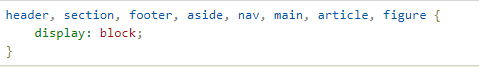
# HTML5新增内容

## 一、新增元素

### 1、为了能让旧版本的浏览器正确显示这些元素，你可以设置 CSS 的 **display** 属性值为 **block**



### 2、HTML新元素

1）新多媒体元素

canvas：画布

audio：音频播放器

video：视频播放器

source：定义多媒体资源audio和video

embed：定义插入的内容，比如插件

track：为诸如 <video> 和 <audio> 元素之类的媒介规定外 部文本轨道

2）新表单元素

datalist：定义选项列表。请与 input 元素配合使用该元素， 来定义 input 可能的值

keygen：规定用于表单的密钥对生成器字段。

output：定义不同类型的输出，比如脚本的输出。

3）新的语义和结构元素

article：定义页面独立的内容区域

aside：定义页面的侧边栏内容

header：定义了文档的头部区域

footer：页脚

nav：导航

progress：进度条

## 二、画布：canvas

### 1、创建canvas标签

设置id、style、width、height等属性

<canvas id="mycanvas"width="400"height="200"style="border:1px solid"></canvas>

### 2、javascript找到canvas标签

var c=document.getElementById("myCanvas")

### 3、创建content对象

var ctx=c.getContext("2d") 表示二位网络

### 4、选择笔画颜色：fillStyle

ctx.fillStyle="#FF0000"

### 5、选择笔画颜色：fillStyle

ctx.fillStyle="#FF0000"

### 6、下笔及抬笔

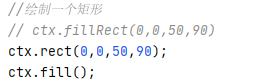
开始下笔：ctx.beginPath()

开始抬笔：ctx.closePath()

### 7、绘制一个矩形

fillRect和rect有4个参数(left,top,宽度,高度)

ctx.fillRect(0,0,150,75)或ctx.rect(50,50,50,50);ctx.fill()

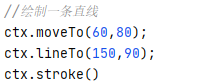


### 8、绘制一条直线

1）moveTo(x,y) 定义线条开始坐标

2）lineTo(x,y) 定义线条结束坐标

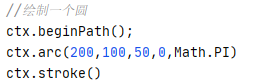
3）stroke()方法来绘制线条



### 9、绘制圆形

1）ctx.arc(圆形x,圆心y,半径,开始角度,结束角度);

2）stroke()，绘制圆形



### 10、绘制文本

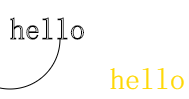
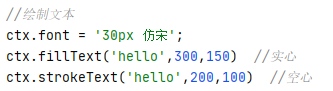
1）设置字体格式

ctx.font = '30px 仿宋'

2）绘制文本

ctx.fillText('hello',300,150) //实心

ctx.strokeText('hello',200,100) //空心

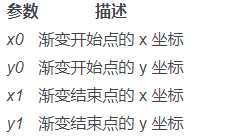


### 11、canvas渐变效果

渐变可以填充在矩形、圆心、线条、文本中，可自行设置不同的颜色。

1）线条渐变

① createLinearGradient(x0,y0, x1,y1)函数



② grd.addColorStop(1,"red");

grd.addColorStop(0,"blue");

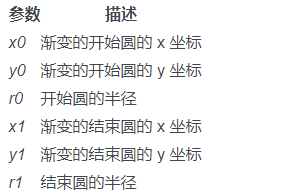
③ 给ctx.fillStyle赋值

④ 绘制图形



2）径向/圆形渐变

① createRadialGradient(x0,y0,r0,x1,y1,r1)函数

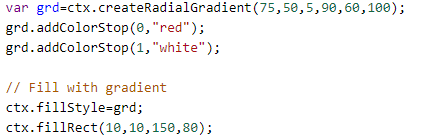


② grd.addColorStop(1,"red");

grd.addColorStop(0,"blue");

③ 给ctx.fillStyle赋值

④ 绘制图形



### 12、图像

将图像放置到画布上

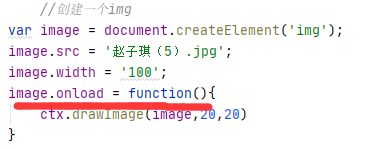
1）drawImage(image,x,y)

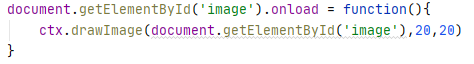
Image：图片

x、y：图片左上角的坐标

2）image.onload = function(){drawImage(image,x,y)}

等待图片加载完毕才能执行drawImage方法





## 三、内联SVG

### 1、什么是SVG

1）SVG 指可伸缩矢量图形 (Scalable Vector Graphics)

2）SVG 用于定义用于网络的基于矢量的图形

3）SVG 使用 XML 格式定义图形

4）SVG 图像在放大或改变尺寸的情况下其图形质量不会有损失

5）SVG 是万维网联盟的标准

### 2、SVG的优势

1）SVG 图像可通过文本编辑器来创建和修改

2）SVG 图像可被搜索、索引、脚本化或压缩

3）SVG 是可伸缩的

4）SVG 图像可在任何的分辨率下被高质量地打印

5）SVG 可在图像质量不下降的情况下被放大

### 3、稍后再做

## 四、MathML

1、稍后再做

## 拖放

拖放是一种常见的特性，即抓取对象以后拖到另一个位置。在 HTML5 中，拖放是标准的一部分，任何元素都能够拖放。

### 1、浏览器支持

Internet Explorer 9+, Firefox, Opera, Chrome, Safari 均支持。

**注意**：Safari 5.1.2不支持拖动

### 2、设置拖拽

1）draggable 属性：设置元素可拖拽

true： 元素可以被拖拽

false：元素不能被拖拽

auto：浏览器自己判断元素是否能被拖拽 ( 默认 )

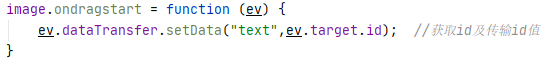
<img draggable="true">

2）对象拖放事件

a、dragstart：按下鼠标键并开始移动时触发

drap的数据传输：

ev.dataTransfer.setData("Text",ev.target.id);



b、drag：在元素拖拽过程中持续触发----相似与mousemove

一般是阻止默认行为

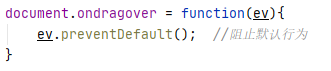
c、dragend：元素拖拽停止时触发

3）投放区事件流程

元素被拖动到有效的放置目标时，下列事件会依次发生：

1. dragenter：当拖拽对象进入投放区时触发

2. dragover：拖拽对象在投放区内移动时持续触发



3. dragleave：元素被拖出了投放区时触发

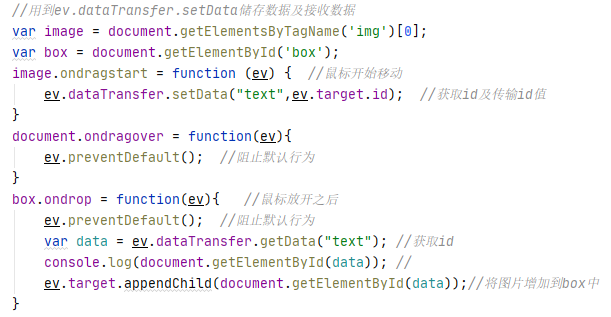
4. drop：拖拽对象投放在投放区时触发

通过ev.dataTransfer.getData("Text")来接收数据，并 对数据进行操作。



4）整体文档结构

a、使用数据传输



b、不使用数据传输



## 六、地理定位（教程在Hbuilder中）

### 1、navigator对象：包含浏览器的信息

1）属性

navigator.[appCodeName](https://www.w3school.com.cn/jsref/prop_nav_appcodename.asp)：返回浏览器的代码名

navigator.[appMinorVersion](https://www.w3school.com.cn/jsref/prop_nav_appminorversion.asp)：返回浏览器的次级版本

navigator.[appName](https://www.w3school.com.cn/jsref/prop_nav_appname.asp)：返回浏览器的名称

navigator.[appVersion](https://www.w3school.com.cn/jsref/prop_nav_appversion.asp)：返回浏览器的平台和版本信息

navigator.[browserLanguage](https://www.w3school.com.cn/jsref/prop_nav_browserlanguage.asp)：返回当前浏览器的语言

navigator.[cookieEnabled](https://www.w3school.com.cn/jsref/prop_nav_cookieenabled.asp)：返回指明浏览器中是否启用 cookie 的布尔值

navigator.[cpuClass](https://www.w3school.com.cn/jsref/prop_nav_cpuclass.asp)：返回浏览器系统的 CPU 等级

navigator.[onLine](https://www.w3school.com.cn/jsref/prop_nav_online.asp)：返回指明系统是否处于脱机模式的布尔值

navigator.[platform](https://www.w3school.com.cn/jsref/prop_nav_platform.asp)：返回运行浏览器的操作系统平台

navigator.[systemLanguage](https://www.w3school.com.cn/jsref/prop_nav_systemlanguage.asp)：返回 OS 使用的默认语言

navigator.[userAgent](https://www.w3school.com.cn/jsref/prop_nav_useragent.asp)：返回由客户机发送服务器的 user-agent 头部的值

navigator.[userLanguage](https://www.w3school.com.cn/jsref/prop_nav_userlanguage.asp)：返回 OS 的自然语言设置

**navigator.geolocation：是否支持定位**

2）方法

navigator.[javaEnabled()](https://www.w3school.com.cn/jsref/met_nav_javaenabled.asp)：规定浏览器是否启用Java

navigator.[taintEnabled()](https://www.w3school.com.cn/jsref/met_nav_taintenabled.asp)：规定浏览器是否启用数据污点 (data tainting)。

### 2、判断浏览器是否支持定位

function loadDemo() {

if(navigator.geolocation) {

document.write("HTML5 Geolocation supported.")  
 } else {document.write("HTML5 Geolocation is not supported in your browser.")  
 }}

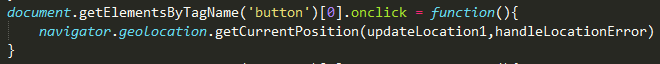
loadDemo()

### 3、位置请求方式

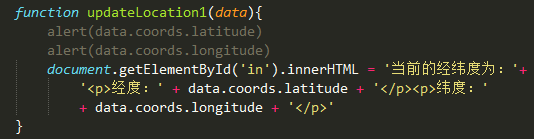
1、单次请求

navigator.geolocation.getCurrentPosition(showPosition, showError)

回调函数showPosition接受一个对象参数



showPosition函数：



showError函数

*function* handleLocationError(*error*){

switch(*error*.code) {

case *error*.PERMISSION\_DENIED:

alert("定位失败,用户拒绝请求地理定位");

break;

case *error*.POSITION\_UNAVAILABLE:

alert("定位失败,位置信息是不可用");

break;

case *error*.TIMEOUT:

alert("定位失败,请求获取用户位置超时");

break;

case *error*.UNKNOWN\_ERROR:

alert("定位失败,定位系统失效");

break;

}

}

2、多次请求

1）navigator.geolocationn.watchPosition (showPositio，showError)

浏览器多次调用showPositio方法，可以持续获取地理位

2）已知两点经纬度，计算两点的距离

使用Haversine公式计算：

3、高德地图应用

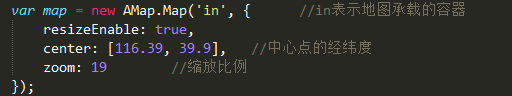
1）搜索高德地图API

2）产品介绍=>地图=>javascriptAPI=>

3）引入高德地图脚本

<scriptsrc="https://webapi.amap.com/maps?v=1.4.15&key=您申请的key值"></script>

4）创建map对象



var map = new AMap.Map('in', {

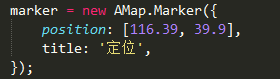
resizeEnable: true,

center: [116.39, 39.9],

zoom: 19

})

5）创建marker对象



marker = new AMap.Marker({

position: [data.coords.latitude, data.coords.longitude)],

title: '定位',

})

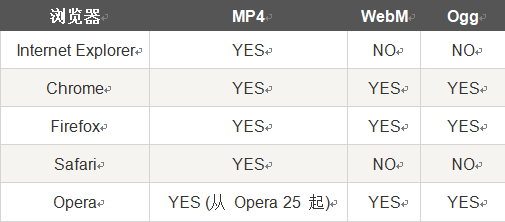
6）将marker标记到地图上

map.add(marker)或marker.setMap(map)

## 七、video（视频）

<video> 元素提供了 播放、暂停和音量控件来控制视频。同时 <video> 元素也提供了 width 和 height 属性控制视频的尺寸，<video>标签中的内容是不会显示在浏览器中的。

