千锋教育
QIAN FENG EDUCATION

做真实的自己，用良心做教育

BOM和DOM上

H5教学部

目录

做真实的自己，用良心做教育

千锋教育
QIAN FENG EDUCATION

BOM介绍

1, BOM的概念

BOM 是Browser Object Model的缩写，简称浏览器对象模型, 提供了独立于内容而与浏览器窗口进行交互的对象，用于访问浏览器的功能。BOM由一系列相关的对象构成，并且每个对象都提供了一些方法与属性. 我们可以通过这些属性和方法去对浏览器进行操作.

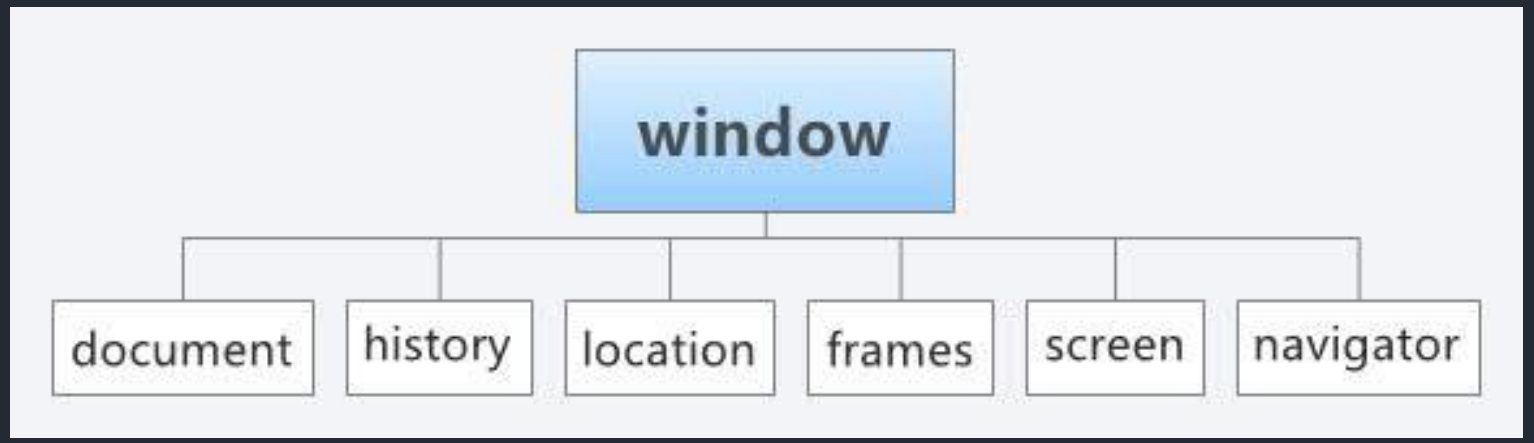
JavaScript语法的标准化组织是ECMA，DOM的标准化组织是W3C, 但是BOM缺乏标准.由于BOM 缺少规范，每个浏览器提供商又按照自己想法去扩展它，那么浏览器共有对象就成了事实的标准。

BOM中的对象和方法

2, window对象

window对象是BOM的核心, window对象表示浏览器窗口的一个对象, 每个浏览器窗口都有一个window对象与之对应.

window对象是BOM的顶层(核心)对象，所有对象都是通过它延伸出来的.



BOM中的对象和方法

window对象的属性对象:

document(核心): 文档对象，让我们可以在js脚本中直接访问页面元素(DOM)

history(重要): 历史对象,包含窗口的浏览历史，可以实现后退

location(重要): 包含浏览器当前的地址信息，可以用来刷新本页面或跳转到新页面

frames: 框架对象，可以获取页面框架内容

screen: 包含有关客户端显示屏幕的信息

navigator: 导航对象, 包含所有有关访问者浏览器的信息

BOM中的对象和方法

window对象的方法:

