1.DOMContentLoaded

Để phòng việc chưa load xong css mà đã load js trước, dẫn tới việc animation tạo bởi js sẽ gây giật lag cho web. Ta dùng DOMContentLoaded để bảo đảm rằng web sẽ load xong css rồi mới load js

document.addEventListener("DOMContentLoaded",function(){

},false)

2.Hiển thị bảng console để check lỗi

Việc check lỗi bằng console xảy ra rất thường xuyên, thế nên dùng console.log hoặc alert để báo lỗi (debug).

    console.log();

    alert('Nội dung thông báo đặt ở đây');

3.Chọn phần tử DOM HTML

Để tương tác được với phần tử HTML thì phải dùng js để lấy được phần tử đó trong js. Có 4 cách lấy :

// lấy bằng ID, trả về 1 phần tử DOM

var x = document.getElementById('nav-top');

// Lấy bằng classname, trả về nhiều phần tử

var y = document.getElementsByClassName('video-items');

// lấy bằng code css, tức là css của phần tử đó như thế nào thì nhập như thể ấy

var z = document.querySelector('.top-nav form');

// Tương tự như querySelector, nhưng lấy nhiều phần tử

var o = document.querySelectorAll('.col-lg-12');

Lưu ý:

Sau khi lấy DOM nên dùng console.log() để hiển thị phần tử đó ra, để chắc chắn là code đúng

var x = document.getElementById('nav-top');

console.log(x);

4.Lấy nd hoặc gán nd cho 1 thẻ HTML

(Không khuyến khích dùng)

// lấy nội dung

var x = document.getElementById('nav-top').innerHTML;

// gán nội dung

document.getElementsByClassName('video-items').innerHTML = 'TieuDe';

5.Quy định đặt tên biến trong js

Bắt đầu bằng ký tự hoặc kí tự gạch dưới \_ (không được bắt đầu bằng số). Không chứa các kí tự đặc biệt như & \* ( , ) . **Tên biến** không được trùng với từ khóa như var let for if,...

Các tên biến dài sẽ viết liên tiếp nhau, chữ đầu tiên viết thường hết, các chữ sau viết hoa chữ cái đầu

// tên biến

var bienNayDungDeLuuGiaTriCuaA;

Tên hằng số được khai báo với từ khóa const, tất cả các chữ cái viết in, cách nhau bằng dấu ‘\_’

// tên hằng số

const MAU\_TOi\_DA = 100;

Nên đặt tên biến bằng Tiếng Anh

6.Thêm bớt classname cho một phần tử HTML

Các bước tạo animation cho 1 thẻ HTML

Bước 1: Viết HTML

    <div *class*="container">

        <button *class*="btn1">Mo menu</button>

        <button *class*="btn2"> tat menuy</button>

        <div *class*="menu">

            Lorem ipsum dolor, sit amet consectetur adipisicing elit. Minus, nihil aut molestiae mollitia hic, dolorem odio id aliquid officia laboriosam architecto blanditiis aspernatur nulla eaque ducimus libero accusamus cum quae!

        </div>

    </div>

Bước 2: Trong css, tạo cho phần tử cần gắn animation 1 class tên là active

*.container*{

    position: relative;

    height: 100%;

    width: 100%;

}

*.menu*{

    height: 100vh;

    width: 300px;

    background-color: red;

    position: absolute;

    transform: translateX(-400px);

    transition: .25s;

    color: white;

}

*.menu.active*{

    transform: translateX(0);

}

Lưu ý: .menu.active -> tức là bình thường thẻ div có class=’menu’ chỉ nhận code css của .menu, nhưng nếu có thêm 1 class active nữa thì code .menu.active sẽ được nhận thêm (ghi đè)

Bước 3: Thêm bớt class active cho thẻ div class=’menu’ bằng js

        //Bước 1 : lấy DOM của các phần tử dùng trong animation

        var btn1 = document.querySelector('button.btn1');

        var btn2 = document.querySelector('button.btn2');

        var menu = document.querySelector('.menu');