### 1、视频可是和浏览器支持



### 2、video的结构

<video controlswidth="300">

<source src="videos.mp4"type="video/mp4">

</video>

### 3、播放视频

1）video.paused：视频播放则返回false，反之返回true。

2）video.play()：控制视频播放

3）video.pause()：控制视频暂停



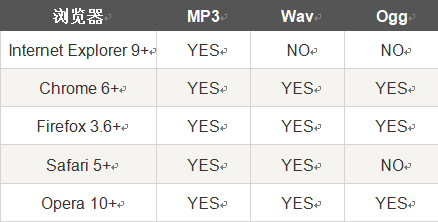
### 4、整体文档结构



## 八、radio（音频）

control 属性供添加播放、暂停和音量控件。在<audio> 与 </audio> 之间你需要插入浏览器不支持的<audio>元素的提示文本。和video很类似

### 1、音频格式及浏览器支持



### 2、文档结构



**注**：audio没有width属性，需要用style来设置。

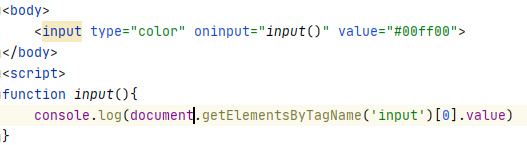
## 九、input类型

### 1、color

1）格式

<input type=””color”/>

2）事件绑定



3）设置默认颜色

value=”默认颜色”

4）获取选择的颜色

alert(document.getElementsByTagName('input')[0].value)

### 2、date

1）格式

<input type=””date”/>

2）事件绑定





3）设置默认日期

value="2019-10-09"

4）获取选择日期



**注**：此时data是String格式

5）格式转换

### 3、与date类似的week、month、time、datetime、datetime-local

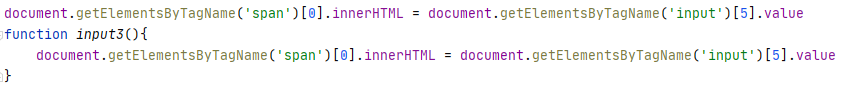
### 4、滑块：range

1）格式

<input type=””range”/>

2）事件绑定





3）设置默认值

value="15"

4）获取选择日期

document.getElementsByTagName('input')[5].value

5）设置最大最小值

min=’’0’ max=’20’

### 5、数字



### 6、搜索





## 十、表单控件

### 1、datalist

1）将输入框与datalist相关联

<input list="browsers"name="browser"oninput="*input*()">

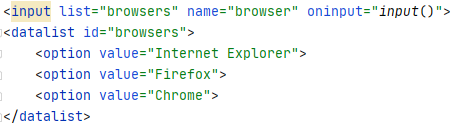
<datalist id="browsers">

<option value="Internet Explorer">

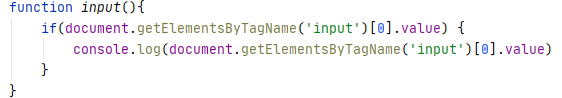
<option value="Firefox">

<option value="Chrome">

</datalist>

. 

2）获取value值



### 2、output标签

1）output标签

<output name="x"for="b c"></output>

Name：output的名称

For：参与计算的表单的id名称，以空格隔开

2）参加计算的表单



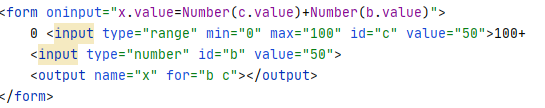
注：需要加id值

3）form标签



用form标签包裹，标签外使用事件并做计算

4）整个文档结构及效果图





### 3、keygen标签

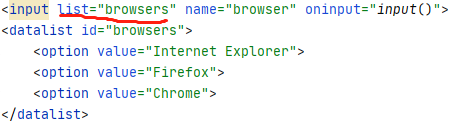
## 十一、表单属性

### 1、autofocus：自动获取焦点



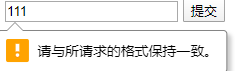


### 2、list属性：配合datalist标签使用



### 3、pattern：规定格式





### 4、multiple

multiple 属性规定<input> 元素中可选择多个值。

**注意:** multiple 属性适用于以下类型的 <input> 标签：email 和 file:

## 十一、语义元素

header：头部区域

nav：导航部分

section：文档中的节。比如章节、页眉、页脚或文档中的其他部分

article：标签定义独立的内容

aside：标签定义页面主区域内容之外的内容（比如侧边栏）

figcaption：<figure> 元素的标题

figure：标签规定独立的流内容（图像、图表、照片、代码等

footer：底部区域

注：以上元素均数据块状元素，使用前需要定义

header, section, footer, aside, nav, article, figure{

display: block;

}

## 十二、web储存

但是Web 存储需要更加的安全与快速. 这些数据不会被保存在服务器上，但是这些数据只用于用户请求网站数据上.它也可以存储大量的数据，而不影响网站的性能。数据以 键/值 对存在, web网页的数据只允许该网页访问使用。

### 1、浏览器支持

Internet Explorer 8+, Firefox, Opera, Chrome, 和 Safari支持Web 存储。

### 2、储存数据的两个对象：localStorage 和 sessionStorage

localStorage：用于**长久保存**整个网站的数据，保存的数据**没有过期时间**， 直到**手动去除**。

sessionStorage - 用于**临时保存**同一窗口(或标签页)的数据，在**关闭窗口** 或标签页之后将会**删除这些数据**。

### 3、检测是否支持web储存

if(Storage){

console.log('支持localStorage,sessionStorage对象!')

} else {

console.log('不支持web存储。')

}



### 4、ocalStorage 对象

1）创建



或



2）查找



或



3）删除

 删除单个数据

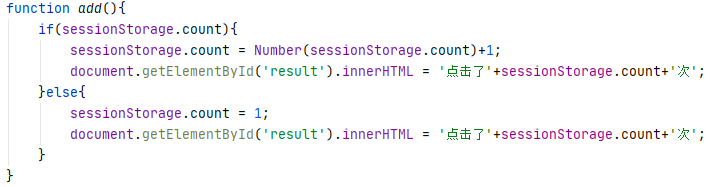
 删除所有数据

4）整个文档结构





### 5、sessionStorage



### 6、Web Storage 开发一个简单的网站列表程序

1）实现功能

* 可以输入网站名，网址，以网站名作为key存入localStorage；
* 根据网站名，查找网址；
* 列出当前已保存的所有网站；

## 十三、web SQL数据库

### 1、核心方法

**1）openDatabase**：使用现有的数据库或者新建的数据库创建一个数据库对象。

**2）transaction**：这个方法让我们能够控制一个事务，以及基于这种情况执行提交或者回滚。

3）**executeSql**：这个方法用于执行实际的 SQL 查询。

### 2、打开数据库：openDatabase方法

1）openDatabase() 方法对应的五个参数说明：

a、数据库名称；

b、版本号；

c、描述文本；

d、数据库大小；

e、创建回调

2、具体操作

## 十四、mate标签



*1、<!-- 为搜索引擎定义关键词-->* <meta name="keywords"content="HTML, CSS, JavaScript">

2、*<!-- 编码-->* <meta charset="UTF-8">  
 *3、<!-- 每10秒钟刷新当前页面:-->* <meta http-equiv="refresh"content="10">  
 *4、<!-- 为网页定义描述内容-->* <meta name="description"content="免费 Web &编程 教程">  
 *5、<!-- 定义网页作者-->* <meta name="author"content="Runoob">

## 十五、[热点链接](https://www.cnblogs.com/jaryup/archive/2012/07/26/2609576.html)：img、map、area简单用法

<img src="图形文件名" usemap="#图的名称">

<map name="图的名称">

<area shape=形状 coords=区域座标列表 href="URL资源地址">

<area shape=形状 coords=区域座标列表 href="URL资源地址">

</map>

1、shape -- 定义热点形状

shape=rect：   矩形

shape=circle：圆形

shape=poly：  多边形

2、coords -- 定义区域点的坐标

1）矩形：必须使用四个数字，前两个数字为左上角座标，后两个数字为 右下角座标例：<area shape=rect coords=100,50,200,75 href="URL">

2）圆形：必须使用三个数字，前两个数字为圆心的座标，最后一个数字 为半径长度：例：<area shape=circle coords=85,155,30 href="URL">  
 3）任意图形（多边形）：将图形之每一转折点座标依序填入，例：<area shape=poly coords=232,70,285,70,300,90,250,90,200,78 href="URL">



## 十六、HTML框架Iframe 标签

多页面网页：在一个网页中可以显示另一个网页的内容





1、frameborder：表示边框，0为无边框

2、src：连接网页的链接

3、width、height：宽高

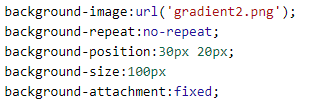
# CSS3内容

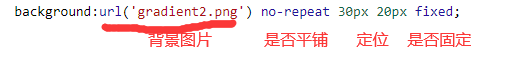
## 一、css基本内容

### 1、类选择器：class

p.center{text-align:center} 表示所有p标签的center类居中

### 2、背景：background





### 3、文本转换：text-transform

属性值：

uppercase：大写

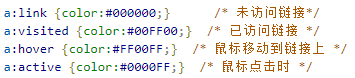
lowercase：小写

capitalize：首字母大写

### 4、文本缩进：text-intend

属性值：50px或2em（2个字符）

### 5、连接样式



一般的顺序为link->visited->hover->active

注意: hover必须在:link和 a:visited之后定义才有效.

注意:active必须在hover之后定义是有效的

### 6、表格

1） border-collapse：collapse：设置表格的边框是否被折叠成一个单一的边框或隔开。

2）文字对齐

Text-align：水平对齐

Vertical-align：垂直对齐

3）表格合并

colspan=”所要合并的单元格列数”：水平合并

rowspan=” 所要合并的单元格行数”：垂直方向合并

### 7、两种隐藏方式

display：none 隐藏但不占位

visibility：hidden 隐藏占位

### 8、定位

1）static：默认值，即没有定位，遵循正常的文档流对象。

2）fixed：元素的位置相对于浏览器窗口是固定位置。

3）relative：相对定位元素的定位是相对其正常位置。

4）absolute：相对于最近的已定位父元素，如果元素没有已定位的父元 素，那么它的位置相对于<html>

5）sticky：基于用户的滚动位置来定位

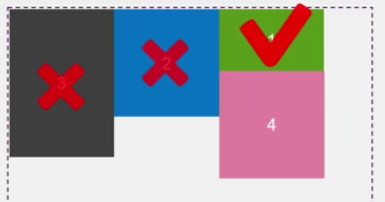
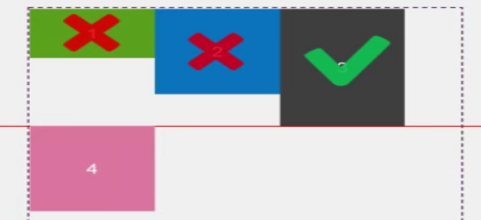
### 9、overflow

|  |  |
| --- | --- |
| **值** | **描述** |
| visible | 默认值。内容不会被修剪，会呈现在元素框之外。 |
| hidden | 内容会被修剪，并且其余内容是不可见的。 |
| scroll | 内容会被修剪，但是浏览器会显示滚动条以便查看其余的内容。 |
| auto | 如果内容被修剪，则浏览器会显示滚动条以便查看其余的内容。 |
| inherit | 规定应该从父元素继承 overflow 属性的值。 |

### 10、浮动

1）浮动元素排列方式

浮动元素排列时，只参考前一个元素即可



2）清除浮动引起的高度塌陷

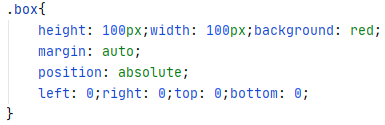
a、给父元素一个固定高度

b、父元素中添加overflow：auto/hidden

c、在浮动元素下边加个空div，并给样式clear：both

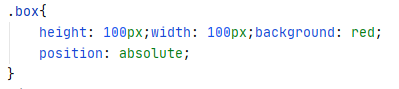
### 11、水平垂直居中

1）纯css----margin：auto

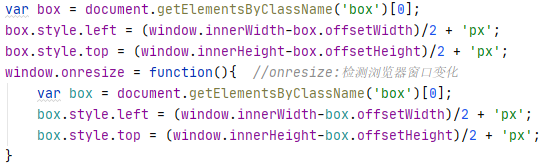




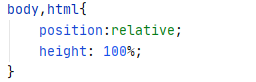
2）用js脚本

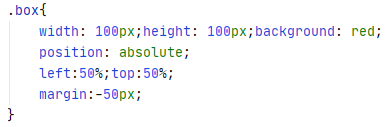






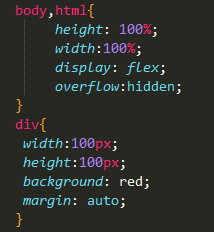
3）纯css---一直div的宽高



或



4）弹性盒



### 12、组合选择符

1）后代选择器(以空格分隔)



