Web前端基础

6月22日-6月24日

前端

互联网的发展

中央控制式网络弱点：
如果中央控制系统受到攻击，整个网络就会瘫痪
包交换：
在网络的不同站点之间数据像接力赛一样传递

互联网信息浏览

table布局

table布局：容易上手，简单快速，兼容不同浏览器但加载缓慢

div+css布局：结构、样式、行为分离，修改简单，加载快，节约空间和流量，提高搜索效率

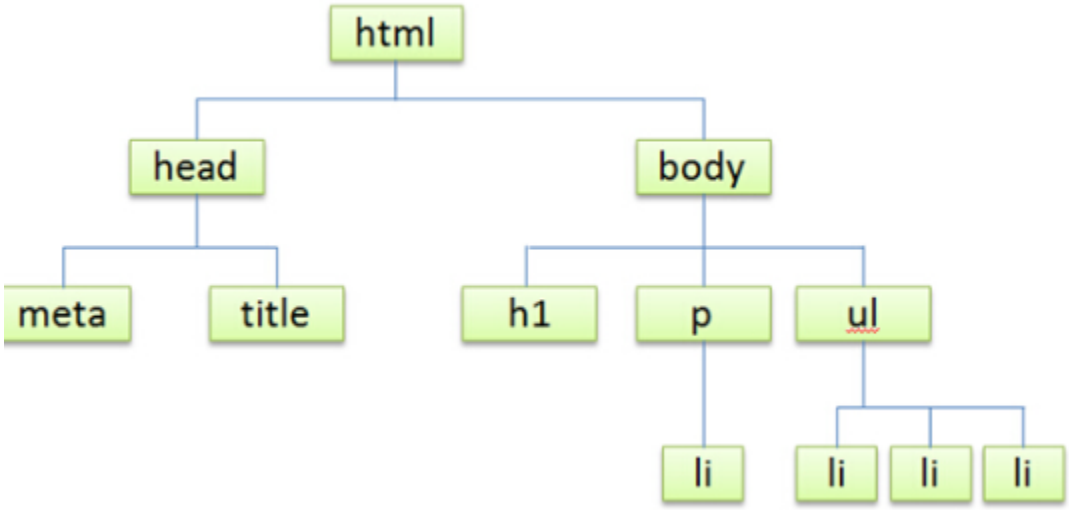
HTML5和CSS3

HTML5提供了网页存储的API，方便离线使用
可以跨平台开发，减轻开发负担

语言简介

标签：封闭的和自封闭的
标签与标签之间可以嵌套，但要有先后顺序

HTML基本结构



标签的常用属性描述

属性	描述	常用于标签
id	区别标签的身份标识，id 值唯一。	div、a、p 等大多数标签
class	定义标签的类名，一个类代表一种样式。	div、a、p 等大多数标签
name	定义标签的名称。	div、a、p 等大多数标签
title	定义标签的显示名称。	a 等锚点标签
href	定义标签的跳转链接。	a 等锚点标签
src	定义标签的源地址链接。	img、script 等标签
value	定义标签的值。	input、textarea 等表单标签

标签

```
<body>
    <a href="http://www.jisuanke.com" title="点击进入计蒜客" target="_blank">计蒜客</a>
    <img src="http://res.jisuanke.com/img/fe/6.1.jpg">
    <h1>1级标题</h1>
    <p><em>汪</em></strong>汪<strong>汪</p>
    <hr>
</body>
```

