

# 第1章 Web前端开发概述

---

## 1.1 W3C

- W3C 指万维网联盟 (World Wide Web Consortium)
- W3C 创建于1994年10月
- W3C 由 Tim Berners-Lee 创建
- W3C 是一个会员组织
- W3C 的工作是对 web 进行标准化
- W3C 创建并维护 WWW 标准
- W3C 标准被称为 W3C 推荐 (W3C 规范)

<https://www.runoob.com/w3c/w3c-intro.html>

<https://www.w3school.com.cn/>

<https://www.w3.org/> 官网

## 1.2 ECMAScript

### 1.2.1 简称 (ES 6等)

ECMAScript是一种由Ecma国际（前身为欧洲计算机制造商协会，European Computer Manufacturers Association）通过ECMA-262标准化的脚本程序设计语言。这种语言在万维网上应用广泛，它往往被称为JavaScript或Script，所以它可以理解是JavaScript的一个标准,但实际上后两者是ECMA-262标准的实现和扩展。

### 1.2.2 版本

各个版本: [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Language\\_Resources](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Language_Resources)

ECMAScript 版本

一、总结

一句话总结:

2015: ECMAScript 6: 添加类和模块,

2016: ECMAScript 7: 增加指数运算符 (\*\*), 增加 Array.prototype.includes

二、ECMAScript 版本

转自或参考: <https://www.runoob.com/js/js-intro.html>

JavaScript 已经由 ECMA (欧洲电脑制造商协会) 通过 ECMAScript 实现语言的标准化。

年份	名称	描述
1997	ECMAScript 1	第一个版本
1998	ECMAScript 2	版本变更
1999	ECMAScript 3	添加正则表达式 添加 try/catch
	ECMAScript 4	没有发布
2009	ECMAScript 5	添加 "strict mode", 严格模式 添加 JSON 支持
2011	ECMAScript 5.1	版本变更
2015	<b>ECMAScript 6</b>	添加类和模块
2016	ECMAScript 7	增加指数运算符 (**) 增加 Array.prototype.includes

ECMAScript 6 也称为 ECMAScript 2015。

ECMAScript 7 也称为 ECMAScript 2016。

## 1.3 Facebook

Facebook改名Meta，进军元宇宙

## 1.4 XML、XHtml、Html

xml一般指**可扩展标记语言**。可扩展标记语言，标准通用标记语言的子集，简称XML。是一种用于标记电子文件使其具有结构性的标记语言。在电子计算机中，标记指计算机所能理解的信息符号，通过此种标记，计算机之间可以处理包含各种的信息比如文章等。用于数据交换（把一种文档格式转换成另一种文档格式）

HTML的全称为**超文本标记语言**([https://baike.baidu.com/item/超文本标记语言/6972570?fromModule=lemma\\_inlink](https://baike.baidu.com/item/超文本标记语言/6972570?fromModule=lemma_inlink))，是一种**标记语言**。它包括一系列**标签**。通过这些标签可以将网络上的**文档**格式统一，使分散的**Internet**资源连接为一个逻辑整体。HTML文本是由HTML命令组成的描述性**文本**，HTML命令可以说明**文字**、**图形**、**动画**、**声音**、**表格**、**链接**等。

XHTML**可扩展超文本标记语言**（英语：eXtensible HyperText Markup Language，XHTML），是一种**标记语言**，表现方式与**超文本标记语言**（HTML）类似，不过语法上更加严格。从继承关系上讲，HTML是一种基于**标准通用标记语言**（SGML）的应用，是一种非常灵活的置标语言，而XHTML则基于**可扩展标记语言**（XML），XML是SGML的一个子集。XHTML 1.0在2000年1月26日成为W3C的推荐标准。

## 第2章 HTML5基础

### 2.1 内部链接、外部链接、空链接、锚点链接

①.内部链接：在自己的网站中，网页之间相互跳转。

[跳转到a.html](#)

href：跳转的路径

target：跳转的时候重新打开一个选项卡

②.外部链接：从自己的网站跳转到别人的网站

[单击后跳转到href指定的网站](#)

③.空链接：当还没有想好跳转到哪里时,在href中写#号。写一个#号会回到本网页的顶部，写两个#号单击了链接也不发生变化。

[回到本网页的顶部](#)

[单击了链接也不发生变化](#)