2）子元素选择器(以大于号分隔）



3）相邻兄弟选择器（以加号分隔）



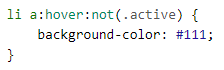
获取em后面的第一个p标签，必须是同级的元素

4）普通兄弟选择器（以破折号分隔）



获取em标签后面所有的p标签，必须是同级的元素

5）相反选择器

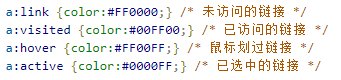


说明：li中的a标签不是active类的被选中

### 13、伪类

1）伪类

a、anchor伪类



b、伪类选择器

first-child/last-child

nth-chihd（n）/ nth-last-chihd（n）

nth-of-type（n）

nth-chihd（odd）奇数/ nth-chihd（even）偶数

nth-child（an+b）

c、lang伪类

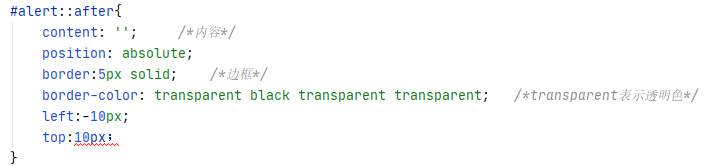
lang 伪类使你有能力为不同的语言定义特殊的规则

2）伪元素

a、:first-line：规定第一行样式

b、:first-letter：第一个字或首字母样式

c、before、after：在元素的前面/后面添加新的内容



|  |  |  |
| --- | --- | --- |
| **选择器** | **示例** | **示例说明** |
| [:link](https://www.runoob.com/cssref/sel-link.html) | a:link | 选择所有未访问链接 |
| [:visited](https://www.runoob.com/cssref/sel-visited.html) | a:visited | 选择所有访问过的链接 |
| [:active](https://www.runoob.com/cssref/sel-active.html) | a:active | 选择正在活动链接 |
| [:hover](https://www.runoob.com/cssref/sel-hover.html) | a:hover | 把鼠标放在链接上的状态 |
| [:focus](https://www.runoob.com/cssref/sel-focus.html) | input:focus | 选择元素输入后具有焦点 |
| [:first-letter](https://www.runoob.com/cssref/sel-firstletter.html) | p:first-letter | 选择每个<p> 元素的第一个字母 |
| [:first-line](https://www.runoob.com/cssref/sel-firstline.html) | p:first-line | 选择每个<p> 元素的第一行 |
| [:first-child](https://www.runoob.com/cssref/sel-firstchild.html) | p:first-child | 选择器匹配属于任意元素的第一个子元素的 <p> 元素 |
| [:before](https://www.runoob.com/cssref/sel-before.html) | p:before | 在每个<p>元素之前插入内容 |
| [:after](https://www.runoob.com/cssref/sel-after.html) | p:after | 在每个<p>元素之后插入内容 |
| [:lang(](https://www.runoob.com/cssref/sel-lang.html)*[language](https://www.runoob.com/cssref/sel-lang.html)*[)](https://www.runoob.com/cssref/sel-lang.html) | p:lang(it) | 为<p>元素的lang属性选择一个开始值 |

### 14、提示工具（tooltip）

效果图：非常的简单

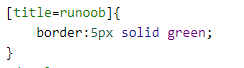


### 15、属性选择器

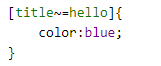
1）属性选择器：



2）属性和值选择器



3）属性和值的选择器-多值



### 16、表单选择器

1）type属性



2）输入框(input) 聚焦



3）checked属性

input[type=cheekbox]:checked

### 17、计数器

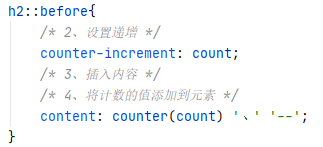
1）counter-reset - 创建或者重置计数器



2）counter-increment - 递增变量

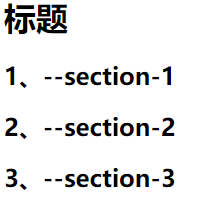
3）content - 插入生成的内容

4）counter() 或 counters() 函数 - 将计数器的值添加到元素



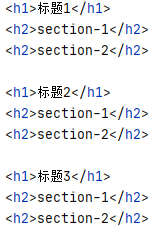
5）整体结构





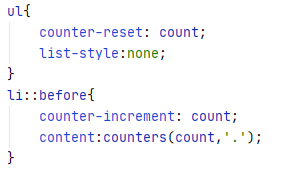
**注：content中的字符串之间用空格连接**

6）计数器的嵌套

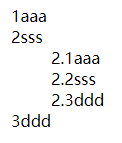
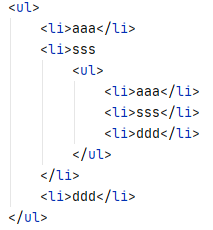




7）计数器在列表中创建



**counters方法会自动计算count值并将count之间用””.”隔开**



## 二、CSS3新增内容

### 1、边框

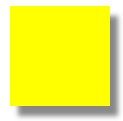
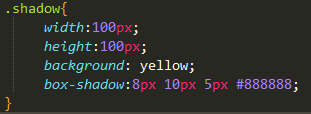
1）圆角（border-radius）



2）盒阴影（box-shadow）







3）边界图片（border-image）

a、设置图片的宽高



b、设置图片的边框**宽度**并给**实线**和**透明色**（边框的宽度决 定了边框的大小）



c、设置图片框



最后一个属性有3个属性值：平铺(repeat)、铺满(roun d)、拉伸(stretch)

### 2、背景

可以给一个元素添加**多个**背景图片，并**分别**设置这些背景图片 的样式，之间用**逗号**隔开。





以后再说

**获取网页背景图片的方法：找到css中的背景图片，右击后点 击open in new tab即可获取到。**

### 3、渐变：background-image:linear-gradient(color1，color2，…)

1）默认情况：从上到下





2）从下到上





3）从左到右





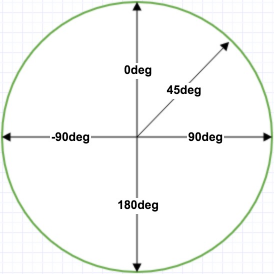
4）对角渐变





5）角度渐变

0deg为从上到下，顺时针为正，逆时针为负





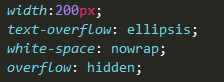
### 4、文本效果

1）文字阴影：text-shadow



2）盒子阴影：box-shadow

3）text-overflow



4）word-wrap



### 5、字体设置

以后再说

### 6、2D和3D转换

1）2D效果

a、transform：translate（100px） 移动

b、transform：rotate（90deg）旋转

c、transform：scale（1.5） 缩放

d、transform：skew（30deg）扭曲

e、组合



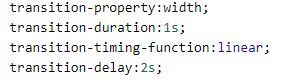
在运动的远胜于上需加上一个动画



2）3D效果

### 7、过度（transition）





[transition-timing-function](https://www.runoob.com/cssref/css3-pr-transition-timing-function.html" \o "CSS3 transition-timing-function 属性)的属性：

linear：以相同速度开始至结束的过渡效果（等于 cubic-bezier(0,0,1,1)）。

ease：慢速-快速-慢速（cubic-bezier(0.25,0.1,0.25,1)）。

ease-in：规定以慢速开始的过渡效果（等于 cubic-bezier(0.42,0,1,1)）。

ease-out：规定以慢速结束的过渡效果（等于 cubic-bezier(0,0,0.58,1)）

ease-in-out：以慢速开始和结束（等于cubic-bezier(0.42,0,0.58,1)）

cubic-bezier(n,n,n,n)：可能的值是 0 至 1 之间的数值

### 8、动画（keyframes）

|  |  |  |
| --- | --- | --- |
| [@keyframes](https://www.runoob.com/cssref/css3-pr-animation-keyframes.html" \o "CSS3 @keyframes 规则) | 规定动画。 | 属性值 |
| [animation](https://www.runoob.com/cssref/css3-pr-animation.html" \o "CSS3 animation 属性) | 所有动画属性的简写属性，除了 animation-play-state 属性。 |  |
| [animation-name](https://www.runoob.com/cssref/css3-pr-animation-name.html" \o "CSS3 animation-name 属性) | 规定 @keyframes 动画的名称。 |  |
| [animation-duration](https://www.runoob.com/cssref/css3-pr-animation-duration.html" \o "CSS3 animation-duration 属性) | 规定动画完成一个周期所花费的秒或毫秒。默认是 0。 | 秒或毫秒数 |
| [animation-timing-function](https://www.runoob.com/cssref/css3-pr-animation-timing-function.html" \o "CSS3 animation-timing-function 属性) | 规定动画的速度曲线。默认是 "ease"。 | 同上[transition-timing-function](https://www.runoob.com/cssref/css3-pr-transition-timing-function.html" \o "CSS3 transition-timing-function 属性)的属性 |
| [animation-fill-mode](https://www.runoob.com/cssref/css3-pr-animation-fill-mode.html" \o "CSS3 animation-fill-mode 属性) | 规定当动画不播放时（当动画完成时，或当动画有一个延迟未开始播放时），要应用到元素的样式。 |  |
| [animation-delay](https://www.runoob.com/cssref/css3-pr-animation-delay.html" \o "CSS3 animation-delay 属性) | 规定动画何时开始。默认是 0。 | 秒或毫秒数 |
| [animation-iteration-count](https://www.runoob.com/cssref/css3-pr-animation-iteration-count.html" \o "CSS3 animation-iteration-count 属性) | 规定动画被播放的次数。默认是 1。 | 规定动画被播放的次数。默认是 1，infinite为无限次 |
| [animation-direction](https://www.runoob.com/cssref/css3-pr-animation-direction.html" \o "CSS3 animation-direction 属性) | 规定动画是否在下一周期逆向地播放。默认是 "normal"。 | Normal：默认值。动画按正常播放  Reverse：动画反向播放  Alternate：奇数次正向播放，偶数次反向播放。  alternate-reverse：奇数次反向播放，偶数次正向播放。 |
| [animation-play-state](https://www.runoob.com/cssref/css3-pr-animation-play-state.html" \o "CSS3 animation-play-state 属性) | 规定动画是否正在运行或暂停。默认是 "running"。 | Running：默认为运行  Paused：暂停动画 |

### 9、多列

1）设置列数

column-count:2



2）设置列之间的距离

column-gap: 30px



3）设置列边框

column-rule-style:solid



4）设置边框颜色

column-rule-color:red



5）设置边框的厚度，样式，颜色



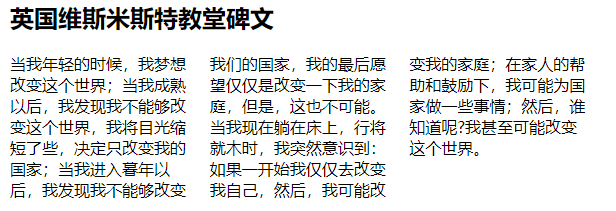
6）设置每个列的宽度



注：当column-count存在值，列数执行column-count，不 存在是，列数执行column-width

7）指定元素跨越多少列





### 10、用户界面

1）resize：用户可以调整







2）box-sizing：调整方框大小

3）outline-offset ：外形修饰



内边框：border

外边框：outline

两边框之间的间距：outline-offset

|  |  |
| --- | --- |
| **属性** | **说明** |
| [appearance](https://www.runoob.com/cssref/css3-pr-appearance.html) | 允许您使一个元素的外观像一个标准的用户界面元素 |
| [box-sizing](https://www.runoob.com/cssref/css3-pr-box-sizing.html) | 允许你以适应区域而用某种方式定义某些元素 |
| [icon](https://www.runoob.com/cssref/css3-pr-icon.html) | 为创作者提供了将元素设置为图标等价物的能力。 |
| [nav-down](https://www.runoob.com/cssref/css3-pr-nav-down.html) | 指定在何处使用箭头向下导航键时进行导航 |
| [nav-index](https://www.runoob.com/cssref/css3-pr-nav-index.html) | 指定一个元素的Tab的顺序 |
| [nav-left](https://www.runoob.com/cssref/css3-pr-nav-left.html) | 指定在何处使用左侧的箭头导航键进行导航 |
| [nav-right](https://www.runoob.com/cssref/css3-pr-nav-right.html) | 指定在何处使用右侧的箭头导航键进行导航 |
| [nav-up](https://www.runoob.com/cssref/css3-pr-nav-up.html) | 指定在何处使用箭头向上导航键时进行导航 |
| [outline-offset](https://www.runoob.com/cssref/css3-pr-outline-offset.html) | 外轮廓修饰并绘制超出边框的边缘 |
| [resize](https://www.runoob.com/cssref/css3-pr-resize.html) | 指定一个元素是否是由用户调整大小 |

### 11、图片

1）border-radius：图片圆角

2）缩略图

3）响应式图片

4）图片文本



5）卡片式图片



6）图片滤镜

filter: grayscale(100%);

