Animate On SCROLL

**First method (with HTML + CSS + JS)**

HTML code :

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <link rel="stylesheet" href="style.css">

</head>

<body>

    <div class="container">

        <div class="section hidden" id="s1">

            <h1>section 1</h1>

            <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Id et nisi veritatis, reiciendis eius nemo?</p>

        </div>

        <div class="section hidden" id="s2">

            <h1>section 1</h1>

            <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Id et nisi veritatis, reiciendis eius nemo?</p>

        </div>

        <div class="section hidden" id="s3">

            <h1>section 1</h1>

            <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Id et nisi veritatis, reiciendis eius nemo?</p>

        </div>

    </div>

    <script src="app.js"></script>

</body>

</html>

CSS code :

\*{

  padding: 0;

  margin: 0;

}

.container{

    background: rgba(128, 128, 128, 0.709);

    display: flex;

    flex-direction: column;

    flex-wrap: wrap;

    /\* flex-flow: column wrap; \*/

}

/\*---------------------------------\*/

.section{

    padding: 25px;

    margin: 25px;

}

#s1{

    width: 40%;

    height: 500px;

    background: rgba(149, 76, 76, 0.831);

}

#s2{

    width: 40%;

    height: 500px;

    background: rgba(51, 15, 15, 0.581);

}

#s3{

    width: 40%;

    height: 500px;

    background: rgba(211, 171, 171, 0.831);

}

/\*------------------ Prancipal Style -----------------------\*/

.hidden{

   opacity: 0;

   transition: all 1s;

   transform: translateX(-100);

   filter: blur(5px);

}

.show{

  opacity: 1;

  filter: blur(0);

  transform: translateX(0);

}

JS code :

const observer = new IntersectionObserver((entries)=>{

    entries.forEach((entry)=>{

        console.log(entry)

        if(entry.isIntersecting){

            entry.target.classList.add('show');

        }else{

            entry.target.classList.remove('show');

        }

    })

})

const allSection = document.querySelectorAll('.section');

 allSection.forEach((el)=> observer.observe(el)) ;

**Second method ( only HTML + CSS)**

HTML code :

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <link rel="stylesheet" href="css.css">

</head>

<body>

<div class="parent">

    <div class="item1 animation">item1</div>

    <div class="item2 animation">item2</div>

    <div class="item3 animation">item3</div>

    <div class="item4 animation">item4</div>

    <div class="item1 animation">item1</div>

    <div class="item2 animation">item2</div>

    <div class="item3 animation">item3</div>

    <div class="item4 animation">item4</div>

    <div class="item1 animation">item1</div>

    <div class="item2 animation">item2</div>

    <div class="item3 animation">item3</div>

    <div class="item4 animation">item4</div>

    <div class="item1 animation">item1</div>

    <div class="item2 animation">item2</div>

    <div class="item3 animation">item3</div>

    <div class="item4 animation">item4</div>

    <div class="item1 animation">item1</div>

    <div class="item2 animation">item2</div>

    <div class="item3 animation">item3</div>

    <div class="item4 animation">item4</div>

    <script src="js.js"></script>

</body>

</html>

CSS code :

.parent{

    background: gray;

    padding: 25px;

    margin: 5px;

    display: flex;

    flex-wrap: wrap;

    flex-direction: row;

    /\* flex-flow: column wrap; \*/

}

.parent div{

    margin: 5px;

    padding: 5px;

}

.item1 , .item2 , .item3 , .item4{

height: 150px;

width: 45%;

}

.item1{

    background: thistle;

    padding: 5px;

}

.item2{

    background: darkred;

}

.item3{

    background: rgb(0, 255, 204);

}

.item4{

    background:coral;

}

/\*-------  Principale Code  -------\*/

.animation{

    animation-name: anim;

    /\* animation-duration: 2s; \*/ /\*---> animation-timeline: view();\*/

    animation-timeline: view();

    animation-timing-function: linear;

    /\* animation-iteration-count: infinite; \*/

    animation-range: entry 0;

}

@keyframes anim {

    0%{

       scale: 0;

       opacity: 0;

       clip-path: inset(100% 100% 0 0);

    }

    100%{

        scale: 1;

        opacity: 1;

        clip-path: inset(0 0 0 0);

    }

}

**Exemple of** addEventListener (scroll)

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Scroll Color Change Example</title>

    <style>

        body {

            height: 2000px; /\* Just to create a scrollable page \*/

        }

        #s1 {

            position: fixed;

            top: 20px;

            left: 20px;

            font-size: 24px;

            margin-top: 50px;

            transition: color 0.3s; /\* Smooth transition for color change \*/

        }

    </style>

</head>

<body>

    <div id="s1">Scroll to change my color</div>

    <script>

        let s1 = document.getElementById('s1');

        console.log(s1);

        window.addEventListener('scroll', function() {

            let scrollPosition = window.scrollY;

            // Define multiple ranges for color change

            if (scrollPosition >= 200 && scrollPosition < 400) {

                s1.style.color = 'blue';

            } else if (scrollPosition >= 400 && scrollPosition < 600) {

                s1.style.color = 'green';

            } else if (scrollPosition >= 600 && scrollPosition < 800) {

                s1.style.color = 'orange';

            } else if (scrollPosition >= 800 && scrollPosition < 1000) {

                s1.style.color = 'purple';

            } else {

                s1.style.color = ''; // Reset the color or apply another style if needed

            }

        });

    </script>

</body>

</html>