1. 常见块级元素：

<div>,<p>,<form>,<address>,<dl>,<ul>,<ol>,<pre>,<h1>,<h2>,<h3>,<h4>,<h5>,<h6>,

<table>,<hr/>

常见行内元素：

<span>,<a>,<i>,<em>,<code>,<sup>,<sub>,<textarea>,<select>

<input />,<img />这两个也可以认为是行内块元素

区别：

（1）块元素默认情况下独占一行，宽度自动填满其父级的宽度；行内元素又叫内联元素，不独占一行，宽度随填充内容而变。紧邻的行内元素会排在同一行中。

（2）块元素的width/height可以进行设置但行内元素的width/height默认情况下不能设定。

（3）块元素的margin,padding值可进行设定，行内元素的左右值会产生边距效果，但上下值没有边距效果。

1. css的选择器包括类选择器（.class）,id选择器（#id）,属性选择器（[attribute]）,元素选择器（element），派生选择器，伪类选择器

优先级：id选择器>类选择器>属性选择器>派生选择器>元素选择器

1. css中position的属性值及区别为：

static:正常值，不做任何定位，元素位于文档流当中。

absolute:绝对定位，相对于document的原点位置（左上角）位置。

relative:相对定位，相对于父级元素进行定位。

fixed:固定定位，相对于浏览器窗口进行定位。

1. createElement() //创建节点

createTextNode() //创建文本节点

createDocumentFragment() //创建一个DOM片段

appendChild() //添加  
　　removeChild() //移除  
　　replaceChild() //替换

getElementById() //通过id查找节点

getElementByTagName() //通过元素查找节点

getElementsByClassName()//通过类名查找节点，返回节点数组

1. （1）定义与用法：当一个函数的返回值是另一个函数，而返回的函数调用了父级函数定义的变量，并且这个函数被调用的时候，就形成了闭包。（阐述正确即可）
2. 表现形式：使函数外部能够调用函数内部定义的变量。
3. 使用闭包的注意点：第一，滥用闭包，会造成内存泄漏：由于闭包会使得函数中的变量都被保存在内存中，内存消耗很大，所以不能滥用闭包，否则会造成网页的性能问题，在IE中可能导致内存泄露。解决方法是，在退出函数之前，将不使用的局部变量全部删除。
4. 会改变父函数内部变量的值。所以，如果你把父函数当作对象使用，把闭包当作它的公用方法，把内部变量当作它的私有属性，这时一定要小心，不要随便改变父函数内部变量的值。
5. 小明,17 小花,18
6. teacher.call(this,args[0],args[1]); 或者 teacher.apply(this,args[0],args[1]);
7. 方法一：使用对象判断是否该值存在

var arr = [0,0,2,3,8,11,8,25];

var obj = {};

var temp = [];

for(var i = 0 ;i< arr.length;i++){

if( !obj[arr[i]] ){

obj[arr[i]] = 1;

temp.push(arr[i]);

}

}

console.log(temp );

方法二：使用循环数组往另一个数组中push不重复元素的方式。

var arr = [2,3,4,4,5,2,3,6],

arr2 = [];

for(var i = 0;i< arr.length;i++){

if(arr2.indexOf(arr[i]) < 0){

arr2.push(arr[i]);

}

}

console.log(arr2);

方法三：使用filter(),对数组中的每个元素都执行一次指定的函数（callback），并且创建一个新的数组，该数组元素是所有回调函数执行时返回值为 true 的原数组元素。

var arr = [2,3,4,4,5,2,3,6];

var arr2 = arr.filter(function(element,index,self){

return self.indexOf(element) === index;

});

console.log(arr2);