Java基础

Java高级、Java实训

数据库 – MySQL Oracle

B/S

Web开发基础（html css javascript jquery）

JavaWeb开发（servlet jsp）

Linux操作系统

Java框架（spring springmvc mybatis）

Web开发基础 –HTML

1. HTML基础语法
   1. HTML概述
      1. web的工作原理

Client客户端/Server服务器架构

**Browser浏览器/Server架构**

B S DB

浏览器🡪服务器—>数据库🡪服务器—>浏览器

* + 1. Web开发的相关技术

浏览器端：

--HTML 页面的内容

--CSS 页面的样式

--JavaScirpt 页面的动态交互

--JQuery JS的框架

服务器端：

Servlet

JSP

**SS**M框架

* + 1. web浏览器

--IE浏览器

--Firefox 火狐浏览器

--Chrome 谷歌浏览器

--Safari 苹果

1.1.4 什么是HTML

HTML（Hyper Text Markup Language）

超文本标记语言。

HTML编写的文件必须以.html结尾

由浏览器解析执行

* 1. HTML基本语法
     1. 标记

--HTML用于描述功能的符号称为标记，比如<p>、<h1>等

--标记在使用时必须使用尖括号括起来

--有封闭类型标记，也有非封闭类型的标记

（1）封闭类型标记

也叫双标记，必须成对出现，

<开始标记>内容</结束标记>

注意：如果一个应该封闭的标记没有封闭，就会出现意想不到的错误。

（2）非封闭类型的标记

也叫单标记：<标记/>

<br/> 单标记，换行的作用

* + 1. 元素和属性

--元素：每一对尖括号包围的部分

三部分构成：开始标记+内容+结束标记

<h1>hello</h1> 称为h1元素

<p>你好</p> 称为p元素

--属性：用来修饰元素的；

属性写在开始标记中；

可以写多个属性，多个属性用空格隔开；

格式是 xx属性=”” 方式

--通用属性：绝大多数都支持

--id

--class

--style 样式，是CSS内联样式写法

1.2.3 注释

语法：<!—注释的内容 -->

* 1. HTML文档的结构
     1. 文档结构

分为三部分：

1. 文档类型的声明—了解
2. **文件头 <head></head>**
3. **文件体 <body></body>**

**说明：**HTML文档要以<html>开始，以</html>结尾

* + 1. 文件头-head

--title标记：设置网页的标题的

<title></title>

--meta标记：1设置页面显示中文的编码

2 设置页面自刷新

--script标记：写JavaScript代码的

--style标记：写CSS代码的

--link标记：1 引入外部的CSS样式

2 设置标题中的图标

2. body体中的标记

2.1 body体中的标记的分类

--特殊字符的标记

空格&nbsp; 表示一个空格

小于&lt; 表示一个小于号

大于&gt; 表示一个大于号

--标题标记

是<hn>表示，n是1-6数字组成

<h1>到<h6>是标题1到标题6

能够将内容加粗，大小**h1最**大,h6最小

--段落标记

用<p>表示，表示一个独立的段落

--换行标记

<br/>一个换行

--字体标记

设置元素的字体、字号和颜色

<font face=”” size=”” color=””>

</font>

--加粗、斜体、下划线标记

<b></b><strong></strong>加粗标记

<i></i> <em></em>斜体标记

<u></u>下划线标记

--居中标记

<center></center>

--分割线标记

在网页中画一条水平直线

<hr/>

--分区标记

常用于页面的布局

1. **块**分区标记：<div><div>
2. **行内**分区标记：<span></span>

2.2标记的另一个分类：

1. 块标记：会另起一行的标记

h1…h6 p center div…

1. 行内标记：不会另起一行的标记

font em i strong u span…

3. 超链接和图片

3.1.1 超链接

标记：<a>内容</a>

属性：href=””超链接的跳转地址

target=”” 超链接的跳转方式

\_blank 新页面打开

\_self 默认，本页面打开

3.1.2 图片

标记：<img />

属性：src=”” 显示的图片的路径名称

width=”” 设置图片的宽度

height=”” 设置图片的高度

title=”” 设置图片的标题

border=”” 设置图片的边框

3.1.3 图片热点分割技术

标记：<map>

<area></area>

</map>

属性：要写在<area>标记中

shape=”” 区域的形状 rect矩形

coords=”” 坐标

3.1.4 超链接的特殊使用1—

返回页面顶部

<a href=”**#top**”>xxxxx</a>

3.1.5 超链接的特殊使用2 –

发送邮件

<a href=”mailto:23434@qq.com”>

xxxxx</a>

4. 列表

列表是将具有相同特征或者具有先后顺序的几行文字进行对齐排列。

分类：

1. 无序列表

标记：无序<ul></ul>

列表<li></li>

1. 有序列表

标记：有序<ol></ol>

列表<li></li>

5、表格

标记：<table></table>表格的开始和结束

<tr></tr>表格的行

<td></td>表格的列

<caption></caption>表格的标题

<marquee></marquee>将内容飘

动的效果

属性：border=”” 设置表格的边框

cellspacing=””设置表格行列之间的空隙

width=”” 设置表格的宽度

height=”” 设置表格的高度

align=”left/center/right”

水平对齐方式

(可用于table tr和td中)

valign=”top/middle/bottom”

垂直对齐方式

(只能用于tr或者td标记中)

不规则表格的处理

跨行：rowspan=”数字”

跨列：colspan=”数字”

