HTML DOM

Method: action you can do (like add or deleting an element html)

Property: value you can get or set (like changing content of an html element)

1. Finding HTML Elements

getElementById(id)

-> return only one element

getElementsByTagName(name) (tìm kiếm phần từ theo tag)

-> return list of all elements

getElementsByClassName(classs)

-> return list of all element

querySelectorAll() (tìm kiếm các element theo bộ chọn Css)

-> return list of all element

1. Changing HTML element

element.innerHTML = new html content -> Thay đổi nôi dung của element

element.attribute = new value -> thay đổi thuộc tính của 1 phần tử

element.style.property = new style -> thay đổi style của 1 element

element.setAttribute(attribute, value)

1. events:

an element is clicked on

the page has loaded

input fields are changed

* events

onload

onchange

onmouseover: khi di chuột vào vùng chọn

onmouseout: khi di chuột ra ngoài vùng chọn

onmousedown:

onmouseup:

onclick:

onfocus: Khi click chuột vào element

resize: khi thay đổi kích thước của window

* event listener

element.addEventListener(“click”, function)

removeEventListener()

1. Naviagtion

ParentNode

childNodes[nodenumber]

firstChild

lastChild

nextSibling

previousSibling

nodeValue -> lấy giá trị

nodeName -> tên của tag

nodeType -> trả về kiểu của 1 nút

createTextNode(“”)

insertBefore()

remove()

removeChild(child)

replaceChild()

1. Thông báo

alert()

1. Adding and Deleting Element

document.createElement(element) (tạo 1 element )

document.createElement(“tên thẻ”)

document.removeChild(element) ( xóa 1 element)

document.appendChild(element) (thêm 1 element )

document.replaceChild(new, old) (thay thế 1 element)

document.write(text)

1. Animations

* Interval: số lần thực hiện funcion là mãi mãi cho tới khi sử dụng hàm clearInterval

var interval = setInterval(function, time)

clearInterval(interval)

* Timeout: thực hiện hàm timeout chỉ 1 lần

var timeout = setTimeout(function, time)

clearTimeout(timeout)

Tạo một nút mới trong một nút đã có



var span = document.createElement("span");

Thêm tên class: span.classList.add("pg-bullet"); (tên class của thẻ)

var pagi = idparentbullets.querySelector('div.pagination');

pagi.appendChild(span);

Xóa một class: element.classList.remove("mystyle");

Active Class

<div id="myDIV">

<button class="btn btn1">1</button>

<button class="btn btn2">2</button>

<button class="btn btn3">3</button>

<button class="btn btn4">4</button>

<button class="btn btn5">5</button>

</div>

for (var i = 0; i < btns.length; i++) {

btns[i].addEventListener("click", function() {

var current = document.getElementsByClassName("active");

current[0].className = current[0].className.replace(" active", "");

this.className += " active";

});

}

1. next.addEventListener('click',nextPos, this);

prev.addEventListener('click',prevPos, this);

function nextPos(ctrl){

click++;

changeLeft(ctrl);

check();

bgBullets();

}

function changeLeft(ctrl){

// console.log(ctrl);

if(click\*pos-(wli\*nli-wd) > 0 ){

ctrl.target.parentElement.querySelector('.jsListGroup').style.left = String(wd-wli\*nli) + 'px';

}else{

ctrl.target.parentElement.querySelector('.jsListGroup').style.left = String(-click\*pos) + 'px';

}

}

3.

id.querySelector(".content-footer").addEventListener("click",function(e){

let old=bt;

let index=parseInt(e.target.getAttribute("data-index"));

console.log(index,index!=NaN);

if(typeof(index)=='number' && index>=0){

console.log(old,index,'ab',e.target.getAttribute("data-index"));

console.log(e)

//return false;

bt=index;

if (bt<2){

sta=a\*wili\*bt

} else{ sta=(nli\*wili -wi) }

idlist.style.transform='translate3d(-'+(sta)+'px, 0px, 0px)';

// Color();

Color(old,index);

}

});

function Color(old,current){

/\*console.log(old,current,footer\_button);

footer\_button[old].style.background= "#000";

footer\_button[old].style.opacity="0.2";

footer\_button[current].style.background= "rgba(251,108,39,1)";

footer\_button[current].style.opacity="1";

\*/

}