

# 21.4.1学习记录

## 一、String对象

前言：在底层字符串是以字符数组的形式保存的

属性：

--length：获取字符串长度

方法：（大部分方法不会改变原字符串）

--charAt：返回指定位置的字符

--charCodeAt：返回指定位置字符的Unicode编码

--fromCharCode：可以根据字符编码去获取字符，用String去调用

--concat：连接字符串

--indexOf：检索字符串，找到返回第一次出现的索引，未找到返回-1；

--参数1：要检索的字符串

--参数2：开始查找的索引

--lastIndexOf：从后向前检索字符串

--slice：截取字符串，前闭后开，参数2可省略，则会截取到末尾；参数2可为负数，将会从后向前计算

--substring：截取字符串，前闭后开，参数2不接受负值，若传递负值则默认为0；若参数2小于参数1，会自动交换参数

--substr：截取字符串，参数1开会索引，参数2截取长度

--split：将一个字符串拆分成数组，参数为一字符串，根据该字符串拆分

--toUpperCase：将字符串转换成大写

--toLowerCase：将字符串转换成小写

## 二、正则表达式

前言：正则表达式用于定义一些字符串的规则

--计算机可以根据正则表达式，来检查一个字符串是否符合规则  
获取字符串中符合规则的内容提取出来

步骤：

--创建正则表达式对象 new RegExp ( “正则表达式” , “匹配模式” )

--参数2： i表示忽略大小写； g表示全局匹配

--字面量创建： /正则表达式/匹配模式

**--字面量方便，但正则对象构建更灵活**

-- 符号 | 是或的关系；中括号【】中的内容也是或的关系【A-z】任意大小写字母

--中括号内写符号 ^ 表示除了 ^ 后的内容，还有别的就返回 true

方法：

--test：可以用来检查一个字符串是否复合正则表达式规则

**字符串与正则：**

--split； 将一个字符串拆分成一个数组， **自动全局匹配**

--方法中可以传递一个正则参数，这样方法会根据正则表达式去拆分。

--search： 可以搜索字符串中是否含有指定内容，搜索到返回第一次找到的索引，没有则返回-1； **不能全局匹配，设置也不行**

--该方法可以接收正则，可根据正则去检索字符串

-match：根据正则，从一个字符串中将符合条件的内容提取出来，并封装到数组中

--replace： 将字符串中的指定内容替换成新的内容

参数1：被替换的内容，可以接受正则

参数2：新的内容

### 量词：

--通过量词可以设置一个内容出现的次数

--量词只对它前面一个内容起作用，多个字符可以用括号（）括起来当作一个

-- 大括号 {n} 表示正好出现n次

--大括号 {x , y} 表示出现x到y次

--大括号 { x, } 表示出现x次以上

--加号 + 表示至少一个

--星号 \* 表示 0个或多个

--问号 ? 表示0个或1个

--小三角 ^表示开头

--\$符号表示结尾，正则同时使用^ 和 \$表示字符串需完全符合

### 元字符：

--点 . 表示任意字符，如果要检查字符串中是否含有点，需要使用转义字符反斜杠\,所以字符串中需要写反斜杠的话，要写两个\\;

注意，使用构造函数时，由于它的参数是一个字符串，而\是转义字符，如果要使用反斜杠则需要使用两个\\来代替

--\w : 任意字母、数字、下划线\_

--\W : 除了字母、数字、下划线\_

--\d: 任意数字

--\D: 除了数字

--\s: 表示空格

--\S: 除了空格

--\b: 单词边界

--\B: 除了单词边界

### 三、DOM (document object model文档对象模型)

前言：学到了宿主对象；

--JS通过DOM来对HTML文档进行操作。

--文档： 整个HTML网页文档

--对象： 将网页的每一个节点都转换成对象

--模型： 使用模型来表示对象之间的关系，方便获取对象

节点：

--Node，是构成我们网页的最基本的组成部分，网页中的每个部分都可以是一个节点，但彼此之间类型不同

--文档节点： 真个HTML文档

--元素节点： HTML文档中的html标签

--属性节点： 元素的属性

--文本节点： HTML标签中的文本内容

nodeName	nodeType	nodeValue
#document	9	null
标签名	1	null
属性名	2	属性值
#text	3	文本内容

--浏览器已提供文档节点对象，这个对象是window属性，可以在页面中直接使用，文档节点代表的是整个网页；

**事件：就是文档或浏览器窗口中发生的一些特定的交互瞬间，即用户和浏览器之间的交互行为**

--我们可以在事件对应的属性中设置一些js代码，这样当事件被触发时，这些代码将会执行

事件写在行内称为结构和行为耦合，不方便维护

--解决，比如为按钮的**对应事件绑定处理函数的形式**来响应事件

重点：文档的加载

--浏览器加载页面时，按照自上而下的顺序加载的，读取到一行就运行一行，如果将JS代码写在页面的上边，在代码执行时，页面还没有加载

## DOM属性：

--innerHTML用于获取元素内部的HTML代码的，对于自结束标签，这个属性没有意义

--innerText 该属性会获取元素内部的文本内容

而读取**元素节点属性**，直接使用元素·属性名；特殊的是class，读取class需要用元素·className；

--childNodes： 返回某个元素节点下的所有节点（包括文本、元素节点），但是在IE8及以下的浏览器中，不会将空白文本当成子节点

--children： 可以获取当前元素的所有子元素

--firstElementChild： 当前元素的第一个子元素，ie8以下不兼容

--firstchild： 当前元素的第一个子节点

--lastElementChild: 最后一个子元素

--lastchild： 当前元素的最后一个子节点

--parentNode 当前节点的父节点

--previousSibling 当前节点的前一个兄弟节点

--previousElementSibling 获取前一个兄弟元素，IE8以下不支持

## DOM方法

1、获取元素节点（document调用）

--getElementById()	id (一个)
--getElementsByTagName ()	标签名 (一组) 返回类数组对象
--getElementsByName ()	name (一组)
--onclick :	单击事件
--onload :	在页面或图像加载完后触发

## 复习

### 1、CSS引入

内部样式表：所谓内部样式表，就是在html内部写样式，但是单独写到style标签内部，控制当前整个页面的元素样式设置

行内样式表:在元素标签内部的style属性中设定css样式

外部样式表：单独写到css文件中，之后吧css文件引入到html中  
两步：

1.新建.css的样式文件

2.在html页面中，link标签引入 <link >

### 2、字体属性

-- font: font-style font-weight font-size/line-height font-family;

--必须保留size和family