

## Chapter 7 - Encryption\caesar\_cipher.html

```
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5      <title>Caesar Cipher Project</title>
6      <style>
7          body {
8              padding: 15px 0;
9              text-align: center;
10             background: rgb(91, 222, 255);
11             font-family: 'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida
Sans', Arial, sans-serif;
12             color: rgb(25, 0, 37)
13         }
14
15         .div1 {
16             text-align: center;
17             font-family: Cambria, Cochin, Georgia, Times, 'Times New Roman', serif;
18             font-size: 150%;
19             background-color: #0077ff;
20             width: 75%;
21             border: 5px solid rgb(8, 0, 116);
22             padding: 25px;
23             margin: auto
24         }
25
26         .customEncrypt {
27             /* background/text */
28             background-color: rgb(196, 75, 75);
29             font: Times;
30             font-size: 60px;
31             color: rgb(23, 239, 192);
32
33             /* border/spacing */
34             border-style: dashed;
35             border-radius: 10px;
36             border-width: 5px;
37             border-color: rgb(65, 109, 253);
38             padding: 15px;
39
40             /* centering */
41             margin: auto;
42             width: fit-content;
43         }
44
45         .customDecrypt {
46             /* background/text */
47             background-color: rgb(0, 255, 0);
48             font: helvetica;
49             font-size: 30px;
50             color: rgb(91, 92, 80);
51
52             /* border/spacing */
```

```
53     border-style: groove;
54     border-radius: 20px;
55     border-width: 5px;
56     border-color: rgb(232, 239, 23);
57     padding: 5px;
58
59     /* centering */
60     margin: auto;
61     width: fit-content;
62 }
63
64 .customPinkRandomKey {
65     /* background/text */
66     background-color: rgb(148, 126, 0);
67     font: helvetica;
68     font-size: 30px;
69     color: rgb(35, 0, 133);
70
71     /* border/spacing */
72     border-style: groove;
73     border-radius: 20px;
74     border-width: 5px;
75     border-color: rgb(65, 0, 54);
76     padding: 5px;
77
78     /* centering */
79     margin: auto;
80     width: fit-content;
81 }
82
83 .customInstructions {
84     /* background/text */
85     background-color: royalblue;
86     font: helvetica;
87     font-size: 40px;
88     color: rgb(255, 255, 255);
89
90     /* border/spacing */
91     border-style: groove;
92     border-radius: 20px;
93     border-width: 15px;
94     border-color: rgb(239, 59, 23);
95     padding: 15px;
96
97     /* centering */
98     margin: auto;
99     width: fit-content;
100 }
101
102 .customForce {
103     /* background/text */
104     background-color: rgb(0, 85, 33);
105     font: helvetica;
106     font-size: 40px;
107     color: rgb(165, 201, 4);
108 }
```

```
109      /* border/spacing */
110      border-style: outset;
111      border-radius: 20px;
112      border-width: 15px;
113      border-color: rgb(196, 0, 121);
114      padding: 15px;
115
116      /* centering */
117      margin: auto;
118      width: fit-content;
119  }
120 </style>
121 <script>
122     // Description: Takes a sentence or word and encrypts it with a key
123     // Citations: None
124     // Input/Parameters: Text input
125     // Output/Return: Encrypted message
126     function encrypt() {
127         let key = idSelectConversion.value;
128         let text = textAreaId.value;
129         key = Number(key);
130         text = text.toUpperCase();
131         let textArray = text.split("");
132         let output = ""
133
134         for (let i = 0; i < textArray.length; i++) {
135             let character = textArray[i]
136             character = character.charCodeAt(0)
137             if (character < 65 || character > 90) {
138                 character = character - key
139             }
140             else if (character + key > 90 || character + key < 65) {
141                 character = character - 26
142             }
143             character = character + key
144             character = String.fromCharCode(character)
145             output += character
146         }
147         outputDiv.innerHTML = output
148         outputDiv.style.background = "red";
149     }
150     // Description: Takes the encrypted message and key and decrypts the message
151     // Citations: None
152     // Input/Parameters: Encrypted message
153     // Output/Return: Decrypted message
154     function decrypt() {
155         let key = idSelectConversion.value
156         let text = textAreaId.value;
157         key = Number(key);
158         text = text.toLowerCase();
159         let textArray = text.split("");
160         let output = ""
161
162         for (let i = 0; i < textArray.length; i++) {
163             let character = textArray[i]