- alert(text): 弹出提示框(警告框)
- confirm(): 创建一个需要用户确认的对话框
- prompt(text,defaultInput) : 创建一个对话框要求用户输入信息
- open(url,name,[options]) : 打开一个新窗口并返回新 window 对象

setInterval(): 设置定时器

clearInterval() : 移除定时器

setTimeOut() : 设置延时器

clearTimeOut(): 移除延时器

close(): 关闭窗口

print(): 调出打印对话框

BOM中的对象和方法

window对象的属性和方法的调用:

window对象的属性和方法的调用方式，可以使用 window.属性、window.方法()或者一般可以直接使用: 属性、方法()的方式调用。

例如：window.alert() 和 alert()效果是一样的

BOM中的对象和方法

3, 系统对话框

alert(“谢谢”); 弹出警告框

```
confirm(“请确定或者取消”);   提示框 :确定和取消
if (confirm((“请确定或者取消”)){
    alert(“您按了确定！”);      //按确定返回true
} else{
    alert(“您按了取消！”)       //按取消返回false
}
```

var num = prompt(“请输入一个数字”, 0); 带输入的提示框: 提示文字和输入框默认值

alert(num);   //点击确定得到输入的值, 点击取消返回null

BOM中的对象和方法

4, window.open()

window.open(): 打开指定的网址url. 一般可以接受三个参数:
(1. 要加载的URL 2. 窗口的名称 或者 窗口的目标 3. 一个特性的字符串)

```
open( “http://www.baidu.com” );                    //新建页面并打开百度
open( “http://www.baidu.com” , “baidu” );          //新建页面并打开百度,窗口命名为baidu
open( “http://www.baidu.com” , “_parent” );        //在本页窗口打开, _blank是新建(默认)
```

注: 不命名会每次打开新窗口，命名的第一次打开新窗口,之后在这个窗口中加载。

第三个字符串参数:
width/height: 新窗口的宽度和高度, top/left: 新窗口的Y坐标和X坐标

```
open( ‘http://www.baidu.com’ , ‘baidu’ , ‘width=400,height=400,top=200,left=200’ );
```

opener: 父窗口对象

```
document.onclick = function(){
    opener.document.write("调用父窗口对象输出!");
}
```

BOM中的对象和方法

5, location对象

location是BOM对象之一，它提供了与当前窗口中加载的文档有关的信息, 还提供了一些导航功能。事实上，location对象是window对象的属性，也是document对象的属性; 所以window.location和document.location等效。

```
alert(window.location);     //获取当前的URL
alert(location);            //window可以省略
alert(window.document.location);
```

BOM中的对象和方法

location对象的属性

属性	描述的 URL 内容
hash	如果该部分存在，表示锚点部分
host	主机名：端口号
hostname	主机名
href	整个 URL
pathname	路径名
port	端口号
protocol	协议部分
search	查询字符串

BOM中的对象和方法

location对象的属性

```
location.hash = '#1'; //设置#后的字符串，并跳转
console.log(location.hash);          //获取#后的字符串
console.log(location.port);          //获取当前端口号
console.log(location.hostname);  //获取主机名(域名)
console.log(location.pathname);  //获取当前路径(服务器地址后部分)
console.log(location.search);  //获取?后面的字符串
location.href = "http://www.baidu.com"; //设置跳转的URL，并跳转
```

location对象的方法

```
location.assign( 'http://www.baidu.com' );  //跳转到指定的URL, 与href等效
location.reload();     //最有效的重新加载,有缓存加载
location.reload(true);   //强制加载,从服务器源头重新加载
location.replace( "http://www.baidu.com" ); //用新的URL替代,可以避免产生历史记录
```

BOM中的对象和方法

6, history对象

history对象是window对象的属性，它保存着用户上网的记录,从窗口被打开的那一刻算起.

history对象的属性

```
history.length;    //history对象中的记录数
```

history对象的方法

```
history.back();     //前往浏览器历史条目前一个URL, 类似后退
history.forward();  //前往浏览器历史条目下一个URL, 类似前进
history.go(-1);     //浏览器在history对象中向前或向后(-1代表后退, 1代表向前,0表示刷新当前页面)
```

BOM中的对象和方法

7, navigator对象(了解)

navigator对象是window对象的属性，它保存了浏览器的信息, 如: 浏览器名称,版本,浏览器所在的电脑操作系统等。

navigator对象的属性

```
console.log(navigator.appName);  //浏览器名称
console.log(navigator.appVersion);  //浏览器版本
console.log(navigator.platform);  //操作系统
//最新的浏览器已经全面放弃以上这些属性
```

navigator对象的属性

```
console.log(navigator.userAgent);  //用户代理信息, 通过该属性可以获取浏览器及操作系统信息
```

