

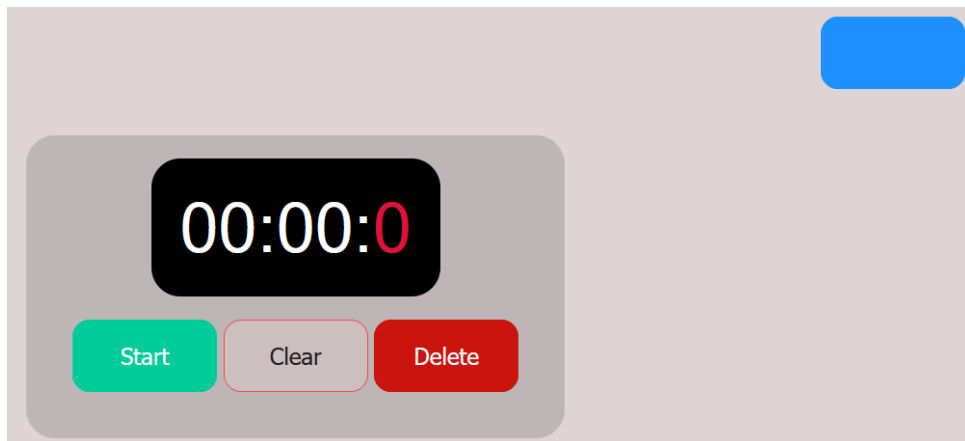
Institut Galilée Paris 13	LANGAGES WEB - JAVASCRIPT Exercice Chronomètre Version Classique	Réf. : WEBC [0.9]
		Page : I - 1

Exercice Chronometre (Version Classique)

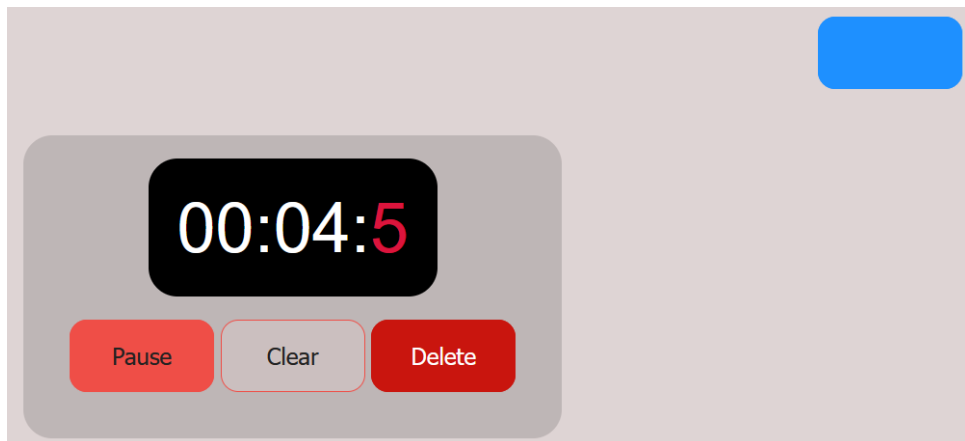
Du code HTML vous est fourni. **Il est interdit d'y toucher**. Nous allons essayer d'afficher et de gérer un chronomètre qui s'égrène en dixièmes de seconde. Le bouton + devra être désactivé à la création du chronomètre et réactiver lorsque l'on supprime le chronomètre.

Voici les étapes "marquantes" de la vie du chronomètre.

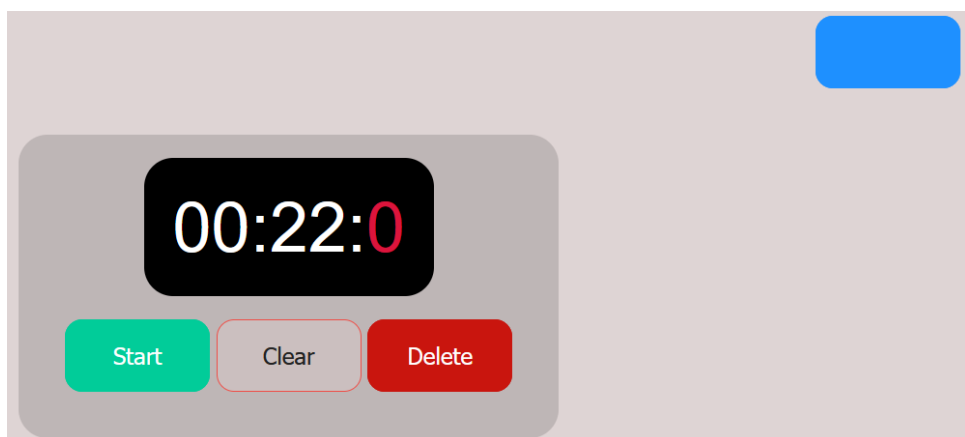
La naissance (après pression sur le bouton +):



Le démarrage (après pression sur le bouton Start):