7）图片Model（模态）

见菜鸟教程

### 12、按钮

动画随后在做

### 13、分页

随后在做

### 14、弹性盒子

1）父元素设置display：flex，表示启动弹性盒

2）子元素设置flex：1，子元素等分父元素的宽度

3）父元素设置flex-direction，子元素在父元素中的排列方式

row：横向从左到右排列（左对齐），默认的排列方式。

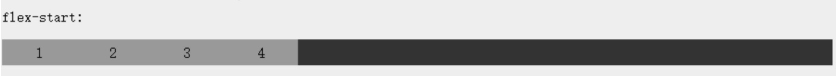
row-reverse：反转横向排列（右对齐，从后往前排，最后一项排在最前面。

column：纵向排列。

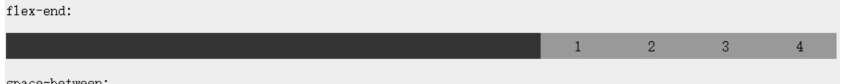
column-reverse：反转纵向排列，从后往前排，最后一项排在最上面。

4）父元素设置justify-content，子元素的对齐方式

flex-start：这个是默认值。横向从左到右。



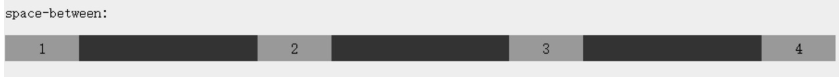
flex-end：横向从左到右。



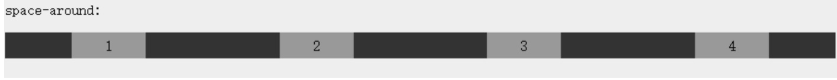
center：横向居中。



space-between：左右两边间距为0，中间的间距相等。

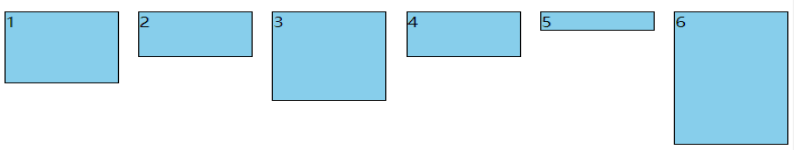


space-around：左右两边的间距为1份，子元素之间的间距为2份。

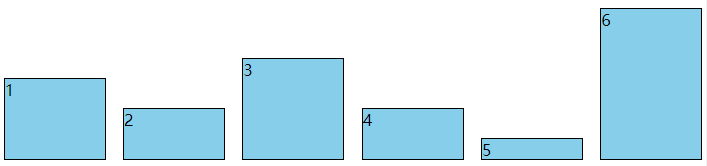


5）父元素设置align-items，

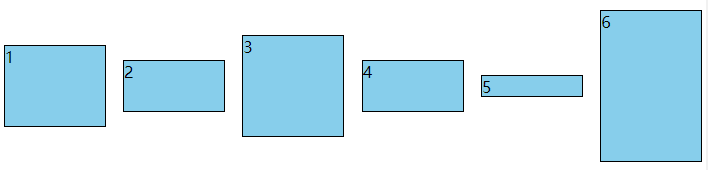
flex-start：弹性盒子元素的侧轴（纵轴）起始位置的边界紧靠住该行 的侧轴起始边界。



flex-end：弹性盒子元素的侧轴（纵轴）起始位置的边界紧靠住该行 的侧轴结束边界。



center：弹性盒子元素在该行的侧轴（纵轴）上居中放置。（如果该 行的尺寸小于弹性盒子元素的尺寸，则会向两个方向溢出相同的长度）。

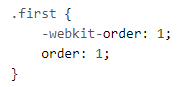


baseline：如弹性盒子元素的行内轴与侧轴为同一条，则该值与 'flex-start'等效。其它情况下，该值将参与基线对齐。

stretch：若指定侧轴大小的属性值为'auto'，则其值会使项目的边距盒的尺 寸尽可能接近所在行的尺寸，但同时遵照'min/max-width/height'属性的限制。 6）父元素添加flex-wrap：子元素之间的换行

nowrap - 默认， 弹性容器为单行。弹性子项可能会溢出容器。 wrap - 弹性容器为多行。该情况下弹性子项溢出的部分 wrap-reverse -反转 wrap 排列。从下到上：3,2,1列会被放置到 新行，子项内部会发生断行，从上到下：1,2,3列

7）子元素添加order：排序，负数表示向左移动，正数享向右



8）子元素添加：margin：auto

子元素全部居中

9）子元素设置：align-self，设置各个子元素的对齐方式

flex-start、flex-end、center，和前面的align-items

### 15、多媒体查询



表示屏幕宽度在700-1000px之间是的样式

# Javascript内容

## 一、基本属性

### 1、数据类型

基本数据类型：数字（number）、字符串（string）、布尔（boolean）、null、underfined、symbol

引用数据类型：对象（object）、数组（array）、函数（function）

### 2、字符串的属性

1）constructor：创建字符串属性的函数



2）length：字符串的长度

3）prototype：向对象添加属性和方法

2、字符串的方法

**常用方法（必背）**

1）charAt()：返回指定索引值的字符

2）charCodeAt()：返回指定索引值的ASCII值

3）concat ()：字符串拼接

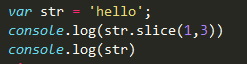
4）indexOf/search()：字符串查找，返回第一次出现的位置， 不存在返回-1，includes()则返回true或false

5）lastIndexOf()：返回字符最后一次出现的位置。

6）replace()：替换子串，返回替换之后的字符串，原字符串不 变。



7）slice/substring ()：字符串截取，参数为索引值，返回截取 的字符串，原字符串不变。



8）substr()：字符串截取，第一个参数为开始下标，第二个参 数为截取的字符串长度。

9）splic()：字符串分割，返回分割之后的数组



10）toLowerCase()、toUpperCase()：转化为大小写

11）trim()：移除字符串首尾的空格

12）toString()：返回字符串对象值

3、数据类型之间的转换

toString()/String()：转化为字符串

toExponential()：把对象的值转换为指数计数法

toFixed()：数字转换为字符串，结果的小数点后有指定位数的数字

toPrecision()：把数字格式化为指定的长度

number()：转化为数字类型

parseInt()：转化为整数型

parseFloat()：转化为浮点型

### 3、正则表达式

1）定义：正则表达式是由一些字符序列组成的搜索模式

2）修饰符

i：执行对大小写不敏感的匹配

g：执行全局匹配（查找所有匹配而非在找到第一个匹配后停止）

m：执行多行匹配

3）正则表达式模式

a、方括号用于查找某个范围内的字符：

[abc]：查找方括号之间的任何字符。

[0-9]：查找任何从 0 至 9 的数字。

(x|y)：查找任何以|分隔的选项。

b、元字符是拥有特殊含义的字符：

\d：查找数字，等同于[0-9]

\D：查找非数字

\s：查找空白字符。

\b：匹配单词边界。

\w：等同于[0-9,A-Z,a-z]

/^：开始匹配

$/：结束匹配

{m,n}：至少匹配m次，至多n次

c、量词

“.”：可匹配回车以外的所有字符



“n\*”：n可重复匹配（n可为0）



“+”：n最少匹配一次



“n?”：可匹配0或1次

4） RegExp 对象

创建一个RegExp对象：var patt1=new RegExp("e");

5）查找字符串中是否含有某个正则表达式

a、test()方法：判断是否存在

/e/.test(“helloworld”)，helloworld中存在e，返回true

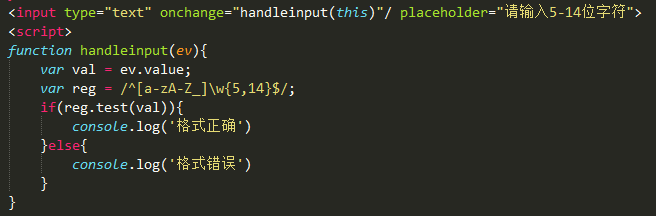
b、exec()方法

/el/.exec(“helloworld”)，返回原正则表达式，不存在返回null。

c、match()方法

'hello el'.match(/el/g)：返回符合条件的数组

6）正则表达式应用



### 4、错误处理

语法：

try{

执行A语句

}catch(error){ //创建或抛出异常

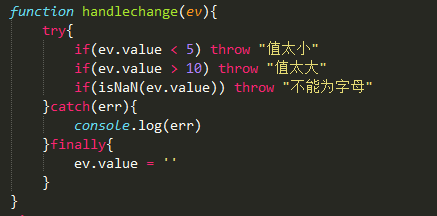
当A语句错误时，执行

}finally{

结束处理：无论执行try还是catch，最后都会执行finally

}

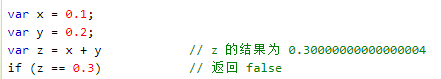
实践：



### 5、js使用误区

1）浮点类型计算

avaScript 中的所有数据都是以 64 位**浮点型数据(float)** 来存储。所有的编程语言，包括 JavaScript，对浮点型数据的精确度都很难确定，解决方法：





2）字符串断行问题

在字符串中直接使用回车换行是会报错的，解决方法：中间加反斜杠。



### 6、表单验证

1）document.forms：获取当前页面的所有表单

2）document.forms[0]：获取当前页面的第一个表单

3）document.forms[“myfrom”]：获取当前页面中name=“myfrom” 的表单

### 7、json数据与js数据之间的转换

[JSON.parse()](https://www.runoob.com/js/javascript-json-parse.html" \t "_blank)：用于将一个 JSON 字符串转换为 JavaScript 对象

[JSON.stringify()](https://www.runoob.com/js/javascript-json-stringify.html" \t "_blank)：用于将 JavaScript 值转换为 JSON 字符串。

### 8、void

Javascript:void(0)与#的区别

**#** 包含了一个位置信息，默认的锚是**#top** 也就是网页的上端。而javascript:void(0), 仅仅表示一个死链接。在页面很长的时候会使用 **#** 来定位页面的具体位置，格式为：**# + id**。如果你要定义一个死链接请使用 javascript:void(0) 。

## 二、函数

### 1、ES6函数

ES6中函数可自带参数，例如function func(a,b=10){函数主体}，若未给定b的值，则b=10，若给定b值，则b为给定的值。

### 2、闭包（使函数拥有的私有变量）

1）概念：是指有权访问另一个函数作用域中的变量的函数，创建闭包的最常见的方式就是在一个函数内创建另一个函数，通过另一个函数访问这个函数的局部变量。调用完一次后，函数没有被释放，整条作用域链上的局部变量都将得到保留。在本质上，闭包就是将函数内部和函数外部连接的一座桥梁。

2）闭包代码

function text(){

var a = 0;

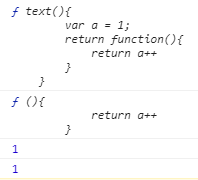
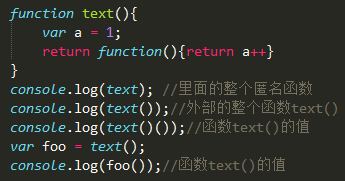
return function(){ alert(a++) }

}

var foo = text()

foo() //或text()() 在这里text()= function(){ alert(a++) }

foo() //或text()() 在这里text= function text(){ }



或

var foo = (function(){

var a = 0;

return function(){ alert(a++) }

})()

foo() //1

foo() //2

当我们不需要引用的时候直接b = null就可以了。

## 三、DOM结构

### 1、addEventListener()方法

1）给定元素添加事件

addEventListener的参数：

第一个参数是事件的类型 (不带””on”).