④.锚点链接：跳转到当前网页指定的位置  
先给要跳转的位置标签设置id属性  
在a标签的href属性中写#id

## 2.2 文本超链接、图片超链接、电子邮件链接、下载歌曲、

%20是空格的HTML编码。

cc: 抄送; bcc: 密抄送; subject: 主题; body: 正文

## 2.3 表格

## 2.4 bootstrap3 框架

```
<!-- 最新版本的 Bootstrap 核心 CSS 文件 -->
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css"
integrity="sha384-
HSMxcRTRXnN+Bdg0JdbxYKrThecOKuH5zCYot1SAcp1+c8xmyTe9GYg1l9a69psu"
crossorigin="anonymous">

<!-- 可选的 Bootstrap 主题文件（一般不用引入） -->
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap-
theme.min.css" integrity="sha384-
6pzBo3FDv/PJ8r2KRkGHifhEocL+1X2rvCTTkuFgk7/0pbek5mMa1upzvbruboz"
crossorigin="anonymous">

<!-- 最新的 Bootstrap 核心 JavaScript 文件 -->
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"
integrity="sha384-
aJ210j1MXNL5UyI1/XNwTMqvzeRMZH2w8c5CRVpZpu8Y5bApTppSuUkhZXN0VxHd"
crossorigin="anonymous"></script>
```

## 2.5 jsDelivr

JSDelivr是一种特殊的CDN。它旨在让用户下载npm和Github上托管JavaScript库。(如果它们托管在Wordpress.org上, 则还可以加载Wordpress插件)。

CDN代表内容交付网络。其主要目的是让用户更快地下载文件。

## 2.6 bootstrap4 框架

```
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css"
integrity="sha384-
zCbkRUCUGaJDKqS1kPbPd7TveP5iyJE0EjAuZQTgFLD2y1zuqkfDKlfg/esRtxUkn"
crossorigin="anonymous">
```

```
<script src="https://cdn.jsdelivr.net/npm/jquery@3.5.1/dist/jquery.slim.min.js"
integrity="sha384-
Dfxdz2htPH01sSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/js/bootstrap.bundle.min.js" integrity="sha384-
fQybjgWLrvvRgtW6bFlB7jaZrFsaBXjsOMm/tb9LTS58ONXgqBR9w8owht/amnpF"
crossorigin="anonymous"></script>
```

## 2.7 端口占用

```
netstat -aon | findstr "5927"
```

```
// console.log(JSON.stringify(accessRoutes, null, 2))
```

# 第5章 音频与视频

## 5.1 HTML5 音频格式

- MP3

MP3是一种[音频压缩技术](#)，其全称是动态影像专家压缩标准音频层面3（Moving Picture Experts Group Audio Layer III），简称为MP3。它被设计用来大幅度地降低音频数据量。利用 MPEG Audio Layer 3 的技术，将音乐以1:10 甚至 1:12 的[压缩率](#)，压缩成容量较小的文件，而对于大多数用户来说重放的音质与最初的不压缩音频相比没有明显的下降。它是在1991年由位于[德国埃尔朗根](#)的研究组织[Fraunhofer-Gesellschaft](#)的一组工程师发明和标准化的。用MP3形式存储的音乐就叫做MP3音乐，能播放MP3音乐的机器就叫作MP3播放器。

- Wav

WAV是最常见的声音文件格式之一，是微软公司专门为[Windows](#)开发的一种标准数字音频文件，该文件能记录各种单声道或立体声的声音信息，并能保证声音不失真。但WAV文件有一个致命的缺点，就是它所占用的[磁盘空间](#)太大（每分钟的音乐大约需要12兆磁盘空间）。它符合资源互换文件格式（RIFF）规范，用于保存Windows平台的音频信息资源，被Windows平台及其应用程序所广泛支持。Wave格式支持MSADPCM、CCITT A律、CCITT μ律和其他压缩算法，支持多种音频位数、采样频率和声道，是PC机上最为流行的声音文件格式；但其文件尺寸较大，多用于存储简短的声音片段。

- Ogg

Ogg全称是OGGVobis(oggVorbis)是一种[音频压缩格式](#)，类似于[MP3](#)等的[音乐格式](#)。Ogg是完全免费、开放和没有专利限制的。OggVorbis文件的扩展名是".ogg"。Ogg文件格式可以不断地进行大小和音质的改良，而不影响旧有的[编码器](#)或播放器。

## 5.2 HTML5 视频格式

- MPEG4

运动图像专家组MPEG 于1999年2月正式公布了MPEG-4（ISO/IEC14496）标准第一版本。同年年底MPEG-4第二版亦告底定，且于2000年年初正式成为国际标准。

