

## INTRODUCTION:

### *La gestion du temps*

#### 1. Les dates:

Une date est un objet dont la base est un nombre de milliseconde et qui s'affiche dans plusieurs formats par conversion des millisecondes en année, mois, jours...

La version JavaScript 1.3 à uniformisé et étendu les possibilités de l'objet. Elle est reconnue par tous les navigateurs mêmes anciens.

```
new Date();
new Date(timestamp);
new Date(dateString);
new Date(année, mois, jour [, heure, minutes, secondes, millisecondes ]);
```

- **S'il n'y a pas d'argument, l'objet correspond au jour et à l'heure lors de la création de l'objet.**
- Dans le cas où l'argument est une date sous forme de texte, son format devra être analysé par la méthode **Date.parse()**. **Le constructeur Date(dateString) appelle implicitement cette méthode.**  
[https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Objets\\_globaux/Date/parse](https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Objets_globaux/Date/parse)
- Le nombre de milliseconde est relatif au 1 janvier 1970. On utilise les millisecondes pour calculer les durées.

Quelques méthodes utiles:

```
getFullYear() : Renvoie l'année sur 4 chiffres ;
getMonth() : Renvoie le mois (0 à 11) ;
getDate() : Renvoie le jour du mois (1 à 31) ;
getDay() : Renvoie le jour de la semaine (0 à 6, la semaine commence le dimanche) ;
getHours() : Renvoie l'heure (0 à 23) ;
getMinutes() : Renvoie les minutes (0 à 59) ;
getSeconds() : Renvoie les secondes (0 à 59) ;
getMilliseconds() : Renvoie les millisecondes (0 à 999).
getTime() : Renvoie le timestamp de la date de votre objet ;
setTime() : Vous permet de modifier la date de votre objet en passant en unique paramètre un timestamp.
```

[http://www.w3schools.com/jsref/jsref\\_obj\\_date.asp](http://www.w3schools.com/jsref/jsref_obj_date.asp)

L'objet Date supporte les opérations arithmétiques. Dans la pratique, on utilise surtout la soustraction pour connaître la durée qui sépare deux dates.

#### Exercice 1:

Vous désirez calculer l'écart entre le début de **l'an 1000** et **l'an 2000**. Testez le script suivant:

```
<script>
var debut = new Date(1000,0,01);
var fin = new Date(2000,0,01);
var diff = new Number((fin.getTime()-debut.getTime())/31536000000).toFixed(0);
document.write(diff);
</script>
```

## Travail à effectuer et questions:

1. Que renvoie la méthode **getTime()** dans l'exemple ci-dessus?
2. Expliquez la relation qu'il existe entre le nombre 31536000000 et le résultat affiché.
3. A quoi sert la méthode **toFixed()**?
4. Ecrire une fonction nommée **jourDiff(deb,fin)** à laquelle on passe la date de début et de fin et qui retourne la différence en nombre de jour écoulé. Fournir un script de démonstration de l'utilisation de cette fonction qui donnera le nombre de jour que vous avez vécu à ce jour.

*Joindre ce travail à votre rapport.*

## 2. Gérer des événements temporels:

**setTimeout()** permet le choix d'un intervalle de temps avant l'exécution d'une fonction:

**setTimeout(maFonction, 3000);** // maFonction sera exécutée au bout de 3 secondes.

On annule une fonction temporelle avec **clearTimeout()**

La méthode **setInterval()** appelle une fonction ou évalue une expression à des intervalles spécifiés (en millisecondes).

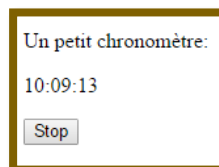
La méthode **setInterval()** continuera d'appeler la fonction jusqu'à ce que **clearInterval()** est appelée, ou que la fenêtre est fermée.

## Exercice 2:

Effectuez une page WEB qui affichera un chronomètre (très simplifié) sous forme d'un texte au chargement de la page:

Le chronomètre utilisera l'heure courante du PC et aura une résolution d'une seconde. La page contiendra bouton "stop" qui arrêtera le chronomètre au clic.

Votre script devra utiliser deux méthodes. L'une définissant le chronomètre, l'autre pour gérer le bouton stop.



