**前端交流平台**

**知乎**

**百度前端吧**

**Imooc讨论区**

**驼峰式拼写法/骆驼拼写法（Camel case）**

**是什么**

**是一种英语复合单词或短语的拼写方法（拼写惯例）。**

**特征**

**复合单词或短语中的每一个单词/缩写的首字母大写。（第一个单词首字母可小写）**

**整个复合单词或短语中没有空格或标点。**

**匈牙利拼写法**

**是什么**

**编程时的一种标识符（变量或函数）命名惯例（命名方法）。**

**特征**

**变量中每个单词的首字母都大写.**

**变量名的最前面再加一个小写字母，表示这个单词的数据类型.**

**备注**

**整型变量的前缀是n,长整型变量是nl,字符型数组变量是ca,以及字符串(以空类型结尾的字符数组)以sz为前缀**

**前端学习计划**

**html，css，js学习→基础良好（html标签、css的盒子模型）**

**堆页面→大部分页面都能做**

**开发框架，类库（如jquery，bootstrap，grunt）**

**往js方向深挖**

**基本的原生js**

**学习这两本书**

**高性能JavaScript**

**JavaScript高级程序设计（第3版）**

**熟练使用jquery**

**ajax，**

**常见浏览器的兼容。**

**前端资源**

**编程练习网站**

**Codecademy**

**Codewars**

**Free Code Camp**

**网易公开课，哈佛大学公开课：构建动态网站**

**知乎**

## 百度前端吧

**Imooc讨论区**

# 字符编码

## 字符编码

# UTF-16（16-bit Unicode Transformation Format）

# 数据结构

## 资料

### 入门

#### 《啊哈！算法》

#### 《大话数据结构》

#### 《算法》第四版，作者塞奇威客

### 初级

#### 《算法导论》

## 是什么：一种关系，数据元素之间存在的一种或多种特定关系的集合。

## 物理结构（存储结构）

### 是什么：数据的逻辑结构在计算机中的存储形式。是面向计算机的。

### 顺序存储结构

#### 是什么：把数据元素存放在地址连续的存储单元中，其数据间的逻辑关系和物理关系是一致的。

### 链式存储结构

#### 是什么：把数据元素存放在任意的存储单元中，这组存储单元可以是连续的，也可以是不连续的。

## 逻辑结构

### 是什么：指数据对象中数据元素之间的相互关系，是面向问题的。

### 集合结构

#### 集合结构中的数据元素除了同属于一个集合外，他们之间没有其它关系。

### 线性结构

#### 数据元素之间是一对一的关系。

#### 线性表（list）：零个或多个数据元素的有限序列。

##### 操作

###### 创建

###### 初始化

###### 清空

###### 判断是否为空

###### 返回i位置的值

###### 新增删除元素

##### 线性表的顺序存储结构（顺序表）

##### 线性表的链式存储结构（链表）

###### 链表

单链表

单向循环链表

双向链表

双向循环链表

静态链表：用数组描述的链表

##### 栈：限定仅在表尾进行插入和删除操作的线性表

###### 栈的顺序存储结构（顺序栈）

###### 栈的链式存储结构（链栈）

##### 队列：只允许在一端进行插入操作，而在另一端进行删除操作的线性表

###### 队列的顺序存储结构（顺序队列）

###### 队列的链式存储结构（链队列）

循环队列

### 树形结构

#### 数据元素之间存在一对多的层次关系。

#### 树

##### 是什么：是n（n≥0）个结点的有限集。

##### 二叉树

###### 特性

第i层最多有2i-1个结点

深度为k的二叉树，至多有2k-1个结点。

###### 二叉树的顺序存储结构

###### 二叉树的链式存储结构（二叉链表）

###### 遍历方法

前序遍历

中序遍历

后序遍历

层序遍历

### 图形结构

#### 数据元素之间是多对多的关系

#### 图

## 数据类型

### 是什么：一组性质相同的值的集合及定义在此集合上的一些操作的总称。

### 抽象数据类型

#### 是什么：一个数学模型及定义在该模型上的一组操作。

### 字符集

字符的编码有两个步骤，一是为字符指定一个唯一的二进制数字，叫编码（如unicode），二是将字符所对应的数字存储在计算机中（如UTF-16），叫存储。

***是什么***

字符的编号的集合

字符(Character)是各种文字和符号的总称，包括各国家文字、标点符号、图形符号、数字等。字符集(Character set)是多个字符的集合

#### 发展：ASCII-7→ASCII-8→unicode-16

#### 常见字符集

##### ASCII（rican Standard Code for Information Interchange）

###### 可显示字符：英文（包括大小写字符、阿拉伯数字和西文符号）

##### GB2312-80（信息交换用汉字编码字符集·基本集）

###### 可显示字符：简化汉字及一般符号、序号、数字、拉丁字母、日文假名、希腊字母、俄文字母、汉语拼音符号、汉语注音字母，共 7445 个图形字符。

##### BIG5

###### 台湾使用

##### GB18030-2000（信息交换用汉字编码字符集基本集的扩充）

##### Unicode（Universal Multiple-Octet Coded Character Set）：一项标准

###### 可显示字符：所有语言，所有字符

###### UTF-8（8-bit Unicode Transformation Format）

表示一个字符是可变的，有可能是用一个字节表示一个字符，也可能是两个，三个。最多不能超过4个字节。

###### UTF-16（16-bit Unicode Transformation Format）

基本平面的字符占用2个字节，辅助平面的字符占用4个字节。

