

一、js的导入方式中外链式和导入式的区别

所谓外链式css指的是 `<link rel="stylesheet" href="css/text.css">`

而导入式css指的是`@import "cssstyle.css"`。

首先，让我们区别一下什么是链接外部样式和导入外部样式：

我们知道，这两种引用外部样式的方式作用几乎是一样的，都是将独立的CSS样式文件引用到网页文件中来，但二者还是有一些细微的的差别的（因为设计都不会无聊到搞出两个完全一样的东西来），下面就根据我所了解到的两个方面进行探讨

1、使用link链接的css是客户端浏览你的网页时先将外部的CSS文件加载到网页当中，然后再进行编译显示，所以这种情况下显示出来的网页跟我们预期的效果一样，即使网速再慢也是一样的效果。而使用@import导入的CSS就不同了，客户端在浏览网页时是先将html的结构呈现出来，再把外部的CSS文件加载到网页当中，当然最终的效果也是跟前者是一样的，只是当网速较慢时会出现先显示没有CSS统一布局时的html网页，这样就会给阅读者很不好的感觉。这也是现在大部分网站的CSS都采用链接方式的最主要原因。

2、导入样式可以避免过多页面指向一个css文件。当网站中使用同一个CSS文件的页面不是非常多时，这两种方式在效果方面几乎是没有什么不同的，但网站的页面数达到一定程度时（比如新浪等门户），如果采用链接的方式可能就会使得由于多个页面调用同一个CSS文件而造成速度下降，但是一般页面能达到这种程度的网站也会有资本用最好的硬盘，所以这方面的因素也不用怎么担心。

二、js提供的输出方式

1.alert：在浏览器中弹出提示框，提示的内容都被转化为**字符串**输出（调用了toString这个方法）。

```
alert('hello');//弹出hello
alert('[23,2]');//弹出23,2
alert({name:(js)});//弹出"[object Object]"
```

2.confirm：在alert基础上增加让用户选择的操作（确定，取消），内容也转化为**字符串**。

```
var st = confirm('are you sure delete');//点击确定接收ture,反之接收false.
alert(st);//点击确定弹出ture,反之弹出false.
```

3.promot：在confirm基础上增加让用户输入的效果

```
var flag = prompt('are you sure delete');//点击取消返回null，点击确定输入有内容返回ture，无内容返回空字符串。  
alert(flag);
```

4.conclse：控制台输出，方便开发人员调试(不会转化数据类型，输出什么格式都可以)

```
console.log('1');//输出1  
console.log('{name:'stru'}');//输出{name:'stru'}  
console.dir('');//输出比.log更详细  
console.table('');//把JSON数据展示成一个表格
```

三、布尔类型

Boolean :把其他数据类型转化为**布尔类型**

1.只有0，null，NaN，空字符串，undenfine 这五个转化为布尔的**false**，其余的都转化为**true**。

```
> Boolean(1)  
true  
-----  
> Boolean(0)  
false  
-----  
> Boolean(null)  
false  
-----  
> Boolean(-1)  
true  
-----  
> Boolean('')  
false  
-----  
> Boolean(NaN)  
false
```

2.!

!= :不等于

!! :取反：先把值**转化为布尔类型**，然后在取反。

```
> ![]  
false  
-----  
> !null  
true  
-----  
> !0  
true
```

优化

```

for (var i = 0; i < oList.length; i++) {
    var n = i % 3,
    bg = oList[i].style.backgroundColor;
    switch (n) {
        case 0:
            bg = '#555';
            break;
        case 1:
            bg = '#eee';
            break;
        default:
            bg = '#ddd';
    }
}

