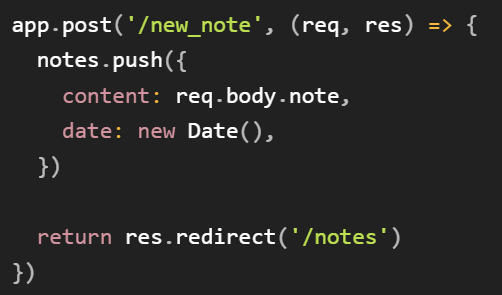
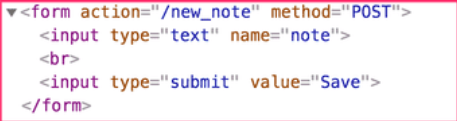
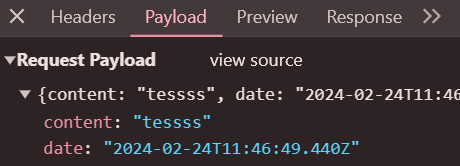
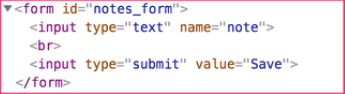
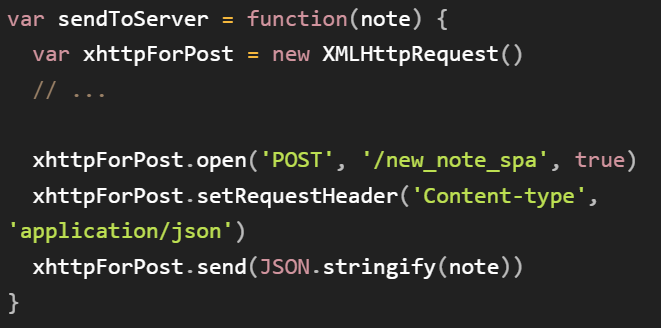
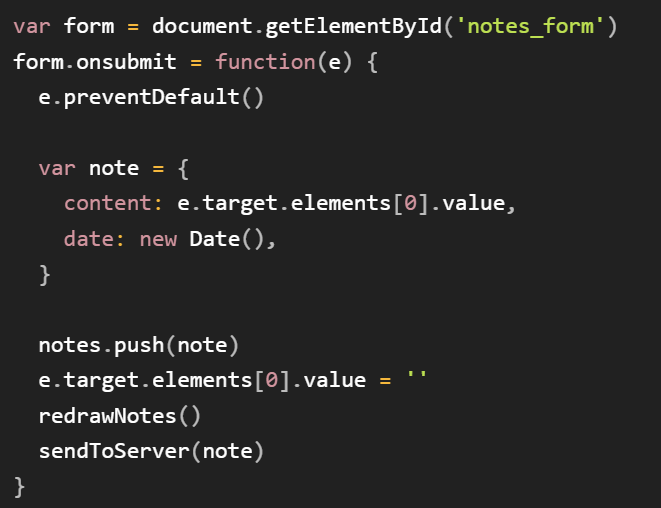
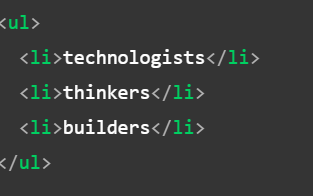
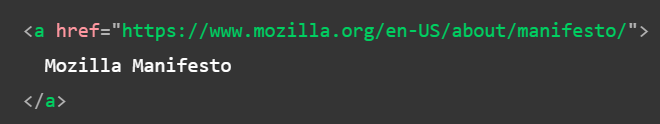
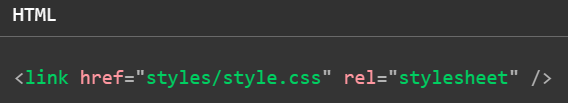
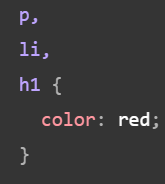
1. Part 0