###### UTF-32（32-bit Unicode Transformation Format）

### 串（字符串）

#### 是什么：由零个或多个字符组成的有限序列

#### 串的顺序存储结构

#### 串的链式存储结构

# 算法

## 是什么：是解决特定问题求解步骤的描述，在计算机中变现为指令的有限序列，并且每条指令表示一个或多个操作。

## 特性

### 输入：具有0个或多个输入。

### 输出：算法至少有一个或多个输出。

### 有穷性：指算法在执行有限的步骤之后，自动结束而不会出现无限循环，并且每一个步骤在可接受的时间内完成。

### 确定性：算法的每一个步骤都具有确定的含义，不会出现二义性。

### 可行性：算法的每一步都必须是可行的，也就是说，每一步都能够通过执行有限次数完成。

## 好算法的特性

### 正确性

#### 算法应该至少具有输入、输出和加工处理无歧义性、能正确反映问题的需求，能够得到问题的正确答案。

### 可读性

#### 算法要便于阅读、理解和交流。

### 健壮性

#### 当输入数据不合法时，算法也能做出相关处理，而不是产生异常或莫名其妙的结果。

### 时间效率高

#### 评价方法

##### 事后统计方法（比较运行时间）

##### 事前分析估算方法

#### 算法（渐进）时间复杂度（大O记法）

##### 是什么：在进行算法分析时，语句总的执行次数T(n)是关于问题规模n的函数，进而分析T(n)随n的变化情况，并确定T(n)的数量级。记作T(n)=O(f(n))

##### O(1)<O(logn)<O(n)<O(nlogn)<O(n2)<O(n3)<O(2n) <O(n!) <O(nn)

##### 平均时间复杂度

##### 最坏时间复杂度

### 存储量低

#### 算法空间复杂度

##### 是什么：算法在计算机上所需的存储空间。S(n)=O(f(n))，n为问题的规模。

## 迭代函数

## 递归函数

## 朴素的模式匹配算法

## KMP模式匹配算法

## 常见算法

### 快速排序算法

#### 步骤

# 结构化程序设计原则

## 每个函数应该只有一个入口点和一个出口点

# ECMA（EuropeanComputerManufacturesAssociation欧洲计算机制造联合会）

## 是什么

### 国际组织，类似国际标准化组织（ISO）、国际电子技术协会（IEC）

## 功能

### 评估，开发和认可电信和计算机标准。

### 建立统一的电脑操作格式标准--包括程序语言和输入输出

# ECMAScript（ECMA-262标准/Standard ECMA-262）

## 是什么

一种脚本程序设计语言标准（规范）。

## 标准 ECMAScript 内置对象

### 全局对象

### object对象

### array对象

### string对象

### 布尔对象

### number对象

### math对象

### date对象

#### 属性

##### Date.prototype

#### 方法

##### Date.UTC()

##### Date.now()

##### Date.parse()

##### Date.prototype.getDate()

##### Date.prototype.getDay()

##### Date.prototype.getFullYear()

##### Date.prototype.getHours()

##### Date.prototype.getMilliseconds()

##### Date.prototype.getMinutes()

##### Date.prototype.getMonth()

##### Date.prototype.getSeconds()

##### Date.prototype.getTime()

##### Date.prototype.getTimezoneOffset()

##### Date.prototype.getUTCDate()

##### Date.prototype.getUTCDay()

##### Date.prototype.getUTCFullYear()

##### Date.prototype.getUTCHours()

##### Date.prototype.getUTCMilliseconds()

##### Date.prototype.getUTCMinutes()

##### Date.prototype.getUTCMonth()

##### Date.prototype.getUTCSeconds()

##### Date.prototype.getYear()

##### Date.prototype.setDate()

##### Date.prototype.setFullYear()

##### Date.prototype.setHours()

##### Date.prototype.setMilliseconds()

##### Date.prototype.setMinutes()

##### Date.prototype.setMonth()

##### Date.prototype.setSeconds()

##### Date.prototype.setTime()

##### Date.prototype.setUTCDate()

##### Date.prototype.setUTCFullYear()

##### Date.prototype.setUTCHours()

##### Date.prototype.setUTCMilliseconds() [我来译!]

##### Date.prototype.setUTCMinutes() [我来译!]

##### Date.prototype.setUTCMonth()

##### Date.prototype.setUTCSeconds()

##### Date.prototype.setYear()

##### Date.prototype.toDateString()

##### Date.prototype.toGMTString()

##### Date.prototype.toISOString()

##### Date.prototype.toJSON()

##### Date.prototype.toLocaleDateString()

##### Date.prototype.toLocaleFormat()

##### Date.prototype.toLocaleString()

##### Date.prototype.toLocaleTimeString()

##### Date.prototype.toSource() [我来译!]

##### Date.prototype.toString()

##### Date.prototype.toTimeString()

##### Date.prototype.toUTCString()

##### Date.prototype.valueOf()

##### Date.prototype[@@toPrimitive]

### regexg对象

### error对象

### JSON对象

## 备注

Javascript是ECMAScript的实现。