164             character = character.charCodeAt(0)
```

```

165         if (character < 97 || character > 122) {
166             character = character + key
167         }
168         else if (character - key > 122 || character - key < 97) {
169             character = character + 26
170         }
171         character = character - key
172         character = String.fromCharCode(character)
173         output += character
174     }
175
176     outputDiv.innerHTML = output
177     outputDiv.style.background = "green";
178
179 }
180 // Description: Return a random integer, n, such that min <= n <= max
181 // Citation: Mozilla Foundation
182 // https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/random
183 // Input/Parameters: Two floating point numbers, min and max
184 // Output/Return: Returns a random integer between min and max
185 function randomInt(min, max) {
186     min = Math.ceil(min);
187     max = Math.floor(max);
188
189     return Math.floor(Math.random() * (max - min + 1)) + min;
190 }
191 function pickRanKey() {
192     ranKey = randomInt(1, 25)
193     idSelectConversion.value = ranKey
194 }
195 // Description: Shows instructions when mouse hovers over them
196 // Citations: None
197 // Input/Parameters: None
198 // Output/Return: Instructions
199 function instructions() {
200     divInstruct.innerHTML = "Enter a plaintext, select an encryption key, and then
press 'Encrypt' to view the ciphertext." + "<br>" +
201     "Or you may enter a ciphertext, select a decryption key, and then press '
Decrypt' to view the plaintext"
202 }
203 // Description: No instructions display when mouse isn't hovering over the button
204 // Citations: None
205 // Input/Parameters: None
206 // Output/Return: Nothing
207 function noInstructions() {
208     divInstruct.innerHTML = ""
209 }
210 // Description: Decrypts the message using every single key and displays
211 // Citations: None
212 // Input/Parameters: Encrypted message
213 // Output/Return: 25 different messages using 25 different keys but only one is
correct
214 function bruteForce() {
215     let text = textAreaId.value;
216     text = text.toLowerCase();
217     let textArray = text.split("")

```

```
218     let output = ""
219     let number = 1
220     for (let i = 0; i < 25; i++) {
221         for (let i = 0; i < textArray.length; i++) {
222             let character = textArray[i];
223             parseInt(character);
224             character = character.charCodeAt(0)
225             if (character >= 65 && character <= 90) {
226                 character = character - number;
227
228                 if (character < 97) {
229                     character = character + 26
230                 }
231             }
232             else if (character >= 97 && character <= 122) {
233                 character = character - number;
234
235                 if (character < 97) {
236                     character = character + 26
237                 }
238             }
239             else{
240                 character = character
241             }
242             character = String.fromCharCode(character)
243             output += character;
244         }
245         output += "<br>"
246         number = number + 1
247     }
248     outputDiv.innerHTML = output
249     outputDiv.style.background = "royalblue";
250
251 }
252
253 </script>
254 </head>
255
256 <body>
257     <h1>Caesar Cipher Project</h1>
258     <textarea id="textAreaId" rows="10" cols="50"></textarea>
259     <br>
260     <br>
261     <br>
262     Key: <select id="idSelectConversion">
263         <option>1</option>
264         <option>2</option>
265         <option>3</option>
266         <option>4</option>
267         <option>5</option>
268         <option>6</option>
269         <option>7</option>
270         <option>8</option>
271         <option>9</option>
272         <option>10</option>
273         <option>11</option>
```

```
274     <option>12</option>
275     <option>13</option>
276     <option>14</option>
277     <option>15</option>
278     <option>16</option>
279     <option>17</option>
280     <option>18</option>
281     <option>19</option>
282     <option>20</option>
283     <option>21</option>
284     <option>22</option>
285     <option>23</option>
286     <option>24</option>
287     <option>25</option>
288 </select>
289 <br>
290 <br>
291 <br>
292 <span class="customEncrypt" onclick="encrypt()">Encrypt</span>
293 <span class="customDecrypt" onclick="decrypt()">Decrypt</span>
294 <br><br><br><br><br>
295 <span class="customPinkRandomKey" onclick="pickRanKey()">Pick Random Key</span>
296 <span class="customInstructions" onmouseover="instructions()" onmouseout="
noInstructions()">Instructions</span>
297 <br><br><br><br><br>
298 <span class="customForce" onclick="bruteForce()">Brute Force Attack</span>
299 <br><br><br><br><br>
300 <div class="div1" id="outputDiv"></div>
301 <div id="divInstruct"></div>
302 </body>
303
304 </html>
```