

## T. D. 3 de javascript, xhtml et css, Info 1B

### Exercice 1. : Recherche de palindrome

L'utilisateur doit rentrer une chaîne de caractères, le programme affiche si la chaîne est un palindrome, un palindrome non parfait (palindrome si on ne s'occupe pas des espaces) ou non.

La page est découpée en cinq zones :

1. un titre le plus grand possible ;
2. une zone permettant d'afficher les résultats, les marges intérieures et la marge extérieure inférieure sont de 0,5 cm ;
3. une zone de bouton permettant d'appeler chacune une fonction Javascript,
  - (a) La fonction `PalindromeSansEspace()` permet d'afficher si le texte rentré est un palindrome en ne tenant pas compte des espaces ;
  - (b) La fonction `PalindromeAvecEspace()` permet d'afficher si le texte rentré est un palindrome en tenant compte des espaces ;
  - (c) La fonction `raz()` permet de remettre à zéro la page, cette fonction est aussi appelée au chargement de la page ;
  - (d) La fonction `exemple()` permet d'afficher les deux exemples « Laval » et « Esope reste ici et se repose » dans la page html.
4. un paragraphe permettant de donner un exemple de palindrome parfait (le palindrome est sur fond cyan, souligné et de couleur rouge) ;
5. un paragraphe permettant de donner un exemple de palindrome non parfait (le palindrome est sur fond cyan, souligné et de couleur rouge).

La figure 1 montre le résultat final.

La figure 2 montre le résultat final.

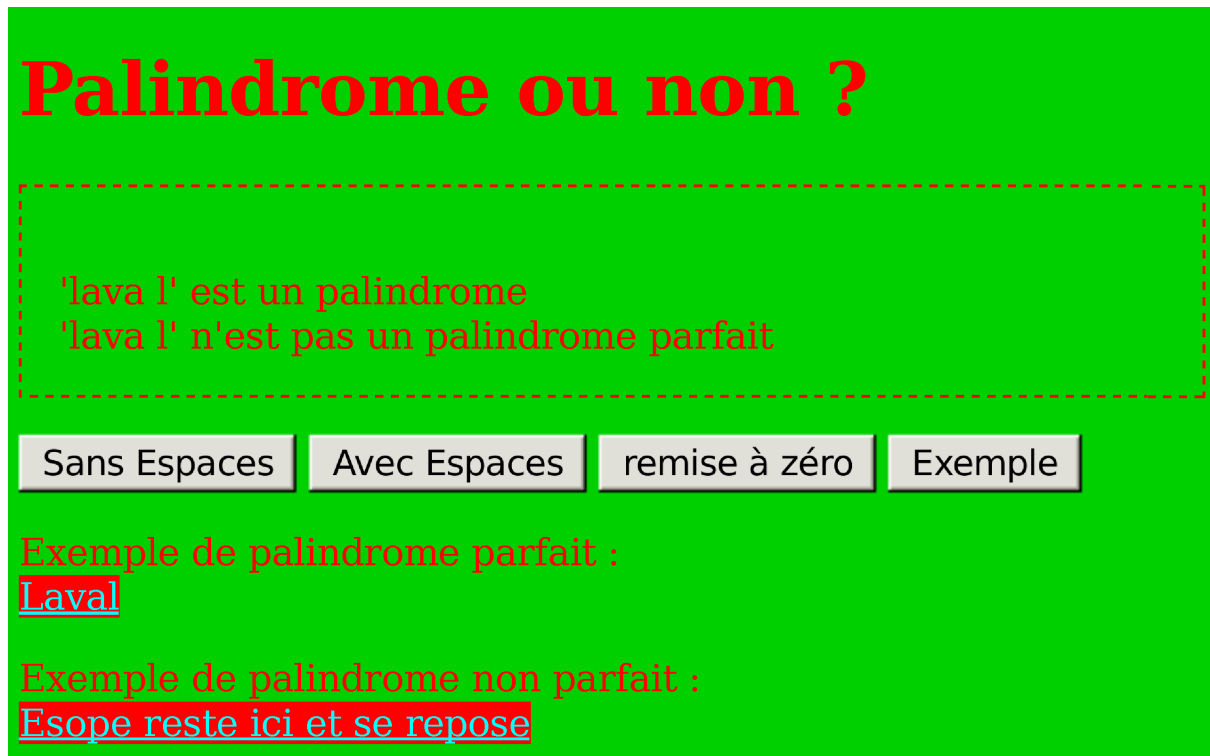


FIGURE 1 – Résultat final de l'exercice sur les palindromes.



FIGURE 2 – Remise à zéro de la page de l'exercice sur les palindromes.

