Snabbdom

是得可以渲染到web以外的平台，ssr。

Snabbdom.ts

Init函数 ：

export function init(modules: Array<Partial<Module>>, domApi?: DOMAPI){

参数：

modules： 包含[module对象](#module)的数组

domApi:

返回值：返回了patch函数

cbs = ({} as ModuleHooks

不同module中的hook根据hook名字存放到cbs对象中

Patch函数：

function patch(oldVnode: VNode | Element, vnode: VNode): VNode {

H函数：

export function h(sel: string): VNode;

export function h(sel: string, data: VNodeData): VNode;

export function h(sel: string, children: VNodeChildren): VNode;

export function h(sel: string, data: VNodeData, children: VNodeChildren): VNode;

data： eg. { on: { click: add } }

返回值是vnode()

vnode函数：

export function vnode(sel: string | undefined,

data: any | undefined,

children: Array<VNode | string> | undefined,

text: string | undefined,

elm: Element | Text | undefined): VNode

createElm函数

函数签名：

function createElm(vnode: VNode, insertedVnodeQueue: VNodeQueue): Node

调用流程：

Hook就是相应的函数，在dom操作周期函数内调用。

VNode

export interface VNode {

sel: string | undefined;

data: VNodeData | undefined;

children: Array<VNode | string> | undefined;

elm: Node | undefined;

text: string | undefined;

key: Key | undefined;

}

data.hook

data.hook.init

VNodeData

export interface VNodeData {

props?: Props;

attrs?: Attrs;

class?: Classes;

style?: VNodeStyle;

dataset?: Dataset;

on?: On; // h函数中用到

hero?: Hero;

attachData?: AttachData;

hook?: Hooks;

key?: Key;

ns?: string; // for SVGs

fn?: () => VNode; // for thunks

args?: Array<any>; // for thunks

[key: string]: any; // for any other 3rd party module

}

Hooks

export interface Hooks {

pre?: PreHook;

init?: InitHook;

create?: CreateHook;

insert?: InsertHook;

prepatch?: PrePatchHook;

update?: UpdateHook;

postpatch?: PostPatchHook;

destroy?: DestroyHook;

remove?: RemoveHook;

post?: PostHook;

}

Module

export interface Module {

pre: PreHook;

create: CreateHook;

update: UpdateHook;

destroy: DestroyHook;

remove: RemoveHook;

post: PostHook;

}

ModuleHooks

type ModuleHooks = ArraysOf<Module>;

type ArraysOf<T> = {

[K in keyof T]: (T[K])[];

}

Tip：Module的属性名字 与 hooks中的元素一一对应

hooks

const hooks: (keyof Module)[] = ['create', 'update', 'remove', 'destroy', 'pre', 'post'];

Vnode

Pr:

Reorder-animation

render();

render();

Questions:

Ts: s is string | number is的用法？