

Caesar Cypher

-Javascript

```
1 // El diccionario que se usa para comparar las frecuencias obtenidas. Se obtuvo contando los caracteres del Quijote.
2 var database = {
3   "porcentajes": {
4     "a": 0.1222088397965167,
5     "b": 0.014718162457500978,
6     "c": 0.03622824458468901,
7     "d": 0.053174824731535404,
8     "e": 0.1396972862797493,
9     "f": 0.004620797183514097,
10    "g": 0.010499041219632018,
11    "h": 0.012141706885054851,
12    "i": 0.05490404272515491,
13    "j": 0.0064182818021901395,
14    "k": 0.0,
15    "l": 0.054334747834058464,
16    "m": 0.027220097694416644,
17    "n": 0.06868292956469055,
18    "o": 0.09905609200390583,
19    "p": 0.02161674872883887,
20    "q": 0.01979914983670867,
21    "r": 0.061534438683770704,
22    "s": 0.07663416281346266,
23    "t": 0.03763746277335602,
24    "u": 0.04849368472765883,
25    "v": 0.010883650509013023,
26    "w": 1.2190468760095233e-06,
27    "x": 0.00022979033612779512,
28    "y": 0.015308181145489588,
29    "z": 0.003956416636088907
30  }
31 };
32
33 // Esta función chequea si el carácter que se va a analizar es una letra minúscula.
34 function isLetter(str) {
35   return str.length == 1 && str.match(/[a-z]/i);
36 }
37
38
39 // Esta función devuelve el mínimo de un conjunto de datos, en este caso se usa para encontrar el valor de la chi cuadrada.
40 function min(data) {
41   var min = 9999999;
42   for(numero in data){
43     min = data[numero] < min ? data[numero] : min;
44   }
45   return min;
46 }
47
48
49 // Esta función devuelve el máximo de un conjunto de datos, en este caso se usa para encontrar el valor de la chi cuadrada.
50 function max(data) {
51   var max = 0;
52   for(numero in data){
53     max = data[numero] > max ? data[numero] : max;
54   }
55   return max;
56 }
57
58
```

```

58
59 // Esta funcion regresa el número de caracter en Unicode.
60 function num(char){
61     return char.charCodeAt(0)-97;
62 }
63
64
65 // Regresa el caracter que se encuentra en la posición "number" del abecedario traducido a Unicode.
66 function character(number){
67     return String.fromCharCode(97 + (26 + number) % 26);
68 }
69
70
71 // Codifica el mensaje usando la llave recibida y regresa el mensaje
72 function encode(key, message){
73     cypheredMessage = "";
74     for(char of message){
75         // Junta todos los caracteres en un único string.
76         // Es una condición que chequea si el caracter está entre la "a" y la "z".
77         // Si es cierto regresa el caracter modificado con la llave.
78         // Si la condición no se cumple solo añade el caracter (Esto para que los espacios se queden en su lugar y no cambiar puntos ni com
79         // condicion verdadera          condicion falsa
80         cypheredMessage += isLetter(char) ? character(num(char) + key) : char;
81     }
82     // Modifica la caja de texto de la página web
83     document.getElementById('txtarea').value = cypheredMessage;
84 }
85
86
87 // Decodifica el mensaje con la llave que se le de
88 function decode(key, message){
89     decypheredMessage = "";
90     for(char of message){
91         // Al igual que la codificacion, junta una cadena con las letras codificadas y los espacios en su lugar.
92         decypheredMessage += isLetter(char) ? character(num(char) - key) : char;
93     }
94     // Modifica la caja de texto de la página web
95     document.getElementById('txtarea').value = decypheredMessage;
96 }
97
98
99 // Funcion que intenta todas las llaves y regresa un mensaje con cada llave probada
100 // Esta funcion se usa cuando el mensaje es muy chico, porque el metodo de frecuencia no es eficaz.
101 function bruteForce(message){
102     messages = "";
103     // For que usa todas las diferentes llaves
104     for(i = 1; i < 27; i++){
105         msg = "";
106         // For que crea los mensajes con cada llave y los añade a una sola cadena
107         for(char of message){
108             msg += isLetter(char) ? character(num(char) + i) : char;
109         }
110         message = msg;
111         messages += msg + ", key = " + (26-i) + ' | ';
112     }
113     // Cambia el area de texto a la cadena con los mensajes con las diferentes llaves
114     document.getElementById('txtarea').value = messages;
115 }
116
117

```

```

118 // Funcion que descifra el mensaje con el uso de chi cuadrada
119 function keyWithChi(message){
120     // Primero se crea un diccionario con todas las letras.
121     var frequency = {};
122     for(var x = 0; x < 26; x++){
123         frequency[character(x)] = 0
124     }
125     // Este for recorre todos los caracteres del mensaje y los cuenta.
126     // El conteo se modifica en el diccionario "frequency" y también cuenta el número total de letras
127     var counter = 0;
128     for(char in message){
129         if(isLetter(message[char])){
130             counter += 1;
131             frequency[message[char]] = frequency[message[char]] + 1;
132         }
133     };
134     // Compara las frecuencias esperadas (Quijote) con las obtenidas (usando lo anterior)
135     expectedFrec = database["porcentajes"];
136     potentialKeys = [];
137     // este for recorre todas las llaves posibles
138     for(var i = 0; i < 26; i++){
139         var chisqrd = 0;
140         // este for obtiene con cada llave todos los valores posibles de chi cuadrada
141         for(var j = 0; j < 26; j++){
142             var e = expectedFrec[character(j)]*counter;
143             var o = frequency[character(i+j)];
144             chisqrd += e != 0 ? ((e-o)**2)/e : 0;
145         }
146         // añade los valores de chi cuadrada donde la posición de la chi cuadrada es la llave utilizada
147         potentialKeys.push(chisqrd);
148     }
149     var maximum = max(potentialKeys);
150     for(i = 0; i < 26; i++){
151         document.getElementById(character(i)).setAttribute('style', "height : "+ 5*potentialKeys[i]/maximum + "vw; width: 3%; margin: auto
152     }
153     // cambia la seed en la página al índice del valor mínimo de chi cuadrada.
154     document.getElementById('seed').value = potentialKeys.indexOf(min(potentialKeys));
155 }
156
157
158 //Funcion que encuentra la llave
159 function keyFinder(message){
160     // Este if es en caso de que el mensaje sea muy corto se utiliza el método de fuerza bruta.
161     // De otra manera se puede hacer con el método de chi cuadrada.
162     return message.length > 6 ? keyWithChi(message) : bruteForce(message);
163 }

```