MPEG-4与MPEG-1和MPEG-2有很大的不同。MPEG-4不只是具体压缩算法，它是针对数字电视、交互式绘图应用（影音合成内容）、[交互式多媒体](#)（WWW、资料撷取与分散）等整合及[压缩技术](#)的需求而制定的国际标准。

MPEG-4标准将众多[多媒体应用集成](#)于一个完整框架内，旨在为多媒体通信及应用环境提供标准算法及工具，从而建立起一种能被多媒体传输、[存储](#)、检索等应用领域普遍采用的统一数据格式

MPEG-4包含了MPEG-1及MPEG-2的绝大部分功能及其他格式的长处，并加入及扩充对虚拟现实模型语言（VRML，VirtualReality Modeling Language）的支持，面向对象的合成档案（包括音效，视讯及VRML对象），以及[数字版权管理](#)（DRM）及其他互动功能。而MPEG-4比MPEG-2更先进的其中一个特点，就是不再使用宏区块做影像分析，而是以影像上个体为变化记录，因此尽管影像变化速度很快、码率不足时，也不会出现方块画面。

- WebM

WebM由[Google](#)提出，是一个开放、免费的媒体文件格式。WebM 影片格式其实是以 [Matroska](#)（即 [MKV](#)）容器格式为基础开发的新容器格式，里面包括了[VP8](#)影片轨和 [Ogg](#) Vorbis 音轨，其中Google将其拥有的VP8视频编码技术以类似[BSD](#)授权开源，Ogg Vorbis 本来就是开放格式。WebM标准的[网络视频](#)更加偏向于开源并且是基于[HTML5](#)标准的，WebM 项目旨在为对每个人都开放的网络开发高质量、开放的视频格式，其重点是解决视频服务这一核心的网络用户体验。Google 说 WebM 的格式相当有效率，应该可以在 netbook、tablet、手持式装置等上面顺畅地使用。

- MP4

MP4是一套用于[音频](#)、视频信息的压缩[编码](#)标准，由国际标准化组织（ISO）和国际电工委员会（IEC）下属的“[动态图像专家组](#)”（Moving Picture Experts Group，即MPEG）制定，第一版在1998年10月通过，第二版在1999年12月通过。MPEG-4格式的主要用途在于网上流、光盘、语音发送（[视频电话](#)），以及电视广播。

- AVI

AVI英文全称为Audio Video Interleaved，即[音频视频交错格式](#)，是[微软公司](#)于1992年11月推出、作为其Windows[视频软件](#)一部分的一种[多媒体](#)容器格式。

AVI文件将[音频](#)（语音）和[视频](#)（影像）数据包含在一个文件容器中，允许音视频同步回放。类似[DVD](#)视频格式，AVI文件支持多个音视频流。AVI信息主要应用在[多媒体光盘](#)上，用来保存电视、电影等各种影像信息。

## 第7章 CSS语法

---

### 7.1 em

任意浏览器的默认字体高都是16px。所有未经调整的浏览器都符合: 1em=16px。那么12px=0.75em, 10px=0.625em。

任意浏览器的默认字体高度16px（16像素）。所有未经调整的浏览器都符合: 1em=16px。那么12px=0.75em, 10px=0.625em。为了简化[font-size](#)的换算，需要在[css](#)中的body[选择器](#)中声明font-size=62.5%，这就使em值变为 16px\*62.5%=10px，这样12px=1.2em, 10px=1em，也就是说只需要将你的原来的px数值除以10，然后换上em作为单位就行了。

### 7.2 p{font:italic bold 12px/30px arial,sans-serif;}

简写方式(变体、字号、行高、字体)，顺序不分，但一般字号和行高写在一起，中间用斜杠分开。

12px 是指 font-size:12px 即字体大小为12像素

30px 是指line-height:30px 即每行字的行高为30px

### 7.3 background:url(bg8.jpg) content-box 0px -10px;

background:url(路径) top(也就是距容器上边距的距离) left(也就是距容器左边距的距离) 重复方式;