BOM中的对象和方法

close() 关闭浏览器

例如: 5秒后关闭浏览器

```
setTimeout(function(){
    window.close();
    alert("窗口关闭");
}, 5000);
```

注意: 火狐浏览器不支持, 如要支持火狐浏览器, 则可以先open指定的页面,然后在该页面调用close()

print(): 调出打印对话框

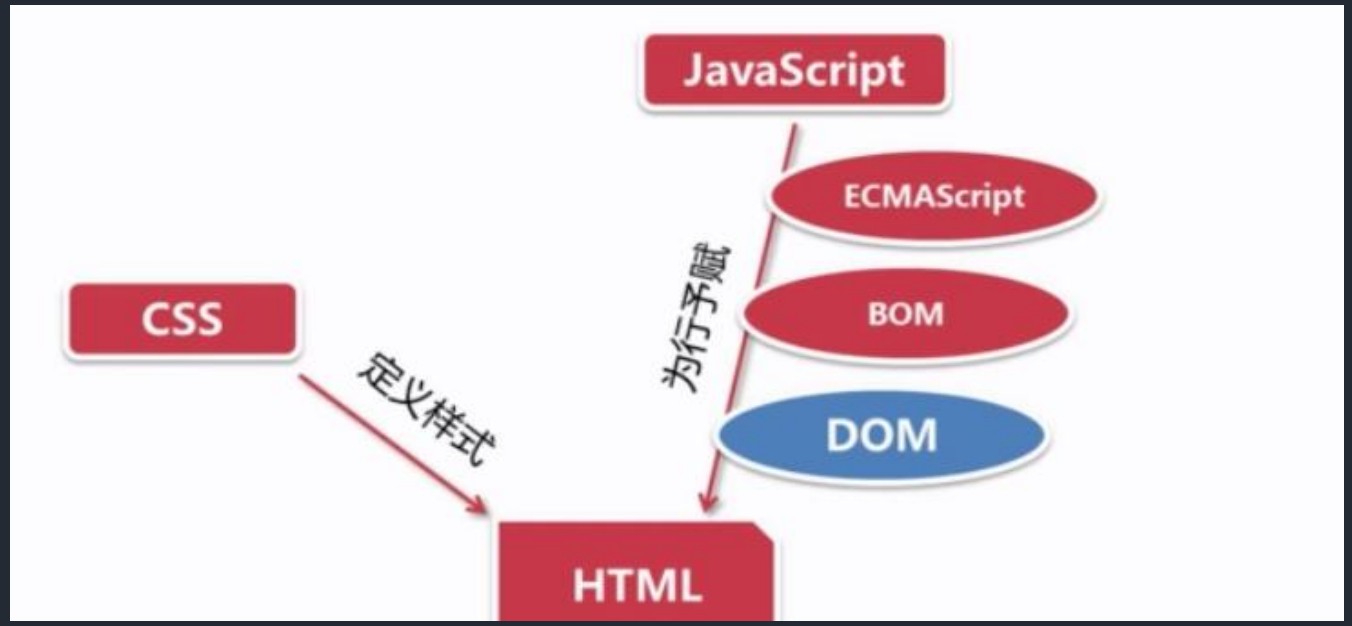
DOM介绍

DOM的概念

DOM就是Document Object Model(文档对象模型)的缩写，DOM 是 W3C（万维网联盟）的标准。

DOM是中立于平台和语言的接口，它允许程序和脚本动态地访问和更新文档的内容、结构和样式。

HTML-页面结构　css-页面样式　javascript –页面行为操作



DOM介绍

DOM概念详解