```

一、

二、

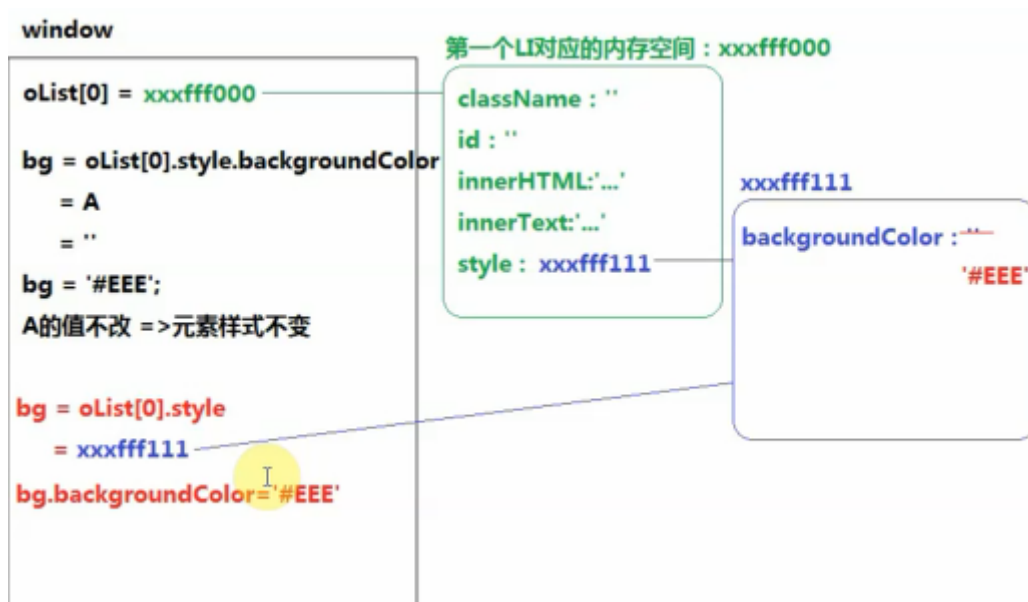
```

for(var i = 0;i<oList.length;i++){
    var n = i%3,
    bg = oListp[i].style;
    switch(n){
        case 0:
            bg.backgroundColor='#eee';
            break;
        case 1:
            bg.backgroundColor='#eff';
            break;
        defsult:
            bg.backgroundColor='#fff';
    }
}

```

第一种不行，第二种可以

区别如下：因为是引用数据类型



区别

1. getElementById()

通过获取**元素的id**获取其属性，获取的结果是一个对象数据类型。

2. getElementsByName()

通过获取**name**获取元素。

3. getElementsByTagName()

通过获取**标签名称**获取其属性。

```
var bookBox = [context].getElementsByTagName() // context表示要获取标签的位置
```

知识点

console.dir();// -> HTMLCollection() 元素集合数

以数字作为属性名，每个属性存储的都是获取到的每一个li，数字是**索引**，同时最后还包括长度。

```
1.newList[0]  
newList[n+1]  
  
2.newList.length // 长度
```

具备以上两种特点特别像数组，但不是数组。称之为**类数组**。

filter: alpha (opacity, finishopacity,
style, startX, startY, finishX, finishY)

opacity代表透明度数,选值0-100,0是完全透明,100是不透明.

finishopacity用来设置结束时的透明度,以达到渐变效果.取值范围也是0-100.style指渐变类型,0是无变化,1是线行渐变,2是放射渐变,3是X型渐变.

filter: alpha(opacity=90,finishopacity=0,style=1,startx=150,starty=0,finishx=0,finisyy=0);

可以实现一张图片从右至左 透明度逐渐变为0的效果
writing-mode :lr-tb (左到右, 上到下) 实现文字排版

```
.animate {
  font-size: 40px;
  margin: 100px 0 0;
}

.animate span {
  display: inline-block;
}

.box span {
  color: #fff;
  opacity: 0;
  transform: translate(30px, 0) scale(.7);
  animation: leftRight 5s forwards;
  /*animation-play-state: paused;*/
}

@keyframes leftRight {
  40% {
    transform: translate(30px,0) scale(.7);
    opacity: 1;
    color: #000;
  }
  60% {
    transform: translate(30px,0) scale(.7);
    opacity: 0;
    color: #000;
  }
  80% {
    transform: translate(30px,0) scale(0);
    opacity: 0;
    color: #000;
  }
  100% {
    transform: translate(30px,0) scale(0);
    opacity: 1;
    color: #000;
  }
}

<div class="animate box">
  <span>不一样的WEB前端工程师</span>
</div>
```

```
<html>
<head>
```

```
<title>鼠标放上链接改变网页背景颜色</title>
<meta http-equiv="Content-Type" mrc="text/html; charset=gb2312">
<SCRIPT LANGUAGE="JavaScript">
function goHist(a)
{
    history.go(a);
}
</script>
</head>
<body>
<center>
<h2>鼠标放到相应链接上看看! </h2>
<table border=1 borderlight=green style="border-collapse: collapse" cellpadding="5" cellspacing="0">
<tr><td align=center><a href="#"
onMouseOver="document.bgColor='skyblue'">天空蓝</a>
<a href="#" onMouseOver="document.bgColor='red'">大红色</a> <a href="#"on
MouseOver="document.bgColor='#0066CC'">清新蓝</a>
</td></tr></table>
</center>

</body>
</html>
```