## 7.4 CSS list-style-type 属性

- 属性值

值	描述
none	无标记。
disc	默认。标记是实心圆。
circle	标记是空心圆。
square	标记是实心方块。
decimal	标记是数字。
decimal-leading-zero	0开头的数字标记。(01, 02, 03, 等。)
lower-roman	小写罗马数字(i, ii, iii, iv, v, 等。)
upper-roman	大写罗马数字(I, II, III, IV, V, 等。)
lower-alpha	小写英文字母The marker is lower-alpha (a, b, c, d, e, 等。)
upper-alpha	大写英文字母The marker is upper-alpha (A, B, C, D, E, 等。)
lower-greek	小写希腊字母(alpha, beta, gamma, 等。)
lower-latin	小写拉丁字母(a, b, c, d, e, 等。)
upper-latin	大写拉丁字母(A, B, C, D, E, 等。)
hebrew	传统的希伯来编号方式
armenian	传统的亚美尼亚编号方式
georgian	传统的乔治亚编号方式(an, ban, gan, 等。)
cjk-ideographic	简单的表意数字
hiragana	标记是：a, i, u, e, o, ka, ki, 等。（日文平假名字符）
katakana	标记是：A, I, U, E, O, KA, KI, 等。（日文片假名字符）
hiragana-iroha	标记是：i, ro, ha, ni, ho, he, to, 等。（日文平假名序号）
katakana-iroha	标记是：I, RO, HA, NI, HO, HE, TO, 等。（日文片假名序号）

## 7.5 CSS list-style-position 属性

- 可能的值

值	描述
inside	列表项目标记放置在文本以内，且环绕文本根据标记对齐。
outside	默认值。保持标记位于文本的左侧。列表项目标记放置在文本以外，且环绕文本不根据标记对齐。
inherit	规定应该从父元素继承 list-style-position 属性的值。

清华教材：edu-9-5-1.html例

## 第8章 CSS页码布局

### 8.1 媒体查询技术

#### 1.什么是媒体查询，如何理解？

就是根据不同的屏幕或设备，自适应布局。一套代码，多个设备应用，合理布局。

#### 2.使用：如何实现一个网站的响应式设计？

##### 1》meta标签

在网页的头部添加这行代码

其中：width=device-width: 网页宽度等于设备宽度

initial-scale = 1.0: 初始缩放比例是1.0。网页初始页面的大小占整个面积的100%

maxmum-scale = 1.0 : 最大缩放比例为1.0

user-scalable: 用于是否可以手动缩放

##### 2》兼容性问题：

因为IE9,IE6,7,8不支持媒体查询，所以必须加载下列文件

```
<!--[if lt IE 9]>
<script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js">
</script>
<script src="https://oss.maxcdn.com/libs/respond.js/1.3.0/respond.min.js">
</script> <script src="http://css3-mediaqueries-js.googlecode.com/svn/trunk/css3-
mediaqueries.js"></script>
<![endif]-->
```

##### 3》设定渲染方式

有时候可能会遇到IE浏览器升级到IE9以上，但是浏览器的文档模式却是IE8，

如何解决这个问题呢？

这句话可以保证让IE的文档模式永远都是最新的

还有一种更高级的写法：

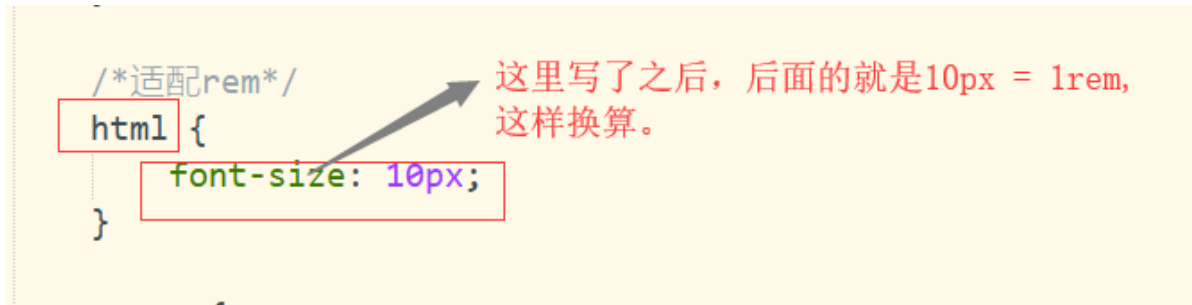
什么意思呢？

怎么这段代码后面加了一个chrome=1，这个[Google Chrome Frame（谷歌内嵌浏览器框架GCF）](#)，如果有的用户电脑里面装了这个chrome的插件，就可以让电脑里面的IE不管是哪个版本的都可以使用Webkit引擎及V8引擎进行排版及运算，无比给力，不过如果用户没装这个插件，那这段代码就会让IE以最高的文档模式展现效果。