### Exercice 2. : Menus déroulants

La page contient deux menus déroulants. Le premier permet de choisir l'une des trois régions : Bourgogne, Franche-Comté ou Basse-Normandie. En fonction du choix de l'utilisateur, le second menu déroulant contient les départements de la région choisie.

Au chargement de la page, le menu déroulant des régions est sur « Choisissez une région » tandis que celui des départements est sur « Choisissez un département ».

Le bouton « choix » permet d'afficher le choix fait par l'utilisateur dans la page html. Il faut vérifier qu'un choix par menu déroulant a bien été effectué.

Lorsque l'utilisateur remet le menu déroulant des régions sur « Choisissez une région », le menu déroulant des départements est remis à zéro (il n'y a plus que « Choisissez un département »).

Lors du changement de choix dans la liste déroulante, il est possible d'appeler la fonction javascript `choix()` via la syntaxe :

```
<select name="boiteRegion" onchange="choix()">
<option>...</option>
</select>
```

### Exercice 3. : examen Janvier 09

Dans le cadre de l'apprentissage en ligne, nous voulons faire vérifier la connaissance des quatre opérations de base aux élèves, figure 3.

#### Vos données

Cliquez sur le bouton "Aléatoire" pour tirer deux nombres entre -100 et 100

A= 
B=

#### Votre solution

A+B	<input type="text"/>
A-B	<input type="text"/>
AxB	<input type="text"/>
A/B	<input type="text"/>

#### La correction

A+B 
A-B 
AxB 
A/B

FIGURE 3 – Simulation d'une calculatrice.

1. Au chargement de la page une fonction `alea()` permet de remplir les deux premières zones de texte avec deux entiers tirés aléatoirement entre -100 et 100. Un clic sur le bouton « aléatoire » permet de faire la même chose. Modifier la page précédente et écrivez la fonction `alea()` ;
2. Après avoir fourni 4 valeurs possibles dans les zones de texte du tableau, l'utilisateur peut vérifier ses réponses en cliquant sur le bouton Vérifier. Ce clic lance une fonction `Vérifier()` qui permet cette vérification. Cette vérification ne peut se faire que si les 4 zones de textes sont remplies avec des nombres. Si une zone n'est pas remplie, elle prend le focus et un message prévient l'utilisateur de son oubli. Modifier la page html et écrire la fonction `Vérifier()` ;
3. Un clic sur le bouton OK permet à l'utilisateur d'obtenir les bonnes réponses. Modifier la page html et la feuille css et écrire la fonction `Solution()` qui affiche la solution dans les 4 zones de texte de la dernière ligne. Attention : les résultats s'affichent en bleu et à droite de la zone de texte.

**Exercice 4.** : Mise en place des fonctions liées aux boutons du Sudoku, figure 4.

**Grille de Sudoku numéro 1**

1		3	
		4	
	3		
	1		3

Solution dans nouvelle page

Remise à zéro

Vérifier ma grille

FIGURE 4 – Sudoku.

1. Préambule à la programmation :

Il est interdit de :

- mettre un `break` dans une boucle (`for`, `do while` ou `while`);
- mettre un `return` dans une boucle (`for`, `do while` ou `while`);
- mettre un `return` dans une exécution conditionnelle (`if...` ou `switch`);
- mettre plusieurs `return` dans une fonction.

2. La numérotation des cases du tableau est donnée par le tableau 1.

1	2	5	6
3	4	7	8
9	10	13	14
11	12	15	16

TABLE 1 – Numérotation des cas du tableau du Sudoku.

3. Le premier bouton permet d'ouvrir une page HTML « grille2Tableau.html » (qui contient la solution) dans une nouvelle fenêtre. Ecrire les lignes HTML et javascript permettant d'ouvrir la nouvelle fenêtre lors du clic sur le bouton « solution dans nouvelle page » avec la page « grille2Tableau.html », figure 5. La nouvelle fenêtre aura comme dimensions 800 par 600.

1	4	3	2
3	2	4	1
2	3	1	4
4	1	2	3

fermer fenetre

FIGURE 5 – Sudoku.

4. La page grille2Tableau.html. contient :

- (a) une solution du Sudoku ;

(b) un bouton permettant de fermer cette nouvelle fenêtre lors d'un clic via la syntaxe :

```
<input type="button" onclick="self.close();"
      value="fermer fenetre" />
```

5. Le bouton « vérifier ma grille » permet seulement de vérifier que la grille est complètement remplie (plus aucune cellule vide). Si c'est le cas, la grille sera envoyée par courrier électronique. Sinon, le « focus » est donné à la première zone vide trouvée.

Modifier si nécessaire votre formulaire de la question 3 et écrire la fonction javascript permettant d'effectuer cette vérification et d'envoyer éventuellement la grille.