

TP N°2 : Messagerie en javascript

TP N°1 : MESSAGERIE EN JAVASCRIPT	1
PREMIERE PARTIE : MESSAGERIE EN HTML CSS ET JS.....	1
Question 1. Créez des documents HTML, CSS, javascript et les liez entre-eux	1
1. FICHIER HTML DE BASE	1
Question 2. Créez un document dont le corps est un élément <i>div</i> qui est de classe <i>conteneur</i>	1
Question 3. Dans celui-ci placez :	1
Question 4. Le titre 1 « Messagerie »	1
Question 5. Un élément <i>div</i> de classe <i>affichage</i>	2
Question 6. Un <i>textarea</i> identifié par <i>clavier</i>	2
Question 7. Un <i>button</i> dont le texte affiché est <i>Ajouter</i>	2
2. LE CSS	2
Question 8. Définissez dans le CSS une classe <i>conteneur</i> qui va créer un conteneur flex avec une direction en colonne, et les éléments à l'intérieur du conteneur sont centrés horizontalement.	2
Question 9. La classe <i>affichage</i> :	2
Question 10. Ajoutez comme propriété à l'affichage que si le texte déborde, est trop long, alors il des barres de défilement s'afficheront	2
Question 11. Le bouton (de type <i>button</i>) ainsi que les titres 1 doivent avoir des marges en bas et en haut de 10 pixels et de type <i>auto</i> à gauche et à droite ; La police de tout le document est <i>monospace</i>	2
3. DEUXIEME PARTIE : JAVASCRIPT UNIQUEMENT	3
a) Déplacer le texte vers l'affichage	3
Question 12. Créez une fonction <i>Deplacer</i> , associée au click sur le bouton (ajoutez dans le HTML), qui va déplacer le texte de la <i>textarea</i> vers un nouveau <i>div</i> de l'affichage. Le contenu du clavier sera alors effacé.	3
b) Modification de la largeur en JS au lancement :	3
Question 13. Créer une fonction <i>ModifierAspectClavier</i> dans lequel vous allez faire les modifications CSS du clavier	3
Question 14. Appelez la fonction <i>ModifierAspectClavier</i> à la fin du chargement de tous les éléments	4
4. ÉCRITURE DANS LA ZONE D'AFFICHAGE (CSS)	4
Question 15. Pour cela créer dans le CSS une classe <i>texte</i> dont voici les propriétés :	4
Question 16. Maintenant pour les textes positionnés à droite de l'affichage, les propriétés de la classe <i>droite</i> sont :	4
Question 17. Pour terminer, les propriétés de gauche sont :	5
5. ÉCRITURE DANS LA ZONE D'AFFICHAGE (JAVASCRIPT)	5
Question 18. Après avoir créée une variable globale <i>compteur</i> Modifiez la fonction <i>Deplacer</i> pour avoir ce comportement	5

Première Partie : Messagerie en HTML CSS et JS

Question 1. Créez des documents HTML, CSS, javascript et les liez entre-eux

1. Fichier HTML de base

Question 2. Créez un document dont le corps est un élément *div* qui est de classe *conteneur*.

Question 3. Dans celui-ci placez :

Question 4. Le titre 1 « Messagerie »

Question 5. Un élément *div* de classe *affichage*

Question 6. Un *textarea* identifié par *clavier*

Question 7. Un *button* dont le texte affiché est *Ajouter*

2. Le CSS

Question 8. Définissez dans le CSS une classe *conteneur* qui va créer un conteneur flex avec une direction en colonne, et les éléments à l'intérieur du conteneur sont centrés horizontalement.

Pour cela voici une explication de chaque propriété utilisée :

1. **display: flex;** : Cette propriété définit l'affichage de l'élément avec la classe **.conteneur** en tant que conteneur flex. Un conteneur flex permet de disposer et d'aligner plus facilement ses éléments enfants (appelés éléments flex) à l'intérieur.
2. **flex-direction: column;** : La propriété **flex-direction** spécifie la direction principale dans laquelle les éléments flex sont disposés à l'intérieur du conteneur flex. Dans ce cas, la valeur **column** indique que les éléments flex sont disposés les uns en dessous des autres, en colonne. Par défaut, la direction principale est définie sur **row**, ce qui signifie que les éléments flex sont disposés horizontalement les uns à côté des autres.
3. **align-items: center;** : La propriété **align-items** définit comment les éléments flex sont alignés sur l'axe transversal (perpendiculaire à l'axe principal) à l'intérieur du conteneur flex. La valeur **center** indique que les éléments flex sont centrés sur l'axe transversal. Dans ce cas, puisque la direction principale est **column**, l'axe transversal est horizontal, et les éléments flex seront centrés horizontalement dans le conteneur.

