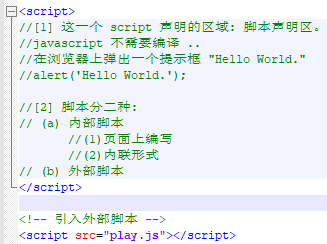
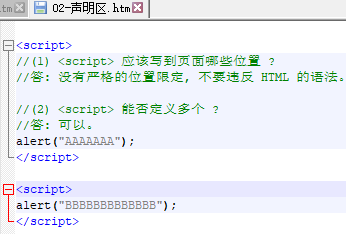
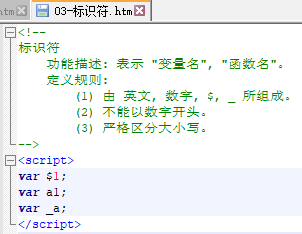
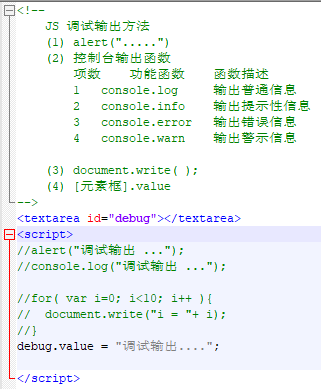
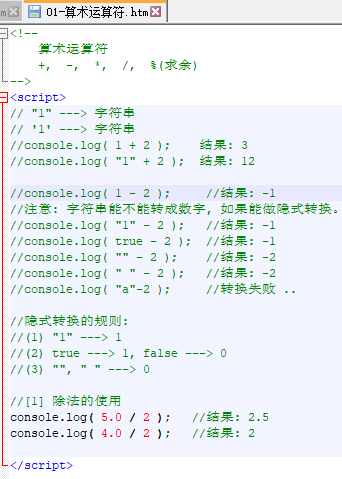
Jsp作业？