D(文档Document): 可以理解为整个Web加载的网页文档；

O(对象Object): 可以理解为类似window对象之类的东西，可以调用属性和方法，这里我们说的是 document 对象；

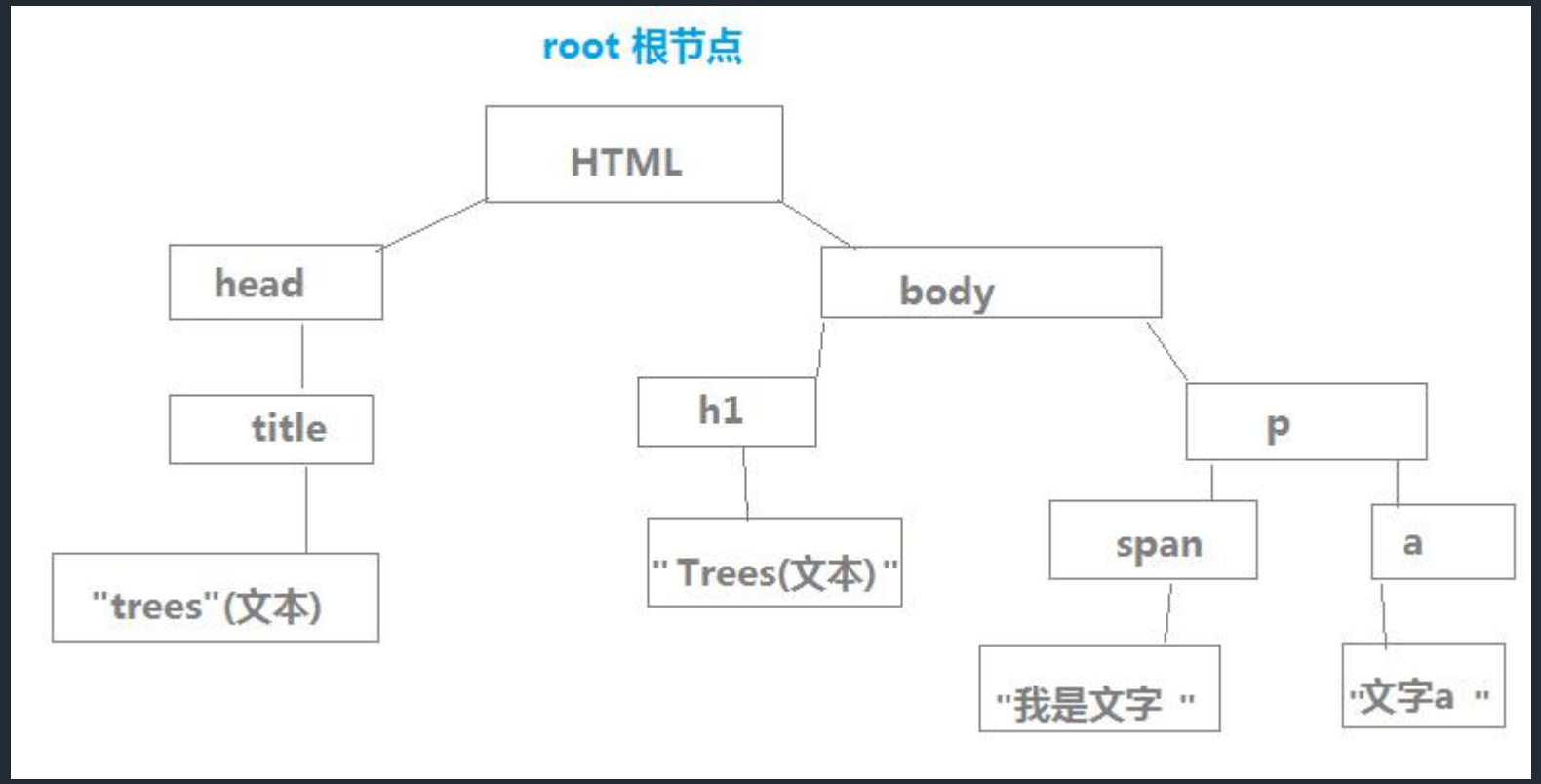
M(模型Model): 可以理解为网页文档的树型结构。

通过DOM，我们可以访问所有的HTML元素，连同它们所包含的文本和属性。可以对其中的内容进行修改和删除，同时也可以创建新的元素

DOM介绍

DOM树(DOM树形结构)

加载 HTML 页面时，Web 浏览器生成一个树型结构，用来表示页面内部结构。DOM 将这种树型结构称为节点组成的节点树。



DOM介绍

DOM节点分类

DOM节点分为三种：元素节点, 属性节点和文本节点

例如: 测试Div

：元素节点(最重要)

title="属性节点"：属性节点

测试Div ： 文本节点

DOM节点和属性

1, 元素节点对象的获取方式:

刚才讲了DOM节点的三种分类, 其中最重要,使用最多的是元素节点;

元素节点的获取方式有多种, 常用的有以下三种:

getElementById(): 获取特定ID元素的节点对象(返回一个对象)

getElementsByTagName(): 获取指定标签名的节点列表(返回一个数组)

getElementsByName(): 获取相同name属性值的节点列表(返回一个数组)

DOM节点和属性

2, getElementById()

getElementById()需要给一个参数: 元素节点的id属性值。如果找到相应的元素则返回该元素的元素节点对象，如果不存在，则返回 null。

例如: 获取 id 为 box 的元素节点

```
var box = document.getElementById('box');
```

注意: 如果id值存在, 但是返回null，则是执行顺序的问题

解决方法:

1. 把<script>标签(JS代码)移到html结构后面；
2. 使用onload事件来处理， onload事件会在html加载完毕后再调用。

```
//加载 html 后执行
window.onload=function() {
    document.getElementById('box');  //id具有唯一性
};
```

DOM节点和属性

3, 元素节点的属性

tagName: 元素节点对象所指向的标签名称

id: 元素节点的id属性值

innerHTML: 元素节点中的内容

className: 元素节点的class属性值

style: css内联样式对象

title: 元素节点的title属性值

例如:

```
document.getElementById( 'box' ).id;    //获取 id
document.getElementById( 'box' ).id = 'person' ;  //设置id
document.getElementById( 'box' ).style;  //获取css样式对象
document.getElementById( 'box' ).style.color;    //获取 style 对象中 color 的值
document.getElementById( 'box' ).style.color= 'red' ; //设置 style 对象中 color 的值
document.getElementById( 'box' ).className;  //获取 class
document.getElementById( 'box' ).className= 'box' ;  //设置 class
```

DOM节点和属性

4, getElementsByTagName()

getElementsByTagName()方法将返回一个数组HTMLCollection(NodeList)，这个数组保存着所有相同标签名的节点列表。

例如:

document.getElementsByTagName(‘li’); 获取所有li元素，返回数组

document.getElementsByTagName(‘li’).length; 获取所有li元素的数量

document.getElementsByTagName(‘li’)[0]; 获取第一个li元素

document.getElementsByTagName(‘li’).item(0); 获取第一个li元素

DOM节点和属性

5, getElementsByName()

getElementsByName()方法可以获取相同名称(name)的元素，返回一个数组 HTMLCollection(NodeList)。

例如:

document.getElementsByName(‘box’); 获取 input 元素

document.getElementsByName(‘box’)[0].value; 获取 input 元素的 value 值

6, getElementsByClassName(): 通过class名称获取元素节点对象(IE9+以上)

DOM节点和属性

7, 获取子元素节点

我们获取元素节点不一定必须使用document.来调用getElementsByTagName()或getElementByName(); 在某些情况下, 我们需要获取指定范围内的子元素节点.

例如:

```
<div id="box">
    <img src="images/1.jpg">
    <img src="images/2.jpg">
    <img src="images/3.jpg">
</div>
<img id="showImg" src="images/4.jpg" />

//通过oBox元素节点, 获取其三个子元素节点img
var oBox = document.getElementById("box");
var aImg = oBox.getElementsByTagName("img");
```

DOM节点和属性

示例:
1, 有一个装有颜色的数组 ["red", "blue", "yellow", "green", "black", "orange"];
点击button 启动定时器,每一秒给它换一次色, 按顺序换色

DOM节点和属性

DOM节点的共有属性

DOM节点可以分为元素节点、属性节点和文本节点，而这些节点又有三个共有的属性，分别为：nodeName、nodeType 和 nodeValue。

节点类型	nodeName	nodeType	nodeValue
元素	元素名称	1	null
属性	属性名称	2	属性值
文本	#text	3	文本内容(不包含 html)

例如：元素节点的nodeName, nodeType和nodeValue的属性值

```
var oDiv = document.getElementById("div");
console.log(oDiv.nodeName); //DIV
console.log(oDiv.nodeType); //1
console.log(oDiv.nodeValue); //null
```

