

LABORATOIRE 5: ANIMATION ET INTERACTIVITÉ

Exercices

1. Pour débiter

- Copiez le dossier **Exercices 5** qui vous est remis sur Léa.
- Dans le fichier **Animation d'une horloge analogique.htm**, remplacez toutes les occurrences de **VotreNom** par votre nom.
- Copiez, dans la fonction **dessinerHorloge()**, le dessin de votre horloge analogique (que vous avez fait dans les exercices 3). **Ne pas enlever .save() et .restore() au début et à la fin de la fonction.**
- Copiez, dans le dossier **Exercices 5**, les images que vous avez utilisées dans le dessin de votre horloge analogique.
- Enlevez toutes les instructions qui chargent votre (vos) image(s) incluant l'instruction suivante :

```
objImage.onload = function() {
```

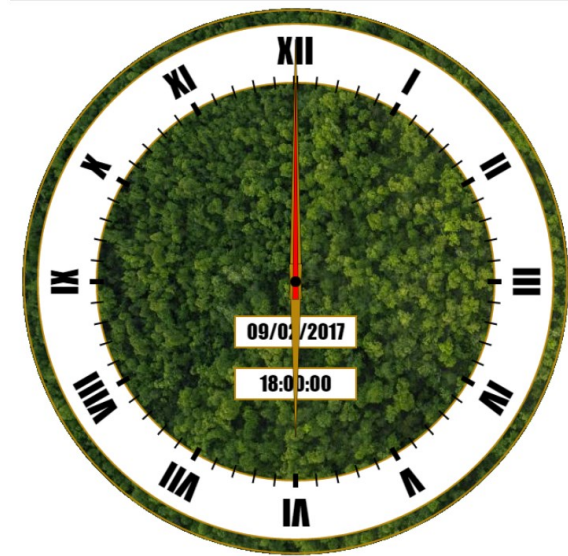
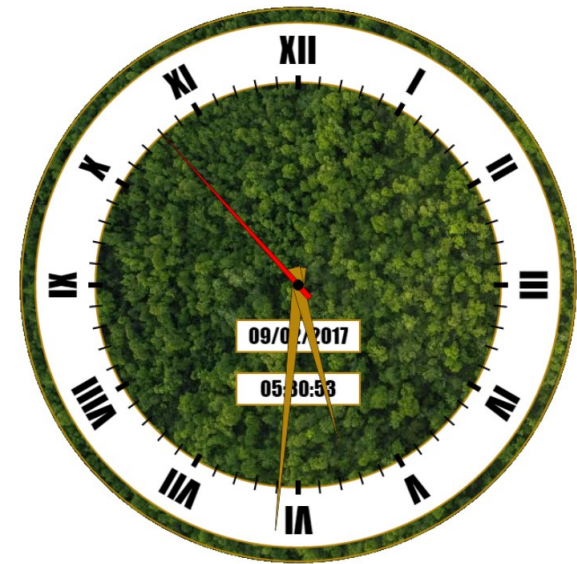
Ici, vous ne devez plus attendre que l'image soit chargée car la fonction **dessinerHorloge()** est appelée 60 fois par seconde. Il est donc évident, qu'à un certain moment, l'image va être complètement chargée.

- Chargez votre (vos) image(s), une seule fois, dans la fonction **initAnimation()**.
- Ouvrez la page Web et vérifiez que votre horloge s'affiche bien.

2. À réaliser – L'animation d'une horloge analogique

- Vous devez animer votre horloge analogique en respectant les contraintes suivantes (voir la vidéo **Démo animation horloge.wmv**).

