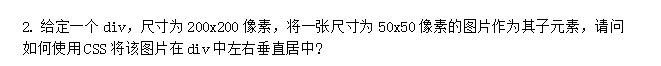


function *A*() {  
 this.id = 0;  
 this.add = function () {  
 setTimeout(function () {  
 this.id += 1; //这里的this指向的是window,而不是A,所以this.id为undefined;运行this.id += 1后,this.id为NaN;  
 }, 100);  
 };  
}  
var ***aInstance*** = new *A*();  
***aInstance***.add(); //A.add()无返回值,因此没有运行结果;

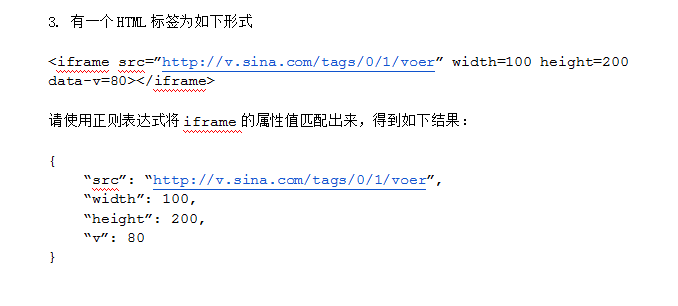


/\*（1）定位：（<div><img src="img1.jpg" alt=""></div>）\*/  
/\*①\*/  
div{  
 *width*: 200**px**;  
 *height*: 200**px**;  
 *position*: **relative**;  
}  
div img{  
 *display*: **block**;  
 *border*: **none**;  
 *width*: 50**px**;  
 *height*: 50**px**;  
 *position*: **absolute**;  
 *top*: 0;  
 *left*: 0;  
 *right*: 0;  
 *bottom*: 0;  
 *margin*: **auto**;  
}  
/\*②\*/  
div{  
 *width*: 200**px**;  
 *height*: 200**px**;  
 *position*: **relative**;  
}  
div img{  
 *display*: **block**;  
 *border*: **none**;  
 *width*: 50**px**;  
 *height*: 50**px**;  
 *position*: **absolute**;  
 *top*: 50%;  
 *left*: 50%;  
 *margin*: -25**px** 0 0 -25**px**;  
}

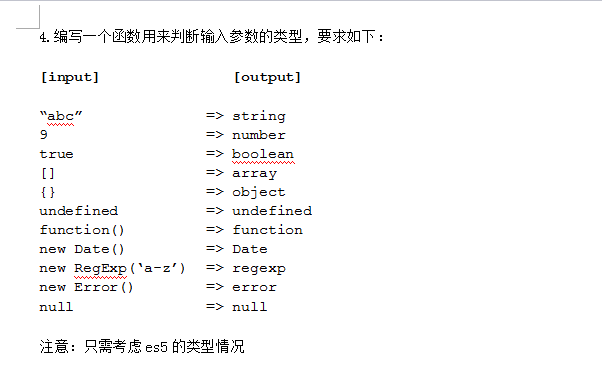
/\*（2）flex布局：（<div><img src="img1.jpg" alt=""></div>）\*/  
 div{  
 *width*: 200**px**;  
 *height*: 200**px**;  
 *display*: **flex**;  
 *flex-flow*: **row nowrap**;  
 *justify-content*: **center**;  
 *align-items*: **center**;  
 }  
div img{  
 *display*: **block**;  
 *border*: **none**;  
 *width*: 50**px**;  
 *height*: 50**px**;  
}

/\*（3）插入span：（<div><img src="img1.jpg" alt=""><span></span></div>）\*/

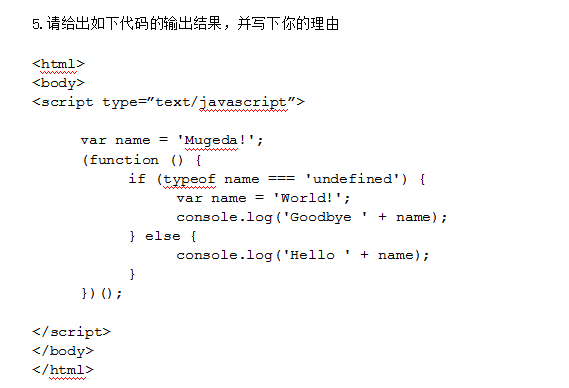
div{*width*:200**px**;*height*:200**px**;*border*:1**px solid red**;*text-align*: **center**;}  
 img{  
 *vertical-align*: **middle**; *border*: **none**;*width*: 50**px**;*height*: 50**px**;}  
 span{*display*: **inline-block**;*vertical-align*: **middle**;*width*:0;*height*:100%;}



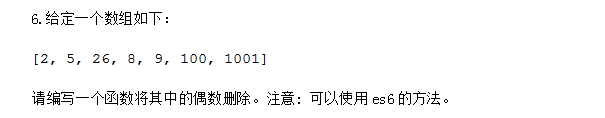
var ***str*** = "<iframe src='http://v.sina.com/tags/0/1/voer' width=100 height=200 data-v=80></iframe>";  
var ***obj*** = {};  
var ***arr*** = ***str***.match(/[a-z]+=\S+/g);  
(function () {  
 for (var i = 0; i < ***arr***.length; i++) {  
 ***obj***[***arr***[i].replace(/=\S+/, "")] = ***arr***[i].replace(/[a-z]+=/g, "").replace(/>\S+/g, "").replace(/'/g, "");  
 }  
 return ***obj***;  
})();  
console.log(***obj***);



function *getType*(*typeObj*){  
 var typeName = typeof *typeObj*;  
 switch(typeName) {  
 case "undefined":  
 case "boolean":  
 case "number":  
 case "string":  
 case "function": return typeName;  
 case "object": if (*typeObj* === null) {  
 return "null";  
 }else if(*typeObj* instanceof Array ){  
 return "Array";  
 }else if(*typeObj* instanceof Date ){  
 return "Date";  
 }else if(*typeObj* instanceof RegExp){  
 return "RegExp";  
 }else if(*typeObj* instanceof Error){  
 return "error";  
 }else{  
 return "object";  
 }  
 default: throw new TypeError("参数不是一个原始值: "+*typeObj*);  
 }  
}  
console.log(*getType*("abc"));  
console.log(*getType*(9));  
console.log(*getType*(true));  
console.log(*getType*([]));  
console.log(*getType*({}));  
console.log(*getType*(undefined));  
console.log(*getType*(new Date()));  
console.log(*getType*(new RegExp('a-z')));  
console.log(*getType*(new Error()));  
console.log(*getType*(null));  
console.log(*getType*(function(){}));



/\*输出结果：Goodbye World!  
 原因：变量提升,就是在函数域中定义一个变量时,变量的声明会提升至第一句;变量提升只是提升变量的声明，并不会把赋值也提升上来;因此,上述自执行函数中实际执行为：\*/  
(function () {  
 var name;  
 if (typeof name === 'undefined') {  
 name = 'World!';  
 console.log('Goodbye ' + name);  
 } else {  
 console.log('Hello ' + name);  
 }  
})();



function *quOu*(*arr*){  
 var newArr=[];  
 for(var i=0;i<*arr*.length;i++){  
 if(*arr*[i]%2!=0){  
 newArr.push(*arr*[i]);  
 }  
 }  
 return newArr;  
}