

```

export class Chronometre {
  #dixieme
  #refTicks
  #idChrono
  #onUpdate
  static #nbChrono = 0

  constructor(onUpdate) {
    Chronometre.#nbChrono++
    this.#idChrono = `chrono${Chronometre.#nbChrono}`
    this.#dixieme = 0;

    this.demarrer = this.demarrer.bind(this)
    this.pause = this.pause.bind(this)
    this.#onUpdate = onUpdate;
  }

  get idChrono() {
    return this.#idChrono;
  }

  /**
   * Supprime l'intervale à la suppression du chronometre
   */
  supprimer() {
    clearInterval(this.#refTicks)
  }

  /**
   * Efface la valeur du chronometre et demande le refresh
   */
  clearChrono() {
    this.#dixieme = 0
    this.#onUpdate(this.#idChrono)
  }

  /**
   * Interrompt le chrono sans l'effacer
   */
  pause() {
    clearInterval(this.#refTicks)
  }

  /**
   * Augmente l'unité de temps et demande le refresh
   */
  #plusUn() {
    this.#dixieme++;
    this.#onUpdate(this.#idChrono)
  }

  /**
   * Démarre de décompte et mémorise l'identifiant du Timer
   */
  demarrer() {
    this.#refTicks = setInterval(this.#plusUn.bind(this), 100)
  }

  /**
   * Retourne la valeur du chronometre dans un tableau pour
   * les minutes, secondes et dixiemes sous forme numérique

```

```
    * @return {number[3]}
    */
    horaireFromDixieme() {
        const minutes = Math.floor(this.#dixieme / 600)
        const secondes = Math.floor((this.#dixieme % 600) / 10)
        const dixieme = (this.#dixieme % 10)

        return [minutes, secondes, dixieme]
    }
}
```