**1 .Websever:**

Web sever apche, nginx , nodejs, microsoft.

**Host & Port**

**Host:** Máy tính có 1 địa chỉ Ip cố định của mạng bạn đang ở. Địa chỉ ip của máy tính vật lý đấy. Chạy nhiều trang web trên cùng 1 máy tính vật lí nào đấy( web sever).

**Port:** Một máy tính có số lương cổng nhất định sever có mỗi cổng khác nhau.

* Cài đặt module để tạo ra một sever

npm -g install static-server

* Để tạo sever: static-server.
* Để đổi port: static-server –p 12312.

**2 .Network:**

Địa chỉ ipaddress: là địa chỉ cục router này: mạng lan

Máy tính, điện thoại được router cấp cho.

Hai máy khác mạng lan:

* Truy cập thông qua địa chỉ ip

**3 .HTML:**

Seo:

Tối ưu hóa cho website:

1.Tìm hiểu cách lên top của bộ máy tìm kiếm Google

1.1: Có 2 cách : SAM( sử dụng dịch vụ quảng cáo của google adwords ) và Seo ( lên tóp một cách tự nhiên )

SAM: Làm cho website lên tóp ngay lập tức, mà chúng ta sẽ phải mấy phí.

SEO: làm cho website lên top mất nhiều thời gian, không mất phí.

2.Tìm hiểu về SEO:

a) Thông báo cho công cụ tìm kiếm xem website bạn đang thể hiện nội dung gì?

<title>, <meta name=”keywords”> và <meta name=”description”>: Được đặt ở thẻ <head>

\*Ví dụ: Website thời trang:

Trang chủ:

<title>Shop thời trang</title> //thẻ tiêu đề.

<meta name=”keywords” content=”áo thun nam, áo thun nữ, quần nam, quần nữ, ...”/> // nó là từ khóa google ,người dùng tìm kiếm.

<meta name=”description” content=”Shop thời trang 2S là cửa hàng thời trang chuyên kinh doanh các mặt hàng abc, xyz chất lượng...”/> //mô tả.

Trang chi tiết sản phẩm: sẽ viết khác.

b) Cách viết keywords khác nhau ở từng trang:

Cắt ghép các file giao diện

c) Phân biệt keywords chính và keywords phụ:

- Từ khóa nào mô tả chung là từ khóa chính: khó seo khó lên top;

- Từ khóa phụ: dễ seo dễ lên tóp hơn;

3.Cách bộ máy tìm kiếm đọc website của bạn

- Đọc nội dung ở những thẻ <title> này trước // từ khóa nào muốn seo thì viết ngắn gọn vào thẻ title

- H1 , H2,H….// nhấn mạnh từ khóa muốn seo nên viết trong thẻ,

- Thẻ <a> : không đọc nội dung, nó đọc trong thuộc tính title

-Thẻ <img>: không đọc nội dung , nó đọc trong thuộc tính alt

**4 .Javascript DOM on Browser;**

**5.Document Object:**

Document: là phần tử root

Giữa HTML và Javascript có liên kết thông qua DOM.

- Properties:

+ document.body:

+ document.head:

* Methods:

**getElementsByClassName:**

* Thao tác với class của html , có kiểu là string , trả về 1 HTML collectionOF ,tập hợp nhiều phần tử HTML.
* Giống với mảng nó cũng là một Objet cũng có length.
* Truy xuất vào từng phần tử như mảng.
* Khác với mảng , Html colectionof bản chất của nó không phải là một array nó chỉ là một dạng gần giống array. Không thể sử dụng method của array.
* Có thể chạy vòng lặp for để lặp phần tử.
* Muốn sử dụng method , function giành riêng cho array phải ép kiểu về array ,dùng Array.from(‘nameColectionof’).

**getElementsByID:** Thao tác js với id của html , có kiểu dữ liệu là string trả lại 1 HTML element. , 1 phần tử HTML.

**getElementsByTagName:** Truy xuất vào tên của thẻ vidu a, p , div… cũng là một Html clolectionOf.

**querySelector:** Trả về một phần tử đầu tiên tìm thấy được mà match với selector.

* **Ví dụ:** querySelector(‘#demo’);

**querySelectorAll:** Trả về nhiều phần tử match với selector , là một notdeList giốn html collectionof. **getElementsByClassName.**

**document.getElementById("myImg").src:** trả về đường dẫn của hình

Hoặc: var img = new **Image**();

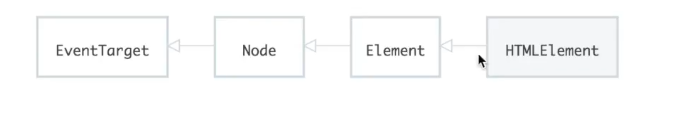
img.src = "path/to/image.jpg"

**6.HTML Element class**

Constructor function khi chạy web nó tự định nghĩa

Khi chạy trên trinh duyệt nó những class HTMLelement , Node thì không có.