6. 表单

6.1 表单的作用

用于显示、收集信息，并提交信息到服务器端，服务器接收到信息后在保存到数据库端。

标记：<form></form>

表单的开始和结束

6.2 表单项的分类

（1）input标记系列

文本框 <input type=”text” />

密码框 <input type=”password”/>

单选<input type=”radio”/>

多选按钮<input type=”checkbox”/>

提交按钮<input type=”submit” />

重置按钮<input type=”reset”/>

普通按钮<input type=”button”/>

--用于执行客户端的脚本，结合JS来使用

隐藏域<input type=”hidden”/>

--在表单中包含不希望用户

看到的信息

文本选择框：上传文件

<input type=”file”/>

1. 非input标记系列

A <label>标记

--语法：<label>文本</label>

--作用:将文本与控件绑定在一起，单击文本，效果如同单击控件一样

B 选项框—下拉框

标记:<select></select>

选项框的开始和结束

<option></option>

选项框中的选项

C 多行文本框

标记:<textarea></textarea>

属性：cols指定多行文本框的列数

rows指定行数

D 控件分组

标记<fieldset></fieldset>

为控件分组

<legend></legend>

给分组指定一个标题

JS Java

If(){

}else{

}

Web开发基础---CSS

1. CSS概述

1.1概念

CSS（Cascading Style Sheets）:层叠样式表，又叫级联样式表，简称样式表

1.2 功能

美化页面

1.3 CSS的使用

1.3.1 CSS样式的使用方式

--内联样式

写在**开始标记**中，针对单独的一个标记元素起作用

--内部样式

写在HTML页面中的<head>

标记里

--外部样式

写在单独定义的一个以.css结尾的文件中，该文件要引入进来

1.3.2 CSS样式的优先级

--浏览器默认设置样式

--外部样式或者内部样式

就近原则

--内联样式

优先级由上往下，由低到高的

* 1. 选择器

针对于外部样式或者内部样式

（1）元素选择器/标记选择器

（2）类class选择器

常规用法：

先在开始标记中定义class=””属性

选择器通过**.className{}**的方式调用

高级用法：

将标记选择器和类选择器结合使用

语法：**标记选择器.className{ }**

（3）id选择器

--根据id的值来查找

--先在开始标记中定义id的值，一般id的值是要保证唯一的

--选择器通过#idName{}的方式来获取

说明：优先级id>class>标记

1. 选择器分组

一般将一些相同的样式作用于多个标记元素中

语法：将多个选择器写在一起，用逗号分隔开

1. 派生选择器

--通过依据标记元素在其位置的上下文关系来定义样式

--语法：选择器使用空格分隔开

例如: h3 div span p{ xxxx }

<h3>

<div>

<span>

<p id=””></p>

</span>

</div>

</h3>

（6）伪类选择器

伪类用于向某些标记添加特殊的效果

语法：使用冒号作为分隔符，左边是选择器，右边是伪类

选择器:link 未访问

选择器:visited 访问后

选择器:hover 鼠标指向时

选择器:active 鼠标按下时

CSS

1. 尺寸和边框

2.1 尺寸/大小（宽度 高度）

width 宽度

height 高度

overflow:当内容溢出时如何处理

--visible 默认，内容超出会溢出

--hidden 超出部分隐藏

--scroll 显示滚动条，不管内容会不会溢出，都有有滚动条

--auto 自动适应，如果内容未溢出，不显示滚动条，如果内容溢出才会显示滚动条

2.2 边框

--简单写法：

border:width style color;

线宽 样式 颜色

--单边写法1：