***Vous commenterez correctement le fichier html de votre page WEB qui contiendra le script et le rendrez en fin de séance.***

## TP: Une horloge en JS, html5 et CSS3

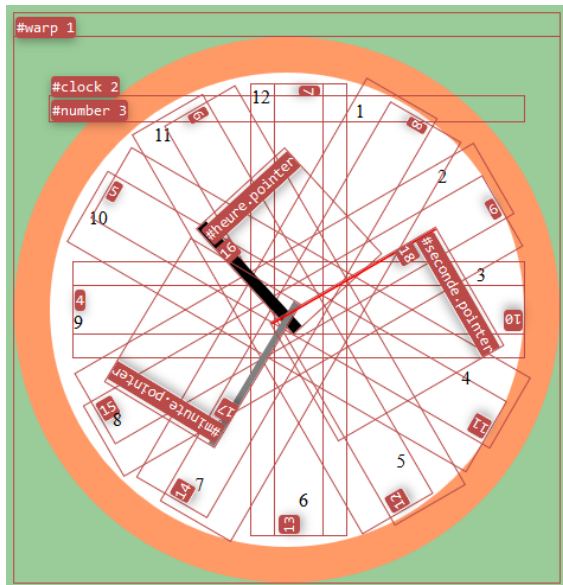


Vous devez obtenir l'horloge ci-dessus. Les aiguilles sont évidemment animées. Les couleurs sont laissées à votre discrétion.

## Guide de travail: (lire la totalité avant de commencer à coder).

- Html, la structure:

Cette horloge contient un grand nombre d'éléments html de type <div>, lorsque l'on en demande l'affichage dans Mozilla Firefox, voilà ce que l'on obtient:



La capture d'écran ci-contre correspond à l'horloge en fonctionnement avec le CSS activé bien sûr.

La numérotation en rouge correspond à l'ordre d'apparition des div dans le document html.

Les nombres en noir, sont les caractères de l'horloge et sont insérés dans des éléments <span>:

```
<div><span>12</span></div>
<div><span>1</span></div>
```

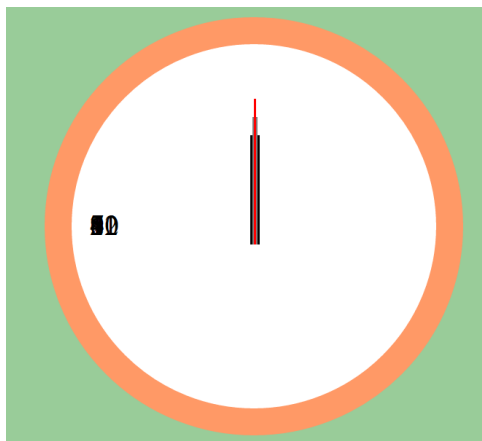
On voit aussi l'utilisation d'une classe "pointer" pour faciliter le positionnement des aiguilles en CSS.

L'élément "div" d'identifiant "warp" est un conteneur carré

de dimensions fixes de 460px.

- Le CSS, le style:

Lorsque le script JavaScript est désactivé, l'on obtient la vue suivante:



Les nombres sont tous superposés à la position du chiffre "9".

Les aiguilles sont alignées sur "midi" et centrée au centre de l'horloge.

Vous devrez faire attention au positionnement CSS, pour obtenir la figure ci-contre. Pour positionner les aiguilles j'ai utilisé la règle CSS transform-origin:

<https://developer.mozilla.org/fr/docs/Web/CSS/transform-origin>

**Conclusion:** C'est avec JavaScript que nous allons modifier la position des éléments pour faire le positionnement initial des "span" et des "div".

- La partie active côté client, le JS:

Pour vous aider on vous donne le départ du script:

```
31 var horlogeNumber=document.getElementById("number");
32 var horlogeDiv=horlogeNumber.getElementsByTagName("div");
33 var horlogeSpan=horlogeNumber.getElementsByTagName("span");
34 for(var i=0;i<horlogeDiv.length;i++){
```

Dans ce script, on utilise successivement deux boucles **"for"**. L'une pour effectuer le positionnement initial des **"div"**, l'autre pour faire le positionnement initial des **"span"**. Ces structures itératives permettent de manipuler le CSS (style) en appliquant une transformation en rotation. Il vous faudra peut-être un peu réviser votre trigonométrie.

Ensuite nous utiliserons une fonction, pour obtenir les valeurs de l'heure courante, dans le but d'effectuer le positionnement en rotation des aiguilles conformément à l'heure lue au niveau de la machine.

Il vous restera à appeler cette fonction à intervalle de temps régulier.

***Rendre une horloge opérationnelle avec des fichiers correctement commentés.***