4》建议使用单位：

rem , em , %

rem的设置：



5》如何编写媒体查询代码

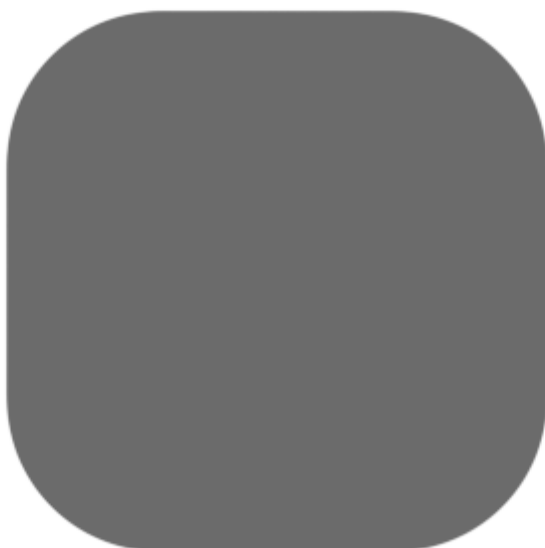
6》如何使用@media规则 <https://www.jianshu.com/p/516c47ef9ac0>

7》如何做到图片自适应

## 8.2 CSS3 border-image-slice 属性



```
border-image:url(border-image2.png) 30 30 30 30 round repeat;
```



## 8.3 CSS3 border-image 属性

**border-image** : url(图像的路径) 图像分割方式/图像边框宽度 图像平铺方式

## 8.4 CSS3 box-shadow 属性

**box-shadow**: *h-shadow v-shadow blur spread color inset*;

**注意：**boxShadow 属性把一个或多个下拉阴影添加到框上。该属性是一个用逗号分隔阴影的列表，每个阴影由 2-4 个长度值、一个可选的颜色值和一个可选的 inset 关键字来规定。省略长度的值是 0。

值	说明
<i>h-shadow</i>	必需的。水平阴影的位置。允许负值
<i>v-shadow</i>	必需的。垂直阴影的位置。允许负值
<i>blur</i>	可选。模糊距离
<i>spread</i>	可选。阴影的大小
<i>color</i>	可选。阴影的颜色。在 <a href="#">CSS颜色值</a> 寻找颜色值的完整列表
<i>inset</i>	可选。从外层的阴影（开始时）改变阴影内侧阴影

## 8.5 CSS 颜色

颜色是由红（RED），绿（GREEN），蓝（BLUE）光线的显示结合。

### 颜色值

CSS中定义颜色使用十六进制（hex）表示法为红，绿，蓝的颜色值结合。可以是最低值是0（十六进制00）到最高值是255（十六进制FF）

3个双位数字的十六进制值写法，以#符号开始。

### 颜色实例

Color	Color HEX	Color RGB
	#000000	rgb(0,0,0)
	#FF0000	rgb(255,0,0)
	#00FF00	rgb(0,255,0)
	#0000FF	rgb(0,0,255)
	#FFFF00	rgb(255,255,0)
	#00FFFF	rgb(0,255,255)
	#FF00FF	rgb(255,0,255)
	#C0C0C0	rgb(192,192,192)
	#FFFFFF	rgb(255,255,255)

## 第9章 Javascript语言基础

### 9.1 ES6变量类型、关键字

#### let 关键字

- 定义一个变量，并且有块级作用域，不能重复声明
- 是对 var 关键字的改进，这个是可以重复声明的

#### const 关键字

- 定义一个常量，不可以重复声明

#### typeof 关键字

- 用于查看变量的数据类型

**JavaScript的关键字				
break	case	catch	continue	default
delete	do	else	finally	for
function	if	in	instanceof	new
return	switch	this	throw	try
typeof	var	void	while	with

JavaScript的保留字				
abstract	boolean	byte	char	class
const	debugger	double	enum	export
extends	final	float	goto	implements
import	int	interface	long	native
package	private	protected	public	short
static	super	synchronized	throws	transient
volatile				

- const

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>const 定义常量</title>
</head>
<body>
  <script>
    //声明常量
    const SCHOOL = '尚硅谷';