border-left/right/top/bottom:

width style color;

--单边写法2

border-left/right/top/bottom-width 只设置边框的线宽

border-left/right/top/bottom-style

只设置边框的类型

border-left/right/top/bottom-color

只设置边框的颜色

1. 框模型—盒子模型

框模型/盒子模型包含：

外边距+边框+内边距

1. 内边距:内容和边框之间的距离

单边写法：

padding-left/right/top/bottom:value值

简单写法：

padding:v1 v2 v3 v4

top right bottom left

1. 外边距:与下一个控件之间的距离

单边写法：

margin-left/right/top/bottom:

value值

简单写法：

margin:v1 v2 v3 v4

top right bottom left

1. 背景

4.1 背景颜色

background-color:red/#ff0000;

4.2 背景图片

background-image:url(“图片名称”);

4.2.1 背景图片的重复问题

--默认情况下，背景图片在水平和垂直方向上重复出现，创建一种称为“墙纸”的效果。

--可以使用background-repeat属性控制背景图片的平铺效果

1）repeat默认,在垂直和水平方向重复

2）repeat-x 仅在水平方向上重复

3）repeat-y 仅在垂直方向上重复

4）no-repeat 仅显示一次

4.2.2 背景定位

background-position属性改变图片在背景中的位置

1. x y 第一个是水平位置

第二个是垂直位置

1. left 左边显示

center 居中显示

right 右边显示

top 顶部显示

bottom 底部显示

4.2.3 背景图片的固定

background-attachment

--默认情况下，背景图片会随着页面的滚动而移动

--可以通过background-attachment

来设置

1)默认值scroll，会随着滚动而移动

2)fixed，固定，背景图片不会移动

1. 文本

5.1 控制字体

font-family:字体; 设置字体的

font-size:数字; 设置字体大小

font-weight:nomal/bold;正常/粗体

font-style:italic 斜体

5.2 控制文本格式

--文本颜色 color:颜色

--文本排列（水平对齐方式）

text-align=”left/center/right”

说明：只能用于块标记

--文本修饰（下划线）

text-decoration:none/underline

--行高

line-height:数字距离

--首行文本缩进

text-indent:数字距离

1. 定位

6.1 定位的概念

定位：定义元素内容相对于其正常位置应该出现的位置。

6.2 分类

--普通流定位（默认）

--浮动定位

--相对定位

--绝对定位

6.3 普通流定位（默认）

块标记

页面中的块级标记从上到下

一个接一个的排列；

每一个块级标记都会出现在

一个新行中。

行内标记

将在一行中从左往右水平排

列；

不需要从新行开始。

6.4、 浮动定位

--浮动定位是指：

1）将元素排除在普通流定位之外

2）将元素浮动以后，可以向左边

或者右边移动，直到碰到边框或者另一个浮动的元素为止

3）浮动定位一般来实现特定的定

位效果

--浮动定位的属性：

float:none/left/right;

在CSS中，任何元素都可以浮动

--clear属性，用于清除浮动带来的

影响。

clear:none/left/right/both;

（定义了元素的那边上不允许出现

浮动元素）

1. 显示方式
2. 设置元素的显示隐藏

display:none;

让元素不再显示，不占用文档的空间

1. 设置块标记为行内标记特性

display:inline;

1. 设置行内标记为块标记特性

display:block;