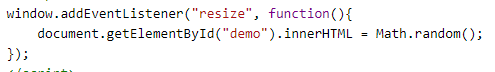
第二个参数是事件触发后调用的函数。

第三个参数是个布尔值用于描述事件是冒泡还是捕获。

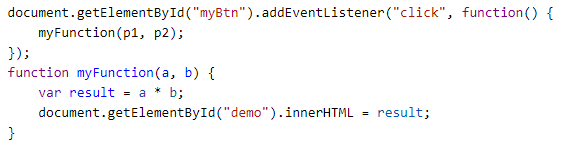
2）给同一个元素添加两个事件



3）向任何 DOM 对象添加事件监听（如window对象）



4）参数传递



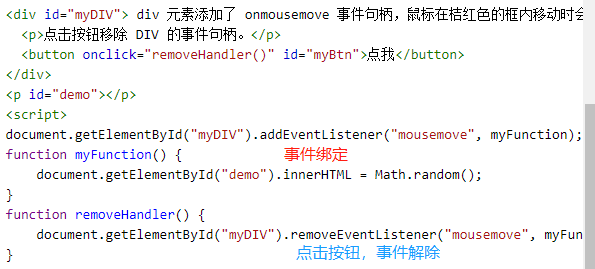
匿名函数不能直接带参数，可以执行带参数的函数

5）事件冒泡与捕获

true表示捕获：由外到内；

false表示冒泡：由内到外，默认是false；

6）事件解除removeEventListener()



### 2、dom元素

1）创建元素

document.createElement(‘p’)：创建一个标签

document.createTextNode(‘这是一段文字’)：创建一段文字

2）将创建的元素添加到文档当中

a、添加到父元素的末尾

父元素.appendChild(新创建的元素)

b、添加到指定元素的前面

父元素.insertBefore(新创建的元素,指定的元素)

3）删除已存在的元素节点

父元素.removeChil(要删除的节点)

4）替换元素

父元素.replaceChild(新节点,旧节点)

5）克隆节点

要克隆的节点.cloneNode() 只克隆标签

要克隆的节点.cloneNode(true) 标签和内容都克隆

6）节点的类型和节点的

|  |  |  |  |
| --- | --- | --- | --- |
| **节点类型** | **元素节点** | **属性节点** | **文本节点** |
| nodeName | 元素名称 | 属性节点 | #text |
| nodeType | 1 | 2 | 3 |
| nodeValue | Null | 属性值 | 文本内容 |

### 3、属性节点

1）获取属性

元素节点.getAttribute('属性')

2）修改属性

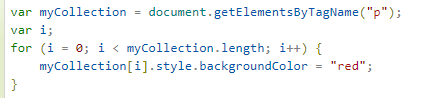
元素节点.setAttribute( ('属性',’属性值’)

3）删除属性节点

元素节点.removeAttribute('属性')

### 4、HTMLcollection对象

document.getElementsByTagName("p")，返回HTMLcollection对 象



### 5、NodeList 对象

document.querySelectorAll("p")，返回NodeList 对象

和HTMLcollection对象类似

## 四、对象

### 1、构造函数

1）构造函数继承

a、属性传递（apply和call）

Person.apply(this,name,age)或Person.apply(Person,name,age)

Person.call(this,[name,age])或Person.call(Person,[name,age])

b、方法传递

Student.prototype = Object.create(Person.prototype)

2）apply和call两种方法的比较

相同点：两种方法产生的效果一样

不同点：方法传递的参数不同， apply传入的是一个参数 数组，也就是将多个参数组合成为一个数组传入，而call则作 为call的参数传入（从第二个参数开始）

3）遍历

构造函数内部的方法是可以直接遍历出来的，外部的方法 必须继承之后，才能被遍历出来

其中i为键，person[i]为值



4）原型对象

添加属性  添加方法 

### 2、Number对象

1）数字

var pi=3.14;    // 使用小数点  
 var x=34;       // 不使用小数点

var y=123e5;    // 12300000（指数）

var z=123e-5;   // 0.00123（负指数）

2）精度（整数最多15位，小数最多17位）



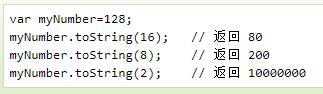


3）进制问题

a、如果前缀为 0，则 JavaScript 会把数值常量解释为八进 制数，如果前缀为 0 和 "x"，则解释为十六进制数。



b、默认情况下，JavaScript 数字为十进制显示。但是你可 以使用 toString() 方法 输出16进制、8进制、2进制。



4）无穷大（Infinity）

经测试310位以后会变成Infinity，且Infinity === Infinity

5）NaN（不是一个数字）

6）number的属性和方法，见菜鸟教程number对象

### 3、日期对象（Date）

1）创建日期对象

var date = new Date()

2）获取年月日等时间

年：date.getFullYear();

getFullYear和getYear的区别：getFullYear返回完整的年份， getYear 返回的数为完整的年份-1900

季度：parseInt(date.getMonth()/3);

月：date.getMonth()+1;

日：date.getDate();

时：date.getHours();

分：date.getMinutes();

秒：date.getSeconds()

星期：date.getDay();

3）获取固定格式的日期

没看懂

### 4、数组对象（Array）

1）数组的属性

a、数组的长相：有长有短，也可以为空

b、数组的长度：arr.length

c、数组的索引（下标）：arr[0]

d、prototype（原型属性）

2）数组的方法

a、连接两个数组：arr1.concat(arr2)

b、将数组拼接成字符串：arr.join(‘+’)

c、将数组转换为字符串：arr.toString()

d、添加数组

末尾添加：arr.push()

末尾删除：arr.pop()

头部添加：arr.unshift()

头部删除：arr.shift()

e、数组排序

升序排列：arr.sort()

降序排列：arr.sort().resvese()

倒序排列：arr.reverse()

f、添加，修改，删除：返回截取下来的元素

arr.splice(开始的下标,截取的长度,替换的数组…)

g、数组截取：返回截取下来的元素

arr.slice(开始下标,结束下标)

**ES6添加的数组方法**

a、数组查找（存在则返回下标，不存在返回-1）

arr.indexOf()

b、遍历数组（forEach）

arr.forEach(function(item,index,array){执行的语句})

item：当前遍历到的元素

index：当前遍历到的元素下标

array：当前的数组

c、映射（map）（遍历—>执行—>返回）

arr. map (function(item,index,array){执行的语句})

item：当前遍历到的元素

index：当前遍历到的元素下标

array：当前的数组

d、过滤（filter）

arr. map (function(item,index,array){过滤要求})

item：当前遍历到的元素

index：当前遍历到的元素下标

array：当前的数组

e、some

arr. some (function(item,index,array){return要求})



f、every

arr. every(function(item,index,array){return要求})

g、归并（reduce）

arr.reduce(function(pre,next,index,array)

pre：开始一个是下标为0的数组元素，后面为return 后面返回的值

next：下标从1开始的数组元素

### 5、算数对象（Math）

1）Math属性

Math.E：自然对数e

Math.PI：圆周率π

2）Math方法

|  |  |
| --- | --- |
| **方法** | **描述** |
| Mat.[ahbs(x)](https://www.w3school.com.cn/jsref/jsref_abs.asp) | 返回数的绝对值。 |
| Math.[ceil(x)](https://www.w3school.com.cn/jsref/jsref_ceil.asp) | 对数进行上舍入。 |
| Math.[floor(x)](https://www.w3school.com.cn/jsref/jsref_floor.asp) | 对数进行下舍入。 |
| Math.[max(x,y)](https://www.w3school.com.cn/jsref/jsref_max.asp) | 返回 x 和 y 中的最高值。 |
| Math.[min(x,y)](https://www.w3school.com.cn/jsref/jsref_min.asp) | 返回 x 和 y 中的最低值。 |
| Math.[random()](https://www.w3school.com.cn/jsref/jsref_random.asp) | 返回 0 ~ 1 之间的随机数。 |
| Math.[round(x)](https://www.w3school.com.cn/jsref/jsref_round.asp) | 把数四舍五入为最接近的整数。 |
| Math.[sqrt(x)](https://www.w3school.com.cn/jsref/jsref_sqrt.asp) | 返回数的平方根。 |

## 五、window对象

### 1、window对象的属性

1）内部窗口宽度和高度：window.innerWidth/innerHeight

2）外部窗口宽度和高度：window.outWidth/outHeight

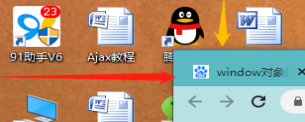
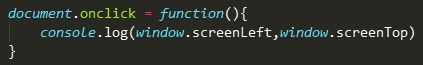
3）返回当前页面相对于窗口显示区左上角的X位置：

window.pageXoffset/ pageYoffset



4）声明了窗口的左上角在屏幕上 的 X 坐标和 Y 的坐标（支 持IE、Safari、Chrome 和 Opera）

window. screenLeft / screenTop



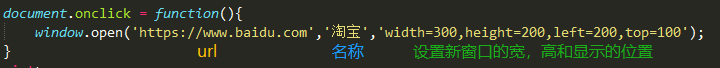
5）支持Chrome、Firefox 和 Safari

window. screenX / screenY

### 2、window对象的方法

1）open()：打开指定链接或页面

参数：1、url；2、名称；3、新窗口的大小和位置



2）close()：关闭窗口



3）scrollBy()：跳到指定的位置（可叠加，可执行多次）

参数：1、跳转的X方向；2、跳转的Y方向



4）scrollTo()：跳到指定的位置（不可叠加，只执行一次）



### 3、window下的screen对象

可用的屏幕高度及宽度： screen.availWidth/ availHeight

### 4、window下的location对象

window.location 对象用于获得当前页面的地址 (URL)，并把浏览器重定向到新的页面。

1）location对象的属性

location.host：主机名+端口号

location.hash：锚点

location.hostname：主机名

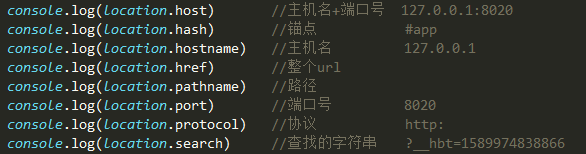
location. Href：整个url

location. Pathname：路径

location. Port：端口号

location. Protocol：协议

location.search：查找的字符串



2）location对象的方法

a、assign()：跳转到指定页面（有history，可返回）

b、reload()：刷新

reload(true)：忽略已缓存的数据，从头开始加载

c、replace()：跳转到指定页面（没有history，不可返回）

### 5、window下的history对象

1）history对象的属性

history.length

2）history对象的方法

history.back()：后退一条

history.forward()：前进一条

history.go()：参数为负，表示后退；参数为0，表示刷新； 参数为正，表示前进

### 6、window下的Navigator对象

1）Navigator对象属性

navigator.[appCodeName](https://www.w3school.com.cn/jsref/prop_nav_appcodename.asp)：返回浏览器的代码名

navigator.[appMinorVersion](https://www.w3school.com.cn/jsref/prop_nav_appminorversion.asp)：返回浏览器的次级版本

navigator.[appName](https://www.w3school.com.cn/jsref/prop_nav_appname.asp)：返回浏览器的名称

navigator.[appVersion](https://www.w3school.com.cn/jsref/prop_nav_appversion.asp)：返回浏览器的平台和版本信息

navigator.[browserLanguage](https://www.w3school.com.cn/jsref/prop_nav_browserlanguage.asp)：返回当前浏览器的语言

navigator.[cookieEnabled](https://www.w3school.com.cn/jsref/prop_nav_cookieenabled.asp)：返回指明浏览器中是否启用 cookie 的布尔值

navigator.[cpuClass](https://www.w3school.com.cn/jsref/prop_nav_cpuclass.asp)：返回浏览器系统的 CPU 等级

navigator.[onLine](https://www.w3school.com.cn/jsref/prop_nav_online.asp)：返回指明系统是否处于脱机模式的布尔值

navigator.[platform](https://www.w3school.com.cn/jsref/prop_nav_platform.asp)：返回运行浏览器的操作系统平台

navigator.[systemLanguage](https://www.w3school.com.cn/jsref/prop_nav_systemlanguage.asp)：返回 OS 使用的默认语言

navigator.[userAgent](https://www.w3school.com.cn/jsref/prop_nav_useragent.asp)：返回由客户机发送服务器的 user-agent 头部的值

navigator.[userLanguage](https://www.w3school.com.cn/jsref/prop_nav_userlanguage.asp)：返回 OS 的自然语言设置

**navigator.geolocation：是否支持定位**

2）Navigator对象方法

navigator.[javaEnabled()](https://www.w3school.com.cn/jsref/met_nav_javaenabled.asp)：规定浏览器是否启用Java

navigator.[taintEnabled()](https://www.w3school.com.cn/jsref/met_nav_taintenabled.asp)：规定浏览器是否启用数据污点 (data tainting)。

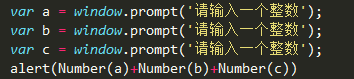
### 7、弹窗

1）alert()

2）confirm()：创建一个有选择的弹窗



3）prompt()：显示可提示用户输入的对话框



### 8、计时事件

1）按照指定的周期(以毫秒计)来调用函数或计算表达式

开启定时器：var timer = setInterval(function(){ },毫秒)

解除定时器：clearInterval(timer)

2）在指定的毫秒后调用函数或计算表达式

开启定时器：var timer = setTimeout(function(){ },毫秒)

解除定时器：clearTimeout(timer)

### 9、Cookie

Cookie 是一些数据, 存储于你电脑上的文本文件中。当 web 服务器向浏览器发送 web 页面时，在连接关闭后，服务端不会记录用户的信息。Cookie 的作用就是用于解决 "如何记录客户端的用户信息"

1）获取cookie信息

document.cookie

2）设置cookie

document.cookie = ‘username=honey’

3）设置cookie的有效期

var oDate = new Date(); //创建日期对象

oDate.setDate(oDate.getDate()+3) //设置日期

document.cookie = ‘username=honey;expires+’date //创建cookie

4）修改cookie（直接覆盖即可）

document.cookie = 'username=xiaomiong;expires+'+oDate

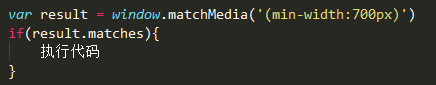
5）删除cookie

var oDate = new Date(); //创建日期对象

oDate.setDate(oDate.getDate()+n) //设置日期（n>过期值即可）

document.cookie = ‘username=honey;expires+’date //删除cookie

## js媒体查询



# jquery内容

## 一、jQuery对象

### 1、jquery加载模式

1）js加载模式

Window.onload = function(){}

特点：只能有一个，后面会覆盖前面；所有dom加载完，img等资源加载完后再加载

2）jquery加载模式

$(document).ready(function(){})

jQuery(document).ready(function(){})