- La date à l'intérieur du premier rectangle doit être la date du jour. Un 0 doit précéder le numéro du jour si le jour est inférieur à 10. Un 0 doit précéder le numéro du mois si le mois est inférieur à 10. Par exemple : **09/02/2020**.
- L'heure à l'intérieur du second rectangle doit être l'heure actuelle. Un 0 doit précéder le nombre d'heures si le nombre d'heures est inférieur à 10. Un 0 doit précéder le nombre de minutes si le nombre de minutes est inférieur à 10. Un 0 doit précéder le nombre de secondes, si le nombre de secondes est inférieur à 10. Par exemple : **05:08:09**.
- La trotteuse doit indiquer le nombre de secondes actuel et doit avancer par coup de 1 seconde. Par exemple, si l'heure actuelle est 05:30:53 alors la trotteuse doit être sur la bonne graduation entre le **X** et le **XI**. Par exemple, si l'heure actuelle est 18:00:00 alors la trotteuse doit être exactement sur le **XII**.
- La grande aiguille doit indiquer le nombre de minutes actuel. **Vous devez tenir compte des secondes**. Par exemple, si l'heure actuelle est 05:30:53 alors la grande aiguille doit être entre le **VI** et très près de la petite graduation suivante. Par exemple, si l'heure actuelle est 18:00:00 alors la grande aiguille doit être exactement sur le **XII**.
- La petite aiguille doit indiquer le nombre d'heures actuel. **Vous devez tenir compte des minutes et des secondes**. Par exemple, si l'heure actuelle est 05:30:53 alors la petite aiguille doit être à mi-chemin entre le **V** et le **VI**. Par exemple, si l'heure actuelle est 18:00:00 alors la petite aiguille doit être exactement sur le **VI**.



- **Contrainte 1:** Pour obtenir la date du jour et l’heure actuelle, utilisez un objet de type **Date** (**const objDate = new Date()**). Allez chercher cet objet puis extrayez le nombre d’heures, le nombre de minutes, le nombre de secondes, le numéro du jour, le numéro du mois et l’année. Pour connaître les méthodes d’un objet de type **Date**, recherchez sur **Internet**. Vous devez programmer cela dans la fonction **mettreAJourAnimation()**. Ne pas programmer cela dans la fonction **dessinerHorloge()**.
- **Contrainte 2 :** Dans la fonction **dessinerHorloge()**, utilisez ces données pour afficher la date et l’heure et pour faire avancer les aiguilles de votre horloge. Pour que la trotteuse, la grande aiguille ainsi que le petite aiguille indiquent la bonne heure, appliquez, à chacune d’elles, la rotation appropriée.
- **Note :** Les variables qui sont utilisées par plusieurs fonctions indépendantes doivent être globales.

3. À réaliser –Une horloge analogique avec une alarme

- Copiez le fichier **Animation d’une horloge analogique.htm** en lui donnant le nom de **Alarme.htm**.
- Sur la page Web, remplacez toutes les occurrences de **Animation d’une horloge analogique** par **Alarme**.
- Juste avant la balise **<canvas>**, à l’intérieur d’une balise **<table>**, ajoutez les 3 balises suivantes (chaque balise doit être située dans une balise **<td>**) :
 - Une case à cocher **Activer l’alarme** :. Au point de départ, cette case ne doit pas être cochée.
 - Une saisie de donnée pour que l’utilisateur puisse choisir une date et une heure.
 - Un élément audio relié à un audio de votre choix. Au point de départ, l’audio ne doit pas jouer.**Note :** Si votre audio n’est pas une adresse URL, copiez votre fichier audio dans le dossier **Exercices 5**.

☐ Activer l'alarme: ▶ 0:00 / 2:39 🔊 —



- L'utilisateur doit sélectionner la date et l'heure de l'alarme (le nombre d'heures et le nombre de minutes). Si la case à cocher est cochée, l'audio doit jouer à répétition à la date et à l'heure indiquée. Il va de soi que l'utilisateur peut arrêter l'audio en utilisant son composant.

☐ Activer l'alarme: ▶ 0:00 / 2:39 🔊 —

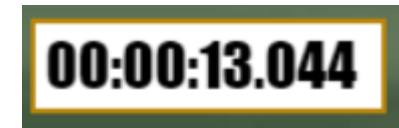
décembre, 2016 ▼

dim.	lun.	mar.	mer.	jeu.	ven.	sam.
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

- Pour faire jouer l'audio, utilisez la méthode **.play()**.
- Ne pas faire jouer l'audio si l'alarme n'est pas activée ou si l'utilisateur n'a pas sélectionné de date et/ou d'heure.
- **Contrainte** : Dans la fonction **mettreAJourAnimation()**, vérifiez si l'alarme est active et si la date ainsi que l'heure sélectionnées correspondent. Si c'est le cas, faites jouer votre audio. Ne faites pas cela dans la fonction **dessinerHorloge()**.
- **Suggestion** : Utilisez la méthode **.split()** pour aller chercher les données de la date et de l'heure.