表单

get方法不安全，数据被放在请求的url中，传输数据量小
post可以传输大量数据

submit为表单的上传按钮，action规定上传的服务器， reset为重置按钮，tyoe说明按钮功能

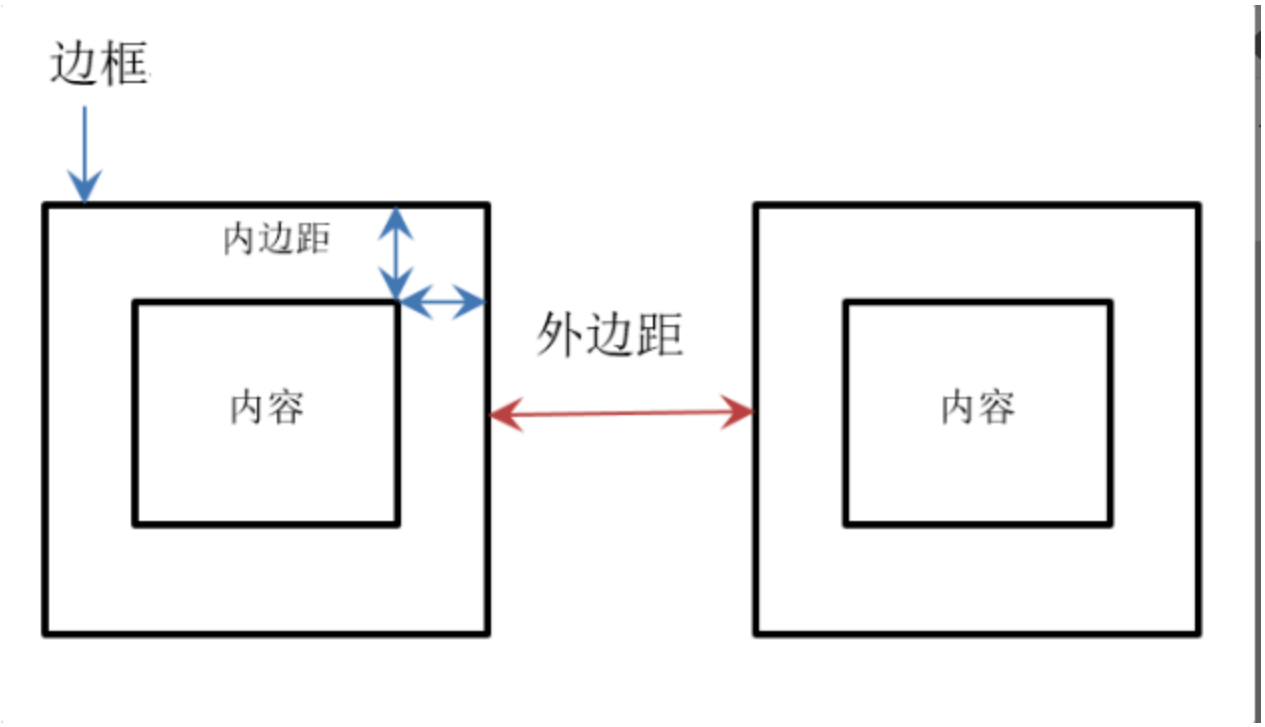
CSS

选择符 {属性: 值;}
网页显示时采用的顺序为内联式>嵌入式>外部式
ID选择器: .ID名 只能一个
类选择器: #类名 可以多个
>符号作用于元素的第一代后代，空格作用于元素的所有后代
a:hover鼠标悬浮
*通用选择器
逗号，都使用同一样式‘

元素类别

块状元素: `<div> <p> <ol> <li> <table> <blockquote> <form>`
内联元素： `<a> <span> <br> <i> <em> <strong> <label> <q>`
内联块状元素： `<img>` `<input>`

盒式模型



顺时针写法：上右下左
1个：上右下左相同
2个：上下，左右
3个：上 左右 下
默认布局模式：流动模型

JavaScript

helloworld

```
console.log('Hello World');
```

变量

字母、下划线或$开头，后面还可以有数字
区分大小写
不可有关键字、保留字
var根据值自动区别是什么类型的变量

+可以拼接字符串或相加数字

各运算符之间的优先级（高到低）：`+ - * /`等算术操作符 → `> = <`等比较操作符 → `&& || !`等逻辑操作符 → `=`赋值符号。

length属性总是比数组最大的数字键值大1，而数字键值不需要连续，length属性可以被赋值

对象

带有属性和方法的数据类型
由若干个键值对构成
键名如果不符合标识符的条件需要加上引号，其他加不加都可以
对象生成方法

```
var obj1 = {};
var obj2 = new Object();
var obj3 = Object.create(null);
```