# CSS

## 是什么：cascading style sheets 层叠样式表

## 继承

### 节点树上的各元素的样式属性将继承其父元素的样式属性。

## 语句类型

### 分类（按所处位置）

#### 优先级

内联样式>内部样式表>外部样式表

内部样式表或外部样式表中

从上往下原则，往下的样式会覆盖上边的样式

important样式>id选择器>其它

#### 选择器

**CSS预处理（器）语言**

**Sass（SCSS）**

**LESS**

**Stylus**

**Turbine**

**Swithch CSS**

**CSS Cacheer**

**DT CSS**

**Sass**

**SCSS**

**是什么：SCSS是SASS的新语法，是Sassy CSS的简写，是CSS3语法的超集**

**区别**

**文件扩展名不同，Sass 是以“.sass”后缀为扩展名，而 SCSS 是以“.scss”后缀为扩展名**

**语法书写方式不同，Sass 是以严格的缩进式语法规则来书写，不带大括号({})和分号(;)，而 SCSS 的语法书写和我们的 CSS 语法书写方式非常类似。**

**是什么**

**CSS预处理器语言的一种**

**是CSS3的一个扩展**

**功能：为CSS增加一些编程的特性**

**安装**

**安装Ruby→安装sass**

**编译**

**方法**

**命令编译**

**实例**

**sass --watch sass/bootstrap.scss:css/bootstrap.css**

**GUI工具编译**

**常见GUI 界面编译工具**

**Koala (**[**http://koala-app.com/**](http://koala-app.com/)**)（推荐）**

**CodeKit（https://incident57.com/codekit/index.html）（推荐）**

**Compass.app（http://compass.kkbox.com/）**

**Scout（http://mhs.github.io/scout-app/）**

**Prepros（https://prepros.io/）**

**自动化编译（Grunt、Gulp）**

**输出方式（风格）**

**嵌套输出方式 nested**

**展开输出方式 expanded**

**紧凑输出方式 compact**

**压缩输出方式 compressed**

**文档对象模型（DOM/documentobjectmodel）**

**是什么**

**文档对象模型(Document Object Model)**

**应用程序接口（application programming interface）**

**特性**

**跨平台**

**独立于各编程语言**

**作用（功能）**

**treats an HTML, XHTML, or XML document as a tree structure**

**节点（对象）**

**节点类型**

***元素节点（nodeType属性值为1）***

***属性***

***elementNode.childNodes***

***elementNode.firstChild***

***elementNode.lastChild***

***方法***

***elementNode.getAttribute(attribute)***

***elementNode.setAttribute(attribute)***

***属性节点（nodeType属性值为2）***

***style属性对元素节点来说是个属性，对特定属性来说，又是一个对象***

***文本节点（nodeType属性值为3）***

***CDATA\_SECTION\_NODE（nodeType属性值为4）***

***ENTITY\_REFERENCE\_NODE（nodeType属性值为5）***

***ENTITY\_NODE（nodeType属性值为6）***

***PROCESSING\_INSTRUCTION\_NODE（nodeType属性值为7）***

***注释节点（nodeType属性值为8）***

***DOCUMENT\_NODE（nodeType属性值为9）***

***DOCUMENT\_TYPE\_NODE（nodeType属性值为10）***

***DOCUMENT\_FRAGMENT\_NODE（nodeType属性值为11）***

***NOTATION\_NODE（nodeType属性值为12）***

***空格节、换行符***

*属性（节点通用）*

*nodeName*

*nodeType*

*nodeValue*

*Document对象*

*方法*

document.getElementById()

document.getElementsByName()

*作用：返回带有指定名称的节点对象的集合。*

*备注*

*因为文档中的 name 属性可能不唯一，所有 getElementsByName() 方法返回的是元素的数组，而不是一个元素。*

*和数组类似也有length属性，可以和访问数组一样的方法来访问，从0开始。*

document.getElementsByTagName()

*作用：返回带有指定标签名的节点对象的集合。返回元素的顺序是它们在文档中的顺序。*

*语法*

document.getElementsByTagName(Tagname)

