Cách sử dụng

Cách sử dụng cơ bản, include file base.js, Tech.js qua tag script

Cách sử dụng có thể thay đổi trong tương lai khi load js qua systemjs, commonjs

Cấu trúc thư viện

Thư viện có 2 kiểu dữ liệu là SingleElement là một đối tượng dom riêng lẻ, và ListElement là 1 danh sách các đối tượng SingleElement

Cách sử dụng

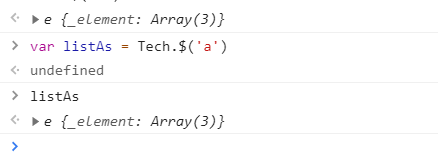
Có thể sử dụng bằng cách

# Khai báo đối tượng

Tương tự như với với sử dụng Jquery ta khai báo

Tech.$(selector truy vấn)

Hoặc Tech.Query.create(selector truy vấn)



Có thể sử dụng từ khóa khai báo để rút gọn cách viết code



# Các hàm đang có trong thư viện

Chú ý không đầy đủ các hàm, do vậy nếu thiếu có thể tự viết thêm

Chú ý các hàm dưới đây đều hoạt động đối với cả SingleElement và ListElement. Đối với các ListElement thông thường sẽ trả về 1 list các element chịu áp dụng, hoặc sẽ gán hành động lên tất cả Element đang xét tới.

## \_element:any;

Đây không phải là 1 hàm

Đây là đối tượng dom gốc của js, trong trường hợp các hàm không hỗ trợ có thể gọi đến đối tượng này để thao tác với js như thông thường.



## text(text?:string): string;

Hàm lấy text

Nếu có tham số truyền vào sẽ gán giá trị text cho element đó

Tương tự hàm text trong Jquery

## html(html?:string): string;

Hàm get/set html, tương tự jquery

## val(html?:string): string;

Hàm lấy giá trị value của input, chỉ hoạt động trên input, nếu sử dụng cho Element khác sẽ báo lỗi

## outerHtml(): string;

Lấy outerHTML

## css(key:string,value:string):Element;

Hàm gán CSS

## find(childSelector:string):Element;

Tương tụ hàm find trong jquery, tuy nhiên nếu tìm các thành phần con trực tiếp với selector “ > ul”

thì cần viết là “:scope > ul”

## addClass(\_class: string): Element;

Hàm Add Class vào Element

## removeClass(\_class: string): Element;

Hàm Remove Class

## hasClass(\_class:string):boolean;

Kiểm tra 1 element có class nào đó không?

## toggleClass(\_class:string):Element;

Luân phiên add/remove class khỏi element

## onClick(callback: FunctionCallBack): void;

hàm click của phần tử

form.onClick(function(e){

e.preventDefault();

…

})

## on(eventName:string,callback:FunctionCallBack):void;

Thêm sự kiện vào phần tử

## off(eventName:string,callback:FunctionCallBack):void;

Xóa sự kiện khỏi phần tử

## index():number;

Lấy vị trí của phần tử đó trong cha phần tử đó

## fadeIn(time:number):void;

Hàm Hiệu ứng chưa tốt, nên viết bằng css

## fadeOut(time:number):void;

## slideUp(time:number):void;

## slideDown(time:number):void;

## toggleSlide(time:number):void;

## hide():void;

Ẩn Element

## show():void;

## item(index:number):Element;

Lấy element vị trí thứ index ra, phần này phục vụ cho ListElement, truy vấn vào mảng element.

Với SingleElement luôn trả về element hiện tại

## append(element:any):void;

## prepend(element:any):void;

## before(element:any):void;

## after(element:any):void;

## next():Element;

## prev():Element;

## parent():Element;

## closest(selector:string):Element;

## clone():Element;

## remove():void;

## empty():void;

## attr(key:string,value?:string|number):string;

## tech5s(key:string,value?:string|number):string|Array<string>;

## removeAttribute(key:string):Element;

## height(height?:any,unit?:string):number;

## outerHeight():number;

## outerWidth():number;

## 

## width(width?:any,unit?:string):number;

## offset():object;

## position():object;

## trigger(eventName:string):void;

## serialize():string;

## isNotNull():boolean;

Trả về phần tử hiện tại có null hay không, tương đương với việc kiểm tra length>0

## length():number;

## forEach(callback:FunctionCallBackTwoParam):void;

## Hàm ready

Tech.Query.ready(function(){

// nội dung viết ở đây

})

Hàm ready này tương đương với hàm document ready của Jquery, xem xét dùng

## Hàm on document

Tech.Query.on(eventName:string,elementSelector:string,callback:FunctionCallBack)

Hàm này tương đương với $(document).on(‘event’,’selector’, callback) trong jquery

## Hàm Tech.Query.extend

extend(out:any,...maps:Array<any>):any

Cho phép gộm nhiều mảng vào 1 mảng, đây là hàm deep extend nghĩa là có thể kế thừa được thuộc tính dạng object

## Hàm getScrollTop

Tech.Query.getScrollTop();

Hàm lấy khoảng cách scrollTop

## Hàm Scroll To

Tech.Query. scrollTo(time:number =300, position:number=0)

Scroll tới vị trí nào đó trong bao lâu

## Hàm ajax

Tech.Query.ajax(obj:any)

Trong đó

Hàm ajax: Tham số nhận vào là đối tượng

{

success:function gọi khi thành công,

fail: function gọi khi error,

always: function gọi khi kết thúc dù thành công hay thất bại,

body: dữ liệu truyền đi dạng đối tượng {name:'tech',age:4}, sẽ merge với dữ liệu qua ajaxSetup

method: GET|POST...,

url: Url gửi request,

headers: header dạng object {}, tham số này sẽ merge với header setup qua ajaxHeader,

global: true, mặc định true, sẽ chịu ảnh hưởng của hàm ajaxStart và ajaxComplete

}

Sử dụng hàm \_ajax nếu muốn dùng Promise

Có thể setup hàm ajaxStart chạy trước mọi request, và ajaxComplete chạy sau mọi request

## Hàm ajaxHeader

Set global header cho mọi request ajax

ajaxHeader(config:object={}):void

## Hàm ajaxSetup

Set tham số mặc định cho mọi body dữ liệu gửi đi

## Hàm ajaxStart

ajaxStart(callback:FunctionCallBackNoParam)

Chạy trước khi mọi hàm ajax được gọi

## Hàm ajaxComplete

ajaxComplete(callback:FunctionCallBackNoParam)

Chạy sau khi request ajax hoàn thành

## Hàm ajaxHelp

Hiển thị thông tin gợi ý cho ajax