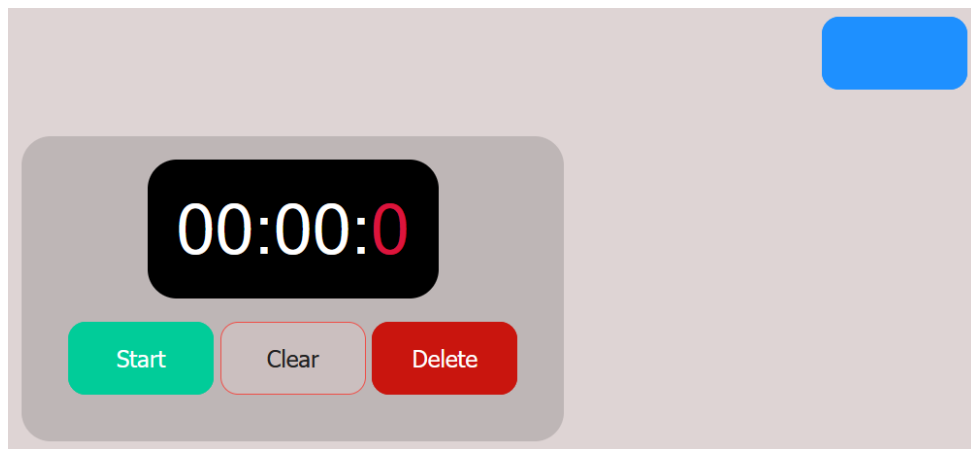
Mise en pause (après pression sur le bouton Pause):



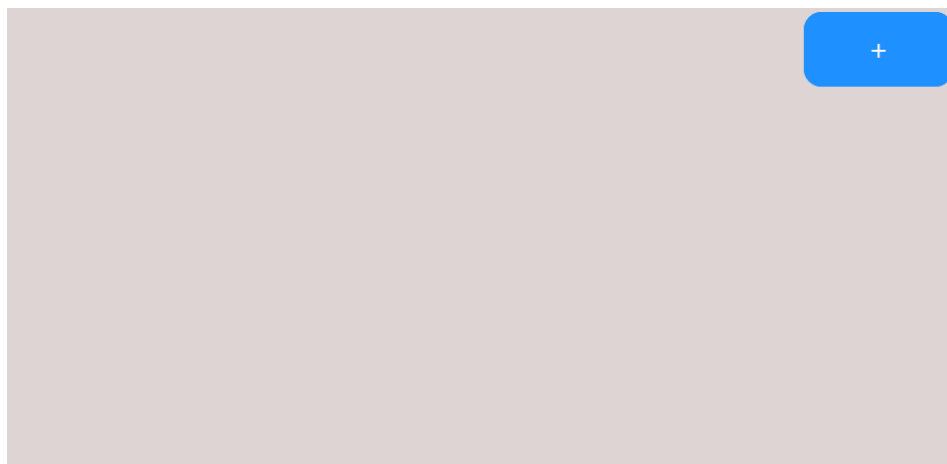
Remarque : Si on clique sur le bouton Start le chronomètre continue sa progression.

Institut Galilée Paris 13	LANGAGES WEB - JAVASCRIPT Exercice Chronomètre Version Classique	Réf. : WEBC [0.9]
		Page : 1 - 2

La remise à zéro (après pression sur le bouton Clear):



La suppression (après pression sur le bouton Delete):



Ainsi, un chronomètre comportera quatre commandes : start/pause, clear et delete.

Normalement, il n'y a pas besoin de variables globales supplémentaires à part celle que j'ai mise dans le code js. Voici la liste des principales fonctions à écrire. On peut en ajouter si vraiment c'est nécessaire.

La fonction `creerChronometre()`:

Elle permet de créer dans le DOM la structure du chronomètre telle que décrite par le "template" présent dans le code HTML. Attention il faudra éliminer l'affichage de l'heure. On s'arrête au minutes. Ne pas oublier d'ajouter l'identifiant "idChrono" au chronomètre. Ça servira plus tard.

Après clonage, le chronomètre est ajouté au conteneur `<main>`. Ensuite on va préparer l'affichage du chronomètre.

La fonction `preparerChronometre()`:

Quelques aspects de la préparation du chronomètre: mettre les bons textes, cacher les heures avec l'attribut "hidden". On attribut aussi les comportements initiaux des 3 boutons avec les méthodes à écrire (par la suite) : **demarrer()**, **raz()**, **supprimer()**.

Il ne faudra pas oublier de mettre à zéro la valeur du chronomètre et de l'afficher (fonction **afficherHoraire()** à écrire aussi).

Institut Galilée	LANGAGES WEB - JAVASCRIPT	Réf. : WEBC [0.9]
Paris 13	Exercice Chronomètre Version Classique	Page : 1 - 3

La fonction `demarrer()`:

Elle utilise un timer pour exécuter la fonction **plusUn()** toutes les 100 millisecondes (donc 1 dixième de seconde). On stocke le timer pour pouvoir l'arrêter plus tard. Elle échange le comportement du bouton qui passe en mode Pause. La classe CSS du bouton doit aussi changer...

Conseil : Vous pouvez utiliser le bouton Delete pour déclencher la pause du chronomètre dans un premier temps. Cela vous permet de coder la fonction arrêtant le timer. Ensuite, vous pouvez vous concentrer sur l'élaboration de l'alternance de comportement entre Start et Pause.

La fonction `raz()`:

Elle remet le compteur de dixièmes à zéro puis elle affiche le chronomètre (fonction **afficherHoraire()**).

La fonction `supprimer()`:

Elle supprime du DOM le chronomètre. Elle arrête le timer et elle remet en fonction le bouton +.

La fonction `plusUn()`:

Elle incrémente de 1 les dixièmes puis elle affiche le chronomètre (fonction **afficherHoraire()**).

La fonction `pause()`:

Elle arrête le timer. Elle échange le comportement du bouton qui passe en mode Start.

La fonction `afficherHoraire()`:

Elle convertit d'abord le compteur de dixièmes en minute/seconde/dixième. Puis, elle intègre ces valeurs sous forme de chaîne de caractères dans le chronomètre. On complétera si besoin avec des '0' grâce à la méthode `padStart()` pour que ça soit "joli".