

//Webkit内核的浏览器，必须加上-webkit前缀。

.box{

display: -webkit-flex; */\* Safari \*/*

display: flex;

}

网页可见区域宽： **document**.documentElement.clientWidth;

网页可见区域高： **document**.documentElement.clientHeight;

网页正文全文宽： **document**.documentElement.scrollWidth;

网页正文全文高： **document**.documentElement.scrollHeight;

网页被卷去的高(**ff**)：**document**.body.scrollTop;

网页被卷去的高(**ie**): **document**.documentElement.scrollTop;

网页被卷去的左：**document**.body.scrollLeft;

网页正文部分上：**window**.screenTop;

网页正文部分左：**window**.screenLeft;

某个元素的宽度：**obj**.offsetWidth;

某个元素的高度：**obj**.offsetHeight;

某个元素的上边界到**body**最顶部的距离：**obj**.offsetTop;（在元素的包含元素不含滚动条的情况下）

某个元素的左边界到**body**最左边的距离：**obj**.offsetLeft;（在元素的包含元素不含滚动条的情况下）

返回当前元素的上边界到它的包含元素的上边界的偏移量：**obj**.offsetTop（在元素的包含元素含滚动条的情况下）

返回当前元素的左边界到它的包含元素的左边界的偏移量：**obj**.offsetLeft（在元素的包含元素含滚动条的情况下）

新增元素 说明

video 表示一段视频并提供播放的用户界面

audio 表示音频

canvas 表示位图区域

source 为video和audio提供数据源

track 为video和audio指定字母

svg 定义矢量图

code 代码段

figure 和文档有关的图例

figcaption 图例的说明

main

time 日期和时间值

mark 高亮的引用文字

datalist 提供给其他控件的预定义选项

keygen 秘钥对生成器控件

output 计算值

progress 进度条

menu 菜单

embed 嵌入的外部资源

menuitem 用户可点击的菜单项

menu 菜单

template

section

nav

aside

article

footer

header

一：location.href 用法:

　　top.location.href=”url”           在顶层页面打开url（跳出框架）  
　　self.location.href=”url”           仅在本页面打开url地址  
　　parent.location.href=”url” 　  在父窗口打开Url地址  
　　this.location.href=”url” 　　  用法和self的用法一致

// 强制刷新页面

setInterval(function () {

    location.href = location.href

}, 1000 \* 60 \* 30);

 //用于监听input的值变化（input的值产生变化才会触发事件）

    (function ($) {

        $.fn.watch = function (callback) {

            return this.each(function () {

                //缓存以前的值

                $.data(this, 'originVal', $(this).val());

                //event

                $(this).on('keyup paste', function () {

                    var originVal = $.data(this, 'originVal');

                    var currentVal = $(this).val();

                    if (originVal !== currentVal) {

                        $.data(this, 'originVal', $(this).val());

                        callback(currentVal);

                    }

                });

            });

        }

    })(jQuery);

    $("input:text").watch(function (value) {

        if(value.length >= 5){

            console.log(1)

            console.log(value)

            $("input").val("")

        }

    });

# [Vue] 生命周期， created，mounted, methods , computed , watched, 通俗易懂

****生命周期：****

beforecreate : 一般使用场景是在加 loading事件 的时候

created ：处于loading结束后，还做一些初始化，实现函数自执行（data数据已经初始化，但是DOM结构渲染完成，组件没有加载）

beforemount：处于组件创建完成，但未开始执行操作

mounted ：处于发起后端请求，获取数据，配合路由钩子执行操作（DOM渲染完成，组件挂载完成 ）

beforeupdate、updated：处于数据更新的前后

beforeDestroy：当前组件还在的时候，想删除组件

destroyed ：当前组件已被销毁，清空相关内容

****created 与 mounted 的区别****

created:在模板渲染成html前调用，即通常初始化某些属性值，然后再渲染成视图。

mounted:在模板渲染成html后调用，通常是初始化页面完成后，再对html的dom节点进行一些需要的操作。

****mounted 与 methods 的区别****

mounted 是生命周期方法之一，会在对应生命周期时执行。

methods 是Vue实例对象上绑定的方法，供当前Vue组件作用域内使用，未调用不会执行，只执行逻辑，返回值可有可无。

****computed 与 watched 的区别****

computed 是计算属性，也可以理解为一个方法。其中计算的结果如果不发生改变就不会触发，且必须返回一个值并在DOM中绑定的才能取得值。他可以自动获取数据的改变。

watched 属性是手动定义的所需监听的值，不同的数据可以在其中多次定义监听值，这时会消耗一定性能，他并不能像computed那样自动改变。



