**GSAP TUTORIALS (SHEHRIYAN CODING SCHOOL YT)**

**Notes:**

**TOPIC: BASIC**

gsap.to (Initial to Final) or gsap.from (Final to Initial)

gsap.to has 2 parameters -> gsap.to(“param1”, {param2}) -> param1 is what we want to animate

properties in camelCase

properties: x, y, duration, delay, rotate, repeat, yoyo, backgroundColor, opacity, stagger, borderRadius, scale

repeat: -1 -> INFINITE REPEAT

**TOPIC: TIMELINE**

Var tl = gsap.timeline()

**TOPIC: SCROLL TRIGGER**

Script tag between GSAP and own script tag

scrollTrigger: ”#page2 #box”

**TOPIC: REACT**

Way 1:

const gsapRef = useRef();

  useGSAP(() => {

    gsap.to(gsapRef.current, {

      x: 1400,

      duration: 2,

      delay: 1,

      rotate: 720,

    });

  });

  return (

    <main>

      <div ref={gsapRef} className="box"></div>

    </main>

  );

Way 2:

useGSAP(

    () => {

      gsap.from(".box", {

        rotate: 360,

        scale: 0,

        duration: 2,

        opacity: 0,

        delay: 0.5,

      });

    },

    {

      scope: ".container",

    }

  );

  return (

    <main>

      <div className="container">

        <div className="circle"></div>

        <div className="box"></div>

      </div>

Way 3:

Use contextSafe

const App = () => {

  const boxRef = useRef();

  const { contextSafe } = useGSAP();

  const rotateBox = contextSafe(() => {

    gsap.to(boxRef.current, {

      duration: 1,

      rotate: 360,

    });

  });

  return (

    <main>

      <button onClick={rotateBox}>Animate</button>

      <div ref={boxRef} className="box">

        Box

      </div>

    </main>

  );

};