Documentation

This API uses the pokdex.json that contains information about the original 151 Pokemon, such as their types, weaknesses, evolutions, etc. At the default URL, the API will get and display all the Pokemon on the home page (displaying their name, pokedex number, and picture). There will be different fields like text boxes and drop-down menus that will allow the users to search for different endpoints.  
  
Endpoints:

/name: This endpoint will take you to a page that displays all of the Pokémon information if that name was typed in the textbox

/type: This endpoint will take you to a page that displays all Pokémon of a designated type, based on what you have selected from a dropdown menu

/finalStageEvolution: This endpoint will take you to a page that displays all Pokémon that are fully evolved (Either last in the evolution stage or have no evolutions)

/random: This endpoint will take you to a page that displays a random Pokémon’s information after a button press

Work completed: Right now I have a basic layout for the different end points and what I want each one to do/information to get. I have mostly just laid the groundwork so I need to actually do some testing and actually fully implement everything.  
  
To do: I need to make my actual html and css file to organize the page and its information. As stated before I also need to fully implement and test my endpoints, as well as calling the proper function and getting urls when certain buttons are pressed and certain fields are filled, as well as accounting for different status error codes.

Above and beyond: I might be planning to have different html files for different pages. I will want the home page to just display a list of all pokemon and their number name type and picture, but when going to a page to view a specific pokemon, I will want all of its information to be displayed (like its evolutions, weight, height, etc). I am not sure if doing this and having a different styling for a new page will be going above and beyond though.

Borrowed code: I have been trying to have similar file, function, and variable naming conventions like the ones used in class. If this is supposed to be accounted for borrowed code then these naming conventions and structured functions are from class.