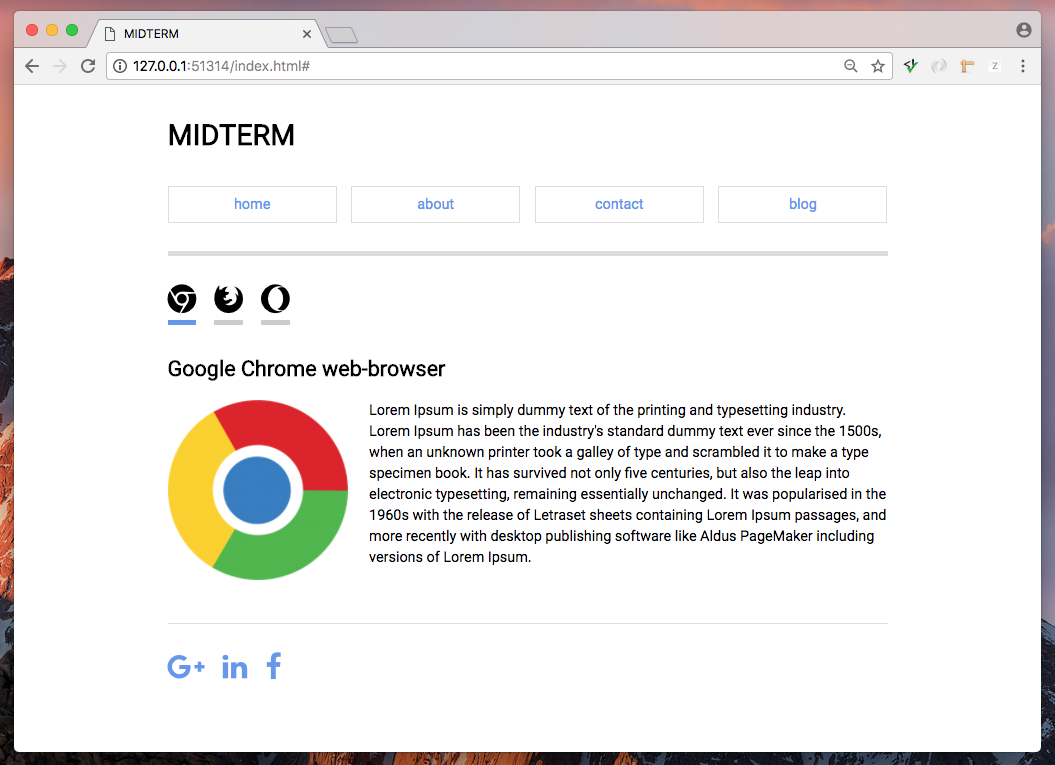
Mobile View < 30rem

The gutter of white space around the content is 2rem. Everything aligned left.

 30rem < Tablet/Desktop View < 60rem

Nav-bar items are floated and horizontally aligned. Main-content image-width is 25%. The image is floated - text wraps around.



Coding Guidelines

// Load your images on page-load  
    function preloader() {  
        const imagesList = [  
           "./img/img-1.jpg",  
           "./img/img-2.jpg",  
           "./img/img-3.jpg"  
        ];  
        const images = [];  
        for (let i = 0; i < imagesList.length; i++) {  
            images[i] = new Image();  
            images[i].src = imagesList[i];  
        }  
  
        // Images ready to be used:  
        console.log(`Preloaded images:\n\t${images[0].src}\n\t${images[1].src}\n\t${images[2].src}`);  
    };      
    window.addEventListener("load", preloader);  
      
      
    /\*   
    Get all buttons in a NODE LIST of buttons (array like structure) \*/  
      
    /\*   
    Complete your resource-object that will store the dynamic content.  
    Resource object should 3 sub-objects. Give your sub-objects  
    meaningful names. Every sub-object should have the following  
    properties headingContent, bodyText, imgUrl and imgAlt. \*/  
  
 /\*   
    Get the reference to your HTML-container  
   that will be dynamically loaded from the resource-object. \*/  
   
 /\*   
 The first button in a NODE LIST of buttons will initially   
 have the id: active-button - this will uniquely style   
 the active button (CSS rule).   
   
 The first content from the  
 resource-object will be loaded on the page load:  
 `<h1>${headingContent}</h1>  
    <img src="${imgUrl}" alt="${imgAlt}">  
     <p>${bodyText}</p>` \*/  
   
    /\*   
    Start your handleSelection function here. \*/   
          
        /\*   
        Remove the id active-button from the element that  
        contains it prior to the click-event.   
  
 This will require the loop throught the NODE LIST of buttons.   
 Inside the loop, use conditional and the element object method  
 hasAttribute() to check if the current button in the loop containes the id.  
 If it does, use element-object property removeAttribute()  
 to remove the id. \*/  
  
        /\*  
        Use the element-object method setAttribute() to set the id active-button   
 to the currently clicked button. \*/  
      
        /\*   
        Use conditional and event-object to check which button is clicked  
 and based on that, create HTML with the data inside the backticks:  
 `<h1>${headingContent}</h1>  
        <img src="${imgUrl}" alt="${imgAlt}">  
        <p>${bodyText}</p>`  
 Assign this content to to your HTML-container that will   
 be dynamically loaded (you already got the reference to   
 this container before you started the function handleSelection) \*/   
      
    /\*   
    Close your handleSelection function here. \*/    
      
    /\*   
    Register all buttons to click event. The event-handler handleSelection will listen   
 for this event to happen. \*/

Submission

Create a GitHub Repository called "green-energy".

