# 面试题向：

## 防抖节流：

（1）通过setTimeout：



1. 通过时间戳的方式：



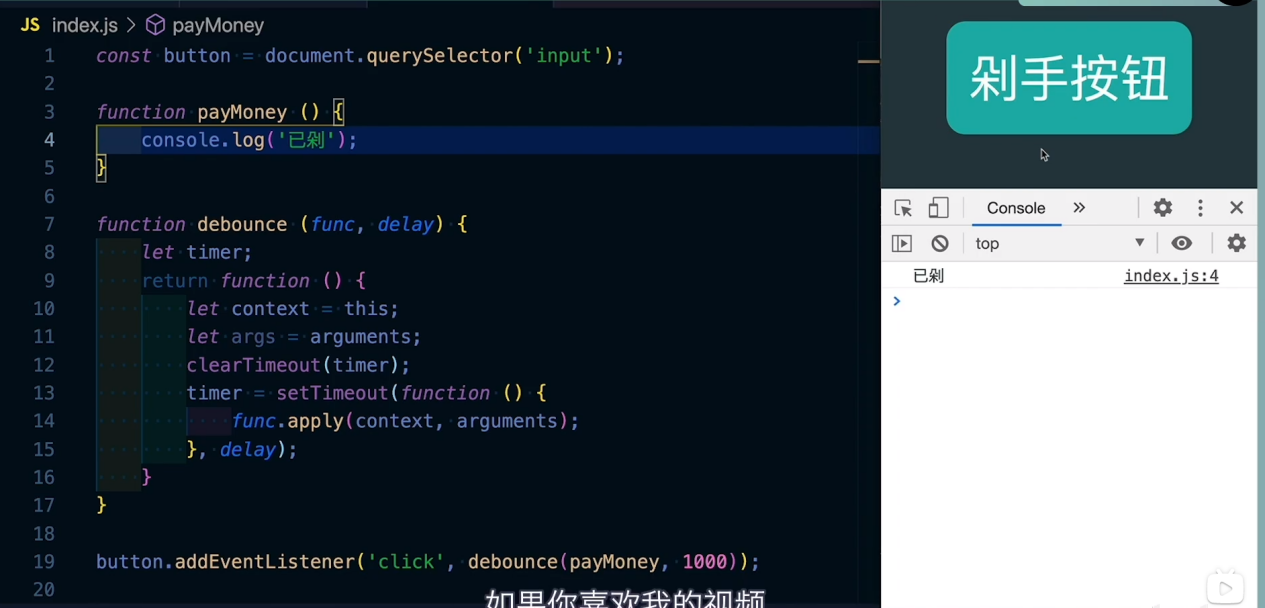
实例：

登录验证：

1. 通过设置倒计时30秒钟，减少用户频繁操作；
2. 不正确的情况下，新旧密码一致，将不再请求；

## 防抖（一定时间内多次操作）

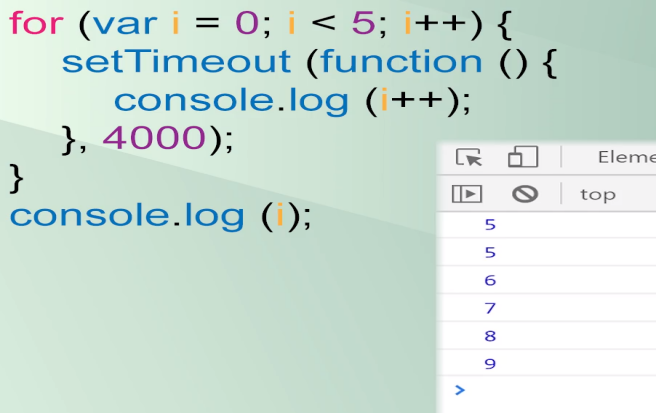
案例：提交，防止短时间内提交多次



## 3.B包

经典面试题：

说明：js单线程从上到下执行，for循环开始，循环一次执行setTimeout一次，但是setTimeout是异步事件，所以会放到任务队列里面去等待执行，循环结束后，继续去执行执行栈里面的任务console.log(),因为没有块级作用域，var 命名的i是全局的，所以输出5，4秒结束后依次执行setTimeout中的内容，依次输出5（++在后面），6,7,8,9；



