Scripting and Computer Environments
Monsoon 2018

ASSIGNMENT 2

[HTML, CSS & JavaScript/jQuery]

General Instructions

Deadline : 9th October (Tuesday), 11:55 PM.

Submission Format

Submit all your codes in a zip file named <RollNumber>_Assignment2.zip (Example 20172010_Assignment2.zip).

Make a folder for each Question (Example **Question1**, **Question2**) containing the respective codes and files. Please properly indent your HTML, CSS and JavaScript Files to make them readable.

If the submission format isn't followed, you will get a straight zero.

You are free to either use JavaScript or jQuery.

Plagiarism and Copying

If anyone is found copying code from the friends or the internet, the one who helps and the who got the help will get a straight zero and any kind of plagiarism will not be tolerated.

All the Best :)

Doubts and Queries

If you have any doubts regarding any questions or submission formats, you can ask them on Moodle

Questions

1. Your Own “Try It Yourself!” Editor

We are pretty sure you must have used “W3Schools” to learn about Web Technologies for your Scripting Course. (Do give them a visit if you haven’t yet). They feature a nice “Try It Yourself!” Editor (Link : https://www.w3schools.com/html/tryit.asp?filename=tryhtml_intro). Your Task will be to implement the same using HTML (+ CSS) and JavaScript/jQuery only. You don’t need to copy the look of the editor, so you can go totally creative and use Bootstrap (or any other CSS Framework) and style it the way you want:

Objectives :

- Like the W3Schools counterpart, it also must have 2 panes, one to write the code and the other where the output will be displayed.
- Your Editor must have 3 buttons : one to run/display the HTML Code, one to clear the display pane and one to clear the code area.
- Use proper CSS to position the elements and buttons.
- Your Editor must support CSS and JavaScript too.
- Only JavaScript/jQuery is allowed to create this.

Hints :

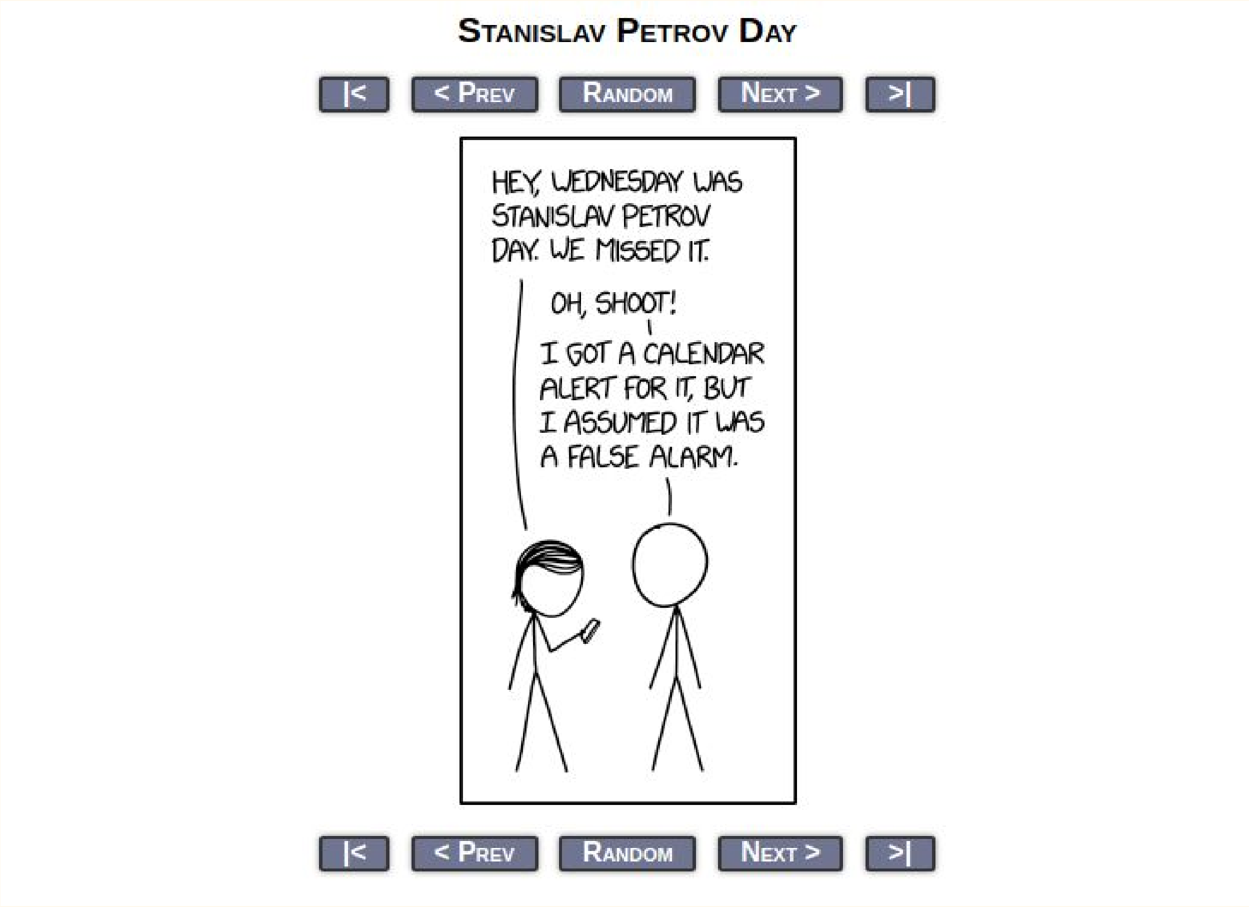
- If you can visualize it, the right side of the editor (the part where output is displayed) is an iframe. You just need to figure out how to get your code/inject your code in that iframe.
- Don’t try to “Inspect Element” the Editor. It’s coded in ASP and you won’t get any help there. But you will get a zero :-)