        // kiểm tra xem lấy được chưa

        console.log(btn1);

        console.log(btn2);

        // khi ấn vào nút thứ nhất thì thêm class active

        btn1.addEventListener('click',function(){

            menu.classList.add('active');

        })

        // khi ấn vào nút thứ hai thì bỏ class active đó đi

        btn2.addEventListener('click',function(){

            menu.classList.remove('active');

        })

7.Ba cách gọi function trong js

7.1.phantuDOM.TenSuKien = function(){}

    phantuDOM.onclick = function(){

// code ...

    }

7.2.addEventListener(‘tên sự kiện bỏ chữ on’, function(){})

phantuDOM.addEventListener('click',function(){

    // code

})

7.3.Arrow Function (hàm mũi tên) dùng trong code ES6

Function sau này có một các viết mới là dùng hàm mũi tên (arrow function), dùng khi hàm đó không cần đặt tên (hàm nặc danh) vì chỉ dùng 1 lần.

var tenFunction = (*thamSo*) => {}

ta còn có thể viết như thế này

phantuDOM.addEventListener('click',() => {})

7.4.Sự kiện trong js

Đã lược lại những event cần biết

| **STT** | **Event Name** | **Description** |
| --- | --- | --- |
| 1 | onclick | Xảy ra khi click vào thẻ HTML |
| 2 | ondbclick | Xảy ra khi double click vào thẻ HTML |
| 3 | onchange | Xảy ra khi giá trị (value) của thẻ HTML đổi. Thường dùng trong các  đối thẻ form input |
| 4 | onmouseover | Xảy ra khi con trỏ chuột bắt đầu đi vào thẻ HTML |
| 5 | onmouseout | Xảy ra khi con trỏ chuột bắt đầu rời khỏi thẻ HTML |
| 8 | onmousemove | Xảy ra khi con chuột di chuyển bên trong thẻ HTML |
| 9 | onkeydown | Xảy ra khi gõ một phím bất kì vào ô input |
| 10 | onload | Sảy ra khi thẻ HTML bắt đầu chạy, nó giống như [hàm khởi tạo](https://freetuts.net/ham-khoi-tao-va-ham-huy-trong-lap-trinh-huong-doi-tuong-php-36.html" \t "_blank" \o "hàm khởi tạo) trong  [lập trình hướng đối tượng](https://freetuts.net/lap-trinh-huong-doi-tuong-la-gi-27.html" \t "_blank" \o "lập trình hướng đối tượng) vậy đó. |
| 11 | onkeyup | Xảy ra khi bạn gõ phím nhưng **lúc bạn nhã phím ra sẽ được kích**  **hoạt** |
| 12 | onkeypress | Xảy ra khi bạn nhấn môt phím vào ô input |
| 14 | onblur | Xảy ra khi con trỏ chuột rời khỏi ô input |
| 15 | oncopy | Xảy ra khi bạn copy nội dung của thẻ |
| 16 | oncut | Xảy ra khi bạn cắt nội dung của thẻ |
| 17 | onpaste | Xảy ra khi bạn dán nội dung vào thẻ |

8.Bài tập thực hành

Dưới đây là bài tham khảo, xem qua để biết thêm, nếu được thì code thử theo nha, có gì không hiểu hỏi qua Zalo :))

8.1.

