**JS Basic**

**1.Arrrays**

* Variable ‘let’ có thể gán lại 1 giá trị khác
* Variable ‘const’ không thể gán lại bởi vì nó là constant
* Phương thức: ‘Push()’ là phương thức sử dụng trong mảng Arrays để them phần tử vào cuối mảng
* Phương thức: ‘Pop()’ là phương thức xóa 1 phần tử cuối cùng của mảng đó và trả về giá trị của phần tử đó.
* Phương thức : ‘Shift()’ loại bỏ phần tử đầu tiên trong mảng và trả về giá trị của phần tử đó.
* Phương thức : ‘UnShift()’ thêm nhiều phần tử vào vị trí đầu tiên của mảng và trả về giá trị của phần tử đó.
* Phương thức: ‘Splice()’ xóa 1 vị trí cụ thể ở trong mảng.  
  splice(index1,index2) là sẽ xóa từ vị trí index1, số bước của index2 hoặc là thay thế nó bằng 1 giá trị ‘a’ spilce(index1,index2,’a’).
* Phương thức concat nối nhiều mảng lại với nhau
* Phương thức: slice() tạo 1 bản sao của 1 chuỗi hoặc chọn 1 phần tử của mảng(từ phần tử thứ 0)
* Phương thức: split() cắt đi những gì được truyền vào mà nó có trong chuỗi
* Reduce(): hàm này được sử dụng lên cách phẩn tử của mảng và thu về 1 biến tích lũy thu về 1 giá trị duy nhất. Cú Pháp: array.reduce(callback,initialValue). Callback là hàm thực thi trên từng phần tử của mảng, hàm callback này nhận 2 tham số là: tham số 1: biến tích luy được chả về sau mỗi lần callback, và tham số thứ 2 là :phần tử của mảng đó đang được xử lý. **initialValue** là giá trị ban đầu của biến tích lũy. Nếu không được chỉ định, biến tích lũy sẽ được khởi tạo bằng phần tử đầu tiên của mảng.

**2.Number**

­ - Ta có thể dung phương thức Number() để ép kiểu 1 chuỗi về 1 số

**3. Template Strings**

- Thay Vì Ta Nối Chuỗi console.log(`abc`+ (a+b)) thì ta thay thế nó bằng console.log(`abc ${a+b} abc`)

**4.DOM(Document Object Model)**

-Có những cách để select vào DOM như:

+)document.getElementById():Sẽ chuyền vào 1 ID và nó trả về 1 element.

+)document.getElementsByClassName():Sẽ chuyền vào 1 Class Name và nó trả về 1 HTML Collection và nó hoạt động như 1 mảng.

+)document.getElementsByTagName():Sẽ chuyền vào 1 Tag Name và nó trả về 1 HTML Collection và nó hoạt động như 1 mảng.

+)document.querySelector():Đây còn được gọi là CSS selector : sẽ chuyền vào syntax như css và nó trả về 1 element.

+)document.querySelectorAll():Đây cũng được gọi là CSS selector: sẽ chuyền vào syntax như css và nó trả về 1 NodeList và nó hoạt động như 1 mảng.

\*Chú ý:Nếu mà cái nào chả về 1 HTML Collection hoặc trả về 1 NodeList , nếu ta muốn them các thuộc tính như innerHTML thì ta phải chỉ rõ và access rõ đến phần tử đó trong mảng.

+)document.forms: sẽ trả về 1 HTML Collectuon và nó action như 1 mảng

-Những các add attributes vào elements:

+)add id: element.id=’’

+) add title: element.title=’’

+) add class: element.className=’’

\*Chú ý: Nếu cái element đó truyền vào các attributes không hợp lệ VD: element đó là của thẻ <h1></h1> thì ta sẽ không thể gán vào attribute là href=’’ được vì vậy nếu muốn gán 1 attribute không hợp lệ thì ta sẽ dùng setAttribute(parameter1,parameter2) paramenter1:Ta sẽ chuyền vào tên attribute và parameter2:Ta sẽ chuyền vào giá trị của nó.

-getAttribute(parameter):sẽ chuyền vào tên 1 attribute lấy cái value của attribute đó ra.

-removeAttribute(parameter):sẽ chuyền vào attribute đó để tiến hành remove attribute

-InnerHTML :thực hiện them 1 element vào trong 1 element được chỉ định

-outerHTML: thực hiện ghi đè luôn cái element được chỉ định

**DOM Events**

**1.Attribute events**

-onclick=bấm chuột vào

-onmouseover = hover vào 1 element

-onmouseout= di chuột ra 1 element

-onchange = nó chỉ say ra khi chúng ta thay đổi giá trị khác giá trị ban đầu của 1 input và sau đó click chuột ra ngoài.

-oninput= lấy liên lục giá trị khi input vào

-onkeyup: là bấm xong nhả lên

-onkeydown:là bấm xuống và dùng event.which để biết đc đang bấm nút nào

-onkeypress:giữ xuống

-preventDefault:bỏ hành vi mặc định của thẻ

-stopPropagation:ngăn chặn sự kiện nổi bọt

**2.Event listener**

**-Eventlistener có thể thực hiện nhiêu công việc được 1 lúc**

-Element.addEventListener(‘sự kiên’,func())

-RemoveEventListener(‘sự kiện’,func())

**JSON(Javascript Object Notation)**

-Stringify:Convert từ JavaScript ra JSON

-Parse:Convert từ JSON ra javascript

-cú pháp khởi tạo 1 JSON là 1 dạng chuỗi VD: let a=’ ”abc” ’ console.log(JSON.parse(a))=> sẽ ra kiểu dữ liệu Javascipt là 1 string

**HTTPRequest**

-Dùng Để Get data từ 1 API

-Khai báo let request =new XMLHttpRequest();

-request.status:trạng thái nếu là 200 thì là thành công, 404 k thấy

-request.open(“phương thức lấy”,”API”,”async”,’user’,’password’)

-request.send()

**CallBack**

-Sẽ được gọi sau khi những thằng ở trên được thực hiện.

**Async Await**

-Promise pending: là trạng thái đang chờ để được thực hiện 1 công việc.

-Promise fulfilled: thì là trạng thái đó đã hoàn thành.

-Fetch(‘API’):là 1 promise => sau khi Fetch(‘API’).then(response=>{})=>thì nó là 1 response

Và cái response này cần được JSON biên dịch lại để biến thành data của chúng ta.

-Khi định nghĩa async ở đầu func thì sẽ thực hiện biến nó thành 1 promises.

-Async: được đặt ở đầu hàm để định nghĩa đó là hàm bất đồng bộ

-Await: được đặt trước 1 promise để định nghĩa rằng ta mong chờ kết quả của nó.

**Destructuring Assignment**

-Ví dụ ta có 1 mảng let a=[1,2,3,4] và ta muốn copy mảng đó và gán vào 1 mảng mới và them được phần tử vào mảng đó ta sẽ dùng let b=[7,…a,9] . Dấu 3 chấm là thể hiện sự copy từ mảng a tương tự như vậy với object là nếu nó là object thì sẽ dùng {} thay vì [] như mảng.

- //đây là cách select nhanh các thuộc tính trong 1 object mà không cần phải select trực tiếp từng thuộc tính một và những key được khai báo

        //đại diện cho từng thuộc tính trong person phải trùng với key của thằng person , sai key thì sẽ không select được.

        let {name,age,major}=person

        console.log(name,age,major)