Assignment 1

1. Find and share 2 chrome extensions that let you be more lazy:

- **Buster: Captcha Solver for humans**
  Link:
  https://chrome.google.com/webstore/detail/buster-captcha-solver-for/mpbjkejclgfgadiemmefgebjfooflfhl/related?hl=en
  Desc:
  As the name suggests, it's a captcha solver that solves any reCAPTCHA, "I'm not a robot" instance, automatically, requiring no user intervention.

- **I don't care about cookies**
  Link:
  https://chrome.google.com/webstore/detail/i-dont-care-about-cookies/fihnjjcciajhdojfnbdddfaoknhalnja?hl=en
  Desc:
  It automatically accepts websites with cookie policies explicitly asking users to accept their policy, which sometimes is very annoying.

- **Fready**
  Link:
  https://chrome.google.com/webstore/detail/fready/fbfecjjfhcgpocehenopdofhkdjfpcgl?hl=en
  Desc:
  Removes add from any web article potentially being read, and presents the article as a beautiful document along with a highlighted area, like a cursor which moves at customizable speed to help assist reading. It does help in achieving greater reading speeds.

- **Video Speed Controller**
  Link:
  https://chrome.google.com/webstore/detail/video-speed-controller/nffaoalbilbmmfgbnbgppjihopabppdk?hl=en
  Desc:
  Maps +0.20x(customizable) speed on keyboard keys, has no upper limit on the speed of any video being played on chrome.

2. Document 2 methods of any open source project.
   Gitlink:
   https://github.com/shawon100/Online-Compiler/tree/master/compilers

   DOC:

   #Contains php scripts for executing C, C++, Java, Python in a linux server.

   # Dependencies:

   Python 3 should be installed
   Java should be installed(jdk-8 and above)
   C/C++ should be installed in the linux system

   # File permissions:

   As you can see in the compiler files that a new file called as input, output, error and code file is being created, thus make sure the directory where these files are going to be saved(the directory containing the compiler files) are chowned by the user.

   #Compiling C code:

   The linux shell command for compiling a c program is:

   ```
   cc filename.c <optional parameters>
   ```

   The optional parameters can include -lm, output file, and error logging file.

   A command cc filename.c 2> errorfile.txt is created.

The code and input taken from the user using a post method is taken from the superglobal array $_POST[] and written in filename.c and input.c file respectively.

Alternatively you can create a unique id and append it on the filename, input output and error files for handling multiple requests.

The files are chmodded to change them to executable for the program file, writable for the error file, input file and output file.

The created command for finding out the error is then executed.

The contents of the error file are checked. If it is not empty, the contents of the error file are printed

If the contents of the error file are empty, the contents of input file are checked, if the input file is empty, “timeout 5s ./a.out” command is executed, the timeout can be changed dynamically by replacing 5 with a variable which will contain the time.

The contents of this command are directed to a variable, the variable, contains error if any or the output, which is then displayed.

If the input file contains some input the “< input.txt” is appended to the a.out file executing command.

The errors or output is displayed using the same process.

Also, before starting and after finishing the execution of the a.out file, the system is noted and subtracted to display the time taken for computation.

#Compiling java code:

The linux shell command for compiling a java program is:

```
javac filename.java <optional parameters>
```

The optional parameters can include output file, and error logging file.

A command javac filename.java 2> errorfile.txt is created.

The code and input taken from the user using a post method is taken from the superglobal array $_POST[] and written in filename.c and input.c file respectively.

Alternatively you can create a unique id and append it on the filename, input output and error files for handling multiple requests.

The files are chmodded to change them to executable for the program file, writable for the error file, input file and output file.

The created command for finding out the error is then executed.

The contents of the error file are checked. If it is not empty, the contents of the error file are printed

If the contents of the error file are empty, the contents of input file are checked, if the input file is empty, “timeout 5s java Main” command is executed, the timeout can be changed dynamically by replacing 5 with a variable which will contain the time.

The contents of this command are directed to a variable, the variable, contains error if any or the output, which is then displayed.

If the input file contains some input the “< input.txt” is appended to the Main .class file executing command.

The errors or output is displayed using the same process.

Also, before starting and after finishing the execution of the a.out file, the system is noted and subtracted to display the time taken for computation.

#Compiling Python code

The linux shell command for compiling a java program is:

```
Python3 filename.py <optional parameters>
```

The optional parameters can include output file, input file, timeouts and error logging file.

A command python3 filename.py 2> errorfile.txt is created.

The code and input taken from the user using a post method is taken from the superglobal array $_POST[] and written in filename.c and input.c file respectively.

Alternatively you can create a unique id and append it on the filename, input output and error files for handling multiple requests.

The files are chmodded to change them to executable for the program file, writable for the error file, input file and output file.

The created command for finding out the error is then executed.

The contents of the error file are checked. If it is not empty, the contents of the error file are printed

If the contents of the error file are empty, the contents of input file are checked, if the input file is empty, the created command is executed again.

The contents of this command are directed to a variable, the variable, contains error if any or the output, which is then displayed.

If the input file contains some input the “< input.txt” is appended to the .py file executing command.

The errors or output is displayed using the same process.

Also, before starting and after finishing the execution of the a.out file, the system is noted and subtracted to display the time taken for computation.

#IMPROVEMENTS:

Since, python is an interpreted language, it expects input in the first line itself. Therefore, if a program is asking for input but as we see here, that we are providing none, it will just throw an error even if the code is correct.

The correct way to do it is, using the command:

```
python3 filename.py < input.txt 2>error.txt 1>output.txt
```

Executing this once, the error if any will be redirected to the error.txt file, output to the output.txt file from where it can be displayed.

Another git link:

https://github.com/john-smilga/javascript-basic-projects/blob/master/1-color-flipper/final/app.js

DOCS>

#Changing the colour of a button with id “btn” on click

A list of potential colors of the button is created.

Using the DOM object methods a button is selected and the color attribute of the button is selected.

An event listener is added to the selected button to listen for the event of a mouse click.

A method is triggered when the mouse is clicked on the button which computes a random number with range equal to the index values of the list of colors.

The random number acts as the index of the list which selects the color to be displayed and the “btn”’s color is changed to that color.

Another git link:

https://github.com/john-smilga/javascript-basic-projects/blob/master/1-color-flipper/final/hex.js

#Changing the colour of a button with id “btn” on click

This time we use hex colour, so all 15 numbers are stored in an array.

Hex value can be generated by choosing a number randomly 6 times and appending the value to a variable with # in the beginning.

Using the DOM object methods a button is selected and the color attribute of the button is selected.

An event listener is added to the selected button to listen for the event of a mouse click.

A method is triggered when the mouse is clicked on the button which computes a random number with range equal to the index values of the list of colors, six times to create hex value as mentioned above

The random number acts as the index of the list which selects the color to be displayed and the “btn”’s color is changed to that color.

Git repo for Assignment PF001:

https://github.com/roya2yush/Assignments.git