a. Créer une variable "I used to get mad at my school (No I can't complain)" et écrire un programme qui remplace :

* les a par des 4
* les e par des 3
* les o par des 0 (zéro)

b. Créer un tableau et, pour chaque mot de ce texte, stocker une entrée correspondant au nombre de lettres de ce mot. Si un chiffre est présent dans un mot, il ne compte pas comme une lettre mais pour sa propre valeur. Ex. "my" vaut 2, "us3d" vaut 6 (3 lettres + 3), "sch00l" vaut 4 (4 lettres + 0 + 0)

c. Afficher cette liste de mots dans une page html sur 2 lignes. La première ligne avec les mots, la seconde, en dessous, avec la valeur associée à chacun d'entre eux

d. Au clic sur une valeur, tous les mots de cette même valeur obtiennent un background jaune

[](https://cloud.githubusercontent.com/assets/606754/12701684/5a8dd32e-c811-11e5-8b17-8bf48acf826c.png)

Légende : exemple du résultat obtenu après un clic sur un 6

|  |
| --- |
| <!DOCTYPE html> |
|  | <html> |
|  | <head> |
|  | <meta charset="utf-8"> |
|  | <title></title> |
|  | <style> |
|  | table { text-align: center; margin: auto; font-family: verdana; } |
|  | td { padding: 10px; cursor: pointer; } |
|  | td.selected { background: yellow; } |
|  | </style> |
|  | </head> |
|  |  |
|  | <body></body> |
|  |  |
|  | <script> |
|  | var text = "I used to get mad at my school (No I can't complain)"; |
|  |  |
|  | var append = ''; |
|  | for (var i = 0; i < text.length; i++) { |
|  | var char = text[i]; |
|  | if (char === 'a') |
|  | char = '4'; |
|  | else if (char === 'e') |
|  | char = '3'; |
|  | else if (char === 'o') |
|  | char = '0'; |
|  | append += char; |
|  | } |
|  |  |
|  | var words = append.split(' '); |
|  | var values = []; |
|  | for (var i = 0; i < words.length; i++) { |
|  | var word = words[i]; |
|  | var value = word.length; |
|  | for (var j = 0; j < word.length; j++) { |
|  | var char = word[j]; |
|  | if (char === '4') |
|  | value += 3; |
|  | else if (char === '3') |
|  | value += 2; |
|  | else if (char === '0' || char === '(' || char === ')' || char === "'") |
|  | value += -1; |
|  | } |
|  | values.push(value); |
|  | } |
|  |  |
|  | var table = document.createElement('table'); |
|  | var tr = document.createElement('tr'); |
|  |  |
|  | for (var i = 0; i < words.length; i++) { |
|  | var td = document.createElement('td'); |
|  | td.innerHTML = words[i]; |
|  | td.classList.add('v' + values[i]); |
|  | tr.appendChild(td); |
|  | } |
|  |  |
|  | table.appendChild(tr); |
|  | tr = document.createElement('tr'); |
|  |  |
|  | for (var i = 0; i < values.length; i++) { |
|  | var td = document.createElement('td'); |
|  | td.innerHTML = values[i]; |
|  | tr.appendChild(td); |
|  |  |
|  | td.addEventListener('click', function() { |
|  | var all = document.querySelectorAll('td.v' + this.innerHTML); |
|  | for (var j = 0; j < all.length; j++) |
|  | all[j].classList.add('selected'); |
|  | }); |
|  | } |
|  |  |
|  | table.appendChild(tr); |
|  | document.body.appendChild(table); |
|  | </script> |
|  | </html> |

Dans le jeu du mastermind, un premier joueur choisit un code de 4 couleurs et un second joueur fait des essais pour trouver celui-ci. À chaque essai, le premier joueur lui indique quelles couleurs sont bien placées, mal placées ou absentes.

a. Créer une variable pour stocker un code de 4 couleurs (utiliser des lettres pour représenter les couleurs, ex. 'B', 'Y', 'G', 'R' pour blue, yellow, green et red)

b. Créer une autre variable pour stocker un essai de 4 couleurs, différent du code

c. Créer une fonction feedback comparant cet essai au code et retournant, pour chaque caractère :

* 2, si il est identique à celui à la même position dans le code
* 1, si il est différent de celui à la même position dans le code, mais présent ailleurs dans le code
* 0, si il est absent du code

Ex. comparer l'essai ['R', 'G', 'Y', 'R'] et le code ['B', 'B', 'Y', 'G'] doit retourner [0, 1, 2, 0] — qui signifie : aucun 'R' dans le code, 'G' mal placé et 'Y' bien placé.