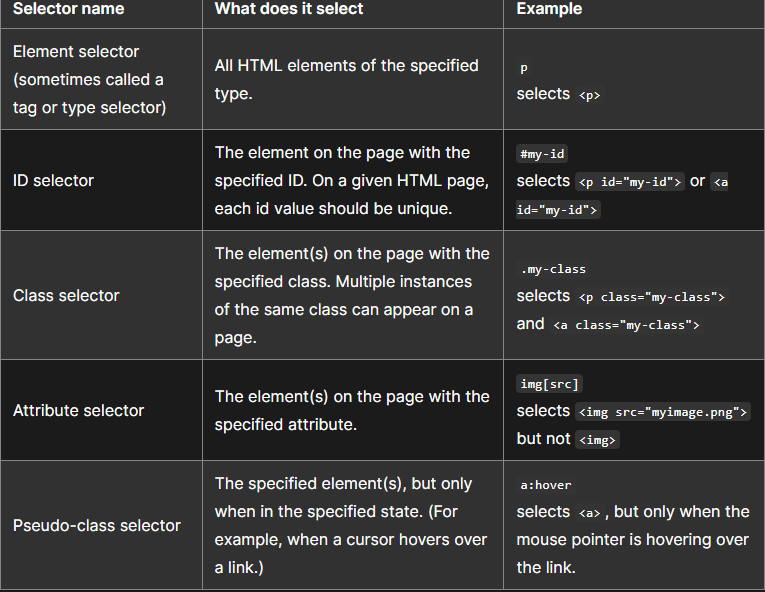
* Do get with the following



* CSS modify style based on class name
* JS do look for with id
* Older style of POST
  + 
  + The server responds the HTML POST with redirect causing page reloads
  + The data is sent to the server via the form's submit, and the server instructs the browser to reload the Notes page with a redirect.
* Single page app style
  + 
  + The browser sends only one request bringing JSON and server doesn't not ask for redirect
  + 
  + e.preventDefault() to prevent the default handling of form's submit. The default method would send the data to the server and cause a new GET request
* HTML
  + <head></head> — the <head> element. This element acts as a container for all the stuff you want to include on the HTML page that isn't the content you are showing to your page's viewers.
  + <meta charset="utf-8"> — This element sets the character set your document should use to UTF-8 which includes most characters from the vast majority of written languages.
  + <meta name="viewport" content="width=device-width"> — This viewport element ensures the page renders at the width of viewport,
  +  In the alt attribute, you specify descriptive text for users who cannot see the image
  +  ul = unordered list, ol = ordered list
  + Adding link, dont omit the http:// or https:// part
* CSS
  + Apply css to html file inside of the head tag
  + To select multiple elements

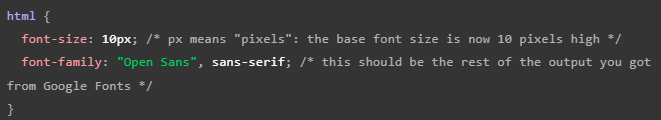


* + Type of selectors



* + Add the following to the HTML head tag to load Open Sans font family along with our webpage





Add obove to CSS file, this rule defines a global base font and font size for the whole page. Since <html> is the parent element of the whole page, all elements inside it inherit the same font-size and font-family.

* + Several properties:

padding, the space around the content. In the example below, it is the space around the paragraph text.

border, the solid line that is just outside the padding.

margin, the space around the outside of the border.

width (of an element).

background-color, the color behind an element's content and padding.

color, the color of an element's content (usually text).

text-shadow sets a drop shadow on the text inside an element.

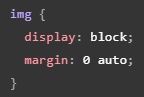
display sets the display mode of an element. (keep reading to learn more)

( Notes:

When you set two values on a property like margin or padding, the first value affects the element's top and bottom side; the second value affects the left and right side. ( auto is a special value that divides the available horizontal space evenly between left and right). You can also use one, two, three, or four values.

border: 5px solid black; This sets values for the width, style and color of the border. In this case, it's a five-pixel–wide, solid black border, on all sides of the body.

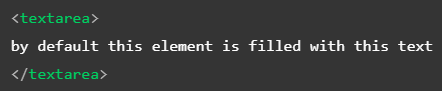
 horizontal offset, vertical offset, blur radius, base color

images are inline elements, for the auto margin trick to work on this image, we must give it block-level behavior using display: block;.

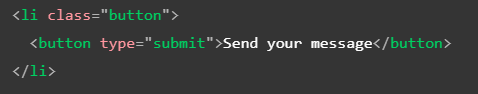
* Designing Form



* + The input field for the name is a single-line text field.
  + The input field for the email is an input of type email: a single-line text field that accepts only email addresses.
  + The input field for the message is a <textarea>; a multiline text field.
  + The ‘for’ attribute on all <label> elements, which takes as its value the ‘id’ of the form control with which it is associated — this is how you associate a form control with its label. ( it associates the label with the form control, enabling mouse, trackpad, and touch device users to click on the label to activate the corresponding control, and it also provides an accessible name for screen readers to read out to their users.)
  + ‘type’ attribute defines how the ‘input’ element appears and behaves
  + 



‘input’ tag is a void element and ‘textarea’ is not, thus the way of defining the default value is different

* + 

add button before the closing /ul tag, ‘type’ attribute accepts one of three values: submit, reset, or button. (submit: send data to web page defined by ‘action’ attribute of ‘form’ element, button: is customizable with js)

* + ‘name’ attribute for each form control is important as server handles each piece of data by name/ value pairs.