Edinburgh Napier UNIVERSITY

Coursework Report

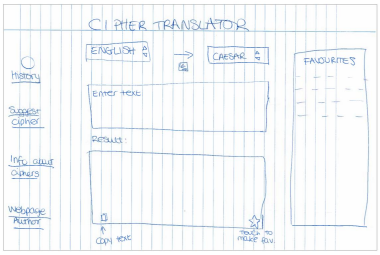
Maria Luque Anguita
40280156@live.napier.ac.uk
Edinburgh Napier University - Web Technologies (SET08101)

1 Introduction

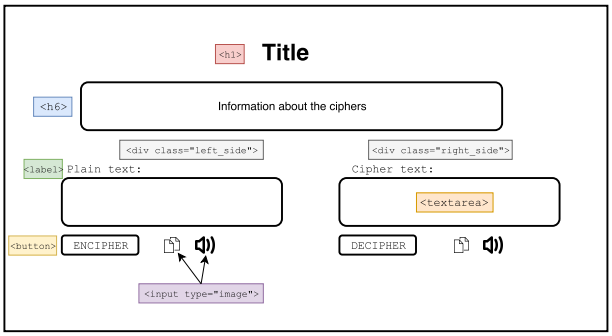
It is known that the design of a web page can be challenging to develop. With the correct implementation of features it can provide a rewarding user experience. The aim of this assignment is to present various ciphers with different types of text encryption in a website.
The first cipher is the Caesar cipher, chosen because it is a very simple cipher, easy to code therefore great to start with. The Morse cipher was chosen because it is a known code, different from most of the other ciphers since it does not use letters or numbers, but instead, a series of on-off tones that represent the different characters. Hence making the website have more variety of ciphers.
The Binary cipher works by representing each character in its binary value. It is very useful to encipher messages since the answer will only contain 1s and 0s, and not many people would think that those numbers are representing text, that is the main reason why it is very cool.
Finally, the Atbash cipher was chosen because it is similar to the Caesar cipher and it involves maths (using the ASCII codes of each characters).
To be able to do the coursework I did a course on StackSkills called "The Full Stack Web Development" which includes video lectures about HTML, CSS and JavaScript that helped me start designing my coursework and know what I was able to do using them. I also read the first chapters of the book recommended by the lecturer Simon Wells "Practical Web Design for Absolute Beginners" which helped me have an understanding on how a common webpage is divided (header, footer and columns in the main body) and from there I started coding the HTML.

2 Software Design

The design of this website was first approached by sketching a basic diagram representing how the website should look. It was influenced by Google to be minimalistic, and since enciphering messages is similar to translating different languages, the first sketch looked similar to the Google translation website:



The final decision was to leave all the ciphers in one page, instead of having links from the main page to every cipher, since that is how most of the websites that have ciphers look (e.g. https://www.braingle.com/brainteasers/codes/).
The inspiration to leave all the ciphers in the same page came from the website http://rogerdudler.github.io/git-guide/ and the format of my website is similar to that one, with plain different background colours for each different part, each part being a different cipher. I did not start coding until I was sure of how the webpage was going to look, since HTML has to follow the display. The website is divided in 6 parts. The first part is the main title and navigation part, the next four are the ciphers and the last one is the contact page. All of the ciphers follow the same design format:

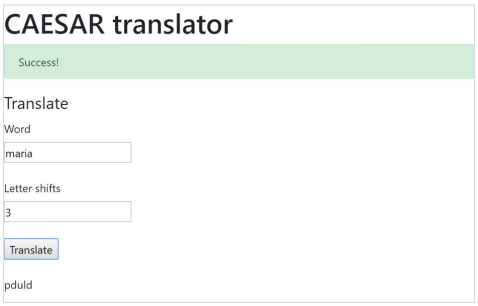


The screen is divided in three parts, the header, which includes the title and the information about the cipher; the left side, where the plain text is and the right side, where the enciphered text is. They both have the option to listen to the message and copy the text.
The ciphers code in JavaScript will all be in separate files. One file for ciphering and one for deciphering. And every other method will have their own JavaScript file.

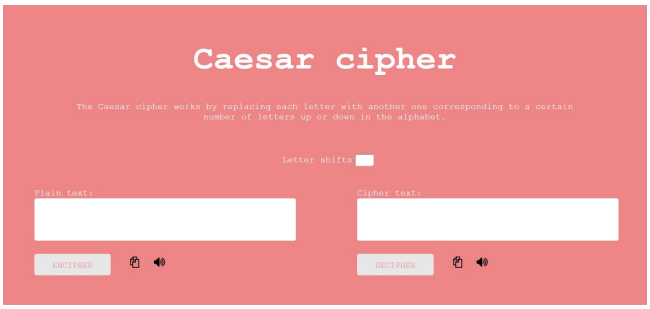
3 Implementation

3.1 Caesar Cipher

The Caesar Cipher was the first thing to be finished, even before starting to code the main HTML document, to see if I was able to make it work. After I finished it, I started setting up the main HTML

