**GSAP**

**GreenSock Animation Platform Animation Library**

* In GSAP, there are **two** ways for applying animations
  1. **Initial to final position:**
     + **gsap.to(“**elemName”, { }**)**
  2. **Final to initial position:**
     + **gsap.from(“**elemName**”,{ })**

**GSAP Timeline**

* When we use **timeline** method with **many elements** with different names, then all elements will be played one by one.
* To **Pause timeline,** we use **tl.pause(),** for playing again, **tl.play(),** for reversing **tl.reverse()**
* **Difference between timeline and delay:**
  + **timeline** is a method that is used on different **elements** with **different** **names.** Different elements, different animations.
  + **delay** is a property that is used on different elements with **same names.** Same name elements, same animations.
* **Example:**

let tl = gsap.timeline()

tl.to("#box1", {

    x: 1000,

    duration: 2,

})

tl.to("#box2", {

    x: 1000,

    duration: 2,

})

**Some GSAP Properties**

1. **x/y:** Moves elements **horizontally/vertivally** in pixels with relative to parent (x: 100)
2. **xPercent/yPercent:** Moves **relative** to the element’s own width/height
3. **rotation:** Rotates the element in **degree**
4. **scale**/
5. **stagger:** Delays animation for multiple elements
6. **repeat: 1 –** the animation will repeat 2 times ( 0 = 1st time, 1 = 2 times , -1 = infinite times).
7. **yoyo: true/false –** The animation will goes **to** to **from** and **from** to **to (**kind of infinity)

**GSAP EASING**

* it is the graph of animations initial to final position

**SCROLLTRIGGER**

**GSAP Plugin for scrolling**

**A white rectangular object with black text

AI-generated content may be incorrect.**

* It is used for scroll trigger animations on our website
* ScrollTrigger properties:
  1. **trigger:”**elemName**” -** The element that will trigger the animation
  2. **scroller:”**parentName**” -** The parent element that acts as the scrollbar (default is body)
  3. **start:”**top 10%**” -** Animation starts when top of the element reaches specified viewport
  4. **end:** Animation end when top of the element reaches specified viewport
  5. **markers:**true -Displays markers to visualize start and end points (for debugging)
  6. **scrub:1-5 -** Smoothly animates based on scroll position instead of a fixed duration (1-5 value)
  7. **pin:**true – Stops/pin the screen on scroll and on that time, animation works. For using this, we have to trigger the parent of the element
     + - **Example:**

gsap.to("#parent element", {

  scrollTrigger: {

    start: "top 0",

    end: "top -100%",

    pin: true,

  },

});

* **IMPORTANT:** Changing gsap animations based on screen sizes using **ScrollTrigger.matchMedia({})**

Example:

    ScrollTrigger.matchMedia({

      "(max-width: 600px)": function () {

        document.querySelector(".btn").addEventListener("click", () => {

          gsap.to(".box", { x: 200 });

        });

      },

    });

Media sizes:

ScrollTrigger.matchMedia({

  // 🌐 Very large desktops (e.g. 2K, 4K)

  "(min-width: 1600px)": function () {

    console.log("🔵 Ultra-wide Desktop");

    // GSAP animations for ultra large screens

//My Desktop

  },

  // 💻 Standard desktops & laptops

  "(min-width: 1200px) and (max-width: 1599px)": function () {

    console.log("🟢 Desktop / Laptop");

    // GSAP animations for normal desktops

  },

  // 📺 Small laptops & large tablets (landscape)

  "(min-width: 992px) and (max-width: 1199px)": function () {

    console.log("🟡 Small Laptop / Large Tab");

    // GSAP animations for small laptops or large tablets

  },

  // 📱 Tablets (portrait)

  "(min-width: 768px) and (max-width: 991px)": function () {

    console.log("🟠 Tablet (Portrait)");

    // GSAP animations for tablets

  },

  // 📲 Mobiles

  "(max-width: 767px)": function () {

    console.log("🔴 Mobile");

    // GSAP animations for mobile

  },

});

**LENIS JS**

**Smooth scrolling library**

* lis a library for smooth scrolling effects on the website

**How to Setup LocoMotive?**

* Need Three **Things** for adding Lenis to our website
  1. **Lenis JS Script src :**
* Go to google, search **“Lenis JS Github”**
* Copy the **Script tag** from the **installation** section
* Paste on the HTML above the main js tag.
  1. **Lenis JS code :**
* Copy the **basic/Custom raf loop** JS code from the **Setup** section
* Paste in the JS main file
  1. **Lenis JS CSS code :**
* Same steps as other. Just paste CSS Link inside the <head> tag