1. 光标

使用cursor属性来设置光标的形状

可使用的值：

--default

--pointer

--crosshair

--text

--wait

--help

9、列表样式

list-style-type属性用于控制列表中列表项的样式

1. 无序列表

--none 无标记

--disc 实心圆，默认

--circle 空心圆

--square 实心方块

1. 有序列表

--none 无标记

--decimal 数字(1,2,3..)默认的

--lower-roman 小写罗马数字

--upper-roman 大写罗马数字

list-style-image属性用图片来替换列表项的符号

取值：url()

1. 关于元素隐藏
2. display:none/block/inline

隐藏 显示

让元素不再显示，不占用文档的空间

1. visibility:hidden/visible

隐藏 显示

占位隐藏

1. 关于列表的嵌套使用

在一个列表中可以嵌套另一个列

表来使用

1. 相对定位、绝对定位和固定定位
2. position属性：更改定位模式为相对、绝对或者固定定位

position:relative/absolute/fixed

（1）相对定位

①设置定位模式relative

②偏移量

left/right水平位置的偏移

top/bottom垂直偏移

③相对定位元素原本所占的空间保留

④相对定位是相对于它原来位置进行偏移的距离

（2）绝对定位

对元素进行绝对定位的设置，会让该元素从普通定位流中移除，不占用空间（理解为浮起来了效果）

1. 设置定位模式为

absolute

②偏移量

left/right水平位置的偏移

top/bottom垂直偏移

1. 堆叠顺序

(1)一旦修改了元素的定位方式，则元素可能会发生堆叠。

(2)可以使用z-index属性来控制元素出现堆叠的顺序。

(3)z-index属性

-- 值为数值，数值越大表示堆叠顺序更高，即离用户更近

-- 数值可以设置为负值，表示离用户更远

3）固定定位

设置固定定位：

1. 设置position的值为fixed
2. 也是通过left top right bottom来定义元素的位置

固定定位的特点：

1. 固定定位会将元素从普通流定位中完全移除，不占用页面的空间。
2. 当页面向下滚动时,不会跟着一起移动

HTML CSS

JavaScript

1. 什么是JavaScript

JavaScript是一种基于对象和事件驱动的解释性脚本语言，具有与C语言和Java类似的语法，简称JS。

--JS是一种网页编程技术，用来向HTML页面添加交互行为；

--由浏览器解析执行

--JS是由网景公司推出的

2、JavaScript的特点

(1)可以使用任何的文本编辑工具编

写；

(2)JS由浏览器内置的JavaScript引擎

执行代码，JS代码不用编译，可以

直接执行；

(3)JS语言内置大量现成的对象可以

直接使用

(4)适用的地方：

--浏览器端的数据计算

--浏览器端表单的合法验证

--浏览器端事件的触发

--浏览器端显示效果的制作

--服务器的异步数据提交

3、使用JavaScirpt

（1）事件定义方式

在定义事件时直接写入JS代码

（2）嵌入式方式

写在页面<head>标记中的

<script>标记里

（3）文件调用方式

①将JS代码写在xx.js结尾的文

件中

②在<head>标记中使用<script>标记引入xx.js文件

4、JS代码编写的规范和注意事项

（1）注释 // 单行注释

/\* \*/ 多行注释

（2）JS是大小写敏感的，区分大小写

（3）关于JS代码的调错

5、变量

(1)声明变量使用关键字var

例如：var x;

var y;

(2)变量初始化

使用=进行赋值初始化

例如：var x=100;

var y=”张三”;

--变量在声明时不需要指定具体的

数据类型，是以赋值的内容为准。

--没有初始化的变量自动取值

为undefined

(3)变量名的命名规则：

变量名由字母、数字、\_、$组成，数字不能开头

6、数据类型

（1）基本类型

number:数字（整数、小数）

科学计数写法:

4.3e23=4.3\*10^23

string：字符串类型 ”” ‘’

boolean: 布尔类型 true false

使用true、fasle表示

也是用1或0表示

在运算中true=1，false=0

（2）特殊类型

null: 空 undefined未定义

--null值在程序中表示”无值”或者”无对象”,可以给一个变量赋值null来清楚变量的内容。

var a=10;

~~a=null;~~

a=”张三”