## 4.bom对象

**Window：**窗口

Bom：浏览器对象模型，让JS可以操作浏览器，也不是语法，是一套接口

#### 4.1 属性

. name 窗口的名字

. innerHeight 窗口的内部高度

. innerWidth 窗口的内部宽度

.outHeight 窗口外部高度

.outWidth 窗口外部宽度

. scrollY 【重点】浏览器纵向滚动条的滚动位置

. onscroll 【重点】浏览器滚动事件

#### 4.2 方法

. alert (呃嫩儿特)： 【重点】弹出一个警告框

. confirm (亢佛姆)： 弹出一个确认框，返回用户选择的值

. prompt (普诺姆普特)：弹出一个输入框，返回用户输入的值，返回的是一个字符串

. scroll ( x , y )： 【重点】让浏览器滚动条滚动的方法

. setInterval (因特儿沃)： 【重点】设置一个周期定时器，到时间就自动执行，可以执行多 次。 返回一个定时器序号作为定时器名称，可以被清除

. setTimeout ( )：设置一个超时定时器，到时间执行一次就自动结束，可以执行1次

. clearInterval ()： 定时器变量名，【重点】清除周期定时器

. clearTimeout ( ) ：清除超时定时器

#### 4.2 子对象

文档对象 document

屏幕对象 screen（斯葵恩）对象（屏幕对象）

availHeight（厄维欧~）：返回显示屏幕的高度 (除 Windows 任务栏之外)。

availWidth（厄维欧~）：返回显示屏幕的宽度 (除 Windows 任务栏之外)。

Width（维茨） 屏幕宽度（分辨率x轴）

height 屏幕高度（分辨率y轴）

地址栏对象

location对象【重点】

. Url：统一资源定位符

. protocal（普诺姆图阔）：协议

. Host: 主机名/域名/IP，全球连接网络的一台电脑的ip地址。

. Port（颇儿特）: 端口，当前软件使用的一个对于一台电脑唯一标识。

. hash 锚点，URL地址后面 #后的部分，页内跳转。

. Pathname：相对服务器的资源地址。

. href 当前页面完整的URL地址，赋值给它就能实现网页的跳转

【完整地址注解】

protocal：host:port / pathname ? search # hash

方法：

. assign(“地址”)：跳转到指定地址

. reload()： 刷新当前页，传参数true为强制刷新，从服务器获取，不使用浏 览器缓存。

.replace(“地址”)： 跳转到指定地址，不可返回。

历史记录对象：

history对象（历史记录对象）

. Length 历史记录长度

. back ( ) 返回上一次访问过的网页

. forward ( ) 进入后一次访问的网页

. go ( n ) n可以是正数（前进forward），也可以是负数（倒退back）

浏览器信息

navigator对象

. userAgent 返回浏览器的基本信息，可以判断用户的浏览器和操作系统

## Dom

Document object model DOM：文档对象模型，让JS可以操作HTML和CSS， 不是语法，是一套接口。

#### 5.1节点：

元素节点、属性节点、文本节点。

【注意】：在js中，所有的节点都是一个对象。

查找节点的方法：

. getElementById() id

. getElementsByTagName() 节点名

. GetElementsByClassName() 类名

. GetElementsByName() 属性名

**Html5**

一个：querySelector() id:“#~”,节点：“div”,属性：[title=”name”]等。

所有：QuerySelectorAll()

可获取/修改节点属性:

Id、title、className、tagName、innerHTML、style、nodeName、nodeType、nodeValue等。

Style（样式获取）:

getComputedStyle(属性名)[“属性”] 获取当前有效样式，ie不支持；

Ie 对应使用 dom.currentStyle[“属性名”]；(克伦特~)

**兼容处理：**

Function(Dom,attr){

Return Dom.currentStyle?Dom.currentStyle[attr]:getComputedStyle(Dom)[attr]