*备注*

返回的是元素的数组，而不是一个元素。

和数组类似也有length属性，可以和访

#### document.write()

## forms对象

# 浏览器对象模型（BOM/browserobjectmodel）

## 是什么

## 作用

### 提供了独立于内容而与浏览器窗口进行交互的对象。

### 用于管理窗口与窗口之间的通讯。

## 备注

### 没有统一的标准

## 对象

### window

#### 方法

##### window.open(url,name,features)

##### window.onload()

### Frame

### Location

### History

### Navigator

#### 属性

##### [activeVRDisplays](https://developer.mozilla.org/zh-CN/docs/Web/API/Navigator/activeVRDisplays)

##### [appCodeName](https://developer.mozilla.org/zh-CN/docs/Web/API/NavigatorID/appCodeName)

##### [appName](https://developer.mozilla.org/zh-CN/docs/Web/API/NavigatorID/appName)

##### [appVersion](https://developer.mozilla.org/zh-CN/docs/Web/API/NavigatorID/appVersion)

##### [battery](https://developer.mozilla.org/zh-CN/docs/Web/API/Navigator/battery)

##### [buildID](https://developer.mozilla.org/zh-CN/docs/Web/API/Navigator/buildID)

##### [connection](https://developer.mozilla.org/zh-CN/docs/Web/API/Navigator/connection)

##### [cookieEnabled](https://developer.mozilla.org/zh-CN/docs/Web/API/Navigator/cookieEnabled)

##### [credentials](https://developer.mozilla.org/zh-CN/docs/Web/API/Navigator/credentials)

##### [doNotTrack](https://developer.mozilla.org/zh-CN/docs/Web/API/Navigator/doNotTrack)

##### *geolocation*

###### getCurrentPosition()

###### watchPosition()

###### clearWatch()

##### [id](https://developer.mozilla.org/zh-CN/docs/Web/API/Navigator/id)

##### [language](https://developer.mozilla.org/zh-CN/docs/Web/API/NavigatorLanguage/language)

##### [languages](https://developer.mozilla.org/zh-CN/docs/Web/API/NavigatorLanguage/languages)

##### [maxTouchPoints](https://developer.mozilla.org/zh-CN/docs/Web/API/Navigator/maxTouchPoints)

##### [mediaDevices](https://developer.mozilla.org/zh-CN/docs/Web/API/Navigator/mediaDevices)

##### [mozBluetooth](https://developer.mozilla.org/zh-CN/docs/Web/API/Navigator/mozBluetooth)

##### [mozNotification](https://developer.mozilla.org/zh-CN/docs/Web/API/Navigator/mozNotification)

##### [mozPermissionSettings](https://developer.mozilla.org/zh-CN/docs/Web/API/Navigator/mozPermissionSettings)

##### [mozSocial](https://developer.mozilla.org/zh-CN/docs/Web/API/Navigator/mozSocial)

##### [mozTime](https://developer.mozilla.org/zh-CN/docs/Web/API/Navigator/mozTime)

##### [mozVoicemail](https://developer.mozilla.org/zh-CN/docs/Web/API/Navigator/mozVoicemail)

##### [onLine](https://developer.mozilla.org/zh-CN/docs/Web/API/NavigatorOnLine/onLine)

##### [oscpu](https://developer.mozilla.org/zh-CN/docs/Web/API/Navigator/oscpu)

##### [permissions](https://developer.mozilla.org/zh-CN/docs/Web/API/Navigator/permissions)

##### [platform](https://developer.mozilla.org/zh-CN/docs/Web/API/NavigatorID/platform)

##### [product](https://developer.mozilla.org/zh-CN/docs/Web/API/NavigatorID/product)

##### [productSub](https://developer.mozilla.org/zh-CN/docs/Web/API/Navigator/productSub)

##### [push](https://developer.mozilla.org/zh-CN/docs/Web/API/Navigator/push)

##### [serviceWorker](https://developer.mozilla.org/zh-CN/docs/Web/API/Navigator/serviceWorker)

##### [userAgent](https://developer.mozilla.org/zh-CN/docs/Web/API/NavigatorID/userAgent)

##### [vendor](https://developer.mozilla.org/zh-CN/docs/Web/API/Navigator/vendor)

##### [vendorSub](https://developer.mozilla.org/zh-CN/docs/Web/API/Navigator/vendorSub)

#### 方法

##### addIdleObserver()

##### getBattery()

##### getGamepads()

##### getUserMedia

##### getVRDisplays()

##### mozIsLocallyAvailable()

##### registerContentHandler()

##### registerProtocolHandler()

##### requestMediaKeySystemAccess()

##### requestWakeLock()

##### sendBeacon()

##### taintEnabled()

##### vibrate()

### Document

#### Layer

#### Link

#### Location

#### Image

#### Area

#### Anchor

#### Applet

#### Plugin

#### Form

##### Textarea

##### Text

##### Fileupload

##### Password

##### Hidden

##### Submit

##### Reset

##### Radio

##### Checkbox

##### Button

##### Select—option

***Console***

方法

*Log()*

# Jscript

## 是什么

### 微软版的JavaScript（javascript是netscape公司的）

### 脚本程序设计语言

## 备注

### 与javascript基本无差异，主要的差异体现在文档建模方面。

# Javascript

## 是什么

一种脚本程序设计语言

是一种面向对象编程语言，但和传统的面向对象语言又有很大区别。

## 代码位置

<head>中

<body>中

外部文件中

## 语法（词法）

实例：<script src="script.js"></script>

字符集：Javascript采用Unicode字符集编写。

### 操作符

#### 算术操作符

+，-，\*，/，%，

\*\*：幂

++：自加一

--：自减一

-=

+=

\*=

/=

#### 比较操作符

##### =：赋值操作符

##### ==：等同于数学上的等于

##### ===：严格等于，即数据类型与值都必须相等

##### !=

##### !==

##### >

##### >=

##### <

##### <=

#### 逻辑操作符

##### &&

##### ||

##### !

#### 操作符优先级

除法、乘法等操作符的优先级比加法和减法高

算术操作符＞比较操作符＞逻辑操作符＞"="赋值符号

如果是同级的运算则按从左到右次序进行,多层括号由里向外。

### unicode转义序列

#### \\

#### \n

#### \r

#### \’

#### \”

#### \t

#### \b

#### \f

### 注释

单行注释，在注释内容前加符号 “//”。

多行注释以"/\*"开始，以"\*/"结束。

### 直接量（程序中直接使用的数据值）

**undefined（未定义）**

**null（空）**

**boolean（布尔型）**

**string（字符串）**

***属性***

Length

**symbol（符号）**

**number（数字）**

**Array（数组）**

**object（对象）**

***Document对象***

***DOM对象***

***浏览器对象（browser）***

Window

Navigator

Screen

History

Location

***Javascript对象（内置）***

Array

*方法（函数）*

*Push()：在后面增加*

*Pop()：在后面删除*

*Shift()：在前面删除*

*Unshift()：在前面增加*

Boolean

Date

Math

Number

String

RegExp

Global

Console

*方法（函数）*

*Log()*

JSON(javascript object notation)

*是什么*

*一种语法。*

基于javascript，但是又有所区别

*一种数据交换格式*

*json简单说就是javascript中的对象和数组*

*JSON 是 JavaScript 原生格式*

*作用：用来序列化对象、数组、数值、字符串、布尔值和 null*

*JSON 可以将 JavaScript 对象中表示的一组数据转换为字符串，然后就可以在函数之间轻松地传递这个字符串*

*方法（函数）*

*JSON.stringify()*

作用：返回指定值的JSON字符串，可以自定义只包含某些特定的属性或替换属性值。

### 标识符、保留字、关键字

#### 标识符必须以字母、下划线（-）或美元符（$）开始

#### 关键字

##### this、

### “;”（分号）：将语句分隔开

#### 如果当前语句和下一行语句无法合并解析，javascript会在第一行后填补分号。

### 条件语句

#### 语法1

|  |
| --- |
| if(condition){ |
| statements; |
| } |

#### 语法2：如果代码块只有一条语句，则花括号可省略。

##### if(1>2)alert("the world has gone mad!");

#### 备注

##### 条件为true时执行代码。

### 循环语句

#### While循环

##### 语法

|  |
| --- |
| while(condition){ |
| statements; |
| } |

##### 备注

###### 条件为true时执行代码块。

#### Do…while循环

##### 语法

|  |
| --- |
| do{ |
| statements; |
| }while(condition); |

##### 备注

###### 代码块至少被执行一次

#### For循环

##### 语法

|  |
| --- |
| for(initial condition; test condition; alter condition){ |
| statements; |
| } |