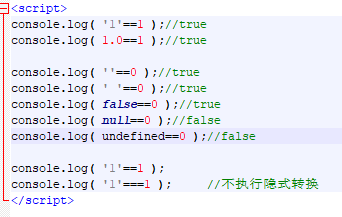


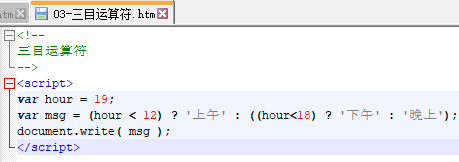


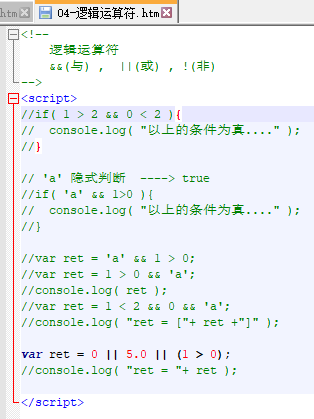


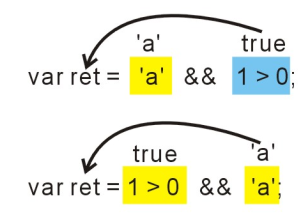


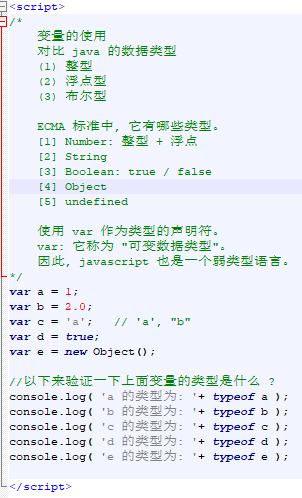


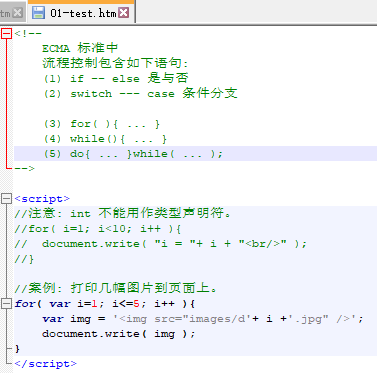




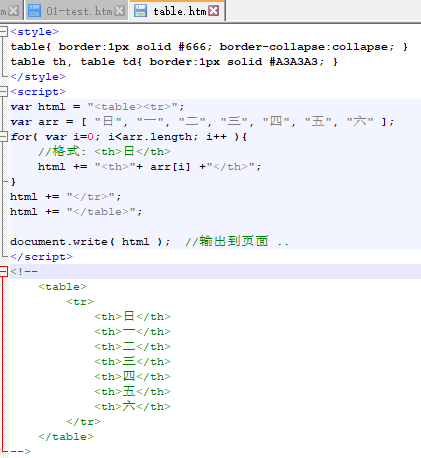


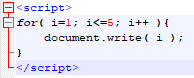


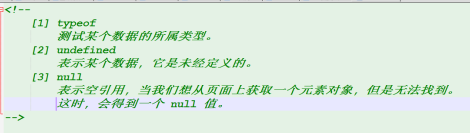


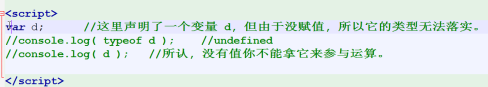
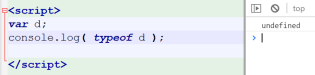
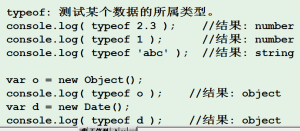


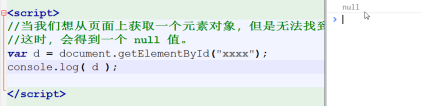
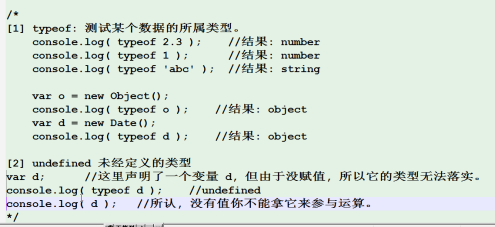
为什么可以没有var？

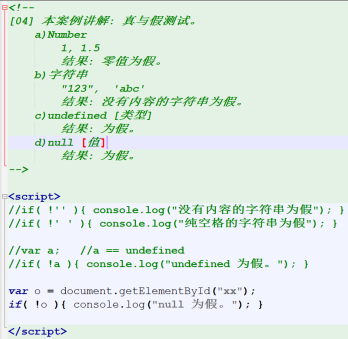
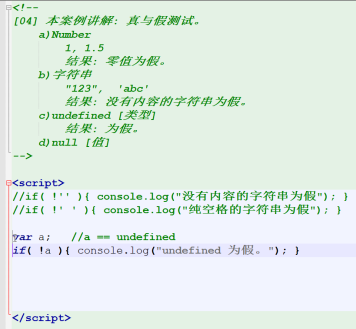
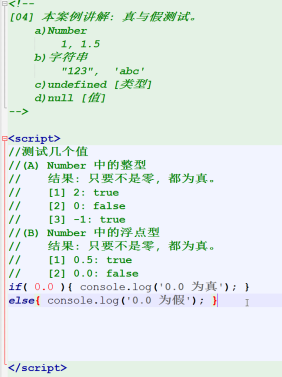


？？

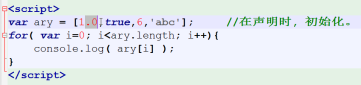
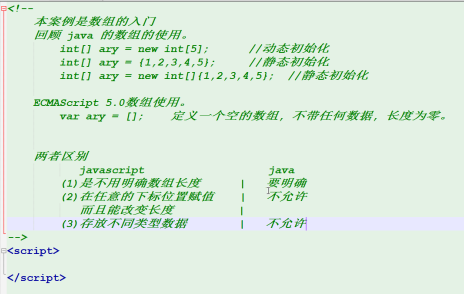
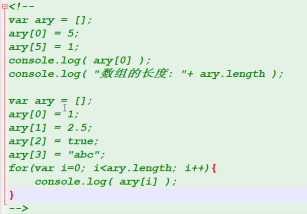