}

attribute（属性获取）:

getAttribute(“属性名”)：获取， 例如Dom.getAttribute(“title”);

setAttritbute(“属性名”)：设置，例如Dom.setAttribute(“title”,”我是title”);

removeAttribute(“属性名”):删除 例如Dom.setAttribute(“title”);

相对于.的方式的区别：

1. set/getAttribute对用户自定义属性的支持
2. removeAttribute 可以删除属性

# html

## html 语义化

意义：根据内容的结构化（内容语义化），选择合适的标签（代码语义化）便于开发者阅读和写出更优 雅的代码的同时让浏览器的爬虫和机器很好地解析。

注意：

1. 尽可能少的使用无语义的标签 div 和 span；
2. 在语义不明显时，既可以使用 div 或者 p 时，尽量用 p, 因为 p 在默认情况下有上下间距，对兼容特殊终端有利；
3. 不要使用纯样式标签，如：b、font、u 等，改用 css 设置。
4. 需要强调的文本，可以包含在 strong 或者 em 标签中（浏览器预设样式，能用CSS 指定就不用他们），strong 默认样式是加粗（不要用 b），em 是斜体（不用i）；
5. 使用表格时，标题要用 caption，表头用 thead，主体部分用 tbody 包围，尾部用 tfoot 包围。表头和一般单元格要区分开，表头用 th，单元格用 td；
6. 表单域要用 fieldset 标签包起来，并用 legend 标签说明表单的用途；
7. 每个 input 标签对应的说明文本都需要使用 label 标签，并且通过为 input 设置id 属性，在 lable
8. 标签中设置 for=someld 来让说明文本和相对应的 input 关联起来。



## canvas相关

使用前需要获得上下文环境，暂不支持 3 d

常用 api:

1. fillRect(x,y,width,height)实心矩形
2. strokeRect(x,y,width,height)空心矩形fillText( "Hello world" , 200 , 200 );实心文字
3. strokeText( "Hello world" , 200 , 300 )空心文字

新标签兼容低版本

ie9 之前版本通过 createElement 创建 html5 新标签引入 html5shiv.js

WEBGL与canvas和openGL的关系：

1. Laya引擎支持WebGL和canvas两种渲染方式，可以在引擎初始化时选择渲染方式
2. Laya.init(width,height,WebGL);//WebGL模式
3. Laya.init(width,height);//canvas默认为canvas模式
4. canvas是一个H5标签，作用是在网页上画图，但是只支持2D，不支持3D。WebGL是一种3D绘图标准，WebGL支持3D，且性能优于canvas。所以现在能用WebGL的都用WebGL,有些项目用canvas是因为部分手机不支持WebGL。
5. OpenGL是 底层的驱动级的图形接口（是显卡有直接关系的） 类似于 DirectX
6. 但是这种底层的OpenGL是 寄生于浏览器的JavaScript无法涉及的
7. 但是为了让Web拥有更强大的 图形处理能力 2010年时候WebGL被推出来
8. WebGL允许工程师使用JS去调用部分封装过的 OpenGL ES2.0标准接口去 提供硬件级别的3D图形加速功能

三者的关系是 JavaScript -> WebGL -> OpenGL ->.... -> 显卡 并把最终渲染出来图形 呈现到Canvas

所以，能使用WebGL的前提是浏览器支持WebGL，且显卡支持OpenGL ES2.0才可以

## svg和canvas的区别？

1. canvas时h5提供的新的绘图方法 ；svg已经有了十多年的历史
2. canvas画图基于像素点，是位图，如果进行放大或缩小会失真 ；svg基于图形，用html标签描绘形
3. 状，放大缩小不会失真
4. canvas需要在js中绘制 ；svg在html绘制
5. canvas支持颜色比svg多
6. canvas无法对已经绘制的图像进行修改、操作 ；svg可以获取到标签进行操作

## html5有哪些新特性？

1. HTML5主要是关于图像，位置，存储，多任务等功能的增加。
2. 拖拽释放(Drag and drop) API
3. 语义化更好的内容标签（header,nav,footer,aside,article,section） 音频、视频API(audio,video)
4. 画布(Canvas) API
5. 地理(Geolocation) API
6. 本地离线存储 localStorage 长期存储数据，浏览器关闭后数据不丢失；
7. sessionStorage 的数据在浏览器关闭后自动删除
8. 表单控件，calendar、date、time、email、url、search