Question 9. La classe *affichage* :

- Largeur minimale 400 pixels et représente 50 % de l'écran.
- Hauteur 200 pixels.
- Une marge, 10 pixels en bas et en haut et de zéro pixel à droite et à gauche
- La bordure est de 1 pixel, solide et noire
- Les coins de la bordure doivent être arrondis avec un rayon de 5 pixels

La propriété CSS **overflow** contrôle le comportement d'affichage du contenu qui dépasse les limites d'un élément, comme une **div** ou un **textarea**. La propriété **overflow** prend plusieurs valeurs, dont **hidden**, **visible**, **scroll**, et **auto**.

Question 10. Ajoutez comme propriété à l'affichage que si le texte déborde, est trop long, alors il des barres de défilement s'afficheront

Question 11. Le bouton (de type *button*) ainsi que les titres 1 doivent avoir des marges en bas et en haut de 10 pixels et de type *auto* à gauche et à droite ; La police de tout le document est *monospace*

3. Deuxième Partie : javascript uniquement

a) Déplacer le texte vers l'affichage

Lors de cette première étape, nous allons coder en JS le comportement suivant : dès que l'utilisateur appuie sur le bouton, le texte se trouvant dans la textarea *clavier* est déplacé dans la zone au-dessus (l'affichage)

Question 12. Créez une fonction *Deplacer*, associée au click sur le bouton (ajoutez dans le HTML), qui va déplacer le texte de la textarea vers un nouveau *div* de l'affichage. Le contenu du clavier sera alors effacé.

Les contraintes pour cette question sont :

- Le code JS doit être dans *script.js*
- Utiliser `getElementById` **ET** `querySelector`
- Utiliser `document.createElement('div')` pour créer une balise `<div>`
- L'attribut `innerHTML` permet d'accéder au contenu de la balise `div`
- Pour ajouter un élément, utilisez `appendChild`

b) Modification de la largeur en JS au lancement :

Vous devez avoir un affichage semblable :



Figure 1 : le largeur du clavier est trop petit

On voudrait modifier l'apparence du clavier. Il serait possible d'ajouter dans le CSS :

```
#clavier {
  resize: vertical;
  overflow: auto;
  width: 50%;
  min-width: 400px;
}
```

On ne veut pas modifier le CSS, mais faire la modification en javascript.

Pour cela :

Question 13. Créer une fonction *ModifierAspectClavier* dans lequel vous allez faire les modifications CSS du clavier

Si on veut appeler une fonction JS (par exemple *Demarre*) lorsque tous les éléments DOM, images etc. sont chargés, il faut utiliser la méthode `addEventListener` sur l'objet `windows` ainsi :

```
window.addEventListener('load', Demarre);
```

Question 14.
éléments

Appelez la fonction *ModifierAspectClavier* à la fin du chargement de tous les

Messagerie



Figure 2 : l'aspect du clavier est modifié

4. Écriture dans la zone d'affichage (CSS)

Pour l'instant tous les textes entrés sont l'un sous l'autre, ce qui n'est pas lisible.

Aussi, nous voudrions avoir un message à droite puis le suivant à gauche comme sur cet exemple :

Messagerie



Figure 3 : les effets recherchés

Question 15.

Pour cela créer dans le CSS une classe *texte* dont voici les propriétés :

- Les coins sont arrondis avec un rayon de 10 pixels (border-radius)
- l'espace entre le contenu d'un élément et sa bordure est de 10 px (padding)
- l'espace entre la bordure d'un élément et les éléments voisins est de 2 pixels

Question 16.

Maintenant pour les textes positionnés à droite de l'affichage, les propriétés de

la classe *droite* sont :

- La bordure (border) a une épaisseur de 1px, un style solid (plein) et une couleur #00FF00 (vert clair).
- la couleur d'arrière-plan (background-color) des éléments de cette est définie sur #AAFFA (vert plus claire)
- la marge gauche (margin-left) est de 100 px

La marge crée un espace vide de 100 pixels à gauche de l'élément, le poussant vers la droite. Cela peut être utilisé pour décaler les éléments ou les séparer des autres éléments à leur gauche.

Question 17. Pour terminer, les propriétés de gauche sont :

- Une bordure d'un pixel, plein de couleur 0000FF
- Une couleur d'arrière-plan de AAAAFF;
- Une marge de 100 pixels cette fois sur la droite

5. *Écriture dans la zone d'affichage (Javascript)*

Le but de cette partie est de mettre par l'alternance à droite puis à gauche le texte. Pour cela, l'idée est de créer un *compteur* et si compteur est pair alors le message sera de classes *texte* et *droite* dans le cas contraire ce sera *texte* et *gauche*

Question 18. Après avoir créée une variable globale *compteur* Modifiez la fonction *Deplacer* pour avoir ce comportement

Aide :

```
nouvelDiv.classList.add(_____);
```