$( function(){}) //推荐

jQuery (function(){})

特点：可有多个，可能在图片加载前加载

### 2、jQuery中$符号的冲突问题

1）释放jQuery的使用

jQuery.noConflict()

释放之后用jQuery

2）自定义jQuery中的符号

var a = jQuery.noConflict()

### 3、jQuery中的核心函数（接收4中参数）

1）函数

入口函数相当于$(document).ready(function(){})

2）字符串

a、字符串选择器

获取dom元素

b、代码片段

创建一个dom节点

3）DOM 元素

返回一个jQuery对象，**jQuery实质上是一个伪数组**

### 4、静态方法和实例方法的区别

1）静态方法

加载类上，通过类来调用

2）实例方法

加载类的原型上，通过实例对象来调用

### 5、jQuery中的静态方法

#### 1）each（能遍历伪数组）

**参数**：1.当前遍历到的索引；2.当前遍历到的元素

**用法**：$.each(arr,function(index,value){})

**返回值**：默认返回遍历的对象

**注意**：参数位置, 能遍历伪数组，回调函数中不可加return定义返回值

#### 2）map（能遍历伪数组）

**参数**：1.当前遍历到的元素；2.当前遍历到的索引

**用法**：$.map(arr,function(index,value){})

**返回值**：默认返回一个空对象

**注意**：能遍历伪数组，回调函数中可加return定义返回值

#### 3）trim（去字符串两端的空格）

用法：$.trim(str)

返回值：去除之后的新字符串

#### 4）isWindow（判断一个对象是不是window对象）

用法：$. isWindow()

返回值：true/false

#### 5）isArray（判断一个对象是不是数组）

用法：$. isArray ()

返回值：true/false

#### 6）isFunction（判断一个对象是不是函数）

用法：$. isFunction ()

返回值：true/false

#### 7）holdReady（暂停，回复ready事件）

用法：$. holdReady (true)：暂停

$. holdReady (false)：恢复

### 6、属性上的方法

object.attr()：添加，修改，获取属性

添加，修改时获取全部的标签，获取时只选中第一个标签

object.removeAttr()：删除属性

删除时获取全部的标签

object.prop()：添加，修改，获取属性

object.removeProp()：删除属性

注：两个都能获取到标签的属性，prop返回true/false，attr返回属性值

### 7、类操作方法

addClass()：添加class

removeClass()：删除class

toggleClass()：切换class

### 8、文本值操作

.text()：添加text文本

.html()：添加html代码片段

.val()：添加value属性值

### 9、操作css样式

#### 1）设置样式

a.简单设置

.css(‘width’,’100px’)

b.链式设置

.css(‘width’,’100px’) .css(‘height,’100px’)

c.对象设置

.css({width,’100px’, height,’100px’})

#### 2）获取css样式

.css(‘width’)

#### 3）操作尺寸的方法

a.获取尺寸

.width()

b.设置尺寸

.width(‘100px’)

4）位置操作方法

a.元素距离窗口的距离.offset()

获取：.offset().left

设置：.offset({left:10})

b.元素距离定位元素的距离.position()

获取：. position ().left

设置：不能设置，可通过 .css({left:’10px’})

### 10、滚动设置（scrollTop）

获取：$(‘div’).scrollTop 或 $(‘document’).scrollTop，元素或页面大的滚动位置

设置：$(‘div’).scrollTop(200)

### 11、事件

#### 1）事件绑定和解绑

绑定事件

a）$(‘button’).click(function(){函数主体})

b）$(‘button’).on(‘click’,function(){})

移除事件

$(‘button’).off() 参数：1.事件；2.函数

传参有三种情况：不传参表示移除所有事件；传一个参数表示移除该事件；传两个参数表示移除该事件上的函数

#### 2）事件冒泡和默认行为

阻止事件冒泡：return false / ev.stopPropagation()

阻止默认行为：return false / ev.preventDefault

#### 3）自动触发事件

$(‘div’).trigger(‘click’)：自动调用div的click事件，但是会产生事件冒泡，也会触发默认行为

$(‘div’).reiggerhandler(‘click’)：不会产生事件冒泡，也不会触发默认行为

注意：调用a标签的自动触发事件且需要触发默认行为时，需要将a标签中的内容永span包裹，触发span的事件即可

#### 4）自定义事件

$(‘div’).on(‘myEvent’,function(){})

$(‘div’).trigger(‘myEvent’)

#### 5）事件命名空间

设置命名空间

$(‘div’).on(‘click.zs’,function(){})

$(‘div’).on(‘click.ls,function(){})

调用

$(‘div’)’).trigger(‘click.ls’)

注意：子元素带有命名空间触发事件后，带有命名空间的父元素会冒泡，而父元素没有命名空间的事件不会冒泡

子元素不带有命名空间触发事件后，带有命名空间的父元素和没有命名空间的父元素都会冒泡

#### 6）事件委托

事件委托：找一个在入口函数执行之前就有的元素来监听动态创建添加的元素

$(‘ul’).delegate(‘li’, ‘click’, function(){console.log($(‘this’))})

#### 7）鼠标移入移出事件的问题

mouseover和mouseout有bug，一般用mouseenter和mouseleave

### 12、元素节点的获取

#### 1）祖先元素

Parent()：返回父元素节点

Parents()：返回所有祖先元素节点

parentsUntil(参数)：返回介于两个给定元素之间的所有祖先元素

#### 2）后代元素

children()：返回子元素

find()：返回后代元素

#### 3）同胞元素

prev()：返回上一个元素

prevAll()：返回前面所有的元素

prevUntil(参数)：返回介于两个给定元素之间的所有同胞元素

next()：返回下一个元素

nextAll()：返回后面所有的同胞元素

nextUntil：返回介于两个给定元素之间的所有同胞元素

siblings()：返回所有同胞元素

#### 4）过滤

first()：返回选中的首个元素

last()：返回选中的末位元素

eq()：返回指定下标的元素

### 13、元素节点的添加和删除

1）添加

开头添加：

prepend()：父元素. prepend (‘要添加的节点’)

before()：同胞元素.before(‘要添加的节点’)

末尾添加：

append()：父元素.append(‘要添加的节点’)

after()：同胞元素. after (‘要添加的节点’)

2）删除

remove()：要删除的元素.remove() -------删除整个元素

empty()：要删除的元素. empty () --------清空元素中的内容

### 14、动画

1）隐藏显示，淡入淡出

hidden()/show()/toggle()：隐藏/显示/取反

fadeIn ()/fadeOut ()/fadeToggle()：淡入/淡出/取反

slideDown()/slideUp()/slideToggle ()：向下/向上/取反

参数：1.动画时间；2.回调函数

2）动画（animate，支持链式调用）

$("div").animate({

         left:'250px',

         top:'200px',

     }).animate({

         width: '200px',

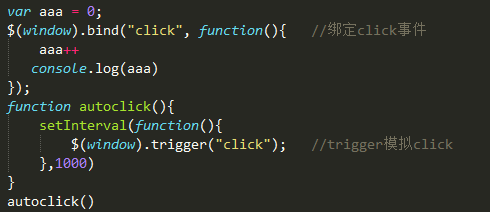
         height: '50px',

         opacity: '0.5'

     });

参数：1.属性；2.动画时间；3.回调函数

一、鼠标事件模拟



# 面试内容

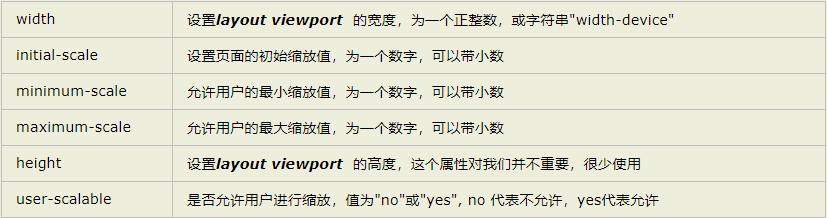
## 一、移动端的meta设置

<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0">

<meta name="viewport" content="width=device-width, initial-scale=1"> //兼容苹果和iPhone

该meta标签的作用是让当前viewport的宽度等于设备的宽度，同时不允许用户手动缩放。

meta viewport 有6个属性，如下



注：当initial-scale＜1时，是缩小；＞1时，是放大。

## 二、跨域问题

### 1、同源策略

同源指的是协议、域名、端口相同的；

同源策略作为浏览器的一项安全策略，是指从一个源中加载的文档或脚本与来自不同源的资源之间的交互会被浏览器所限制。

### 2、跨域的方式（随后在做）

## 三、页面的渲染优化

## 四、事件的各个阶段



addEventListener 的第三个参数设置为 true 和 false 的区别已经非常清晰了：true 表示该元素在事件的"捕获阶段"（由外往内传递时）响应事件；false 表示该元素在事件的"冒泡阶段"（由内向外传递时）响应事件。

## 五、let、var、const

let：生命一个块级变量，不存在变量提升，

var：存在变量提升

const：基本数据类型当数据类型改变时，会报错

const a = 10; a = true; alert(a); //报错Assignment to constant variable

## 六、箭头函数

箭头函数的this始终指向window

## 七、快速打乱数组

var a = [1,2,3,4,5];

arr.sort(function(){

return Math.random() – 0.5

})

console.log(arr)

注意：sort的参数必须是一个回调函数

## 八、浏览器兼容前缀

-moz- //火狐等使用Mozilla浏览器引擎的浏览器

-webkit- // Safari, 谷歌浏览器等使用Webkit引擎的浏览器

-ms- //IE浏览器

-o- //欧朋浏览器

## 九、单行文本溢出显示…

overflow:hidden; text-overflow:ellipsis; white-space:nowrap





## 十、多行文本溢出显示…

overflow:hidden; //溢出隐藏

display: -webkit-box;

-webkit-box-orient: vertical;

-webkit-line-clamp: 2;

## 十一、禁止用户复制

-moz-user-select:none;

-ms-user-select:none;

-webkit-user-select:none;

-khtml-user-select: none;

user-select:none;

## 十二、盒子水平垂直居中

前面有

## 十三、改变input中placeholder字体颜色

input::-moz-placeholder{ }

input:-ms-input-placeholder{ }

input::-webkit-input-placeholder{ }

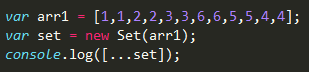
## 十四、最快捷的数组取最大值

var arr= [1,2,3,5,6];

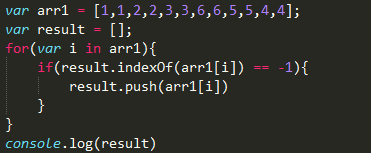
Math.max(…arr)

