
TP6 : Arbres DOM

Exercice 1 :

1. Récupérer le dossier [tp6ex1Input](#) et afficher la page HTML
2. Modifier le code de Films.html et proposer un code JS (accesdomelmnt.js) qui permet de récupérer les informations suivantes et les afficher dans la console

```
=====
Titre de la page : Films
Type du contenu de la page : [object HTMLBodyElement]
La troisième bande d'annonce est : http://www.allocine.fr/video/player\_gen\_cmedia=19561982&cfilm=144185.html
=====
Le titre GONE GIRL est de type : [object HTMLHeadingElement]
=====
Notre page contient 19 paragraphes et 4 images
=====
Notre page contient 3 synopsis de films.
=====
La première rubrique des films date de l'année --2014--
Il existe 2 films sous la rubrique 2016
=====
Synopsis du film :IMITATION GAME:
1940 : Alan Turing, mathématicien, cryptologue, est chargé par le gouvernement Britannique de percer le secret de la célèbre machine de
cryptage allemande Enigma, réputée inviolable.
=====
```

Pour ce faire, il est demandé de varier les solutions en utilisant

- Les raccourcis
- La recherche par propriété
- La recherche par sélection CSS

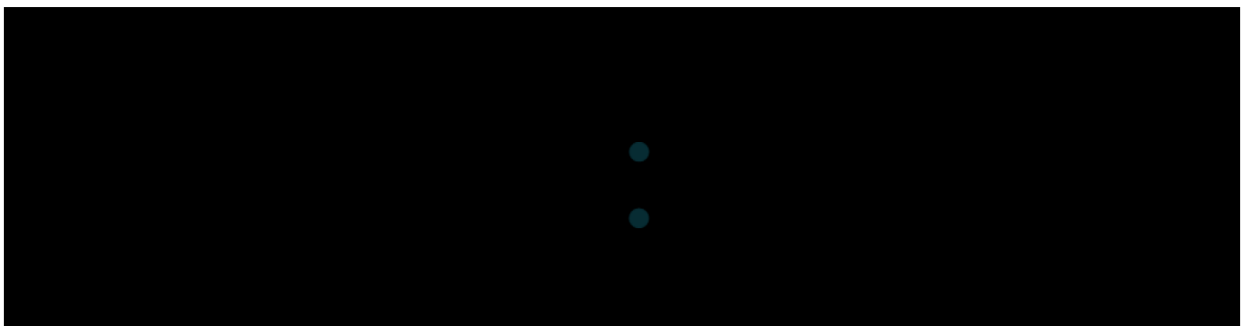
Exercice 2 :

L'objectif de cet exercice est de récupérer quelques informations à partir d'un tableau des notes des étudiants.

1. Récupérer dossier [tp6ex2Input](#)
2. créer un fichier JS qui affiche dans la page HTML (au dessous du tableau) les informations suivantes :
 - 1) titre de la page
 - 2) nombre de ligne de tableau
 - 3) nombre de colonnes de tableau
 - 4) affiche le nombre de semestre
 - 5) affiche la note de S5 de E1
 - 6) affiche la note de S5 de E2
 - 7) ajoute deux points à la note de S5 de E1, agrandi sa taille et la marque en rouge
 - 8) ajoute une zone de texte

Exercice 3 :

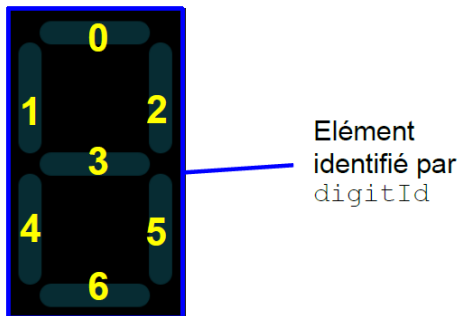
1. Récupérer dossier [tp6ex3Input](#). Le code génère l'affichage suivant :



Le but de ce TP est d'ajouter du code dans le fichier digitalClock.js

pour aboutir à une horloge fonctionnelle. Il est interdit de modifier les fichiers HTML et CSS

2. Complétez la fonction `addSegments(digitId)` afin qu'elle ajoute 7 segments dans l'élément HTML identifié par `digitId`, tels que représentés ci-dessous :



- Les segments sont numérotés de 0 à 6
 - Les segments sont des div ayant les classes CSS "segment" et "off". La classe "off" joue sur l'opacité du segment pour donner l'apparence "éteint".
 - Le segment numéro N possède la classe CSS segmentN (par exemple le segment 0 possède la classe segment0).
 - Pensez à utiliser des boucles et les gabarits (voir [CM3](#))
3. Dans la fonction init(), appliquez la fonction addSegments() aux 4 chiffres de l'horloge. Vous devriez obtenir le résultat suivant :



4. Complétez la fonction `updateDigit(digitId, value)` afin que l'élément identifié par `digitId` affiche la valeur `value`, en jouant sur

état des segments (allumé/éteint). Un segment ne possédant pas la classe "off" apparaît comme allumé. Exemple d'utilisation : l'appel `updateDigit("hours-tens", 2)` doit aboutir à l'affichage :



Pour vous aider, le tableau `segmentStates` vous est fourni, il définit l'état des 7 segments pour chaque valeur :

- L'état d'un segment est modélisé par un entier (1 = allumé, 0 = éteint)
- Les segments sont numérotés de 0 à 6 comme défini précédemment.