--undefined是在声明一个变量以后未初始化(赋值)，该变量是undefined，表示不存在

var a;

alert(a); undefined

（3）复杂类型

Array: 数组

Object:对象（Date…）

7、数据类型的隐式转换

JS属于松散类型的程序语言

--变量在声明时不需要指定数据类型

--变量由赋值操作后确定数据类型

不同类型数据在计算过程中会自动进行转换

1. 数字+字符串：数字会转为字符串
2. 数字+布尔值：true转为1，false转为0
3. 字符串+布尔值：布尔值转换为字符串的true或false
4. 布尔值+布尔值:布尔值会转换为数值1或0

8、数据类型的显示转换

（1）toString

语法：value.toString()

将vlaue的类型转换为字符串类型

所有的数据类型均可以转换为

string类型

（2）parseInt

语法: parseInt(value)

强制转换为整数类型；

如果不能转换，会返回NaN

(Not a Number)

（3）parseFloat

强制转换为浮点数

如果不能转换，会返回NaN

9、查看数据类型

typeof：查看数据的类型

语法：typeof value

判断value数据类型是什么

isNaN：判断是否为数值

isNaN(value)：如果value是数字返回结果为false，如果value不是数字类型，返回true

10、运算符

（1）算数运算符

+ - \* / % ++ --

（2）关系运算符

> >= < <= == !=

关系运算符的运算结果为

boolean类型

=== 全等(类型相同，数值相同)

!==不全等(类型和数值都不等)

（3）逻辑运算符

&&(逻辑与) ||(逻辑或)

!(逻辑非)

条件1 && 条件2

true false

false true

true true

false false

!条件 !true-->false

!false-->true

（4）条件运算符/三目运算符

语法:条件?代码1:代码2

如果条件为true，执行代码1

如果为false，执行代码2

11、分支结构

(1)一种情况的分支结构

if(条件){

语句

}

(2)两种情况的分支结构

if(条件){

语句1

}else{

语句2

}

(3)多种情况的分支结构

if(){ //1次

}else if(){ //0-n次

}else if(){

}else{ //0-1次

}

分支结构的第二种语法：

switch…case语句结构

switch(表达式){

case 值1:

语句1;

break;

case 值2:

语句2;

break;

…

default:

其他语句;

}

12、循环结构

(1)for循环

for(起始条件1;结束条件2;步进语句3){

循环体的内容4

}

1 2 4 3 2 4 3… 2 4 3… 2条件不成立

(2)while循环

语法：

循环的起始条件;

while(结束条件){

循环体的内容;

步进语句;

}

(3)do…while循环

循环的起始条件;

do{

循环体;

步进语句;

}while(结束条件);

13、JS中的对象

类（属性、方法） 对象

JFrame f; f.setVisible();

函数（属性、方法）

Math.floor();

--对象是JavaScirpt中最重要的内容

--JavaScript中包含了多种对象：

(1)内置对象

(2)自定义对象

(3)浏览器对象

(4)DOM对象

--对象的使用

(1)对象由属性和方法构成的

(2)调用：对象名.属性

对象名.方法()

13.1 内置对象

(1)简单数据对象

String Number Boolean

(2)组合对象

Array Math Date

(3)高级对象

Function(自定义) RegExp(正则表达式)

13.2 简单数据对象

（1）String对象

字符串对象

String对象中的属性：.length

（2）String对象中的方法

1）大小写转换

x.toLowerCase() 转小写

x.toUpperCase() 转大写

2）获取指定的字符

x.charAt(index)

返回指定位置(index)的字符

3）查询指定的字符串位置

x.indexOf(findStr)

findStr要查找的字符串

该方法查找指定字符串首次出现的位置，如果查找不到，返回-1。

x.indexOf(findStr,[index])

[index]开始查找的位置索引，可以忽略，如果忽略不写，默认从0开始查找。

x.lastIndexOf(findStr,[index])

4）获取子字符串

x.substring(start,[end])

--start：开始位置

--end：结束位置，可以忽略，如果忽略默认到末尾结束。

5）替换子字符串

x.replace(findstr,tostr)