## 十五、数组去重

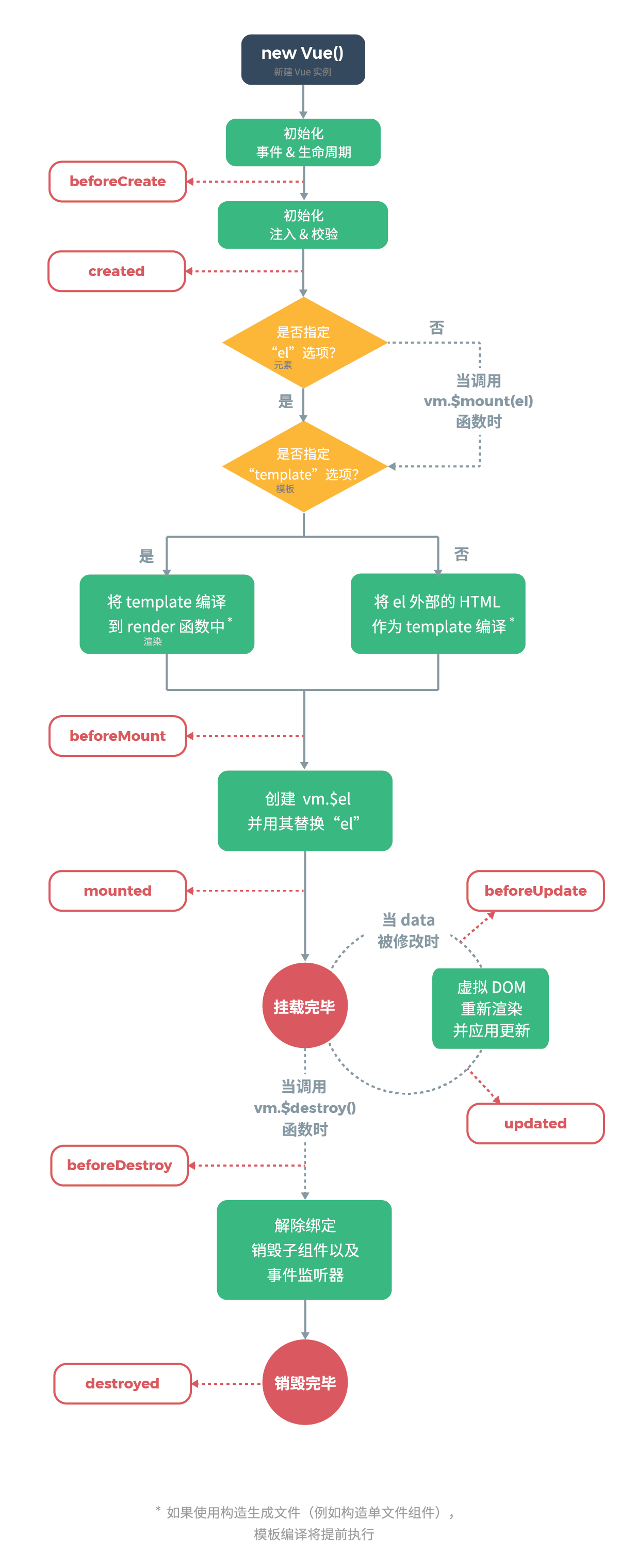
### 1、set对象

****

### 2、push方法



## 十六、vue的生命周期



**当执行destroyed时，组件已经全部销毁，此时组件中的data，methods，过滤器，指令都已不可使用**

**当执行beforeDestroy时，vue实例已经从运行阶段进入到销毁阶段**

**当执行beforeDestroy时，实例中所有的data，methods，过滤器，指令都处于可用状态，此时还没有正真执行销毁过程**

**现根据data的中新的数据在内存中重型渲染一份最新的内存DOM树，然后会把新的内存DOM树重新渲染到页面当中的，**

**这时候就完成了从data（model）->视图（view）的更新**

**update：data数据是最新的，页面数据也是最新的**

**beforeUpdate：data数据是最新的，但是页面数据任然是旧数据，还未同步**

**这是组件运行阶段的生命周期**

**这两个事件会随着data的改变触发0-n次**

**挂载结束，vue初始化完成**

**将内存中编译好了模板挂载到页面中去**

**这里表示vue开始编译模板**

**↓**

**把vue代码中的那些指令进行编译执行**

**↓**

**在内存中生成一个编译好的模板字符串**

**↓**

**把这个模板字符串渲染成为内存中的DOM**

**↓**

**此时只是在内存中渲染好了模板，并没有将**

**模板挂载到页面中**

**Data和methods均已初始化**

**注入指的是定义的data等数据**

**New一个vue实例对象**

**刚初始化一个空的vue实例，这时候这个对象上只有默认的一些收起函数和事件，其他的东西未创建，data数据和methods均未初始化**

**此函数执行的时候，模板已经在内存中编译好了，但尚未挂载到页面中去**

红色部分代表生命周期

## 十七、cookie、localStorage、sessionStorage区别

相同点：都存在于客户端中。

1、储存大小

cookie一般不超过4K，localStorage、sessionStorage可以达到5M

2、储存期限

localStore：永久储存；

sessionStorage：浏览器关闭后自动删除；

cookie：可设置期限，过期后自动删除

3、数据与服务器交互方式

cookie：将数据自动传到服务器中，服务器端也可以将cookie写到客户端

## 十八、document.write和innerHTML的区别

document.write是直接写入到页面的内容流，会导致页面被重写。

innerHTML则是DOM页面元素的一个属性，代表该元素的html内容。不会导致页面全部重绘。

innerHTML很多情况下都优于document.write，其原因在于其允许更精确的控制要刷新页面的那一个部分。

## 十九、ajax（稍后做）

1、什么是ajax

异步的javascript和XML，是指在不重新加载整个网页的情况下更新网页部分内容的一种技术。

2、ajax的步骤

1）创建ajax对象

2）

3）

4）

5）

6）

## 二十、XML和json的区别

1、体积

Json的体积小，传递速度快

2、交互

Json金额javascript的交互更加方便，更容易解析处理，更好的数据交互

3、传递速度

Json的传递速度更快

## 二十一、js有几种数据类型

基本数据类型：Number，String，Boolean，underfined，symbol，null，

引用数据类型：function，array，object

## 二十二、underfined和null的区别

null： Null类型，代表“空值”，代表一个空对象指针，使用typeof运算得到 “object”。

underfined：underfined是underfined类型，当申明一个变量为申明是，得到的就是underfined

javaScript权威指南： null和 undefined 都表示“值的空缺”，你可以认为undefined是表示系统级的、 出乎意料的或类似错误的值的空缺，而null是表示程序级的、正常的或在意料之中的值的空缺。

## 二十三、http和https的区别

http是HTTP协议运行在TCP之上。所有传输的内容都是明文，客户端和服务器端都无法验证对方的身份。

https是HTTP运行在SSL/TLS之上，SSL/TLS运行在TCP之上。所有传输的内容都经过加密，加密采用对称加密，但对称加密的密钥用服务器方的证书进行了非对称加密。

## 二十四、常见的状态码

2开头：（请求成功）表示成功处理了请求的状态代码。

3开头：（请求被重定向）表示要完成请求，需要进一步操作。 通常，这些状态代码用来重定向。

4开头：（请求错误）这些状态代码表示请求可能出错，妨碍了服务器的处理。

5开头：（服务器错误）这些状态代码表示服务器在尝试处理请求时发生内部错误。 这些错误可能是服务器本身的错误，而不是请求出错。

## 二十五、优化页面性能

### 1、优化的目

从用户角度而言，优化能够让页面加载得更快、对用户的操作响应得更及时，能够给用户提供更为友好的体验。从服务商角度而言，优化能够减少页面请求数、或者减小请求所占带宽，能够节省可观的资源。

### 2、优化的措施

1）减少HTTP请求数。

2）使用缓存

3）脚本的无阻塞加载。

4）内联脚本的位置优化等

5）Javascript中的DOM 操作优化、CSS选择符优化。

6）图片编码优化，懒加载。

7）使用css stripes（css精灵），减少图片加载

8）将样式表放到页面顶部

9）将javascript和css从外部引入

10）压缩javascript和css

## 二十六、mvvm和mvc的区别

MVC是比较直观的架构模式，用户操作->View（负责接收用户的输入操作）->Controller（业务逻辑处理）->Model（数据持久化）->View（将结果反馈给View）。MVC使用非常广泛，比如JavaEE中的SSH框架。

如果说MVP是对MVC的进一步改进，那么MVVM则是思想的完全变革。它是将“数据模型数据双向绑定”的思想作为核心，因此在View和Model之间没有联系，通过ViewModel进行交互，而且Model和ViewModel之间的交互是双向的，因此视图的数据的变化会同时修改数据源，而数据源数据的变化也会立即反应view。微信小程序前端使用mvvm。

## 二十七、渐进增强和优雅降级

渐进增强：一开始就针对低版本浏览器进行构建页面，完成基本的功能，然后再针对高级浏览器进行效果、交互、追加功能达到更好的体验。

优雅降级：一开始就构建站点的完整功能，然后针对浏览器测试和修复。

## 二十八、px和em的区别

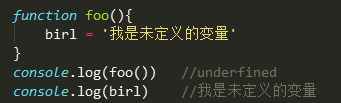
px：代表屏幕上的一点，是绝对单位

em：表示相对父元素的字体大小，是相对单位

## 二十九、js操作造成的内存泄漏

1、闭包引起

2、意外的全局变量



在 JavaScript 非[严格模式](https://lz5z.com/JavaScript%E4%B8%A5%E6%A0%BC%E6%A8%A1%E5%BC%8F/" \t "_blank)中，未定义的变量会自动绑定在全局对象上

3、没有及时清理的dom结构

4、被遗忘的定时器或回调

## 三十、浏览器是怎样渲染页面的

1、解析HTML文件，创建DOM树。自上而下，遇到任何样式（link、style）与脚本（script）都会阻塞（外部样式不阻塞后续外部脚本的加载）。

2、解析CSS。优先级：浏览器默认设置<用户设置<外部样式<内联样式<HTML中的style样式。

3、将CSS与DOM合并，构建渲染树（Render Tree）。

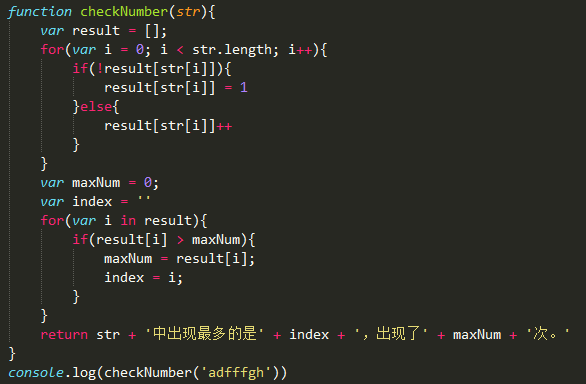
4、布局和绘制，重绘（repaint）和重排（reflow）。

## 三十一、display:none和visibility:hidden

1、display:none是彻底消失，不在文档流中占位，浏览器也不会解析该元素；visibility:hidden是视觉上消失了，可以理解为透明度为0的效果，在文档流中占位，浏览器会解析该元素；

  2、使用visibility:hidden比display:none性能上要好，display:none切换显示时页面产生回流，而visibility切换是否显示时则不会引起回流。

## 三十二、判断字符串中出现次数最多的字符



result是键值对：

## 三十三、$(document).ready方法和window.onload

1、window.onload方法是在网页中所有的元素(包括元素的所有关联文件)完全加载到浏览器后才执行的。

  2、$(document).ready() 方法可以在DOM载入就绪时就对其进行操纵，并调用执行绑定的函数。

## 三十四、get和post的区别

1、请求方式不同：$.get() 方法使用GET方法来进行异步请求的。$.post()方法使用POST方法来进行异步请求的。

2、参数传递方式不同：get请求会将参数跟在URL后进行传递，而POST请求则是作为HTTP消息的实体内容发送给Web服务器的，这种传递是对用户不可见的。

3、数据传输大小不同：get方式传输的数据大小不能超过2KB 而POST要大的多。

4、安全问题： GET 方式请求的数据会被浏览器缓存起来，因此有安全问题。

## 三十五、从浏览器地址栏输入url到显示页面的步骤

1、浏览器根据请求的URL交给DNS域名解析，找到真实IP，向服务器发起请求；

2、服务器交给后台处理完成后返回数据，浏览器接收文件（HTML、JS、CSS、图象等）；

3、浏览器对加载到的资源（HTML、JS、CSS等）进行语法解析，建立相应的内部数据结构（如HTML的DOM）；

4、载入解析到的资源文件，渲染页面，完成。

## 三十六、HTML的全局属性有哪些

id、class、title、draggable、style、lang

## 三十七、浏览器内核的理解

主要分成两部分：渲染引擎(layout engineer或Rendering Engine)和JS引擎

渲染引擎：负责取得网页的内容（HTML、XML、图像等等）、整理讯息（例如加入CSS等），以及计算网页的显示方式，然后会输出至显示器或打印机。

JS引擎则：解析和执行javascript来实现网页的动态效果

## 三十八、link与@import的区别

本质上，这两种方式都是为了加载css文件，但还是存在细微的差别。

1、方式不同：link是HTML方式，@import是CSS方式

2、加载顺序：link会和页面同时加载，而import等待页面加载完成后才会加载

3、兼容问题：由于@import是CSS2.1提出的，所以老的浏览器不支持，@import只有在IE5以上的才能识别，而link标签无此问题，完全兼容。

4、使用dom控制样式时的差别：当时用JavaScript控制dom去改变样式的时候，只能使用link标签，因为@import不是dom可以控制的（不支持）。

## 三十九、块级格式化上下文（BFC）

1、什么是BFC

BFC是特殊盒子（容器）所具有的的特性（属性），这种特殊盒子在样式上和普通盒子没有区别；其从功能上可以将其看作是隔离了的容器，容器里面的布局不会影响到外面的元素，并且该容器有一些普通容器没有的特殊能力（作用），如解决高度塌陷、解决margin值叠加等。