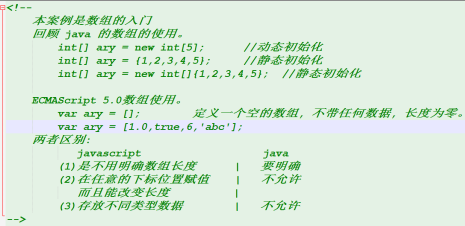


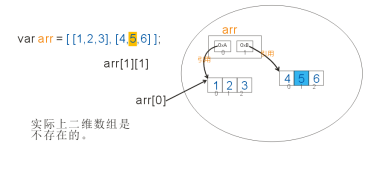
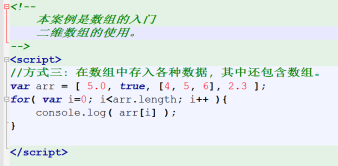
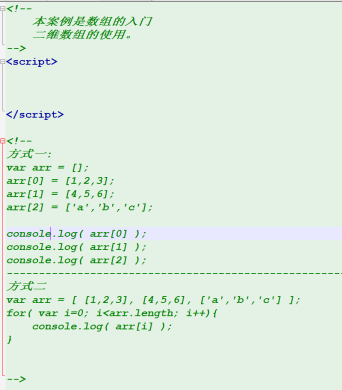
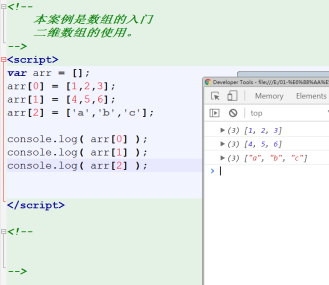




数组

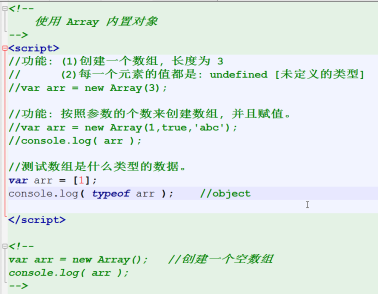


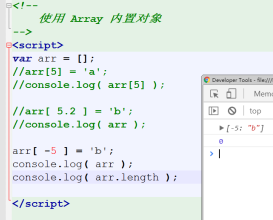


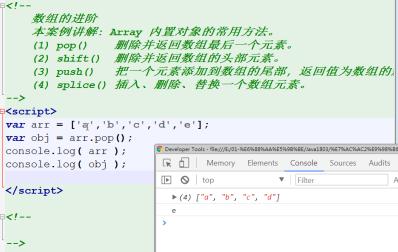


二维数组内存分布是连续的...？

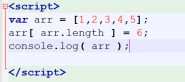
Array

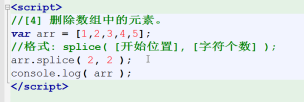






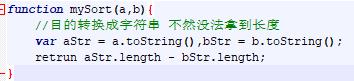
传统的做法：

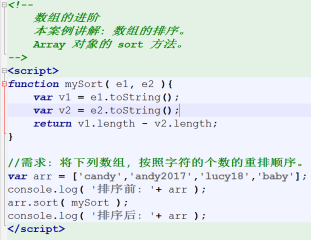


返回删除的元素

插入 替换

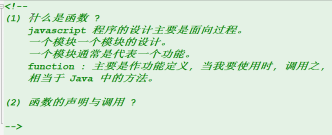


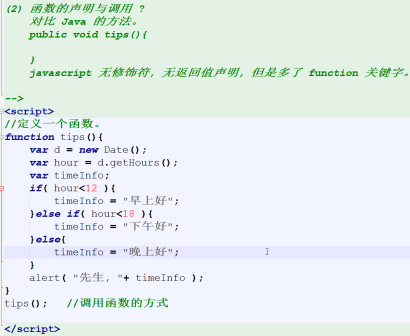


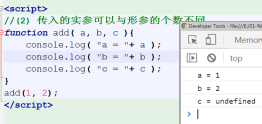


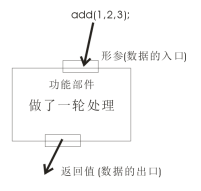
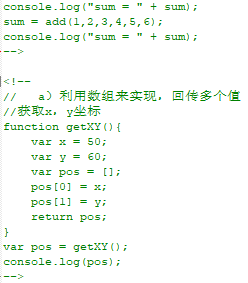
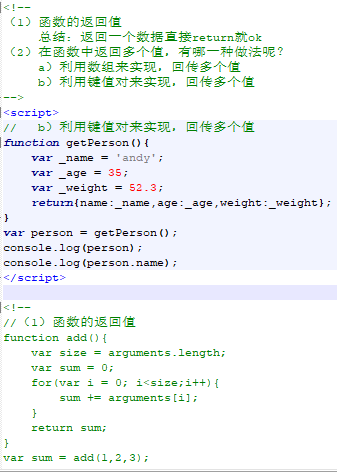