d. Si des couleurs sont dupliquées dans l'essai, et que le code n'en contient qu'un exemplaire, le feedback retourné ne doit l'indiquer qu'une seule fois. Ex. comparer l'essai ['R', 'G', 'Y', 'R'] et le code ['G', 'G', 'Y', 'R'] doit retourner [0, 2, 2, 2]

|  |
| --- |
| <!DOCTYPE html> |
|  | <html> |
|  | <head> |
|  | <meta charset="utf-8"> |
|  | <title></title> |
|  | </head> |
|  |  |
|  | <body></body> |
|  |  |
|  | <script> |
|  | var feedback = function(guess, code) { |
|  | var feedback = [0, 0, 0, 0]; |
|  | var letters = code.slice(0); |
|  |  |
|  | for (var i = 0; i < guess.length; i++) |
|  | if (guess[i] === code[i]) { |
|  | feedback[i] = 2; |
|  | var index = letters.indexOf(guess[i]); |
|  | letters.splice(index, 1); |
|  | } |
|  |  |
|  | for (var i = 0; i < guess.length; i++) |
|  | if (feedback[i] === 0 && letters.indexOf(guess[i]) !== -1){ |
|  | feedback[i] = 1; |
|  | var index = letters.indexOf(guess[i]); |
|  | letters.splice(index, 1); |
|  | } |
|  |  |
|  | return feedback; |
|  | } |
|  |  |
|  | console.log(feedback(['R', 'G', 'Y', 'R'], ['B', 'B', 'Y', 'G'])); // [0, 1, 2, 0] |
|  | console.log(feedback(['R', 'G', 'Y', 'R'], ['G', 'G', 'Y', 'R'])); // [0, 2, 2, 2] |
|  | </script> |
|  | </html>  a. Créer un tableau contenant 8 paires de mots (les mots 'rain', 'sun', 'storm', 'desert', 'mountain', 'ocean', 'forest', 'cosmos' répétés 2 fois)  b. Afficher ces 16 mots dans une page html en 4 lignes et 4 colonnes  c. Cliquer sur un mot lui donne un background jaune  d. Lorsque deux mots sont sélectionnés (background jaune), ils sont comparés et sont supprimés s'il sont identiques  e. Cliquer sur un troisième mot retire le background des deux mots précédents (s'ils n'ont pas été supprimés) et donne un background jaune à ce nouveau mot ; l'opération est possible jusqu'à ce que tous les mots aient été supprimés  [doubles](https://cloud.githubusercontent.com/assets/606754/12701707/d0cb8a68-c811-11e5-9f6b-a4a36927efd8.png)  Légende : exemple d'un début de partie   |  | | --- | | <!DOCTYPE html> | |  | <html> | |  | <head> | |  | <meta charset="utf-8"> | |  | <title></title> | |  | <style> | |  | ul { list-style-type: none; font-family: verdana; width: 400px; text-align: center; margin: auto;} | |  | li {display: inline-block; width: 100px; padding: 20px 0; cursor: pointer; } | |  | li.selected {background-color: yellow; } | |  | </style> | |  | </head> | |  |  | |  | <body></body> | |  |  | |  | <script> | |  | var eightPairs = function() { | |  | var values = ['rain', 'sun', 'storm', 'desert', 'mountain', 'ocean', 'forest', 'cosmos']; | |  | return values.concat(values); | |  | }; | |  |  | |  | var append = function(pairs) { | |  | var ul = document.createElement('ul'); | |  | for (var i = 0; i < pairs.length; i++) { | |  | var li = document.createElement('li'); | |  | li.innerHTML = pairs[i]; | |  | ul.appendChild(li); | |  | } | |  | document.body.appendChild(ul); | |  | }; | |  |  | |  | var last; | |  | var play = function() { | |  | if (last && last.innerHTML === this.innerHTML) { | |  | last.remove(); | |  | this.remove(); | |  | last = null; | |  | } else if (last) { | |  | this.classList.add('selected'); | |  | last = null; | |  | } else { | |  | for (var i = 0; i < lis.length; i++) | |  | lis[i].classList.remove('selected'); | |  |  | |  | this.classList.add('selected'); | |  | last = this; | |  | } | |  | }; | |  |  | |  | var pairs = eightPairs(); | |  | append(pairs); | |  |  | |  | var lis = document.querySelectorAll('li'); | |  | for (var i = 0; i < lis.length; i++) | |  | lis[i].addEventListener('click', play); | |  | </script> | |  | </html> | |