2、BFC触发条件

1）根元素（html、body）

2）float不会none（left、right）

3）绝对定位元素（absolute、fixed）

4）display设置为inline-block、flex、inline-flex

5）overflow不为visible，可看见的（hidden、scroll、auto）

## 三十九、相对定位和绝对定位

绝对定位：是相对于元素最近的已定位的祖先元素

相对定位：是相对于元素在文档中的初始位置

## 四十、解决高度塌陷

1、父元素设置固定高度

2、父元素给个overflow：hidden

3、父元素末尾加个空div，给个clear：both

4、用after伪元素设置

## 四十一、为什么要初始化css样式

因为浏览器的兼容问题，不同浏览器对有些标签的默认值是不同的，如果没对CSS初始化往往会出现浏览器之间的页面显示差异。

初始化样式会对SEO有一定的影响，但鱼和熊掌不可兼得，但力求影响最小的情况下初始化

## 四十二、栈和堆

原始数据类型储存在栈中，栈的体积小，大小固定，属于频繁使用的数据。

引用数据类型是储存在对堆中的，占据内存大，引用数据类型在栈中指明了自己的所在地，当代吗解析时，会从栈中获取地址，再从堆中寻找实体。

## 四十三、当函数内部没有a这个变量的时候，才会向上一级查找

## 四十四、 apply和call两种方法的比较

相同点：两种方法产生的效果一样

不同点：方法传递的参数不同， apply传入的是一个参数数组，也就是将多个参数组合成为一个数组传入，而call则作为call的参数传入（从第二个参数开始）

## 四十五、继承

## 四十六、闭包（使函数拥有的私有变量）

1、概念：是指有权访问另一个函数作用域中的变量的函数，创建闭包的最常见的方式就是在一个函数内创建另一个函数，通过另一个函数访问这个函数的局部变量。调用完一次后，函数没有被释放，整条作用域链上的局部变量都将得到保留。在本质上，闭包就是将函数内部和函数外部连接的一座桥梁。

2、闭包代码

function text(){

var a = 0;

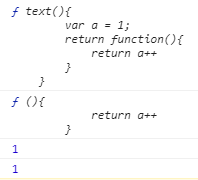
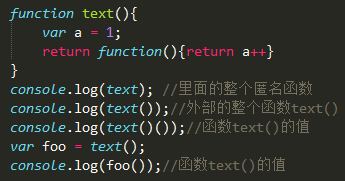
return function(){ alert(a++) }

}

var foo = text()

foo() //或text()() 在这里text()= function(){ alert(a++) }

foo() //或text()() 在这里text= function text(){ }



或

var foo = (function(){

var a = 0;

return function(){ alert(a++) }

})()

foo() //1

foo() //2

当我们不需要引用的时候直接b = null就可以了。

## 四十七、异步编程

## 四十八、去掉字符串中的空格

### 1、首尾空格

str.trim()

### 2、全部空格

str.replace(/\s/g,'')

## 四十九、判断应用数据的数据类型

var a = [];console.log(a instanceof Array)

五十、css的属性

### 1、calc()：动态计算长度

width: calc(100% - 100px)

### 2、@support：检测浏览器知否支持css的某个属性

1）基本语法



浏览器支持display:flex属性，div的样式就是display:flex

2）“not”用法



浏览器不支持display:flex属性，div的样式就是display:right

3）“and”用法



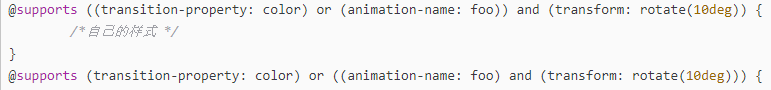
浏览器支持display:flex和box-shadow，执行里面样式

4）“or”用法



如果浏览器支持其中一个就可以执行里面自己的样式

5）混用



### 3、媒体：针对不同的媒体类型定义不同的样式

# Webpack

## nrm

### 1、安装nrm

npm i nrm -g

### 2、查看所有可用的镜像

nrm ls

### 3、切换使用的镜像

nrm use 镜像名称

## 二、webpack

全局安装webpack：npm install webpack(-cli) -g

### 1、文档结构

dist

src

js ------------js文件

css -----------css文件

index.html -----------html文件

main.js ------------js入口文件

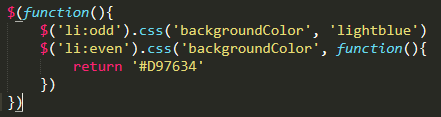
**生成package.js文件：npm init -y**

1）main.js中引入jquery

下载jquery资源包：npm install jquery

在入口文件中引入

写jq代码



**注：由于main.js中有浏览器无法识别的es6的新语法，因 此需要webpack将main.js编译成bundle.js供浏览器识别**

2）编译入口文件main.js

webpack版本号4.0以下

webpack ./src/main.js ./dist/bundle.js

webpack版本号4.0以上

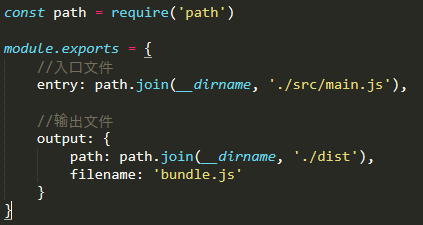
webpack ./src/main.js -o ./dist/bundle.js

3）在index.html中引入bundle.js文件



Webpack的功能：a、能够处理js文件的相互依赖关系；b、能够处理js的兼容问题，把高级的，浏览器识别不了的语法转化为浏览器能识别的语法

### 2、webpack.config.js文件的配置（不需要运行webpack命令了）



在命令行输入webpack，回车即可

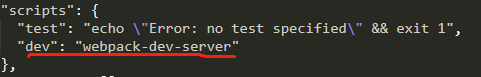
此时webpack的运行步骤：a、发现命令行中没有入口和出口文件；b、webpack去项目的根目录webpack.config.js文件中去找；c、当这个文件存在时，回去执行这个配置文件，当解析执行配置文件后，就得到了配置文件中导出的配置对象；d、此时就得到了入口和出口，然后打包构建。

### 3、webpack-dev-server（自动打包编译，实现**热替换**）

1）安装webpack-dev-server

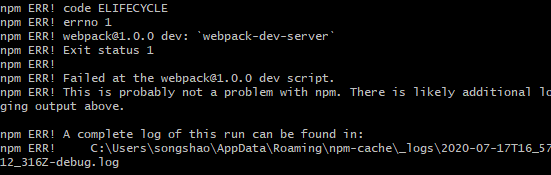
npm install webpack-dev-server -D

2）安装后和webpack的用法一样，但是需要借助package.json文件来实现



3）运行npm run dev

报错



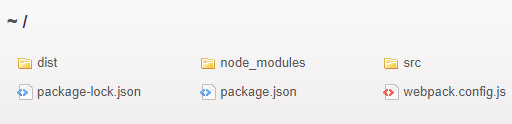
4）报错处理

将mode文件删掉重新配置

在项目中重新安装webpack（npm i webpack -D）

在项目中重新安装webpack-cli（npm i webpack-cli -D）

5）重新运行，localhost:8080



6）修改index.html中的script地址（改为根路径）

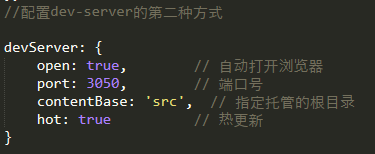


注意：根路径下的bundle.js并没有存放在实际的物理磁盘 中而知存放在内存中。

### 4、自动开启浏览器、设置端口号、设置初始路径、**热更新**（1）



### 5、自动开启浏览器、设置端口号、设置初始路径、**热更新**（2）



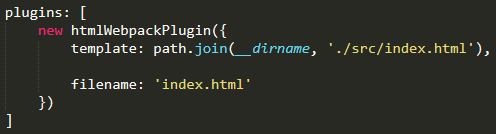
6、替换index.html中的script文件

1）安装webpack插件

npm install html-webpack-plugin –D

2）在webpack.config.js文件中导入html-webpack-plugin

 3）使用插件



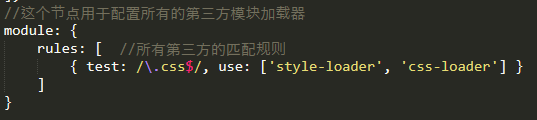
## 三、编译css文件的loader

Webpack默认只能打包js类型的文件，无法处理非js类型的文件，如果要处理非js类型的文件，需要手动安装一些合适的第三方loader加载器。

1、安装loader

npm i style-loader css-loader -D

2、在webpack-config.js中添加配置



**注意：use后面的数组顺序不能颠倒，处理的时候从后往前**

**当最后一个loader调用完毕后，会把处理的结果交给webpack进 行打包编译，最重输入bundle.js中去。**

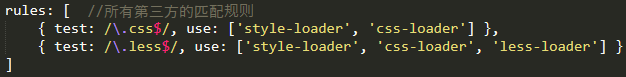
## 四、编译less文件的loader

1、安装less-loader

npm i less-loade –D

npm i less -D

2、配置文件

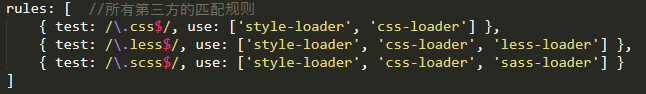


## 五、编译scss文件的loader

### 1、安装sass-loader

npm i sass-loader node-sass –D

### 2、配置文件



## 六、css中的url编译

### 1、安装编译工具

cnpm i url-loader film-loader -D

### 2、配置文件

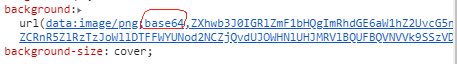


1）将默认base64的图片转换为本地图片

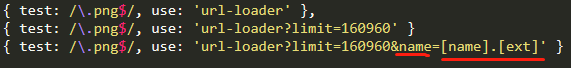


参数limit表示限定，若图片大于limit给定的是，则图片不 会转换成base64，若小于给定的limit值，则图片会转换成base64





2）将图片的url地址设置为本地路径的url（固定写法）





**注：可以认为图片托管在项目的根目录中**

**3）如果两张图片的名称和后缀都一样，可以设置为**



name=[hash:8]-[name].[ext]表示截取hash值的前8位放在 原名称前面

## 七、字体样式



## 八、高级的js语法转换

一些高级的ES6的语法及ES7以上的语法webpack是识别不了的，所以需要借助第三方的插件来实现

1、安装相关的loader（两套方法）

cnpm i babel-core babel-loader babel-plugin-transform-runtime -D

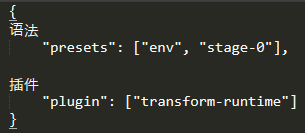
cnpm i babel-preset-env babel-preset-stage-0 -D

2、配置文件

1）webpack的配置文件webpack.config.js



2）babel的配置文件.babelrc



## 九、总结

### 1、webpack.config.js



### 2、用到的命令

初始化项目：npm init -y

安装webpack：cnpm iwebpack -D

安装webpack-cli：cnpm iwebpack-cli -D

安装webpack-dev-server：cnpm iwebpack-dev-server–D

替换html的插件：cnpm ihtml-webpack-plugin -D

安装css-loader： cnpm icss-loaderstyle-loader -D

安装less-loader：cnpm iless-loaderless–D

安装sass-loader：cnpm isass-loadernode-sass–D

编译css中url：cnpm i url-loaderfile-loader -D

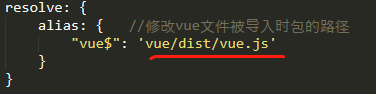
编译js高级语法：

cnpm i babel-corebabel-loaderbabel-plugin-transform-runtime -D

cnpm i babel-preset-envbabel-preset-stage-0 -D

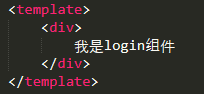
## 十、用webpack构建vue项目

### 1、import默认引入的包是不全的，需要配置文件来处理



注：可以不用配置，组件必须用.vue文件的形式

### 2、将组件写成.vue的形式



### 3、vue实例用render来渲染



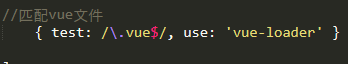
### 4、此时需要安装vue-loader来处理vue文件

1）vue-loader安装

cnpm i vue-loader vue-template-compiler -D

2）配置文件

a、配置.vue结尾的文件



b、vue-loader-plugin





注：Vue-Loader v15需要配合一个 webpack 插件才能 正常使用

### 5、vue中引入和导出数据的问题

引入：

import vue from ‘vue’

import ‘./css/index.css’

导出：

export default ------一个文件只能有一个，接收是可以自定义参数

export ------可以有多个，接收时按需导入{ 需要导出的数 }

### 5、路由配置

1）安装vue-router

cnpm i vue-router –D

2）main.Js中引入router



3）手动安装路由



4）创建路由实例

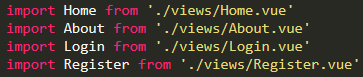


5）在vue实例中挂载路由

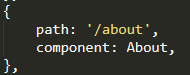


6）路由基本使用

a、引入路由组件



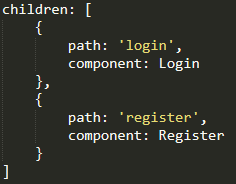
b、创建路由匹配规则



c、路由容器



d、子路由



### 6、style样式

1）普通的style只支持css，如果需要用到sass或less需要加 lang属性

2）避免局部组件样式影响全局组件样式，需加上scoped属性

scoped实现原理：自动给标签加了一个属性，通过属性选 择器，避免了样式干扰

十一、Mint UI

1、安装Mint UI

cnpm i mint-ui -S

# node

一、node的内置模块

fs：文件操作

文件追加：fs.appendFile(‘test.txt’,’hello’,function(err){if(err) throw err;})

文件写入：fs.writeFile(‘test.txt’,’hello’,function(err){if(err) throw err;})

文件读取：fs.readFile()