## 如何处理HTML5新标签的浏览器兼容问题？

IE8/IE7/IE6支持通过document.createElement方法产生的标签，可以利用这一特性让这些浏览器支持HTML5新标签，当然最好的方式是直接使用成熟的框架、使用最多的是html5shim框架

（1）使用document创建标签

（2）采用田间编译

a）使用html5shim：在<head>中调用以下代码：

<!--[if lt IE 9]>

<script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>

<![endif]-->

b）使用kill IE6: 在</body>之前调用以下代码：

<!--[if lte IE 6]>

<script src="http://letskillie6.googlecode.com/svn/trunk/letskillie6.zh\_CN.pack.js"></script>

<![endif]-->

## 说说 title 和 alt 属性

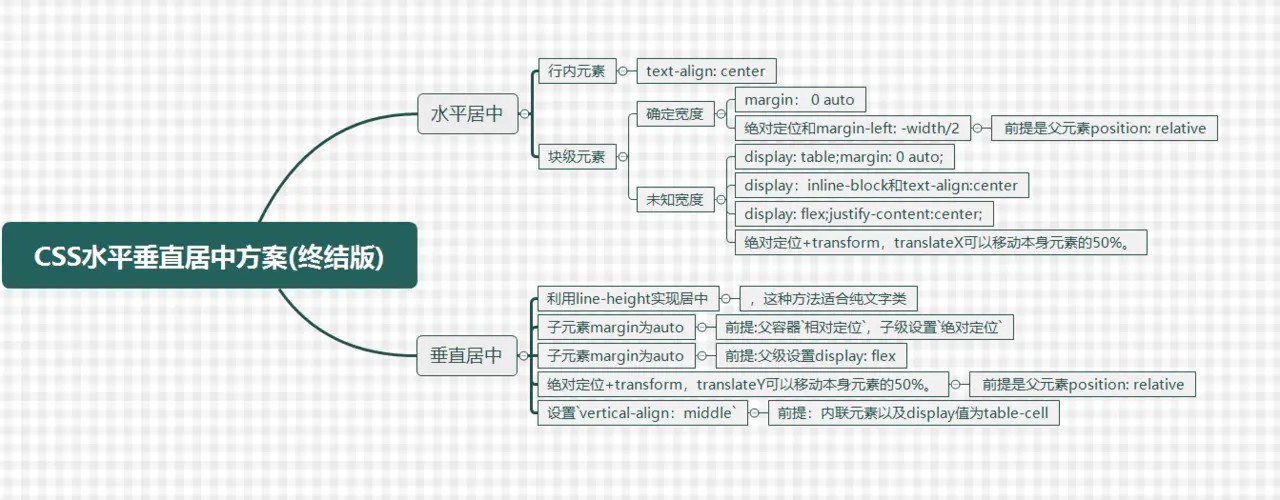
1. 两个属性都是当鼠标滑动到元素上的时候显示
2. Alt是img的特有属性，是图片内容的等价描述，图片无法正常显示时候的替代文字。
3. title属性可以用在除了base，basefont，head，html，meta，param，script和title之外的所有标签，是对dom元素的一种类似注释说明;

## HTML全局属性(global attribute)有哪些

1. class :为元素设置类标识
2. data-\* : 为元素增加自定义属性
3. draggable: 设置元素是否可拖拽
4. id: 元素 id ，文档内唯一
5. lang: 元素内容的的语言
6. style : 行内 css 样式
7. title: 元素相关的建议信息

# CSS

## 让一个元素水平垂直居中，到底有多少种方案？



#### 水平居中

对于 行内元素 : text-align: center;

对于确定宽度的块级元素：

（1）width和margin实现。margin: 0 auto;

（2）绝对定位和margin-left: -width/2, 前提是父元素position: relative

对于宽度未知的块级元素

1. table标签配合margin左右auto实现水平居中。使用table标签（或直接将块级元素设值为display:table），再通过给该标签添加左右margin为auto。
2. inline-block实现水平居中方法。display：inline-block和text-align:center实现水平居中。
3. 绝对定位+transform，translateX可以移动本身元素的50%。
4. flex布局使用justify-content:center