##### 备注

###### 这种语句格式更为紧凑，可能这也是发明它的原因。

### 变量

#### 备注

##### 先声明，再赋值。

#### 变量的作用域

##### 全局变量：可以在文档的任何位置被引用

##### 局部变量：只存在于对它做出声明的那个函数的内部。外部无法应用。

### 其它

#### 区分大小写

#### 忽略空格、换行符

## javascript数据类型（值）

### undefined（未定义）

### null（空）

### Boolean（布尔值）

### string（基本字符串）

### number（数字）

### 对象

是什么：由一些彼此相关的属性和方法集合在一起而构成的一个数据实体。

#### 属性

是什么：属性是隶属于某个特定对象的变量

访问语法：Object.property

#### 方法

##### 是什么

方法是只有某个特定对象才能调用的函数

方法其实也是一种属性值，只不过该属性值为函数。如array.split()方法，属性名为split()，属性值为function split()

##### 访问语法：object.method()

#### 创建（新实例）

语法：var instance\_name = new object\_name

#### 分类（按归属）

##### 宿主对象

是什么：即由javascript语言的运行环境（即各种浏览器）提供的对象。

###### Form

###### Image

###### Element

###### Windows对象

备注：这个对象的属性和方法通常被称作BOM（浏览器对象模型），但其实叫window object model（窗口对象模型）更准确。

##### javascript内建对象

###### boolean对象

###### number对象

***string对象***

属性

*Length*

方法

String.fromCharCode()：根据unicode创建字符串→字符串（非string对象）

String.fromCodePoint()

String.prototype.anchor()

String.prototype.big()

String.prototype.blink()

String.prototype.bold()

String.prototype.charAt()

String.prototype.charCodeAt()：获取字符utf-16编码→整数（小于 65,536）

String.prototype.codePointAt()

String.prototype.concat()

String.prototype.endsWith()

String.prototype.fixed()

String.prototype.fontcolor()

String.prototype.fontsize()

String.prototype.includes()

String.prototype.indexOf()

String.prototype.italics()

String.prototype.lastIndexOf()

String.prototype.link()

String.prototype.localeCompare()

String.prototype.match()

String.prototype.normalize()

String.prototype.padEnd()

String.prototype.padStart()

String.prototype.quote()

String.prototype.repeat()

String.prototype.replace()

String.prototype.search()

String.prototype.slice()

String.prototype.small()

String.prototype.split()

String.prototype.startsWith()

String.prototype.strike()

String.prototype.sub()

String.prototype.substr()

String.prototype.substring()

String.prototype.sup()

String.prototype.toLocaleLowerCase()

String.prototype.toLocaleUpperCase()

String.prototype.toLowerCase()

String.prototype.toSource()

String.prototype.toString()

String.prototype.toUpperCase()

String.prototype.trim()

String.prototype.trimLeft()

String.prototype.trimRight()

String.prototype.valueOf()

String.prototype[@@iterator]()

String.raw()

###### String.prototype

String.prototype.charAt()

String.prototype.split()→数组对象实例

String.prototype.replace()→字符串对象实例

String.prototype.toLowerCase()→字符串对象实例

###### Array.prototype

方法

Array.from()

→数组

Array.isArray()

→true/false

~~Array.observe()~~

Array.of()

→数组实例

Array.prototype.concat()：合并数组

→数组实例

Array.prototype.copyWithin()：复制数组元素→数组实例

Array.prototype.entries()

Array.prototype.every()

Array.prototype.fill()

Array.prototype.filter()

Array.prototype.find()

Array.prototype.findIndex()

Array.prototype.forEach()：对数组的每个元素执行一次函数

Array.prototype.includes()

Array.prototype.indexOf()

Array.prototype.join()

Array.prototype.keys()

Array.prototype.lastIndexOf()

Array.prototype.map()

Array.prototype.pop()

Array.prototype.push()

Array.prototype.reduce()

Array.prototype.reduceRight()

Array.prototype.reverse()

Array.prototype.shift()

Array.prototype.slice()

Array.prototype.some()

Array.prototype.sort()

Array.prototype.splice()

Array.prototype.toLocaleString()

Array.prototype.toSource()

Array.prototype.toString()

Array.prototype.unshift()

Array.prototype.values()

Array.prototype[@@iterator]()

Array.unobserve()

get Array[@@species]

###### Math

###### Date

方法

getDay()

getHours()

getMonths()

###### regExp

###### function

***是什么：将代码块打包，随时调用。***

***事件处理函数***

是什么：一种在浏览器中注册的函数

作用：在预定事件发生时让预先安排好的javascript代码开始执行。

###### arguments

备注

arguments对象不是一个 Array 。它类似于数组，但除了 长度之外没有任何数组属性。

arguments 对象仅在函数内部有效

###### Array对象

是什么：是一种特殊的对象（有序，且有专用的语法）

属性

Length

方法

concat()

split()

