Framework Friday: Anime.JS

Steps to Install

1. `npm install vite --save-dev`
2. `npm install animejs`
3. Add below to package.json in place of scripts:

   ```
   "scripts": {
       "dev": "vite",
       "build": "vite build",
       "preview": "vite preview"
   }
   ```

4. Run website using `npm run dev`.

Objects

You can find objects that anime.js has by using their website: https://animejs.com/documentation. - Importing - Functions can be imported with the import command in the js file.

```
import { createTimeline, utils, createDraggable} from 'animejs';
```

- Utils
  - A Collection of utility functions for animation tasks.
    * This library works by targeting class or html type. you target them like this:

      ```
      const [ $logo ] = utils.$('.logo.js');
      const [ $button ] = utils.$('button');
      ```

      ```
      <div class="largeCenteredRow">
          <svg class="logo js" preserveAspectRatio="xMidYMid meet"
           ↪  viewBox="0 0 630 630">
      <path fill="currentColor" d=""/></svg>
      </div>
      <div class="mediumRow">
          <fieldset class="controls">
              <button>rotations: 0</button>
          </fieldset>
      </div>
      ```

- Draggable

  - Adds draggable items to DOM Elements
  - Create the Draggable Item using `createDraggable(target, parameters)`
    - Has axes parameters such as:
      - locking x and y directions

        ```
        import { createDraggable } from 'animejs';

        createDraggable('.square.enabled', {
          x: true
        });

        createDraggable('.square.disabled', {
          x: false
        });
        ```

      - snapping to an area
      - mapping to an area
  - Methods
    - enable, disable
    - setX, setY
    - reset, stop, revert

- Timer
  - createTimer:

    ```
    const timer = createTimer({
      duration: 2000,
      onUpdate: self => $time.innerHTML = self.iterationCurrentTime,
      loop: true,
    });
    ```

  - Callbacks: Can execute functions during a timer playback
    - onBegin, onComplete, onUpdate, onLoop, onPause
  - Methods: control over the timing, behavior, and progression of the timer.
    - resume, pause, alternate

      ```
      const resumeTimer = () => timer.resume();
      const pauseTimer = () => timer.pause();
      const alternateTimer = () => timer.alternate();

      $resumeButton.addEventListener('click', resumeTimer);
      $pauseButton.addEventListener('click', pauseTimer);
      $alternateButton.addEventListener('click', alternateTimer);
      ```

- Animations
  - Can target CSS and DOM elements. Ways you can animate are translations, scaling, opacity, rotations and more!
  - Takes in a target and parameters:
    * Keyframes: When things happen and what happens
    * Property specific: Like rotate, delay, ease, loop
  - Created with `animate(targets, parameters)`
  - Methods:
    * Pause, play, start, reverse, etc
- Timeline
  - Synchronize animations, timers, and callbacks together.
  - Methods:
    * Pause, play, restart, resume
    * add, sync, call, label, set
  - Creation example

    ```
    const tl = createTimeline({ defaults: { duration: 750 } });

    tl.label('start')
    .add('.square', { x: '15rem' }, 500)
    .add('.circle', { x: '15rem' }, 'start')
    .add('.triangle', { x: '15rem', rotate: '1turn' }, '<-=500');
    ```