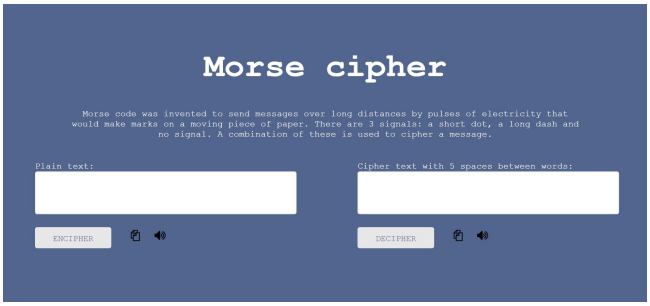


The first try at it was very simple, just finding the ASCII code of each letter and adding the amount shifted to it, and then finding the new letter that appeared from that code. The problem was that for the last letters of the alphabet, once you surpass the numbers of the letters, the program prints the code of other ASCII characters, so I added 'if statements' to make sure it would only work with letters and it would ignore the other characters, as well as going back to A once it reached Z. The decipher is the same but changing the '+' to a '-' in the amount. It is very easy to use once you have the ciphering code. The final design was added to the HTML document and with some CSS this is how it looks:



3.2 Morse Code Cipher

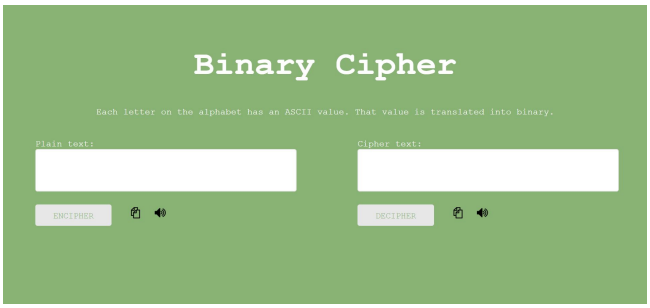
The Morse cipher is easy to implement if a dictionary is used to translate each letter to Morse. A 'for loop' will go through each letter of the original text translating it. As an additional feature, I added the sound that the Morse code would produce, so in this cipher, if the user clicks the sound button it wouldn't hear a woman's voice, but the sound of dots and dashes. However, for this cipher I implemented it directly in the main HTML rather than doing a separate document since the display was the same as for the Caesar cipher.



Deciphering is a bit more difficult because there is a backwards slash between each letter and 5 spaces between words, so the code has to split the words and then the letters and look for the translation of each letter in an array of key:value pairs.

3.3 Binary Cipher

Translating characters to binary in HTML can be done using the function '.toString(2)' that will turn a number into its binary value. Therefore finding the ASCII code of each character and turning it into binary will output a ciphered text of the plain text. The display was the same as the others but with a different background colour:



To decipher the same could be done but using 'parseInt(letter, 2);' which parses a binary number to an integer.

3.4 Atbash Cipher

The Atbash cipher works by taking the alphabet and mapping it to its reverse, so that 'A' is now 'Z' and so on. The way to code it is very similar to the Caesar cipher, it uses the ASCII codes of each letter and by using some simple mathematics it finds the corresponding letter, then just convert that code to the actual letter.

As every letter has a corresponding value, the same exact code can be used to decipher the message, with no alternations needed.

4 Critical evaluation

4.1 Comparison against requirements

The requirements for this assignment are to design and implement a website with a couple classical ciphers using HTML, JavaScript and CSS. Another requirement was to evaluate the design using appropriate techniques. As Matera (2013) studied, a way to evaluate a website is by looking at its usability. If any software fails to meet the user's needs and satisfaction, then the user will lose interest and the website is considered less usable. The first thing to look at is learnability, how easy it is to navigate through the webpage and how easy it is to get the job done. The start of this site is a simple image with a title and a navigation bar with the names of the ciphers so it is quite easy to navigate through it. The next point to take into consideration is how efficient it is, or

in this case, how long it takes to encipher and decipher the messages. As it is a very simple website, the translation is almost instant and easy to read and even hear the ciphered message by clicking the sound button. Memorability is also important to evaluate a website, how easy it is to remember how to use the website after a period of non-use without needing to learn again how to use it, but as the point before said, it is a very simple website so it does not have much to remember. The purpose of this website is met, since it allows the user to encipher text in different ways.

4.2 Possible improvements

There is always room for improvement, even if the website meets its requirements, updates can always make it better. In this case, I feel some features could be added to different parts of the website.
Regarding CSS (the design of the website), the homepage could be different, new, peculiar, give it a different touch, instead of making the typical website homepage. Also I would like to add a blending visualization effect between the different colours of the cipher backgrounds, to make it look like they are all together, sliding into each other. As well as adding more movement to the web page, make it more dynamic and entertaining.
It would be a good idea to add a mini game related to deciphering code or something that lets the user create their own cipher and send it to friends to decipher the messages. Additionally, there could be links promoting sites that have good cyber-security courses for the users that are more interested in the topic.

5 Personal evaluation

Before this module, I was aware of what HTML was and I had a very basic understanding o how it worked, but thanks to it, I learned what CSS and JavaScript could do, how to implement the three of them together to make a website look

6 References

- Matera, M. (2013). Web Usability: Principles and Evaluation Methods. Retrieved from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.462.3115&rep=rep1&type=pdf