



Langage de Script JavaScript

TP 2 : DOM

Pr. Mohammed SALIHOUN

salihoun.med@gmail.com

3ème Année - Ingénierie en Informatique et Réseaux

27 octobre 2021



Table of Contents

1. TP 2 : DOM

- Exercice 1 : Diaporama avec des flèches
- Exercice 2 : Diaporama cliquable
- Exercice 3 : Diaporama (Galerie)
- Exercice 4 : Jeu de puzzle

- Exercice 5 : Diaporama (Automatique)
- Exercice 6 : Diaporama (Play/Pause)
- Exercice 7 : Horloge Digitale
- Exercice 8 : QCM Chronométré
- Exercice 9 : Jeu de mémoire



Table of Contents

1. TP 2 : DOM

- Exercice 1 : Diaporama avec des flèches
- Exercice 2 : Diaporama cliquable
- Exercice 3 : Diaporama (Galerie)
- Exercice 4 : Jeu de puzzle

- Exercice 5 : Diaporama (Automatique)
- Exercice 6 : Diaporama (Play/Pause)
- Exercice 7 : Horloge Digitale
- Exercice 8 : QCM Chronométré
- Exercice 9 : Jeu de mémoire



TP 2 : DOM

Exercice 1 : Diaporama avec des flèches

Complétez le fichier **exo1.js** de façon que le document HTML contenu dans le fichier **exo1.html** présente un diaporama simple à l'utilisateur.

- En cliquant sur (l'image de) la flèche de droite, on affiche l'image suivante, et en cliquant sur (l'image de) la flèche de gauche, on affiche l'image précédente.
- Cet affichage est circulaire : quand on a affiché la dernière image et qu'on demande à afficher la suivante, on affiche la première image, et quand on a affiché la première image et qu'on demande à afficher la précédente, on affiche la dernière image.



TP 2 : DOM

Exercice 2 : Diaporama cliquable

Reprenez l'exercice précédent, mais cette fois en rendant l'image elle-même **cliquable**.

- Dans cette version, on affiche l'image suivante en cliquant sur l'image elle-même. Remarquez que dans cette version on ne peut afficher que l'image suivante.
- L'affichage est toujours circulaire : quand on a affiché la dernière image et qu'on demande à afficher la suivante, on affiche la première image.



TP 2 : DOM

Exercice 3 : Diaporama (Galerie)

Complétez le fichier **exo3.js** de façon que le document HTML contenu dans le fichier **exo3.html** affiche :

- une galerie d'images de taille réduite ;
- afficher l'image en taille normale ainsi que sa légende, respectivement dans l'élément IMG et dans l'élément P prévus à cet effet, lors du survol de l'une de ces images avec la souris.

La légende est la valeur de l'attribut **title** des images de taille réduite.



TP 2 : DOM

Exercice 4 : Jeu de puzzle

Pour cet exercice, vous devez travailler dans le serveur web (et non pas en mode file).

- Le but de l'exercice est de réaliser un jeu de puzzle.
- Une grande partie de l'exercice est fournie et vous n'avez qu'à compléter le fichier puzzle.js.

Pour vous expliquer le fonctionnement du jeu, une démonstration sera faite durant le cours. Suivez attentivement cette démonstration et analysez le code HTML avec soin avant de vous lancer dans l'écriture du code JavaScript.



TP 2 : DOM

Exercice 5 : Diaporama (Automatique)

Reprenez l'exercice 2, mais dans cette nouvelle version, le diaporama est **automatique** :

- Dès le chargement de la page, le diaporama se lance automatiquement et les images défilent à intervalle régulier.
- Comme dans l'exercice 2, les images défilent de façon circulaire.

Indication

- 1 La fonction utile pour réaliser cet exercice : `setInterval`.



TP 2 : DOM

Exercice 6 :Diaporama (Play/Pause)

Reprenez l'exercice précédent, mais dans cette nouvelle version, le diaporama peut être stoppé et redémarré simplement en cliquant sur l'image elle-même ! Autrement dit :

- Au chargement de la page, le diaporama démarre automatiquement et on peut l'arrêter en cliquant sur l'image.
- Une fois arrêté, il suffit de cliquer à nouveau sur l'image pour le redémarrer.



TP 2 : DOM

Exercice 7 : Horloge Digitale

Complétez le fichier **exo7.js** de façon que le code HTML du fichier **exo7.html** affiche une horloge digitale. Cette horloge est affichée à l'aide de six images :

- Deux images (pour les unités et les dizaines) pour les heures ;
- Deux images (pour les unités et les dizaines) pour les minutes ;
- Deux images (pour les unités et les dizaines) pour les secondes ;
- Chaque seconde, l'horloge est mis à jour avec les nouvelles valeurs pour le nombre de secondes, de minutes et d'heures.

Pour obtenir ces valeurs, il suffit de calculer la date courante et de les extraire de cette date via les fonctions **getHours**, **getMinutes** et **getSeconds**.



TP 2 : DOM

Exercice 8 : QCM Chronométré

Complétez le fichier **exo8.js** de façon que le code HTML du fichier **exo8.html** affiche une question type QCM à temps limité :

- Après le chargement de la page, l'utilisateur dispose d'un temps limité pour cocher la bonne réponse.
- S'il coche une mauvaise réponse avant la limite, le message adéquat est affiché et il peut faire une nouvelle tentative.
- S'il coche la bonne réponse avant la limite, le message adéquat est affiché.
- S'il n'arrive pas à cocher la bonne réponse avant la limite, le message adéquat s'affiche.
- Enfin, s'il coche une réponse après la limite, un message s'affiche lui signalant qu'il a soit déjà bien répondu, soit déjà mal répondu.



TP 2 : DOM

Exercice 9 : Jeu de mémoire

Pour cet exercice, vous devez travailler dans le serveur web (et non pas en mode **file**).

- Le but de l'exercice est de réaliser un jeu de type **memory**.
- Une grande partie de l'exercice est fournie et vous n'avez qu'à compléter le fichier **memory.js**.

Pour vous expliquer le fonctionnement du jeu, une démonstration sera faite durant le cours. Suivez attentivement cette démonstration et analysez le code HTML avec soin avant de vous lancer dans l'écriture du JavaScript.

Essayez de comprendre l'architecture proposée et complétez le fichier **memory.js** en conséquence.