Brownie Points :

- Since you are making your own version of this editor, you can style the text, input area, display frames and the buttons in your own creative ways and creativity will fetch you extra points. You are free to use any CSS Framework or Styles.
- Try making the editor to work like the “Atom Live Server” plugin (automatically re-run the code when it is updated), that will also fetch you extra points.
- You can add your extra functionalities or buttons if you can think of any.

2. The xkcd clone

Randall Munroe, a Nasa contract programmer turned cartoonist, is the creator of the webcomic xkcd. All hail Randall Munroe! We wouldn't want the world to lose out on these amazing webcomics if Munroe decides to go rogue and removes them from his site. Create an xkcd clone using the xkcd api.


Stanislav Petrov Day
|<
< Prev
Random
Next >
>|
HEY, WEDNESDAY WAS STANISLAV PETROV DAY. WE MISSED IT.
OH, SHOOT!
I GOT A CALENDAR ALERT FOR IT, BUT I ASSUMED IT WAS A FALSE ALARM.
|<
< Prev
Random
Next >
>|

This is a typical xkcd page. Link to the api: https://xkcd.com/json.html

Objectives :

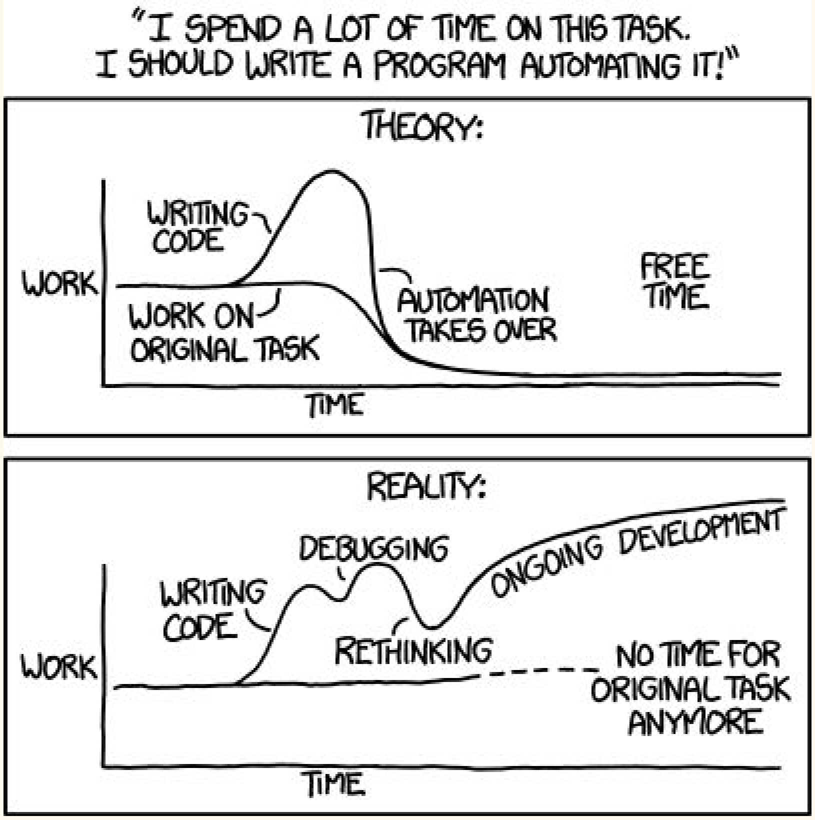
- The clone should have the title, the comic image, and the navigation buttons.
- Implement the First, Prev, Random, Next, and Last comic features.
- Make sure you don't reload the page to change the comic when you click on the navigation buttons.

API explanation:

- The response of the api consists of all the information that you may require in a JSON format.
- The *i* th comic' information would be found at the link:
  - https://xkcd.com/*i*/info.0.json
- The last comic information would by default be found at:
  - https://xkcd.com/info.0.json

Note: We are not looking for a web scraped/database based approach. You site should make an appropriate request to the api everytime a comic has to be fetched.

If you made it this far, here's something to cheer your mood:



3. The Search of Ice and Fire

Imagine how a modern day citadel would work with the power of the internet.

Implement a Search on the Characters and Houses from the ‘The Song of Ice and Fire’ using the api: https://anapioficeandfire.com/

Why you ask?

“Because that’s what we do. We click and we know things.”