#### 垂直居中

1. 利用 line-height 实现居中，这种方法适合纯文字类
2. 通过设置父容器 相对定位 ，子级设置 绝对定位 ，标签通过margin实现自适应居中
3. 弹性布局 flex :父级设置display: flex; 子级设置margin为auto实现自适应居中
4. 父级设置相对定位，子级设置绝对定位，并且通过位移 transform 实现
5. table 布局，父级通过转换成表格形式，然后子级设置 vertical-align（维题阔~） 实现。（需要注意的是：vertical-align: middle（密都）使用的前提条件是内联元素以及display值为table-cell（舍欧）的元素）。

## 浮动布局的优点？有什么缺点？清除浮动有哪些方式？

浮动布局简介:

当元素浮动以后可以向左或向右移动，直到它的外边缘碰到包含它的框或者另外一个浮动元素的边框为止。元素浮动以后会脱离正常的文档流，所以文档的普通流中的元素好像察觉不到浮动元素一样。

优点

1. 这样做的优点就是在图文混排的时候可以很好的使文字环绕在图片周围。
2. 另外当元素浮动了起来之后它有着块级元素的一些性质例如可以设置宽高等，但它与inline-block还是有一些区别的，第一个就是关于横向排序的时候，float可以设置方向而inline-block方向是固定的；还有一个就是inline-block在使用时有时会有空白间隙的问题。

缺点

最明显的缺点就是浮动元素一旦脱离了文档流，就无法撑起父元素，会造成父级元素高度塌陷。

清除浮动的4种方式

1. 额外标签法（在最后一个浮动标签后，新加一个标签，给其设置clear：both；）（不推荐）

<div class="parent">

//添加额外标签并且添加clear属性

<div style="clear:both"></div>

//也可以加一个br标签

</div>

**优点：**通俗易懂，方便

**缺点：**添加无意义标签，语义化差

1. 父级添加overflow属性（父元素添加overflow:hidden）（不推荐）

<div class="parent" style="overflow:hidden">//auto 也可以

//将父元素的overflow设置为hidden

<div class="f"></div>

</div>

**优点**：代码简洁

**缺点**：内容增多的时候容易造成不会自动换行导致内容被隐藏掉，无法显示要溢出的元素

1. 使用after伪元素清除浮动（推荐使用）

//在css中添加:after伪元素

.parent:after{

/\* 设置添加子元素的内容是空 \*/ content: '';

/\* 设置添加子元素为块级元素 \*/ display: block;

/\* 设置添加的子元素的高度0 \*/ height: 0;

/\* 设置添加子元素看不见 \*/ visibility: hidden;

/\* 设置clear：both \*/ clear: both;

}

<div class="parent">

<div class="f"></div>

</div>

**优点：**符合闭合浮动思想，结构语义化正确

**缺点：**ie6-7不支持伪元素：after，使用zoom:1触发hasLayout.

## 使用display:inline-block会产生什么问题？解决方法？

问题复现：

两个display：inline-block元素放到一起会产生一段空白。

如例：

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Document</title>

<style>

.container { width: 800px; height: 200px;

}

.left {

font-size: 14px; background: red; display: inline-block; width: 100px;

height: 100px;

}

.right {

font-size: 14px; background: blue; display: inline-block; width: 100px;

height: 100px;

}

</style>

</head>

<body>

<div class="container">

<div class="left">

左

</div>

<div class="right">

右

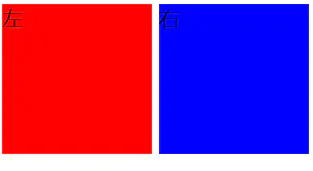
</div>

</div>

</body>

</html>

效果如下:



产生空白的原因

元素被当成行内元素排版的时候，元素之间的空白符（空格、回车换行等）都会被浏览器处理，根据 CSS中white-space属性的处理方式（默认是normal，合并多余空白），原来HTML代码中的回车换行被转成一个空白符，在字体不为0的情况下，空白符占据一定宽度，所以inline-block的元素之间就出现了空隙。