DOM节点和属性

childNodes: 某元素的所有子节点，这些子节点包含元素节点和文本节点。

例如:

```
var box = document.getElementById( ‘box’ );  获取元素节点对象
console.log(box.childNodes.length);  获取这个元素节点的所有子节点
console.log(box.childNodes[0]);  获取第一个子节点对象
```

children: 某元素的所有子元素节点

DOM节点和属性

忽略空白文本节点

在非IE中, 标准的DOM具有识别空白文本节点的功能，而 IE自动忽略了, 如果要保持一致的子元素节点, 需要手工忽略掉它.

```
//忽略nodes数组中的空白节点
function filterSpaceNode(nodes){
    for(var i=0; i<nodes.length; i++) {
        //如果是文本节点, 且文本节点的内容为一个或多个空格
        if ( nodes[i].nodeType==3 && /^\s+$/.test(nodes[i].nodeValue) ){
            nodes[i].parentNode.removeChild(nodes[i]);  //找到空白文本节点后, 将其删除
        }
    }
    return nodes;
}
```

注意: /^\s+$/表示一个或多个空格

DOM节点和属性

innerHTML和nodeValue的区别
1, 取值的区别
nodeValue 可以获取文本节点的内容和属性节点的值。
innerHTML只能获取元素节点中的内容。

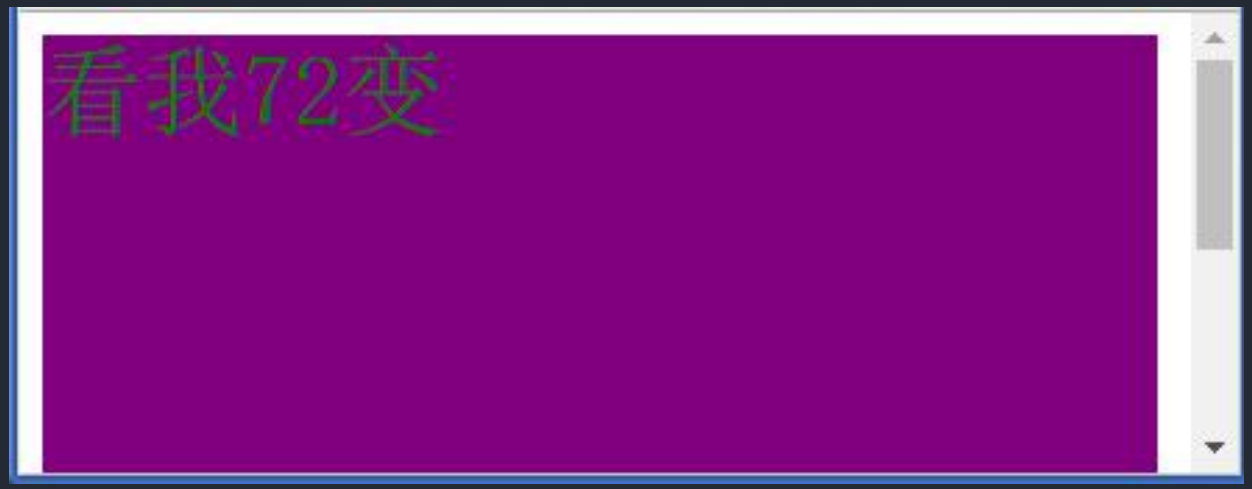
2, 赋值的区别
nodeValue 会把包含在文本里的HTML标签按原样输出。
innerHTML 可以解析内部的HTML标签

例如:
文本节点使用nodeValue,结果为：abc
box.childNodes[0].nodeValue='abc';

//元素节点使用innerHTML,结果为：**abc**(加粗的)
box.innerHTML='abc';