--findstr表示要查找的字符串

--tostr要替换的字符串

--方法返回替换后的新字符串

6）拆分子字符串

x.split(bystr)

--bystr表示分割用的字符串

--方法返回分割后的字符串数组

7）String和正则表达式regexp

x.replace(regexp,tostr)

根据regexp来替换tostr的内容

x.match(regexp)

返回根据regexp匹配的字符串数组

x.search(regexp)

返回根据regexp匹配的首字符出现的索引(下标)位置

（3）Array对象 --数组对象

1）创建数组对象—动态创建

var arr1=new Array(10);

var arr2=new Array();

--往数组中存储内容：

数组名[下标]=内容

--获取数组大小的属性：

xx.length

2）静态初始化数组

var arr1=[100,200,300];

var arr2=new Array(100,200,300);

3）数组常用的方法

(1)数组转换为字符串的方法

--x.join(bystr)

--x.toString()

说明：

--返回连接后的字符串

--bystr作为连接数组中元素的字符串，可以省略

--toString方法返回就是默认用逗号连接的字符串

(2)连接数组的方法

--x.concat(value1,value2,…)

--value是作为数组元素连接到数组的末尾

--方法返回连接以后的新数组

(3)获取子数组

--x.slice(start,[end])

--start：开始位置的索引

--end:可忽略,结束位置的索引

(注意:end的前一位)

(4)数组的倒转

--x.reverse()

(5)数组的排序

--x.sort() 默认升序排列

--x.sort(function)

function可选项，用来确定元素顺序的函数名称

（4）Math对象

常用于执行数学中的操作

1. 属性：数学常数

Math.PI 圆周率

1. 方法：

--[abs(x)](https://www.w3school.com.cn/jsref/jsref_abs.asp) 返回 x 的绝对值

--ceil(x)返回 x，向上舍入为最接近的整数

--floor(x) 返回 x，向下舍入为最接近的整数(取整)

--max(v1,v2,v3….) 找最大值

min(v1,v2,v3….) 找最小值

--pow(x,y) 返回 x 的 y 次幂值

--random() 返回 0 到 1 之间的随机数

--sqrt(x) 返回 x 的平方根

--round(x) 将 x 舍入为最接近的整数

（5）Number对象

--Number对象是原始数据的包装对象

--创建Number对象

var myNum=3.14; number

var myNum=new Number(value)

var myNum=Number(value);

--Number对象中的方法

toString()方法：数值转换为字符串

toFixed(num)方法：数值转换为字符串，并保留小数点后num的位数

（6）正则表达式对象：RegExp

1）创建正则表达式对象

var rgExp=/正则表达式/[flag];

var rgExp=

new RegExp(“正则表达式”,[flag]);

--正则表达式

--flag是可以忽略的，有两个值

g表示全局匹配

i 表示匹配中忽略大小写

2）RegExp对象中的方法

x.test(string)

x表示正则表达式对象

string表示要匹配的字符串

该方法返回true或者false

（7）Date日期对象

1）创建Date对象

--var d1=new Date();

当前系统时间

--var d2=

new Date(“2013/3/20 11:12”);

指定创建的时间

2）Date对象中的方法

(1)getDate()返回月中的第几天

(从1开始)

(2)getDay()返回星期几(1-7)

(3)getFullYear()返回年

(4)getHours() 返回小时(0-23)

(5)getMonth() 返回月份(0-11)

(6)getSeconds() 返回秒数

(7)getMinutes() 返回分钟

(8)toString() 将Date类型转换为字符串类型

(9)toLocaleString() 将默认日期格式修改为xxx/xx/xx格式输出

(10)setXXX()

（8）Function对象

--函数是一个可以重复执行的

代码块

--Function对象可以表示开发者定义的任何函数

1）函数的定义

--关键字是function开始

--函数名定义规则和变量名一直，大小写敏感

--可以使用变量作为函数的参数，多个参数用逗号隔开

--如果函数需要有返回值，使用关键字return

function 函数名([参1,参2…]){

函数体的内容

[return xxx;]