Push()：在后面增加

Pop()：在后面删除

Shift()：在前面删除

Unshift()：在前面增加

作用：用于构造数组实例

*ArrayBuffer*

*AsyncFunction*

*Atomics*

*Boolean*

*DataView*

*Date*

*Error*

*EvalError*

*Float32Array*

*Float64Array*

*Function*

*Generator*

*GeneratorFunction*

*Infinity*

*Int16Array*

*Int32Array*

*Int8Array*

*InternalError*

*Intl*

*Intl.Collator*

*Intl.DateTimeFormat*

*Intl.NumberFormat*

*Iterator*

*JSON*

*Map*

*Math*

*NaN*

*Number*

*Object*

*ParallelArray*

*Promise*

*Proxy*

*RangeError*

*ReferenceError*

*Reflect*

*RegExp*

*SIMD*

*SIMD.Bool16x8 [我来译!]*

*SIMD.Bool32x4 [我来译!]*

*SIMD.Bool64x2 [我来译!]*

*SIMD.Bool8x16 [我来译!]*

*SIMD.Float32x4 [我来译!]*

*SIMD.Float64x2 [我来译!]*

*SIMD.Int16x8 [我来译!]*

*SIMD.Int32x4 [我来译!]*

*SIMD.Int8x16 [我来译!]*

*SIMD.Uint16x8 [我来译!]*

*SIMD.Uint32x4 [我来译!]*

*SIMD.Uint8x16 [我来译!]*

*Set*

*SharedArrayBuffer*

*StopIteration [我来译!]*

*String*

*Symbol*

*SyntaxError*

*TypeError*

*TypedArray*

*URIError [我来译!]*

*Uint16Array [我来译!]*

*Uint32Array*

*Uint8Array*

*Uint8ClampedArray [我来译!]*

*WeakMap*

*WeakSet*

*WebAssembly*

*decodeURI()*

*decodeURIComponent()*

*encodeURI()*

*encodeURIComponent()*

*escape()*

*eval()*

*isFinite()*

*isNaN()*

*null*

*parseFloat*

*parseInt*

*undefined*

*unescape()*

*uneval()*

*转义字符*

|  |  |
| --- | --- |
| Code | Output |
| \0 | 空字符 |
| \' | 单引号 |
| \" | 双引号 |
| \\ | 反斜杠 |
| \n | 换行 |
| \r | 回车 |
| \v | 垂直制表符 |
| \t | 水平制表符 |
| \b | 退格 |
| \f | 换页 |
| \uXXXX | unicode 码 |
| \u{X} ... \u{XXXXXX} | unicode codepoint |
| \xXX | Latin-1 字符(x小写) |

## 继承

### 原型链继承（对象间的继承）

### 类式继承（构造函数间的继承）

## JSON(javascriptobjectnotation)

### 是什么

#### JSON是JS对象的字符串表示法，它使用文本表示一个JS对象的信息，本质是一个字符串。

#### 一种语法，基于javascript，但是又有所区别。

#### 一种数据交换格式。

#### json简单说就是javascript中的对象和数组

#### JSON是JavaScript原生格式

### 作用：用来序列化对象、数组、数值、字符串、布尔值和null

#### JSON可以将JavaScript对象中表示的一组数据转换为字符串，然后就可以在函数之间轻松地传递这个字符串

### 方法

#### JSON.parse()：将字符串→javascript对象。

#### JSON.stringify()：将javascript对象→字符串。

##### 作用：返回指定值的JSON字符串，可以自定义只包含某些特定的属性或替换属性值。

**正则表达式javascript实现**

**是什么**

**是一种规则，一种对字符串特征的描述。**

**用一个“字符串”来描述一个特征，然后去验证另一个“字符串”是否符合这个特征。**

**备注**

**Javascript中使用regexp对象来支持正则表达式。一个正则表达式直接量会在执行到它时转换为一个regexp对象。**

**采用NFA引擎**

**作用**

**验证字符串是否符合指定特征**

**查找字符串，从一个长的文本中查找符合指定特征的字符串，比查找固定字符串更加灵活方便。**

**用来替换。**

**创建方法**

**直接量法**

**构造函数法**

**字符**

**直接量字符**

**非直接量字符**

**转义字符“\”**

**空白字符**

**" "(空格符)**

**\r (回车符)**

**\n (换行符)**

**\t (制表符)**

**\f (换页符)**

**元字符**

**定义：字符类型的集合**

**重复字符“{}”**

**非贪婪的重复字符“?”**

**选择符“()”、分组符“()”、引用符**

**制定匹配位置字符**

**修饰符**

**字符类“[]”“\”（等价字符）**

**特殊字符类（等价字符）**

**“javascript:”伪协议**

**在html文档中通过“javascript:”伪协议调用javascript代码的做法非常不好。**

**字面量（字面值）**

**在所处上下文中，表达某个特定的值的字符序列。**

**隐藏元素方法**

**CSSopacity的指是0。**

**CSSdisplay的值是none。**

**type="hidden"的表单元素。**

**宽度和高度都显式设置为0。**

**一个祖先元素是隐藏的，该元素是不会在页面上显示**

**CSSvisibility的值是hidden.**

**javascript编程良好习惯原则**

**预留退路：确保网页在没有javascript的情况下也能正常工作。**

**分离javascript**

**向后兼容性**

**实践**

**字符串单词首字母大写**

**方法1**

**使用toLowerCase()方法将字符串字母小写化**

**使用split()方法将字符串分解为单词数组**

**使用toUpperCase()方法将数组中的每一项的首字母大写**

**使用replace()将大写的首字母替换进原单词**

**使用join()方法组合成一个新的字符串**

**方法2**

**使用“/(|^)[a-z]/g”配出整个字符串最开始的首字母以及以空格分隔的每个单词首字母。**

**使用replace()方法进行首字母大写化替换。**

## 事件

### 事件流

#### IE：事件冒泡流（逐级向上）

#### Netscape：事件捕获流（逐级向下）

***HTML事件处理程序***

***DOM0级事件处理程序***

***DOM2级事件处理程序***

***IE/opera事件处理程序***

***跨浏览器的事件处理程序***

## 判断变量数据类型

### typeof

|  |
| --- |
| var a="string"; console.log(a); //stringvar a=1; console.log(a); //numbervar a=false; console.log(a); //booleanvar a; console.log(typeof a); //undfinedvar a = null; console.log(typeof a); //objectvar a = document; console.log(typeof a); //objectvar a = []; console.log(a); //objectvar a = function(){}; console.log(typeof a) //function 除了可以判断数据类型还可以判断function类型 |