**解决办法**

1. 将子元素标签的结束符和下一个标签的开始符写在同一行或把所有子标签写在同一行

<div class="container">

<div class="left">

左

</div><div class="right">

右

</div>

</div>

1. **父元素中设置font-size: 0，在子元素上重置正确的font-size**

.container{ width:800px; height:200px; font-size: 0;

}

1. 为子元素设置float:left

.left{

float: left; font-size: 14px; background: red;

display: inline-block; width: 100px;

height: 100px;

}

//right是同理

## **布局题：div垂直居中，左右10px，高度始终为宽度一半**

问题描述:

实现一个div垂直居中, 其距离屏幕左右两边各10px, 其高度始终是宽度的50%。同时div 中有一个文字A，文字需要水平垂直居中。

1. 思路一：利用height:0; padding-bottom: 50%;
2. 思路二**:** 利用**calc**和**vw**

其他布局/考点：

1. 盒子模型：

标准盒子：width：contentWidth+padding+border;

怪异盒子：width:contentWidth

1. CSS如何进行品字布局？
2. CSS如何进行圣杯布局

完成一些图形：

1. 三角形



// 四个参数对应 ：上 右 下 左

border-color: transparent transparent #ccc transparent;

1. 对话框



.dialog {

width: 300px;

line-height: 2; background: lightblue;

color: #fff;

}

.dialog::before{

content: ''; position: absolute; border: 8px solid;

border-color: transparent lightblue transparent transparent;

left: -16px;

top: 8px;

}

<div class="dialog">这是一个对话框鸭！</div>

1. 画一个平行四边形



.parallel {

margin-top:50p; margin-left:50p;

width: 200px; height: 100px;

background: lightblue; transform: skew(-20deg,0);

}

<div class="parallel"></div>

1. 五角星



#star {

position: relative; margin:200px auto;

width: 0;

height: 0;

border-style: solid;

border-color: transparent transparent red transparent; border-width: 70px 100px;

transform: rotate(35deg);

}

#star::before {

position: absolute; content: '';

width: 0;

height: 0; top: -128px; left: -95px;

border-style: solid;

border-color: transparent transparent red transparent; border-width: 80px 30px;

transform: rotate(-35deg);

}

#star::after { position: absolute; content: '';

width: 0;

height: 0; top: -45px; left: -140px;

border-style: solid;

border-color: transparent transparent red transparent; border-width: 70px 100px;

transform: rotate(-70deg);

}

<div id="star"></div>

## 什么是BFC？

W3C对BFC的定义如下：

浮动元素和绝对定位元素，非块级盒子的块级容器（例如 inline-blocks, tablecells, 和 table-captions），以及overflow值不为"visiable"的块级盒子，都会为他们的内容创建新的

BFC（Block Fromatting Context， 即块级格式上下文）。

触发条件:

一个HTML元素要创建BFC，则满足下列的任意一个或多个条件即可： 下列方式会创建块格式化上下文：

1. 根元素()
2. 浮动元素（元素的 float 不是 none）
3. 绝对定位元素（元素的 position 为 absolute 或 fixed）
4. 行内块元素（元素的 display 为 inline-block）
5. 表格单元格（元素的 display为 table-cell，HTML表格单元格默认为该值）
6. 表格标题（元素的 display 为 table-caption，HTML表格标题默认为该值）
7. 匿名表格单元格元素（元素的 display为 table、table-row、 table-row-group、table-headergroup、table-footer-group（分别是HTML table、row、tbody、thead、tfoot的默认属性）或inline-table）overflow 值不为 visible 的块元素 -弹性元素（display为 flex 或 inline-flex元素的直接子元素）
8. 网格元素（display为 grid 或 inline-grid 元素的直接子元素） 等等。

## Flex布局

# JavaScript

## JS原始数据类型有哪些？引用数据类型有哪些？

1. 栈: (原始数据) string/number/boolean/null/undefined/symbol
2. 堆: (引用数据类型)object（array和函数属于object）

数据类型一共7（6种基本类型+1种引用类型）种