<!DOCTYPE *html*>

<html *lang*="en"><head>

    <title>Hiệu ứng 3D</title>

    <meta *charset*="utf-8">

    <meta *name*="viewport" *content*="width=device-width, initial-scale=1">

    <script *type*="text/javascript" *src*="vendor/bootstrap.js"></script>

    <link *rel*="stylesheet" *href*="vendor/bootstrap.css">

    <link *rel*="stylesheet" *href*="5.css">

    <link *rel*="stylesheet" *href*="fontawesome/css/fontawesome.css">

    <link *rel*="stylesheet" *href*="fontawesome/css/brands.css">

    <link *rel*="stylesheet" *href*="fontawesome/css/solid.css">

</head>

<body >

    <div *class*="pusheffect">

        <div *class*="nentrang">

            <button *class*="btn btn-success nut">Push</button>

            <button *class*="btn btn-warning nut1">Open Door</button>

        </div>

        <!-- Sử dùng nền đen để ẩn menu đi -->

        <div *class*="nenden"></div>

        <div *class*="menutrai">

            <ul *class*="list-group">

                <h2>NAVBAR MENU</h2>

                <li *class*="list-group-item">Cras justo odio</li>

                <li *class*="list-group-item">Dapibus ac facilisis in</li>

                <li *class*="list-group-item">Morbi leo risus</li>

            </ul>

        </div>

    </div> <!-- hết pusheffect -->

    <script *type*="text/javascript" *src*="5.js"></script>

</body>

</html>

8.2.

\*{margin: 0; padding: 0}

*.menutrai* h2 {

    padding: 17px;

    text-align: center;

}

html,body{

    width: 100%;

    height: 100%;

    /\*background:transparent;\*/

        overflow-x: hidden;

    /\*overflow-y:hidden;\*/

}

*.pusheffect*{

    position: relative;

    width: 1920px;

    height: 100%;

    perspective: 3200px;

}

*.xamlai*{

    background: gainsboro;

}

*.nentrang*{

    position: absolute;

    z-index: 2;

    height: 100%;

    width: 100%;

    transition: 0.4s;

}

*.sangphai*{

    transform: translateX(300px);

}

*.pusheffect* *.nut*{

    position: absolute;

    top: 50%;

    left: 600px;

    font-size: 30px;

}

*.nut1*{

    position: absolute;

    left: 700px;

    top: 50%;

    font-size: 30px;

}

*.nenden*{

    position: fixed;

    z-index: 3;

    background: #00000057;

    height: 100%;

    width: 100%;

    cursor: pointer;

    transition: 0.4s;

    opacity: 0;

    visibility: hidden;

}

*.ra*{

    opacity: 1;

    visibility: visible;

}

*.menutrai*{

    position: fixed;

    top: 0;

    left:0;

    z-index: 3;

    height: 100%;

    width: 300px;

    background: #5cb85c;

    color: white;

    transition:0.4s;

    transform: translateX(-100%);

}

*.tutraisang*{

    transform: translateX(0);

}

*.menutrai* *.list-group-item*{

    background: #5cb85c;

    font-size: 17px;

    color: white;

    /\*border: none;\*/

}

*.opendoor*{

    transform-origin: 100% 0%;

    transform: rotateY(-25deg);

    background: white;

}

8.3.

document.addEventListener("DOMContentLoaded",function(){

    var nut = document.querySelector('.nut');

    var nenden = document.querySelector('.nenden');

    var menutrai = document.querySelector('.menutrai');

    var nentrang = document.querySelector('.nentrang');

    var nut1 = document.querySelector('.nut1');

    var khung = document.querySelector('.pusheffect');

    // console.log(nut1);

    //hiệu ứng khi click vào nút PUSH

    nut.onclick = function(){

        nenden.classList.add("ra");

        menutrai.classList.add("tutraisang");

        nentrang.classList.add("sangphai");

    }

    //trả về như cũ khi nhấn vào nền đen

    nenden.onclick = function(){

        nenden.classList.remove("ra");

        menutrai.classList.remove("tutraisang");

        nentrang.classList.remove("sangphai");

                nentrang.classList.remove("opendoor");

        khung.classList.remove("xamlai");

    }

    nut1.onclick = function(){

        menutrai.classList.add("tutraisang");

        nenden.classList.add("ra");

        nentrang.classList.add("opendoor");

        khung.classList.add("xamlai");

    }

},false)