**instanceof**

**constructor**

**Jquery**

**是什么：是一个javascript脚本库**

**文件链接**

**链接位置**

***<head>之间***

**链接语法**

***<scriptsrc="http://libs.baidu.com/jquery/1.9.1/jquery.min.js"></script>***

**对象**

**jQuery对象与DOM对象是不一样的**

**DOM对象就是一个单独的DOM元素，而jQuery是一个类数组对象。**

**jQuery库本质上还是JavaScript代码，它只是对JavaScript语言进行包装处理，为的是提供更好更方便快捷的DOM处理与开发中经常使用的功能。我们使用jQuery的同时也能混合JavaScript原生代码一起使用。**

**选择器（节点集？）**

***参考jquery官方文档***

***XPath***

是什么

*即XML路径语言，是一种用来确定XML文档中某部分位置（查找信息）的语言。XPath用于在XML文档中通过元素和属性进行导航。*

*XPath使用路径表达式在XML文档中进行导航。*

*XPath包含一个标准函数库。*

*XPath是XSLT中的主要元素。*

*XPath是一个W3C标准。*

***基本选择器（basic）***

\*

.class

Element

#id

***层级选择器（hierarchy）***

parent > child

ancestor descendant

prev + next

prev ~ siblings

***基本筛选选择器（basic filter）***

:animated

:eq()

:even

:odd

:first

:focus

:gt()

:lang()

:last

:lt()

:not()

:root

:target

***属性选择器（attribute）***

[attribute|='value'](前缀—)

*描述： 选择指定属性值等于给定字符串或以该字符串为前缀（该字符串后跟一个连字符“-” ）的元素。*

*参数*

*attribute:：属性名。*

*value：属性值。可以是一个 有效标识符或带引号的字符串。*

[attribute\*='value']（含有）

*描述：Selects elements that have the specified attribute with a value containing a given substring.*

*参数*

*Att：属性名。*

*Val：属性值。可以是一个有效标识符或带引号的字符串。*

[attribute~='value']（含有该词（空格））

*描述：Selects elements that have the specified attribute with a value containing a given word, delimited by spaces.*

*参数*

*attribute: An attribute name.*

*value: An attribute value. Can be either a valid identifier or a quoted string.*

[attribute$='value']（结尾）

[attribute='value']（等于）

[attribute!='value']（不等于）

[attribute^='value']（开头）

[attribute]

***子元素筛选选择器（child filter）***

:first-child //长子

:last-child //幼子

:nth-child(n) //第n子

*描述：Selects all elements that are the nth-child of their parent.*

:only-child //独子

:nth-last-child(n) //倒数第n子

:first-of-type //同类第一个

:last-of-type //同类最后一个

:nth-of-type(n) //同类第n个

:only-of-type //同类单独一个

:nth-last-of-type(n) //同类倒数第个

***表单选择器***

:button

:checkbox

:file

:focus

:image

:input

:password

:radio

:reset

:submit

:text

:checked

:selected

:disabled

:enabled

***内容筛选选择器***

:contains()

:empty

:has()

