* Tiếp theo, sau khi hiệu ứng thực hiên xong phải remove class active của slide hiện tại và thêm active vào slide kế tiếp (bổ sung code sau)

var ketThucAnSlide = function(){

            console.log('slide hiện tại đã ẩn rồi');

            this.classList.remove('active');

        }

        pthientai.addEventListener('webkitAnimationEnd',ketThucAnSlide);

        var ketThucHienSlide = function(){

            console.log('slide tiếp theo đã hiện ra');

            this.classList.add('active');

        }

=>kiểm tra Element xem đối tượng đã remove hoặc add class chưa?

Hoàn thiện hiệu ứng khi click nút Next (remove class active, class anSlide cho slide hiện tại, Thêm Class active cho slide kế tiếp đồng thời remove class hienSlide của slide kế tiếp)

var ketThucAnSlide = function(){

            console.log('slide hiện tại đã ẩn rồi');

            this.classList.remove('active');

            this.classList.remove('anslideNext');

        }

        pthientai.addEventListener('webkitAnimationEnd',ketThucAnSlide);

        var ketThucHienSlide = function(){

            console.log('slide tiếp theo đã hiện ra');

            this.classList.add('active');

            this.classList.remove('hienslideNext');

        }

        pttieptheo.addEventListener('webkitAnimationEnd',ketThucHienSlide);

        pthientai.classList.add('anslideNext');

        pttieptheo.classList.add('hienslideNext');

= = = = >>>>>> ***Hoàn thành hiệu ứng chuyển slide cho nút Next*** <<<<<< = = = =

Xử lý trường hợp khi người dùng click nhiều lần liên tiếp (sẽ có 1 slide nào đó ko hiện ra, và có thể bị giựt màn hình)

(lý do xảy ra trường hợp trên: khi người dùng tác động thì hiệu ứng sẽ được gọi thực hiện, trong khi hiệu ứng đang gọi chưa thực hiện xong, người dùng click tiếp , do đó sẽ thực hiện add class liên tục và chương trình sẽ xử lý ko kịp, để khắc phục: trong lúc đang thực hiện (hiệu ứng chưa xong) sẽ cấm ko cho làm gì hết (người dùng click), bằng cách thêm return false => muốn vậy phải nắm bắt được hiệu ứng đang thực hiện (*sử dụng biến trung gian để nắm bắt hiệu ứng đang thực hiện*) mà người dùng tác động, ta cho chương trình dừng luôn (return false), khi nào kết thúc hiệu ứng thì mới nhận sự kiện (click)

 var trangthai='TamDung'; //hiệu ứng đang tạm dừng

   var moveslideRight = function(){

    console.log('đã click');

       //Kiểm tra trạng thái hiệu ứng đang hoạt động

       if(trangthai=='chuyendong'){return false}

//Nếu hiệu ứng đang hoạt động thì dừng không làm gì hết

       trangthai='chuyendong';

        var pthientai=slides[chisohientai];

        ...

   }

Giải thích: sử dụng biến trạng thái để biết được hiệu ứng đang chuyển động, khi chuyển động xong phải gán trangthai về TamDung,

Chuyển động xong : có nghĩa tất cả các hiệu ứng (ẩn và hiện slide) phải hoàn thành xong

if(trangthai=='chuyendong'){return false}

//Nếu hiệu ứng đang hoạt động thì dừng không làm gì hết

       trangthai='chuyendong';

       var Trangthaicua2cd=0; //sử dụng để đếm số lần của 2 chuyển động

        var pthientai=slides[chisohientai];

         ...

        var ketThucAnSlide = function(){

            ...

            this.classList.remove('anslideNext');

            Trangthaicua2cd ++;

            if(Trangthaicua2cd==2){trangthai='TamDung';}

        }

        pthientai.addEventListener('webkitAnimationEnd',ketThucAnSlide);

        var ketThucHienSlide = function(){

            ...

            this.classList.remove('hienslideNext');

            Trangthaicua2cd ++;

            if(Trangthaicua2cd==2){trangthai='TamDung';}

        }

Ok xong.