**instanceof :** kiểm tra xem thành phần này có phải kế thừa thành phần khác không.



**Mẹo:** để chuột vào chỗ cần làm việc $ 0 để chọn nhanh element.

**.Children:** properties của class **Element** mà **HTMLElement** thừa kế vào nên cũng có. Hiển thị element con.

.**textContent:** của **Node , HTMLElement** lại thừa ké của **Element** nên nó cũng có. Hiển thị nội dung ở dạng string.Có thể gán nội dung string cho elemetn.

.**appendChild**(): method của **Node**. Thêm một thằng con vào cuối.

.**remove**(): method của **Element.** Xóa phần tử được chọn.

**Document.createElement():** Tạo HTML Element mới truyền vô tag name.

**7. innerHTML property**

Thay đổi nội dung của một tag , element và chèn **HTMLElement** vào dưới dang text ,cả thẻ bên trong luôn.

**8.onClick:**

Input: có một property là value.

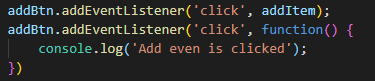
**9.Event listener:**



Onclick đang kí 1 listener cho sự kiện này thôi,

**Cách 2:**

**addEvenListener:** đăng kí nhiều even listener cho 1 sự kiện.



**10. “Change” event.**

Thay đổi giá trị bên trong của nó input ,select.

**11: LocalStorage:**

Bộ nhớ trên trình duyệt như một object , Mỗi domen mỗi host có một localStorage khác nhau.

Giới hạn: LocalStorage: không chia sẻ được.

* Có giới hạn, nó không phải là vô hạn.

Syntax: **localStorage**: kiểm tra xem xem nó có lưu bao nhiêu phần tử

* **localStorage.setItem(‘key’ , ’value’):** Thêm vào bộ nhớ. Tự chuyển đổi sang string.
* **localStorage.getItem(‘key’):** Lấy từ bộ nhớ ra.
* **localStorage.removeItem(‘key’):** Xóa ra khỏi bộ nhứ trình duyệt.

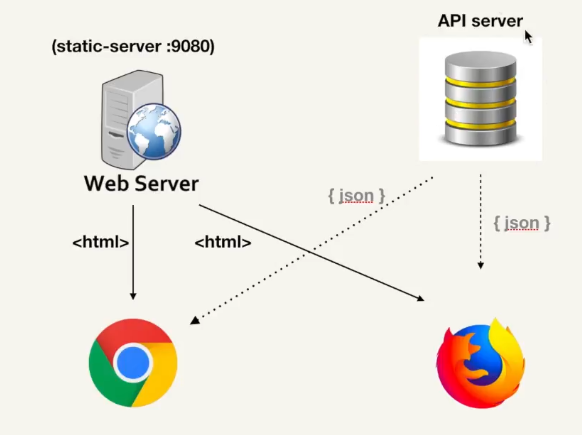
**12:** **sessionStorage:**

Là một phiên ngắn hạn , kết thúc trong một khoảng thời gian nào đấy , đóng trình duyệt thì kết thúc.

**13: AJAX:**

Mở một trang web gửi 1 request lên sever nhận responsive về client.

AJAX tạo những request không nhìn thấy sau khi nhân responsive.



Chuyển hết dữ liệu sang một cái API server: gửi request nhẫn dữ liệu từ API server.

Sử dụng module: json-server.

* Để chạy thì: json-server –watch db.json.

Method Get : lấy thông tin có sẵn trên server.

Method Post: là gửi.

Method Put: là thay thế.

Method Patch: chỉnh sửa.

Method Delete: là xóa

**14: Debug**

**Call stack:** Thực thi một hàm nào đấy, nó sẽ nhét hàm vào call stack la một kiểu cấu trúc dữ liệu.

* Giống như kiểu array nó có thể push khi hàm được thực thi. Nếu hàm này gọi hàm khác nữa thì nó sẽ nhét đè lên trên
* Sau đó hàm thứ hai thực hiện xong rồi thì bỏ ra tới hàm thứ nhât.

**15: Firebase hosting**

Chia sẻ trang web cho mọi người cùng xem.

Cách truyền thống : thuê một host ting của 1 công ty và một domain , chạy ra của hàng bảo người ta mở lên.

Firebase hosting: của google mua lại không cần code backend vẫn làm một trang web.

**16: XSS(Cross-stite scriptin) hacking:**

Dữ liệu nhập vào ô tìm kiếm chưa được lọc tag, kiểm tra dữ liệu nhập vào.

Google xss filter: kiếm thư viện đẻ loại bỏ html tag. Thường được lọc ở server site, hoặc một số dùng ở client.

**17: data-atributes:**

Khai báo attributes có dạng data- ngày thì ngay lập tức javascript dom lưu lại

Mỗi thằng ở trong html Element nó có biến gọi là **dataset** là một object bất cứ những gì khai báo với data- thì nó sẽ được lưu vào dataset.

Giá trị data- luôn là string.

Chúng ta có thể lưu dữ liệu trong html thông qua data-.

**18:Event’s target: target(mục tiêu).**

Event có một property là target

Các hàm có được truyền đâu vào là even đứng đầu danh sách tham số truyền vào, còn nhận hay không là việc của nó.

* **Event( even object ):** Được gọi ở callback , constructor function của nó là mouseEvent . Event là một object của sự kiện này thôi.
* **Event có một properties là target:** Event.target