    //1. 一定要赋初始值
    // const A;
    //2. 一般常量使用大写(潜规则)
    // const a = 100;
    //3. 常量的值不能修改
    // SCHOOL = 'ATGUGU';
    //4. 块级级作用域
    // {
    //   const PLAYER = 'UZI';
    // }
    // console.log(PLAYER);
    //5. 对于数组和对象的元素修改，不算做对常量的修改，不会报错
    const TEAM = ['UZI','MXLG','Ming','Letme'];
    // TEAM.push('Meiko');
```

```
    </script>
</body>
</html>
```

- let

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>let</title>
</head>
<body>
  <script>
    //声明变量
    let a;
    let b,c,d;
    let e = 100;
    let f = 521, g = 'iloveyou', h = [];

    //1. 变量不能重复声明
    // let star = '罗志祥';
    // let star = '小猪';

    //2. 块级级作用域 全局，函数，eval
    // if else while for
    // {
    //   let girl = '周扬青';
    // }
    // console.log(girl);

    //3. 不存在变量提升
    // console.log(song);
    // let song = '恋爱达人';

    //4. 不影响作用域链
    {
      let school = '尚硅谷';
      function fn(){
        console.log(school);
      }
      fn();
    }

  </script>
</body>
</html>
```

- 案例

```
<!DOCTYPE html>
<html lang="en">
```

```

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>点击 DIV 换色</title>
  <link crossorigin="anonymous" href="https://cdn.bootcss.com/twitter-bootstrap/3.3.7/css/bootstrap.min.css"
    rel="stylesheet">
  <style>
    .item {
      width: 100px;
      height: 50px;
      border: solid 1px rgb(42, 156, 156);
      float: left;
      margin-right: 10px;
    }
  </style>
</head>

<body>
  <div class="container">
    <h2 class="page-header">点击切换颜色</h2>
    <div class="item"></div>
    <div class="item"></div>
    <div class="item"></div>
  </div>
  <script>
    //获取div元素对象
    let items = document.getElementsByClassName('item');

    //遍历并绑定事件
    for(let i = 0;i<items.length;i++){
      items[i].onclick = function(){
        //修改当前元素的背景颜色
        // this.style.background = 'pink';
        items[i].style.background = 'pink';
      }
    }

  </script>
</body>

</html>

```

## 9.2 函数的定义

```

//函数的定义
function test() {
  var name = "此处为局部变量的信息"; //定义局部变量
  alert(name); //弹出信息: "此处为局部变量的信息"
}

var test = function()
{
  var name = "此处为局部变量的信息"; //定义局部变量
  alert(name); //弹出信息: "此处为局部变量的信息"
}

```

```
}  
var test = () =>{  
    var name = "此处为局部变量的信息"; //定义局部变量  
    alert(name); //弹出信息：“此处为局部变量的信息”  
}
```

## 9.3 运算符

### JavaScript 算数运算符

算数运算符用于对数字执行算数运算：

运算符	描述
+	加法
-	减法
*	乘法
/	除法
%	取模（余数）
++	递加
--	递减

**注释：**[JS 算数](#)这一章对算数运算符进行了完整描述。

### JavaScript 赋值运算符

赋值运算符向 JavaScript 变量赋值。

运算符	例子	等同于
=	x = y	x = y
+=	x += y	x = x + y
-=	x -= y	x = x - y
*=	x *= y	x = x * y
/=	x /= y	x = x / y
%=	x %= y	x = x % y

加法赋值运算符（+=）向变量添加一个值。

### JavaScript 比较运算符

运算符	描述
==	等于
===	等值等型
!=	不相等
!==	不等值或不等型
>	大于
<	小于
>=	大于或等于
<=	小于或等于
?	三元运算符

注释：[JS 比较](#)这一章中完整描述了比较运算符。

### JavaScript 逻辑运算符

运算符	描述
&&	逻辑与
	逻辑或
!	逻辑非

注释：[JS 比较](#)这一章中完整描述了逻辑运算符。

### JavaScript 类型运算符

运算符	描述
typeof	返回变量的类型。
instanceof	返回 true，如果对象是对象类型的实例。

注释：[JS 类型转换](#)这一章完整描述了类型运算符。

### JavaScript 位运算符

位运算符处理 32 位数。

该运算中的任何数值运算数都会被转换为 32 位的数。结果会被转换回 JavaScript 数。





## 9.5 Js 中的反引号

```
var a = 5;
var b = 10;
document.write('Fifteen is ' + (a + b) + ' and not ' + (2 * a + b) +
'.');

var a = 5;
var b = 10;
document.write(`Fifteen is ${a + b} and
not ${2 * a + b}.`);
```