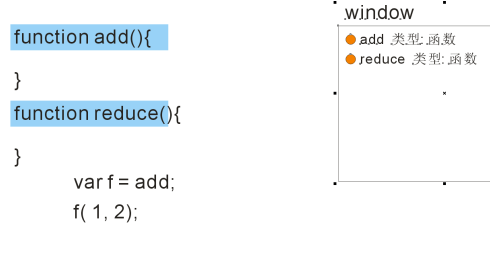
函数使用

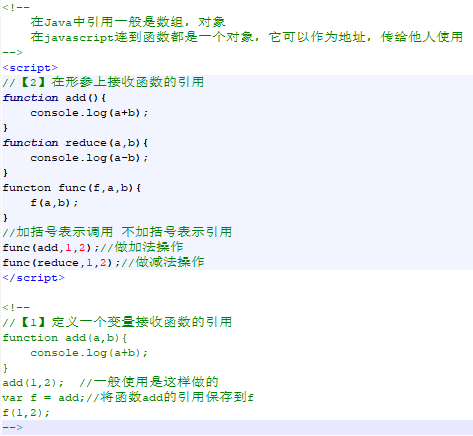


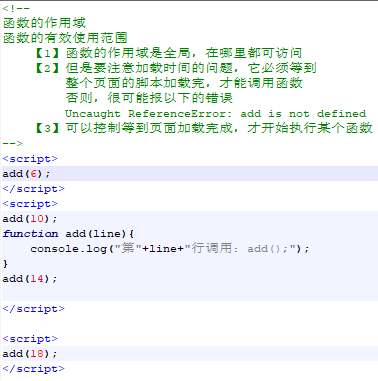


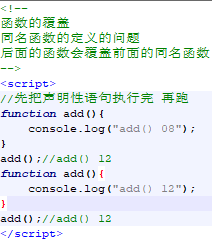


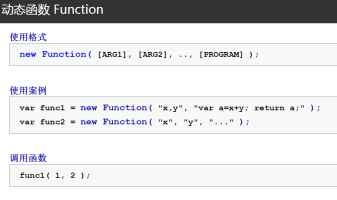




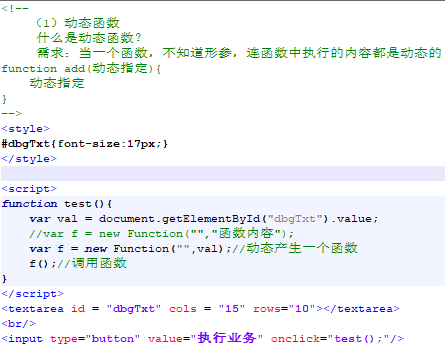


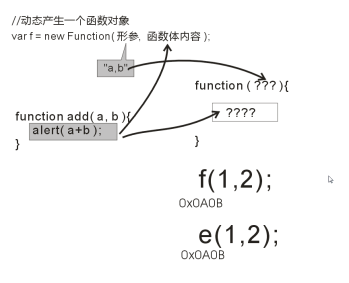
...

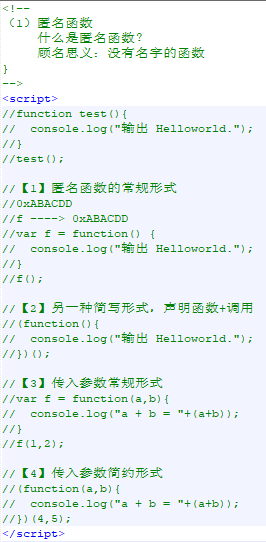












？？？

作业大题

<script>

function getSum(n){

var result = 0;

var j = 1;

for(var i = 1;i<=n;i++){

if(i%2==0){

result = result-1/j;

}else{

result = result + 1/j;

}

j= j+2;

}

console.log(result);

}

getSum(10);

</script>

<!--

var result = 0;

var j = 1;

var i = 1;

function getSum(){

if(i%2==0){

result = result-1/j;

}else{

result = result + 1/j;

}

j= j+2;

if(i<10){

i++;

getSum();

}

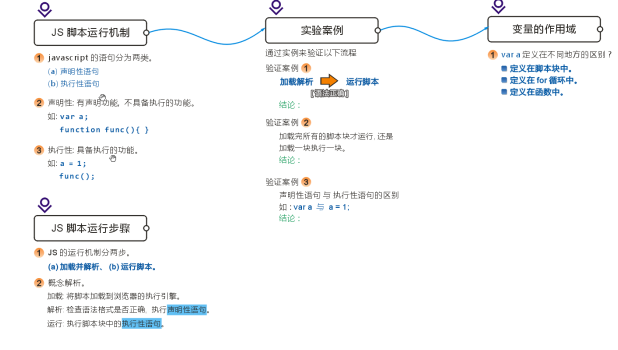
}

getSum();