Add the gh-pages branch to the green-energy repo.

index.html has to be in the root of your repo.

Submit the link to your gh-pages rendered site.

Eg: https://abcd0001.github.io/green-energy/index.html

(Change the abcd0001 to your own username)

TEST your link to make sure it works before submitting the link.

Links that return a 404 error are worth zero.

Rubric

**Clean Energy**

| Clean Energy | | |
| --- | --- | --- |
| **Criteria** | **Ratings** | **Pts** |
| This criterion is linked to a Learning OutcomeGitHub repo | |  |  | | --- | --- | | 1.0 pts  GitHub Repo named as requested and gh-pages link works | 0.0 pts  One of the requirements is missing. | | 1.0 pts |
| This criterion is linked to a Learning OutcomeFile structure | |  |  |  | | --- | --- | --- | | 2.0 pts  File structure created as requested | 1.0 pts  One directory is missing or is not properly named | 0.0 pts  More than one directory is missing or is not properly named. | | 2.0 pts |
| This criterion is linked to a Learning OutcomeValidity of HTML and CSS | |  |  |  | | --- | --- | --- | | 1.0 pts  HTML and CSS files don't contain errors by HTML5 and CSS3 standards | 0.5 pts  One of the files containes error(s) | 0.0 pts  Both files contain error(s) | | 1.0 pts |
| This criterion is linked to a Learning OutcomeWireframe and mockup | |  |  |  | | --- | --- | --- | | 1.0 pts  Wireframe and mockup screenshots are placed inside their directory | 0.5 pts  One of the screenshots is missing or is not placed inside requested directory | 0.0 pts  More than one requirement is missing | | 1.0 pts |
| This criterion is linked to a Learning OutcomeSwitch content code functionality | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 4.0 pts  There is no logical nor syntax errors. Reveal-hide content runs as requested. | 3.5 pts  One logical or syntax error or reveal-hide content doesn't run properly | 3.0 pts  Two logical or syntax errors or reveal-hide content doesn't run properly | 2.5 pts  Three logical or syntax errors or reveal-hide content doesn't run properly | 2.0 pts  Four logical or syntax errors or reveal-hide content doesn't run properly. | 0.0 pts  More than four logical or syntax errors | | 4.0 pts |
| This criterion is linked to a Learning OutcomeHTML, CSS and JS code formatting | |  |  |  |  |  | | --- | --- | --- | --- | --- | | 2.0 pts  There is no misalignments in code. JavaScript file doesn't contain the coding guidelines. | 1.5 pts  One misalignment in code | 1.0 pts  Two misalignments in code | 0.5 pts  Three misalignments in code | 0.0 pts  More than three misalignments in code | | 2.0 pts |
| This criterion is linked to a Learning OutcomeBreak-points and layouts | |  |  |  | | --- | --- | --- | | 1.0 pts  Break points set as requested | 0.5 pts  One break-point is missing required value | 0.0 pts  More than one requirements is missing | | 1.0 pts |
| This criterion is linked to a Learning OutcomeAlignments | |  |  |  |  |  | | --- | --- | --- | --- | --- | | 2.0 pts  Entire content is properly aligned (vertically and horizontally) | 1.5 pts  One instance of misalignment | 1.0 pts  Two instances of misalignment | 0.5 pts  Three instances of misalignment | 0.0 pts  More than three instances of misalignment | | 2.0 pts |
| This criterion is linked to a Learning OutcomeHorizontal and vertical gutters between HTML elements | |  |  |  |  |  | | --- | --- | --- | --- | --- | | 2.0 pts  All HTML elements have proper and consistent gutters | 1.5 pts  One instance of missing gutter | 1.0 pts  Two instances of missing gutter | 0.5 pts  Three instances of missing gutter | 0.0 pts  More than three instances of missing gutter | | 2.0 pts |
| This criterion is linked to a Learning OutcomeFont-size, type and line-height | |  |  |  | | --- | --- | --- | | 1.0 pts  Font-sizes are set as requested for every view-port. Serif font is used for headings and sans-serif is used for body text. Line-height is between 1.4 and 1.6 | 0.5 pts  One requirement is missing. | 0.0 pts  More than one requirement is missing | | 1.0 pts |
| This criterion is linked to a Learning OutcomeColor-scheme and images | |  |  |  | | --- | --- | --- | | 1.0 pts  Color scheme is analogous. Hexadecimal values of colors are listed inside CSS style-sheet as CSS-comment placed at the top of your stylesheet. Images have to be optimized for web. | 0.5 pts  One requirement is missing | 0.0 pts  More than one requirement is missing | | 1.0 pts |
|  | | |