## 9.6 Js数组

JavaScript中的数组类似于Python的列表

```
a = [11,22,33,44]
```

`obj.length`      数组的大小

`obj.push(ele)`      尾部追加元素  
`obj.pop()`      尾部获取一个元素  
`obj.unshift(ele)`      头部插入元素  
`obj.shift()`      头部移除元素  
`obj.splice(start, deleteCount, value, ...)` 插入、删除或替换数组的元素  
    `obj.splice(n,0,val)` 指定位置插入元素  
    `obj.splice(n,1,val)` 指定位置替换元素  
    `obj.splice(n,1)`    指定位置删除元素

`obj.slice( )`      切片  
`obj.reverse( )`    反转  
`obj.join(sep)`      将数组元素连接起来以构建一个字符串  
`obj.concat(val,...)` 连接数组  
`obj.sort( )`      对数组元素进行排序

## 9.7 Js字典

定义：

```
let dict = {"k1":"v1","k2":"v2"}
{k1: "v1", k2: "v2"}
```

字典取值：

```
dict = {"k1":"v1","k2":"v2"}
{k1: "v1", k2: "v2"}
dict["k1"]
"v1"
dict["k2"]
"v2"
```

## 9.8 Js数组排序sort()

sort() 方法以字母顺序对数组进行排序：默认地，sort() 函数按照字符串顺序对值进行排序。

### 实例

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.sort();           // 对 fruits 中的元素进行排序
```

### 对于数字

使用比值函数完成；

### 实例-降序

```
var points = [40, 100, 1, 5, 25, 10];
points.sort(function(a, b){return b - a});
```

## 9.9 Js操作JSON常用方法

<https://blog.csdn.net/xujie3/article/details/52954940>

### 一、概念简介

JSON(JavaScript Object Notation) 是一种轻量级的数据交换格式，采用完全独立于语言的文本格式，是理想的数据交换格式。同时，JSON是 JavaScript 原生格式，这意味着在 JavaScript 中处理 JSON数据不须要任何特殊的 API 或工具包。

在JSON中，有两种结构：对象和数组。

#### 1.对象

```
var packJson = {"name":"Liza", "password":"123"};
```

一个对象以“{”开始，“}”结束，“key/value”之间运用“,”分隔。

#### 2.数组

```
var packJson = [{"name":"Liza", "password":"123"}, {"name":"Mike", "password":"456"}];
```

数组是值的有序集合。一个数组以“[”开始，“]”结束。值之间运用“,”分隔。

### 二、JSON对象和JSON字符串的转换

在数据传输过程中，JSON是以字符串的形式传递的，而JS操作的是JSON对象，所以，JSON对象和JSON字符串之间的相互转换是关键。例如：

JSON字符串：

```
var jsonStr = '{"name":"Liza", "password":"123"}' ;
```

JSON对象：

```
var jsonObj = {"name":"Liza", "password":"123"};
```

## 1、JSON字符串转化JSON对象

```
var jsonObject= jQuery.parseJSON(jsonstr);
```

## 2、JSON对象转化JSON 字符串

```
var jsonstr =JSON.stringify(jsonObject );
```

# 三、遍历JSON对象和JSON数组

## 1、遍历JSON对象代码如下：

```
var packJson  = {"name":"Liza", "password":"123"} ;

for(var k in packJson ){//遍历packJson 对象的每个key/value对,k为key
    alert(k + " " + packJson[k]);
}
```

## 2、遍历JSON 数组代码如下：

```
var packJson = [{"name":"Liza", "password":"123"}, {"name":"Mike",
"password":"456"}];

for(var i in packJson){//遍历packJson 数组时, i为索引
    alert(packJson[i].name + " " + packJson[i].password);
}
```

## 9.10 Date 对象

new Date(year,mon,day,hour,min,sec)中, mon月份是从0开始的, 0表示1月, 11表示12月。其他都正常。

# 第13章 JavaScript对象编程

## 13.1 HTML DOM nextElementSibling 属性

nextSibling 属性与 nextElementSibling 属性的差别：

- nextSibling 属性返回元素节点之后的兄弟节点（包括文本节点、注释节点）；
- nextElementSibling 属性只返回元素节点之后的兄弟元素节点（不包括文本节点、注释节点）；