Objectives :

- Take some time to learn about the api and how it may be used for your problem.
- You are expected to implement search given a character or a house and print the appropriate result in a presentable manner. You may choose to filter out some of the results from the response of the api.
- Use your creativity and discretion. There is no specific format of the page that we expect. Just make sure it is presentable and you are not just populating the page with the results.
- Appropriate error message should be provided if no such character or house is present.

API explanation: https://anapioficeandfire.com/Documentation

Features Expected :

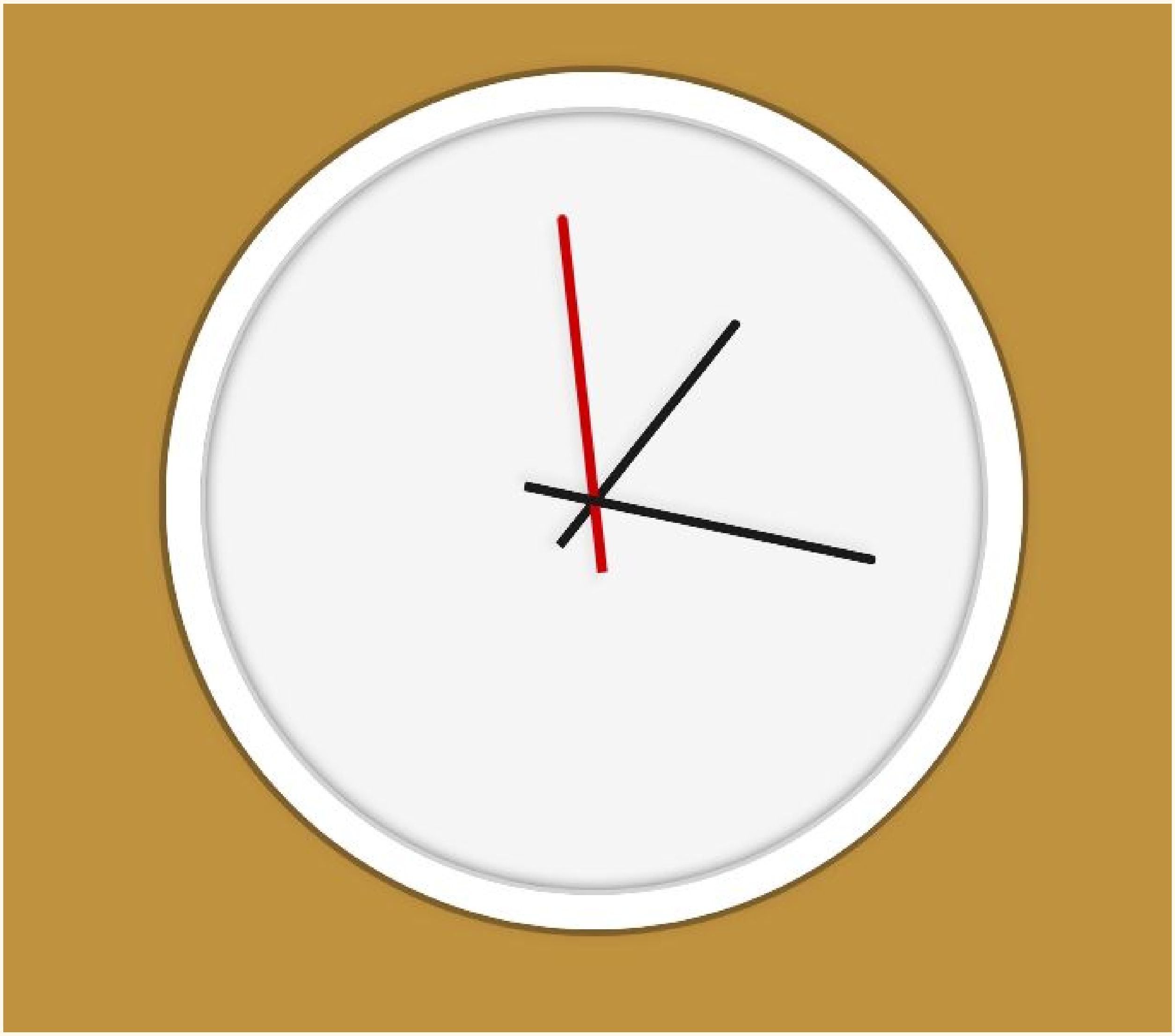
- Search character by name.
- Search house by name.
- Random character
- Random house
- Random character from house taken as input. (Considerable weightage)

4. Losing Time

Among all these assignments and deadlines, it’s been hard to keep track of time. So let’s design a simple clock using pure JavaScript/jQuery to let you keep track of the time.

You can use your creativity to design the clock in whichever way you want and creative and beautiful designs will be appreciated (and will get marks for creativity).

Here is a simple example of the same : (Check the attached GIF file too)



The clock must move and update in real time (See hints on how to accomplish this)

It’s not necessary to have the same design for the clock and creative designs are welcome.

Hints :

- You can draw the face and the clock hands using CSS
- You can get the date and time using JavaScript's "Date" object.
- To rotate the hands, use the **transform** property of CSS (You need to change it dynamically using JavaScript)
- Also consider all the cases of clock looping around.

Plagiarism will leave you with no time and a good zero.