

```

/**
 * Représentation d'un Album.
 * @typedef {Object} Album
 * @property {number} userId - identifiant Artiste.
 * @property {string} title - titre de l'album.
 * @property {number} id - identifiant Album.
 */

/**
 * Représentation d'un Album avec vue.
 * @typedef {Album} AlbumWithView
 * @property {number} vue - Nombre de vues d'un album.
 */

/**
 * Retourne un tableau d'albums avec une nouvelle propriété vue et une valeur aléatoire
 * @param {Album[]} desAlbums
 * @returns {AlbumWithView[]}
 */
export function ajouterVues(desAlbums) {
  return desAlbums.map((album) => {
    return {
      ...album,
      vue: Math.floor(Math.random() * 100 + 1) * 100
    }
  })
}

/**
 * Filtre les albums avec au moins 9000 vues
 * @param {AlbumWithView[]} desAlbums
 * @param {number} limiteDeVue
 * @return {AlbumWithView[]}
 */
export function bestOf(desAlbums, limiteDeVue = 9000) {
  return desAlbums.filter( (album) => {
    return album.vue >= limiteDeVue
  })
}

/**
 * Retourne un tableau des meilleurs albums
 * @param {AlbumWithView[]} desAlbums
 * @param {number} finTop
 * @return {AlbumWithView[]}
 */
export function topAlbum(desAlbums, finTop = 3) {
  const debutTop = 0
  return Array.from(desAlbums)
    .sort((album1, album2) => album2.vue - album1.vue)
    .slice(debutTop, finTop)
}

```