```
<!DOCTYPE html>
<html>
  <body>
    <p>实例列表:</p>
    <ul>
      <li id="item1">咖啡（第一项）</li>
      <li id="item2">西瓜（第二项）</li>
    </ul>
    <p>点击以下按钮，获取列表中第一个li元素节点之后的兄弟元素的内容。</p>

    <button onclick="myFunction()">点我</button>
    <p><strong>注意:</strong> IE8 及以下版本不支持 nextElementSibling 属性。</p>
    <p id="demo"></p>
```

```
<script>
    function myFunction() {
        var x =
document.getElementById("item1").nextElementSibling.innerHTML;
        document.getElementById("demo").innerHTML = x;
    }
</script>
</body>
</html>
```

## 13.2 href="javascript:void(0);"

为什么要使用href="javascript:void(0);"

javascript:是伪协议，表示url的内容通过javascript执行。void(0)表示不作任何操作，这样会防止链接跳转到其他页面。这么做往往是为了保留链接的样式，但不让链接执行实际操作，

```
<a href="javascript: void(0)" onclick="window.open()">
```

点击链接后，页面不动，只打开链接

## 13.3 MIME类型

**媒体类型**（通常称为 **Multipurpose Internet Mail Extensions** 或 **MIME 类型**）是一种标准，用来表示文档、文件或字节流的性质和格式。它在[IETF RFC 6838](#)中进行了定义和标准化。

### 语法

#### 通用结构

```
type/subtype
```

MIME的组成结构非常简单；由类型与子类型两个字符串中间用 '/' 分隔而组成。不允许空格存在。*type* 表示可以被分多个子类的独立类别。*subtype* 表示细分后的每个类型。

MIME类型对大小写不敏感，但是传统写法都是小写。

#### 独立类型

```
text/plain
text/html
image/jpeg
image/png
audio/mpeg
audio/ogg
audio/*
video/mp4
application/*
application/json
application/javascript
application/ecmascript
application/octet-stream
...
```

独立类型表明了对文件的分类，可以是如下之一：

类型	描述	典型示例
text	表明文件是普通文本， 理论上是人类可读	text/plain, text/html, text/css, text/javascript
image	表明是某种图像。不包括视频，但是动态图（比如动态gif）也使用image类型	image/gif, image/png, image/jpeg, image/bmp, image/webp, image/x-icon, image/vnd.microsoft.icon
audio	表明是某种音频文件	audio/midi, audio/mpeg, audio/webm, audio/ogg, audio/wav
video	表明是某种视频文件	video/webm, video/ogg
application	表明是某种二进制数据	application/octet-stream, application/pkcs12, application/vnd.ms-powerpoint, application/xhtml+xml, application/xml, application/pdf

## 13.4 JQuery UI

<https://jqueryui.com/>

Query UI is a curated set of user interface interactions, effects, widgets, and themes built on top of the jQuery JavaScript Library. Whether you're building highly interactive web applications or you just need to add a date picker to a form control, jQuery UI is the perfect choice.

## 第14章 Bootstrap-table

<https://bootstrap-table.com/>

```
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet" href="https://unpkg.com/bootstrap-table@1.19.1/dist/bootstrap-table.min.css">

<!-- Latest compiled and minified JavaScript -->
<script src="https://unpkg.com/bootstrap-table@1.19.1/dist/bootstrap-table.min.js"></script>

<!-- Latest compiled and minified Locales -->
<script src="https://unpkg.com/bootstrap-table@1.19.1/dist/locale/bootstrap-table-zh-CN.min.js"></script>
```

## 反引号作用

模板字符串（template string）是增强版的字符串，用反引号（`）标识。它可以当作普通字符串使用，也可以用来定义多行字符串，或者在字符串中嵌入变量。

模板字符串中嵌入变量，需要将变量名写在`\${}`之中。

## 第15章 HBuilder

下载: <https://www.dcloud.io/hbuilderx.html>

## 一、解决中文乱码

1. 首先，在头部里加上代码`< meta charset="utf-8"/>`，保存

2. 然后，修改编辑器的编码方式（文件——以指定编码重新打开）

3. 正如上面所说，会发现编辑区的代码也乱码了，别急，`ctrl+z`撤销一下就行了。

怎么确定撤销到哪一步呢？我这里是一次就行了。你看到代码中`< meta charset="utf-8"/>`存在，右下角显示的编码方式也是`utf-8`，同时编码区域不出现乱码就保存一次。可能此时右边的web浏览器显示的中文还是乱码，刷新一下就行。