1. classList

a. chỉ làm việc với element node

b. element.classList.lenght 🡪 số lượng phần tử trong list class

b1. Nếu tồn tại các tên class trùng nhau thì nó tính các class trùng nhau là một

c. lấy ra một phần tử trong classdList

c1. Element.classList[2]🡪 phần tử thứ 2

c. lấy ra dạng chuỗi classList

ex: class = 'box-1 box-2 box-3';

element.classList.value( === 'box-1 box-2 box-3') ( nó tính luôn cả dấu cách)

2. các thuộc tính làm việc với classList

2.1 add

a. thêm class vào classList

b.element.classList.add('box-3','box-4'…v.vv.); 🡪 chỉ thêm vào chứ không ghi đè các phần tử class đã có sẳn

2.2 contains

a. kiểm tra xem class đó có tồn tại trong classList hay không

b. return về boolen

c.element.classList.contains('box-2') (=== true)

2.3 remove

a.xoá phần tử class trong classList

b. element.classList.remove('box-3') ( box-3 bị xoá)

2.4 toggle

a. kiểm tra xem class đó đã có trong classList hay chưa , nếu có thì remove , nếu không thì add nó vào

b. element.classList.toggle('box') ( 🡪 remove box)

3. nextElementSibling

a. thuộc tính này cho phép select element đằng sau liền kề element đã select

b. nếu không có element nào liền sau thì return về null

//////

DOM EVENT

🡪 Attribute events

3.onclick

a.Bên trong vùng hoạt động attribute được sử dụng this để lấy ra element đang được click vào

b. đặt onclick ở thẻ mẹ, click vào vùng hoạt động của thẻ con thì

b1. Ăn vào onclick thẻ con

b2. Ăn ra onclick thẻ mẹ

🡪 hành vi sự kiện nỗi bọt

🡪 assigned even using the element node ( sử dụng file thay vì inline)

4. onclick

a. dùng document lấy tới element

( element này là một node list )

b.element.onclick = function(e){

console.log(e.target); ( lấy ra được element đã click vào)

}

B1. trong function , hỗ trợ một tham số e ( mouse Event, dùng mouseEvent.target 🡪 xuất ra element vừa click)

DOM EVENT INPUT (

🡪 thanh input text

5. onchange: ( lắng nghe sự thay đỗi value của thẻ input, khác chỗ mỗi loại thẻ input có cách cập nhật khác nhau)

a.element.onchange = function(e){

console.log(e.target.value); ( e.target🡪 lấy được element , từ element.value🡪 lấy được giá trị element)

}

b. lấy những thông tin được cập nhật trong input khi , mỗi lần click ra ngoài lại cập nhật một lần

c. return về string

6. oninput

a. giống hệt cấu trúc onchange

b. khác onchange: khi nội dung trong ô input khác thì nó tự cập nhật

🡪 checkbox click

a. dùng onchange

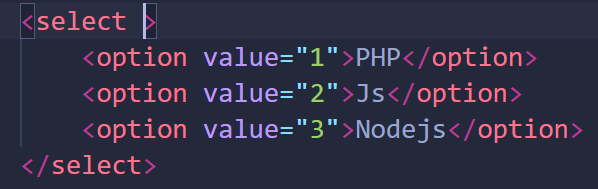
b. e.target.checked

b1. Return true ( clicked)

b2. Return false ( non-clicke)

🡪 check value Option

Kể cấu input Option có dạng:



B. dùng onchange :

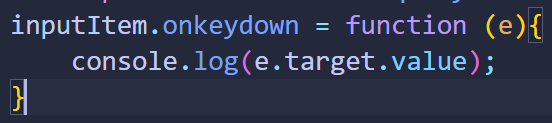
Element.onchange = function(e){

Console.log(e.target.value) ( cái value này là cái value trong thẻ value của option)

}

8. onkeydown && onkeyup

🡪 onkeydown: ( nhấn xuống , phím bật lên 🡪 mới nhận một giá trị)



a. hiển thị trên console phím vừa nhập với độ trể 1 phím ( nhập abc🡪 ab, abcd--. Abc)

🡪 onkeyup: (nhấn xuống 🡪 nhận giá trị)

a. giống hệt cấu trúc onkeydown nhưng mà không có độ trể phím

8.1 .which

a. element.which ( lấy ra mã id của phím vừa nhập)

b. ex: esc có mã là một number : 27, …v.v.v.

🡪 onkeypress: ( nhấn xuống 🡪 nhận giá trị|| nhấn giữ xuống 🡪 spam liên tục nhận giá trị đó)

Cấu trú I hệt hai ông trên , nhưng không nhận phím ESC

///////

PREVENTDEFAULT && STOPPROPAGATION:

Prevendefault : bỏ các thuộc tính mặc định của trình duyệt

stopPropagation : bỏ hiệu ứng sự kiện nỗi bọt

9. HTML selection

Document.anchors 🡪 lấy ra các thẻ a có attribute( name:'tên');

Document.links🡪 lấy ra toàn bộ thẻ a

10/ prevendefault( ngăn chặn thuộc tính click vào thẻ a là chuyển dến link)



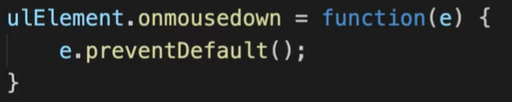
a/startsWith('string') 🡪 kiểm tra xem string truy cập đến có bắt đầu bằng cái này hay không

a1/ return về boonlen

b.preventDefault()

a1. Event.preventDefault() 🡪 ngăn chặn sự kiện Event sảy ra ( sự kiên e dược tham chiếu ở trên là sự kiện click vào thẻ a)

11. onmousedown ( event click chuột vào )



Ngăn chặn sự kiện (e) vừa click chuột vào ulElement

12. ngăn chặn hiện tượng sự kiện nỗi bọt

a.trong function sự kiện của thẻ con 🡪 e. stopPropagation();

//////////

EVENT LISTENER

13. setTimeOut (function(){}, thời gian ms);

14. các onclick.. onmousedown..v.v. cũng là các key trong object

Vì vậy cũng có thể gán rỗng nó lại

Ex:

Var btn = document.querySelector('button');

btn.onclick = function(){

console.log('cong viec 1')

}

Thì cái onclick cũng là một key ( thuộc tính của object)

Tiếp sau đó nếu có một đoạn mã

btn.onclick = function(){} 🡪 đoạn function trống

🡪trình tự thực hiện : thực hiện đoạn mã 1 🡪 in ra 'cong viec 1' 🡪 click lại 🡪 ko còn gì sảy ra nữa vì lúc sau đã bị ghi đè

15. addEventListener

a. dùng khi có nhiều sự kiện , và muốn remove sự kiện nào đó

b. addEventListener(click',fucntion);

b.removeEvenListener('click',function) 