4. À réaliser –Un chronomètre

- Copiez le fichier **Animation d'une horloge analogique.htm** en lui donnant le nom de **Chronomètre.htm**.
- Sur la page Web, remplacez toutes les occurrences de **Animation d'une horloge analogique** par **Chronomètre**.
- Votre horloge analogique doit se transformer en chronomètre.
- Dans la fonction **mettreAJourAnimation()**, allez chercher, en plus, le nombre de millisecondes à l'intérieur de la date.
- Dans votre rectangle, au lieu d'afficher l'heure dans le format **hh:mm:ss**, vous devez afficher l'heure dans le format **hh:mm:ss.iii** (**iii** est le nombre de millisecondes). Deux 0 doivent précéder le nombre de millisecondes si le nombre de millisecondes est inférieur à 10. Un 0 doit précéder le nombre de millisecondes si le nombre de millisecondes est entre 10 et 99. Par exemple : **03:05:07.009** représente 3 heures, 5 minutes, 7 secondes et 9 millisecondes.
- La trotteuse doit toujours indiquer le nombre de secondes et doit tenir compte du nombre de millisecondes. Par exemple, si l'heure actuelle est 05:30:55.500, alors la trotteuse doit être au bon endroit entre le **XI** et la petite graduation suivante. En d'autres mots, l'avancée de la trotteuse doit être progressive et non pas par coup de 1 seconde.
- Au point de départ, le chronomètre n'est pas en marche. Par conséquent, l'heure doit indiquer minuit (les trois aiguilles doivent pointer sur le **XII** et dans le rectangle, on doit voir **00:00:00.000**).
- Lorsque l'utilisateur clique sur le bouton gauche de la souris, si le chronomètre n'est pas en marche, le chronomètre doit se mettre en marche. Dans le rectangle, on doit voir le temps s'écouler et l'horloge doit être fonctionnelle sauf qu'elle doit indiquer le temps écoulé (et non pas l'heure).



- Lorsque l'utilisateur clique de nouveau sur le bouton gauche de la souris, si le chronomètre est en marche, celui-ci doit s'arrêter à cet endroit. Le chronomètre doit se remettre en marche lorsque l'utilisateur clique de nouveau sur le bouton gauche de la souris.
- Lorsque l'utilisateur double-clique sur le bouton gauche de la souris, le chronomètre doit s'arrêter (s'il est en marche) et doit se remettre à 0.
- Voir la vidéo **Démo chronomètre.wmv**. Dans cette démo, je clique sur le bouton gauche de la souris à plusieurs reprises. A la fin, je double-clique sur ce bouton, ce qui a comme conséquence de remettre le chronomètre à 0.
- **Note importante** : Il existe plusieurs manières pour programmer cela. La méthode la plus précise est d'utiliser des objets **Date**. Voici comment :
 - Au début du programme, initialisez le nombre de millisecondes cumulées à 0 (dans une variable globale) et indiquez que le chronomètre n'est pas en marche (dans une autre variable globale).
 - Dès que l'utilisateur met en marche le chronomètre, à l'aide de l'objet **Date**, allez chercher la date et l'heure actuelle (appelons-la **objDateHeure1** qui doit être une variable globale). **Ne faites pas cela si le chronomètre est déjà en marche.**
 - Dans la fonction **mettreAJourAnimation()**, si le chronomètre est en marche :
 - A l'aide de l'objet **Date**, allez chercher la date et l'heure actuelle (appelons la **objDateHeure2** qui doit être une variable locale).
 - Soustrayez **objDateHeure1** de **objDateHeure2**. Cela va vous donner le temps écoulé, en millisecondes, entre les 2 dates/heures (**let intMsEcoulees = objDateHeure2 - objDateHeure1**).
 - Affectez à **objDateHeure1** la valeur de **objDateHeure2** (pour le prochain cycle d'animation).
 - Additionnez ces millisecondes écoulées aux millisecondes cumulées.
 - **Ne faites rien si le chronomètre n'est pas en marche.**

- Que le chronomètre soit en marche ou non, transformez le nombre de millisecondes cumulées en heures, minutes, secondes et millisecondes (pour que votre horloge puisse s'en servir). Par exemple, 5000010 millisecondes donnent 1 heure, 23 minutes, 20 secondes et 10 millisecondes. Utilisez ces données pour l'affichage du temps écoulé et pour le positionnement des aiguilles.
- Pour remettre le chronomètre à 0, vous n'avez qu'à réinitialiser le nombre de millisecondes cumulées à 0.

5. À remettre

- Le dossier **Exercices 5** compressé.
- Au plus tard, **Mercredi, le 12 février 2020 à 16:00**
- **Aucun retard ne sera accepté**
- Sur LÉA