1.Sử dụng Jquery resize cho reponsive:

- $(window).height();// chiều cao màn hình;

- $(window).resize();//co to nhỏ màn hình thì sẽ bắt sự kiện;

- $(‘body’).animate({scrollTop:$(‘.top2’).offset().top});//cuộn chuột tới vị trí nào đó;

- $(window).scroll();// cho biết bao giờ cuộn chuột;

- $(body).scrollTop();//vi trí của người dùng trong giao diện;

- offset().top;// lấy vị trí của 1 thành phần nào đó: top, left, bottm, right.

- return false; //đã tồn tại không làm gì cả.

//animate là hàm chuyển động;

* [hienthi=’modi’] có thể gọi theo tên tự đặt;

2.Json

1. Nó giúp cho chúng ta trao đổi dữ liệu giữa mảng được viết ở trong PHP với lại mảng được viết ở Jquery ( jquery <=> PHP )

2. Ung dụng lớn hơn có yêu cầu trao đổi dữ liệu giữa website này với website khác (web này <=> web khác);

3. Trao đổi giữa viết web PHP với ngôn ngữ chuyên viết phần mềm như là Java, C# (PHP<=> Java ⬄ C#);

\*Trương hợp 2,3: chúng ta sao lưu (backup) dữ liệu thành một file SQL, rồi sau đó copy cho web khác mà không cần đến json;

Ví dụ: Chia sẻ tiêu đề , hình ảnh, mô tả cho một người khác :

Sử dụng Json là một chuẩn chung để chúng ta có thể trao đổi thông tin, dữ liệu

3.FrontEndFunn javascripts

- outerHeight: lấy chiều cao của phần tử bao gồm padding, border;

- outerHeight(true): lấy chiều cao của phần tử bao gồm cả thêm margin;

- innerHTML : lấy nội dung html; <b>xin chao</b>

- innerText: lấy nội dung text;

- getElement : truy xuất các thành phần con trong thành phần;

Hoặc jquerySelectorAll(‘ div#hello ’);

Event trong Javascript: Gán sự kiện vô thẻ html cần thực hiện

- onClick : hành động click chuột.

- onmouseover= “show( )”: hover vô thẻ thực hiện hàm show;

This : đại diện cho đối tượng hiện tại đang thao tác.

element.addEventListener(‘ tên sư kiện bỏ ‘on’ ’,functionName)”: Thêm sự kiện cho DOM

element.removeEventListener(‘ eventName ’,functionName, option): Xóa bỏ sự kiên.

+ option: Các giá trị true | false ( có thể bỏ trống ).

**\*Các thuộc tính thường dùng trong DOM Document:**

- document.activeElement.tagName : Lấy ra thẻ đang được focus.

- document.anchors.length: Trả về một mảng đối tượng các thẻ a ( không chưa href ).

- document.links.length: Trả về một mảng đối tượng chứa các thẻ a,area có thuộc tính href trong trang.

- document.URL: Trả về đường dẫn của trang hiện tại.

- document.title: Trả về title trong heade của trang.

- document.script.length: Trả về một mảng đối tượng chứa thể script trong trang.

- document.readyState: Trả về trạng thái của trang.

- document.images.length: Trả về một mảng đối tượng chứa các thẻ img trong trang.

**\*Đôi tương window trong javascript**

- documentElement : phần tử gốc của một trang , sẽ trả về đối tượng thẻ <html>

Lấy chiều rộng của trình duyệt:

+ var width = window.innerWidth.

+ document.documentElement.clientWidth.

+ document.body.clientWidht.

Lấy chiều cao cũng tương tự thay width bằng height.

Mở một cửa sổ mới:

- window.open( url,name,specs, replace ):

+url: địa chỉ muốn mởi;

+name: tên bạn muốn đặt cho cửa sổ;

+specs: thông số cho cửa sổ height, width, cách top, cách left;

Đóng một cửa sổ:

- wObject.close():

wObject : là đối tượng mà chúng ta muốn đóng.

Di chuyển trình duyệt:

wObject.moveTo(top,left): di chuyển trình duyệt;

Thay đổi kích thước:

wObject.resizeTo(width,height);

**\* Đối tượng location trong javascript:**

Đối tượng window luôn bao gồm rất nhiều đối tượng con đi kèm theo nó document, location, …

Đối tượng location này chuyên dùng để xử lý, điều hướng url của trang web.

Các thuộc tính trong location:

- window.location.hash: gán hoặc lấy phần sau dấu # của URL.

- window.location.host: gán hoặc lấy hostname và port number của URL.

- window.location.hostname: gán hoặc lấy hostname.

- window.location.href: gán hoặc lấy URL.

- origin: trả về đường dẫn đầy đủ của trang hiện tại.

- pathname: gán hoặc lấy path name ( tên đường dẫn ) của URL.

- port: gán hoặc lấy port của URL.

- search: lấy phần query string của URL.

- window.location.href = “<http://trungkientraning.com>”;

Các phương thức trong location:

Phương thức reload() có tác dụng tải lại trang web, giống như khi bạn ấn F5.

- window.location.reload();

Phương thức replace() có tác dụng như thuộc tính href, nhưng khi sử replace() thì trình duyệt sẽ không lưu vào lịch sử, còn href thì có lưu.

- window.location.replace(url);

Phương thức assign(): nhường lại, người kế quyền, người thừa nhận.

- window.location.assign(url): Không khác gì thuộc tính href.

**\* Cookie trong javascript: DOM là con của BOM,**

Cookie là một file được lưu trữ ở mỗi máy của client. Khi tắt trình duyệt đi thì cookie thời gian sống của nó vân còn

Cookie trong javascript:

Tạo cookie:

- window.document.cookie = ‘option’;

Hoặc

- document.cookie =’option’;

Tạo cookie có tên là name và có giá trị là ‘DamTrungKien’.

- document.cookie =’name=DamTrungKien’;

Tạo cookie kèm với thời gián sống expires và kèm theo path.

- document.cookie = ‘name=DamTrungKien; expires=Thu , 22 Dec 2019 19:55:20 UTC;path=/’;

Đọc cookie

Var data = document.cookie; trả về biến data dưới dạng string và cookie cách nhau bởi một dấu ;.

**\* Đối tượng history trong javascript**

History lưu trữ toàn bộ lịch sử bật trình duyệt đến khi tắt đi.

History trong javascript:

Window.history; hoặc history;

Các thành phần trong history:

- History.length: tổng số trang mà bạn đã duyệt.

- History.back(): Phương thức lưu lại trang web chúng ta vừa truy cập gần nhất.

- History.forward(): di chuyển lại trang vừa rôi, khi chúng ta vừa back();

- History.go(-n): quay về load cách đây n lần;