指向同一对象的不同变量名修改任一属性都会影响到其他变量

时间方法

```
var now = new Date();
now.setTime(now.getTime() + 60 * 60 * 1000);
var myyear = now.getFullYear(); // 四位数年份，如 2015
var mymonth = now.getMonth(); // 月份 [0，11]，要加 1，如 7 (8 月)
var mydate = now.getDate(); // 月中日期，如 1 (1 号)
var myhours = now.getHours(); // 小时，24 小时制
var myminutes = now.getMinutes(); // 分钟
var myseconds = now.getSeconds();
var mynum = now.getDay();
```

String对象

```
var mystr = "I like JavaScript!";
var mylen = mystr.length;
var myup = mystr.toUpperCase();
var mylow = mystr.toLowerCase();
console.log(mystr.charAt(2));
console.log(mystr.indexOf("I"));
console.log(mystr.split("."));
console.log(mystr.split(".",2));
console.log(mystr.substring(7));
console.log(mystr.substring(2,6));
console.log(mystr.substr(7));
console.log(mystr.substr(2, 4));
```

Math对象

```
console.log(Math.PI);
console.log(Math.abs(-15));
onsole.log(Math.ceil(0.8));
console.log(Math.floor(0.8));
console.log(Math.round(0.8));
console.log(Math.random() * 10);
console.log(Math.round(Math.random() * 10));
console.log(Math.min(0.8, 6.3, 5, 3.5, -5.1, -5.9));
console.log(Math.max(0.8, 6.3, 5, 3.5, -5.1, -5.9));
```

数组

```
var mya1 = new Array("hello!");
var mya2 = new Array("I", "love");
var mya3 = new Array("JavaScript", "!");
console.log(mya1.concat(mya2, mya3));
console.log(mya1);
var myarr = new Array(3);
myarr[0] = "www";
myarr[1] = "jisuanke";
myarr[2] = "com";
console.log(myarr.join("."));
var myarr = new Array(3);
myarr[0] = "www";
myarr[1] = "jisuanke";
myarr[2] = "com";
console.log(myarr.reverse());
console.log(myarr);
r myarr = new Array(3);
myarr[0] = "www";
myarr[1] = "jisuanke";
myarr[2] = "com";
console.log(myarr.slice(0,2));
console.log(myarr1.sort());
```

函数

```
function add(a, b){
    return this + (a + b);
}
console.log( add.apply(null, [3, 6]));
```

获取元素

通过ID：`document.getElementById`
通过标签名：`document。getElementsByTagName`
通过名称：`document.getElementsByName`
鼠标单击： .onclick
鼠标按下：onmousedown
鼠标抬起: onmouseup
鼠标经过：onmouseover

鼠标移出：onmouseout

计时器

计时器有四种方法：

setTimeout()： 在指定的毫秒数后调用函数或计算表达式。

clearTimeout()： 取消由 setTimeout() 方法设置的 timeout。

setInterval()： 按照指定的周期（以毫秒计）来调用函数或计算表达式。

clearInterval()： 取消由 setInterval() 设置的 timeout。

DOM

方法	说明
createElement('tagName')	创建节点。
createTextNode('text');	创建文本节点。
appendChild(o)	在父节点末尾附加子节点，其中 o 为节点对象。
createDocumentFragment()	创建文档片断。
removeChild(oP)	删除节点。
replaceChild(newOp, targetOp)	替换节点。
insertBefore(newOp, targetOp)	已有的子节点前插入一个新的子节点。
insertAfter(newOp, targetOp)	已有的子节点之后插入一个新的子节点。
get/setAttribute('key', 'value')	设置或得到属性节点。
cloneNode(true/false)	复制节点。

页面尺寸

属性	说明
dientWidth / dientHeight	窗口的当前宽度 / 高度
scrollWidth / scrollHeight	文档内容的宽度 / 高度
offsetWidth / offsetHeight	文档内容的宽度 / 高度

在坐标位置里，分为滚轴距离和偏移距离，顺带着我们也说一下事件触发时鼠标指针相对于窗口的。

属性	说明
scrollLeft / scrollTop	滚轴的水平偏移距离 / 垂直偏移距离
offsetLeft / offsetTop	对象与页面左边距 / 对象与页面上边距
event.dientX / event.dientY	事件触发时鼠标指针对于窗口的水平坐标（左边距） / 垂直坐标（上边距）