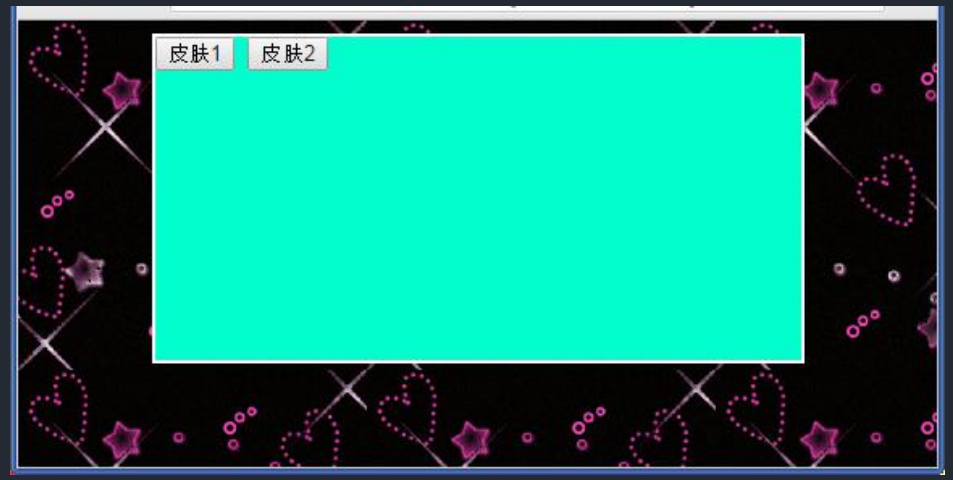
练习

1，写一个定时器，每隔0.1秒修改一次div内文字颜色和文字大小．最开始这个文字是默认大小，开启定时器后文字大小开始增大，当增大了6次以后，文字大小开始缩小，缩小6次，文字再开始增大...效果如下图:



练习

2, 网页换肤, 点击皮肤1, 切换到第一张图的效果, 点击皮肤2切换到第二张图的效果



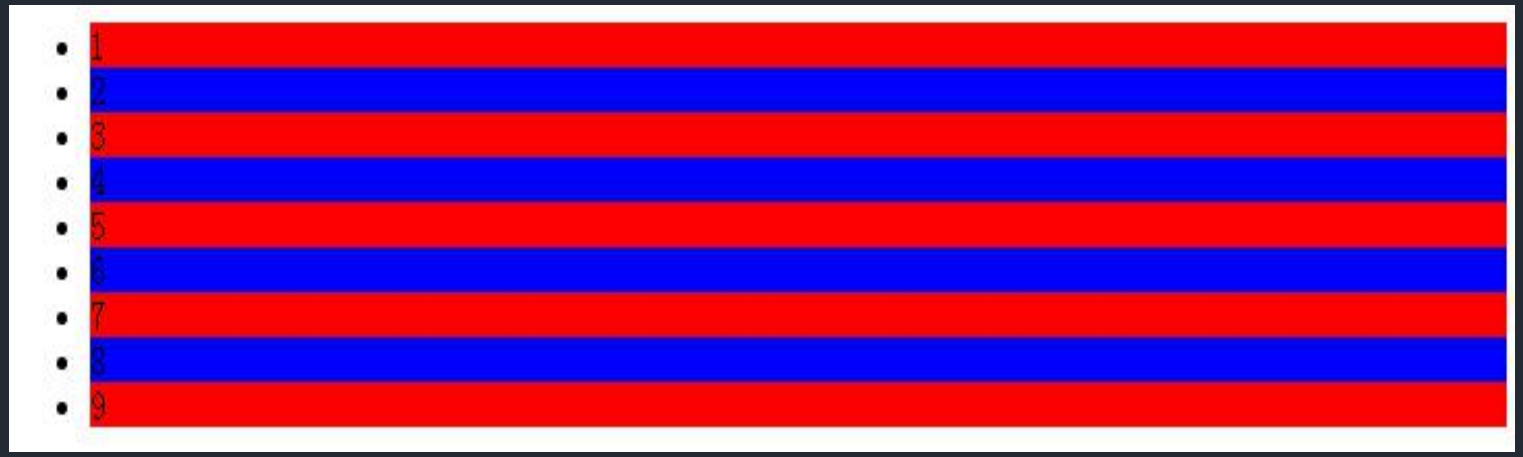
练习

3, 实现一个个人信息页面　姓名　年龄　学历　专业, 可以获取到输入的信息, 点击保存按钮以后, 输出信息(信息显示形式: 姓名:张三,年龄:33,学历:本科,专业:计算机)

要求: 分别使用通过id, tagName, name三种获取节点的方式实现

练习

4，隔行变色



THANK YOU

千锋教育
QIAN FENG EDUCATION

做真实的自己，用良心做教育