

WORDLE

Le numéro de chaque consigne est désigné en fin de paragraphe.

```
export const transformWord = (word) => {  
  return String(word).replace(regex.space, ' ').toUpperCase();  
}
```

Voici un exemple de fonction qui manipule un Objet de type String. Son rôle est de mettre en majuscule toutes les lettres de la chaîne de caractères. Le but est de travailler uniquement avec des Majuscules dans l'application pour simplifier la vérification des mots. (1)

```
const createScoreMessage = () => {  
  var message = `Score : <span>${score.current}</span>`;  
  if (score.current === score.best)  
    message += "<br>Vous avez battu votre meilleur score !";  
  return message;  
};
```

Cette fonction permet d'afficher le message de "meilleur score" si le score de la partie actuelle dépasse le meilleur score stocké dans l'attribut *best* de l'objet *score*. (2 / 3)

```
word = txt.toLowerCase();  
wordLength = word.length;  
input = "";  
rows = [];  
nRow = 0;  
nLetter = 0;  
// Suppression des éléments HTML existants.  
while (board.firstChild) board.removeChild(board.firstChild);  
// Création des éléments HTML.  
let j;  
for (let i = 0; i < maxRows; i++)  
{  
  let row = document.createElement("div");  
  row.classList.add("row");  
  rows.push(row);  
  board.appendChild(row);  
  for (j = 0; j < wordLength; j++)  
  {  
    let c = document.createElement("div");  
    if (word[j].match(regex.space)) c.classList.add(spaceClass);  
    else c.classList.add(cellClass);  
    row.appendChild(c);  
    animateCell(rows[i], j);  
  }  
}
```

La première boucle (while) permet de supprimer toutes les cases de la grille Wordle lors de l'initialisation. La seconde boucle recrée toutes les cases de la grille vierge pour la nouvelle

partie. Chaque case (désignée par une div dans le HTML) est stockée dans le tableau rows. Ce tableau nous sert à accéder aux éléments HTML des cases du Wordle facilement. (4 / 5 / 8.1 / 8.2)

```
api.dicos.forEach(dico => {
  const option = document.createElement("option");
  option.value = dico;
  option.innerText = dico;
  if (dico === api.currentDico) option.selected = true;
  dicoField.appendChild(option);
});
// Sélection du formulaire.
setSelectionEvents(dicoField);
setSelectionEvents(maxLengthField);
setSelectionEvents(startButton);
// Actions du formulaire.
dicoField.onChange = () => dico = dicoField.childNodes[dicoField.selectedIndex + 1].value;
maxLengthField.value = api.currentMaxLength;
maxLengthField.onChange = () => maxLength = maxLengthField.value;
startButton.onclick = () => {
  api.setDico(dico);
  api.setMaxLength(maxLength);
  wordle.restart();
};
```

Cette partie du code manipule le formulaire pour les options du jeu. Par exemple, le code déclenché par appui sur le bouton start sauvegarde les options de jeu dans le fichier api.js et relance l'application avec celles-ci. (6)

```
const mandatoryValues = {
  "expires": 0,
  "path": "/",
  "SameSite": "None",
  "Secure": undefined
};
```

Cet objet déclaré dans cookie.js contient les attributs obligatoires que doit posséder un cookie HTTP. Il est utilisé lors de l'écriture du cookie contenant, entre autres, le meilleur score du joueur et sa configuration de jeu. L'attribut expires, qui définit la durée de validité du cookie, est alors modifié. (7)

```
if (nLetter < wordLength)
{
  input += letter.toLowerCase();
  rows[nRow].childNodes[nLetter].innerHTML = input[nLetter];
}
```

Ces lignes permettent de rajouter une lettre au clavier en modifiant la valeur de la case correspondante sur la grille HTML du Wordle. (8.3)

Site web : pweb-wordle.vercel.app