}

2）arguments对象

--是一种特殊的对象，在函数代码中，表示函数的参数数组

--在函数代码中，可以使用arguments访问所有的参数

arguments.length求参数的个数

arguments[i] 第i个参数，从0开始

3）使用Function对象创建函数

--使用Function对象直接创建创建

var funcName=new Function

(arg1,arg2,…,functBody);

--funcName表示函数名

--arg1,arg2,…表示参数变量

--functBody函数体

4）**匿名方式创建函数**

正常：function f1(xx){ xxx }

调用f1()

匿名：

var funcName=

function([参数]){ 函数体 }

调用funcName()

（9）eval函数

--eval函数用于计算某个字符串，

来得到结果；也可以用于执行其

中的JS代码。

14、JS中的对象模型

(1)BOM：浏览器对象模型

Browser Object Model,用来访问和操纵浏览器窗口

具体通过window对象来实现浏览器窗口的操纵。

(2)DOM：文档对象模型

Document Object Model,用来操纵文档，主要定义了访问和操纵HTML文档的方法，具体功过document对象进行操作。

14.1 window对象

--window对象表示浏览器窗口

--window对象中的常用方法：

(1)window.alert()

只带确定的对话框

window.confirm()

带确定和取消的对话框

window.prompt()

带确定、取消和输入框的对话框

(2)close() 关闭当前浏览器窗口

(3)open() 打开新的浏览器窗口

open(url,name,config)

--url 打开新窗口的地址

--name 打开窗口的名字

“\_blank”

“\_self”

名字

--config 窗口的配置参数

(4)moveTo(x,y) 移动窗口

(5)resizeTo(width,height)

调整窗口的大小

(6)focus() 获得焦点

14.2 定时器

(1)周期性定时器

以一定的时间间隔执行代码，循环

往复

window.setInterval(exp,time)

--exp 执行代码/函数

--time 时间间隔，单位毫秒

window.clearInterval(tID);

--tID 定时器名

(2)一次性定时器

在一个设定的时间间隔后执行代

码，而不是在函数被调用后立刻执

行。

window.setTimeout(exp,time)

time时间后一次性执行exp函数

--exp执行的函数

--time 时间，单位毫秒

window.clearTimeout(tID);

停止一次性定时器

14.2 document对象

是DOM对象模型中的最重要的一

个对象，提供了操作HTML文档的

功能和方法。

1）操作元素节点的内容

(1)innerText

以纯文本的方式获取

(2)innerHTML

以HTML文本方式获取

可以获取也可以设置元素中内容

2）操作元素节点的属性

(1)getAttribute() 获取属性

(2)setAttribute() 设置属性

(3)removeAttribute() 删除属性

3）操作元素节点的样式 style

元素节点.style.样式

xx.style.color=””;

xx.style.fontSize=””;

xx.style.textDecoration=””;

说明：CSS样式如一个单词属性，JS也是一个单词属性；

如果CSS中是-隔开的两个单词，JS是默认-去掉，第二个单词首字母大写

14.3、document 对象查询节点

--如果要操作HTML元素，必须首

先要找到该元素

--查询节点的方法

(1)通过id查询

document.getElementById();

(2)通过层次(节点关系)查询

parentNode 父节点

childNodes 孩子节点

firstChild 第一个孩子节点

lastChild 最后一个孩子节点

previousSibling

前一个同级别节点

nextSibling

后一个同级别节点

1. 通过标签名称查询

getElementsByTagName()

根据指定的标签名返回所有的元素节点

(4)通过name属性查询

getElementsByName()

通过标签中的name属性的值

进行查询

14.4 document对象中的增加节点

(1)创建新节点

document.createElement(elName)

--elName：要创建的元素节点名称

--返回创建的新节点

(2)添加新节点

xx.appendChild(newNode)

将新节点newNode添加xx里面来，默认在xx末尾添加。

xx.insertBefore(newNode,refNode)

--refNode：参考节点

newNode会在refNode之前添加到xx里面。

14.5 document中的删除节点

(1) xx.removeChild(childNode)

--删除childNode这个节点

--childNode必须是xx的子节点

(2)cNd.parentNode.removeChild(cNd)

--cNd是要删除的节点

以上是DOM的增删改查操作

14.6、关于Select和Option对象