:parent

***可见性筛选选择器***

:hidden

:visible

***this***

***选择器并集***

selector1, selector2, selectorN

**基本**

**\***

**.class**

**element**

**#id**

**selector1, selectorN, ...**

**层级**

**parent > child**

**ancestor descendant**

**prev + next**

**prev ~ siblings**

**基本筛选**

**:animated**

**:eq()**

**:even**

**:first**

**:gt()**

**:header**

**:lang()**

**:last**

**:lt()**

**:not()**

**:odd**

**:root**

**:target**

**内容筛选**

**:contains()**

**:empty**

**:has()**

**:parent**

**可见性筛选**

**:hidden**

**:visible**

**属性**

**[name|="value"]**

**[name\*="value"]**

**[name~="value"]**

**[name$="value"]**

**[name="value"]**

**[name!="value"**

**[name^="value"]**

**[name]**

**[name="value"][name2="value2"]**

**子元素筛选**

**:first-child**

**:first-of-type**

**:last-child**

**:last-of-type**

**:nth-child()**

**:nth-last-child()**

**:nth-last-of-type()**

**:nth-of-type()**

**:only-child**

**:only-of-type()**

**表单**

**:button**

**:checkbox**

**:checked**

**:disabled**

**:enabled**

**:focus**

**:file**

**:image**

**:input**

**:password**

**:radio**

**:reset**

**:selected**

**:submit**

**:text**

**事件对象**

#### 属性

**event.currentTarget**

**event.namespace**

**event.data**

**event.pageX**

**event.pageY**

**event.relatedTarget**

**event.result**

**event.target**

**event.type**

**event.which**

**event.timeStamp**

#### 方法

**event.isDefaultPrevented()**

**event.isImmediatePropagationStopped()**

**event.isPropagationStopped()**

**event.preventDefault()**

**event.stopImmediatePropagation()**

**event.stopPropagation()**

**方法**

**属性/CSS**

**属性&特性**

**.attr()**

**.prop()**

**attributes是HTML元素（标签）的属性，而properties 是 DOM 对象的属性。**

###### **attr()和prop()区别**

**具有true和false两个属性的属性，如checked,selected或者disabled使用prop()，其他的使用attr()。**

**对于HTML元素本身就带有的固有属性，在处理时，使用prop方法。**

**对于HTML元素我们自己自定义的DOM属性，在处理时，使用attr方法。**

**.removeAttr()**

**.removeProp()**

**.val()**

**CSS**

**.hasClass()**

**.toggleClass()**

**.addClass()**

**.removeClass()**

**.css()**

**jQuery.cssHooks**

**jQuery.cssNumber**

**jQuery.escapeSelector()**

**尺寸**

**.height()**

**.innerHeight()**

**.outerHeight()**

**.width()**

**.innerWidth()**

**.outerWidth()**

**位置**

**.offset()**

**.offsetParent()**

**.position()**

**.scrollLeft()**

**.scrollTop()**

**数据**

**jQuery.data()**

**.data()**

**jQuery.hasData()**

**jQuery.removeData()**

**.removeData()**

**DOM操作（元素节点&文本节点）**

**拷贝**

**.clone()**

**DOM插入,包裹**

**.wrap()**

**.wrapAll()**

**.wrapInner()**

**DOM插入,**

**内部插入**

**.append()**

**.appendTo()**

**.prepend()**

**.prependTo()**

**.html()**

**.text()**

**外部插入**

**.after()**

**.before()**

**.insertAfter()**

**.insertBefore()**

**DOM 移除**

**.detach()**

**.empty()**

**.remove()**

**.unwrap()**

**DOM 替换**

**.replaceAll()**

**.replaceWith()**

**遍历**

**筛选**

**.eq()**

**.filter()**

**.first()**

**.has()**

**.is()**

**.last()**

**.map()**

**.not()**

**.slice()**

**各种遍历**

**.add()**

**.andSelf()**

**.contents()**

**.each()**

**.end()**

**树遍历**

**.next()**

**.nextAll()**

**.nextUntil()**

**.prev()**

**.prevAll()**

**.prevUntil()**

**.parent()**

**.parents()**

**.parentsUntil()**

**.children()**

**.siblings()**

**.addBack()**

**.closest()**

**从自身开始，向上**

**.find()**

**事件**

**浏览器事件**

**.resize()**

**.scroll()**

**文档加载**

**.ready()**

**.holdReady()**

**事件绑定**

**.on()：绑定事件处理函数**

**.off()：删除绑定的事件处理函数**

**.bind()**

**.unbind()**

**.delegate()**

**.undelegate()**

**.die()**

**.live()**

**.one()**

**.trigger()**

**.triggerHandler()**

**表单事件**

**.blur()：输入域失去焦点时触发**

**.focus()：获得焦点时触发**

**.change()：value值变化时触发**

**.select()：textarea 或文本类型的 input 元素中的文本被选择时触发**

**.submit()：提交表单时触发**

**键盘事件**

**.keydown()：按下字母键时触发**

**.keyup()：松开键盘按键时触发**

**.keypress()**

**鼠标事件**

**.click()**

**.contextmenu()**

**.dblclick()**

**.focusin()**

**.focusout()**

**.hover()**

**.mousedown()**

**.mouseenter()**

**.mouseleave()**

**.mousemove()**

**.mouseout()**

**.mouseover()**

**.mouseup()**

**.toggle()**

**事件对象**

*属性*

**event.type：事件的类型**

**event.pageX：鼠标当前相对于页面的坐标**

**event.pageY：鼠标当前相对于页面的坐标**

**event.which：在鼠标单击时，单击的是鼠标的哪个键**

**event.currentTarget：在事件冒泡过程中的当前DOM元素**

**event.namespace**

**event.data**

**event.relatedTarget**

**event.result**

**event.target**

**event.timeStamp**

*方法*

**event.preventDefault()：阻止默认行为**

**event.stopPropagation()：阻止事件冒泡**

**event.isDefaultPrevented()**

**event.isImmediatePropagationStopped()**

**event.isPropagationStopped()**

**event.stopImmediatePropagation()**

**效果**

**基础**

**.hide()：隐藏元素**

**.show()**

**.toggle()**

**自定义**

**.animate()**

**.clearQueue()**

**.delay()**

**.dequeue()**

**jQuery.dequeue()**

**.finish()**

**jQuery.fx.interval**

**jQuery.fx.off**

**.queue()**

**jQuery.queue()**

**.stop()**

**渐变**

**.fadeIn()**

**.fadeOut()**

**.fadeTo()**

**.fadeToggle()**

**滑动**

**.slideDown()**

**.slideToggle()**

**.slideUp()**

### Ajax

#### 全局Ajax事件处理程序

##### .ajaxComplete()

##### .ajaxError()

##### .ajaxSend()

##### .ajaxStart()

##### .ajaxStop()

##### .ajaxSuccess()

#### 辅助函数

##### jQuery.param()

##### .serialize()

##### .serializeArray()

#### 底层接口

##### jQuery.ajax()

##### jQuery.ajaxSetup()

#### 快捷方法

##### jQuery.get()

##### jQuery.getJSON()：从服务器加载JSON编码（httpget请求）

##### jQuery.getScript()

##### .load()

##### jQuery.post()

# Bootstrap

***是什么***

一种前端框架

***位置***

***Head之间，<link rel="stylesheet" href="//cdn.bootcss.com/bootstrap/3.3.1/css/bootstrap.min.css"/>***

***图标库***

Font-awesome icon（fa）

Glyphicon icon（glyphicon）

***步骤***

把所有的HTML内容放在class为container-fluid的div下。

给图片添加 img-responsive class属性。

居中h元素：把text-center class属性添加给 h2元素。

Button按钮风格：button元素class属性：btn，btn-block，btn-primary，btn-info，btn-danger

栅格系统

Span

# web API

## position

### 属性

#### coords

#### timestamp