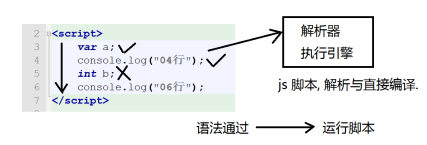
console.log(result);

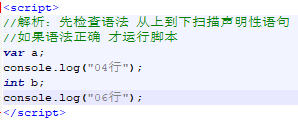
-->

？？

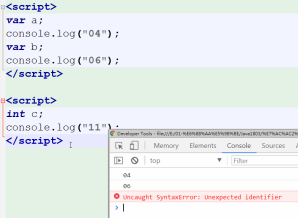
...

案例1：

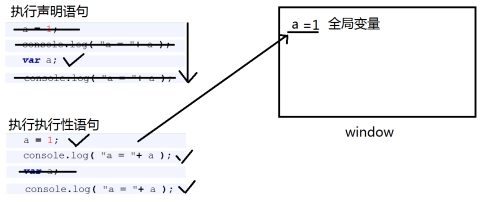


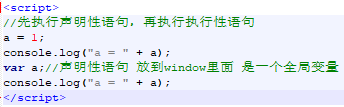


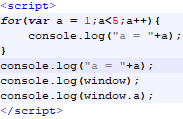
案例2：加载一块执行一块

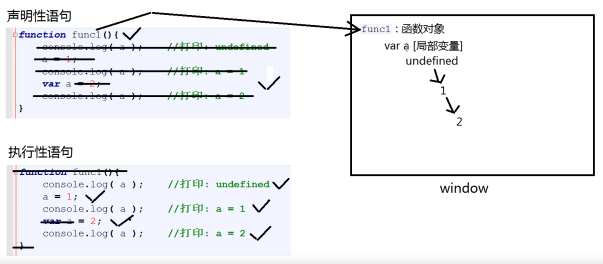


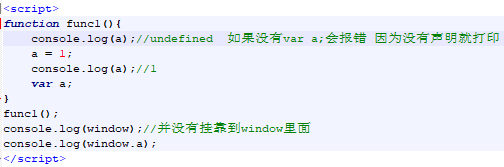
案例3：

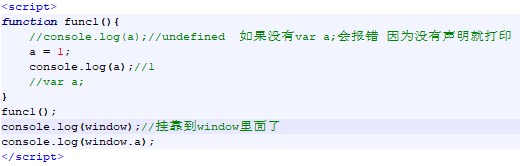




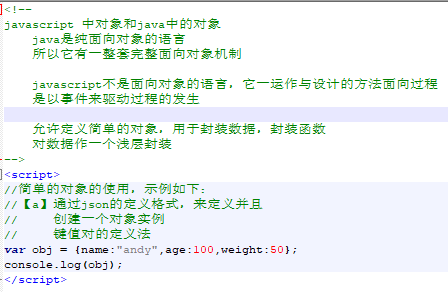
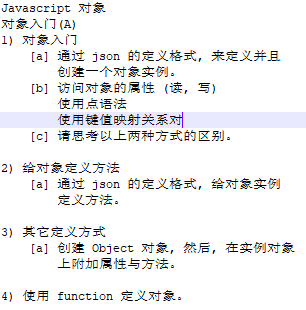
图...



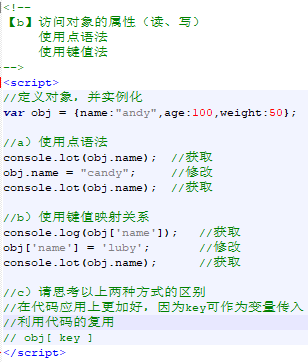
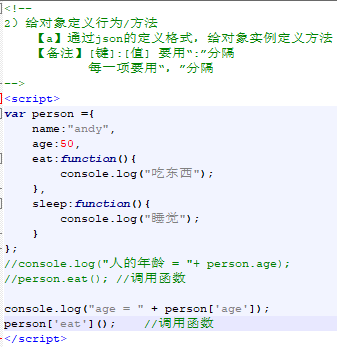




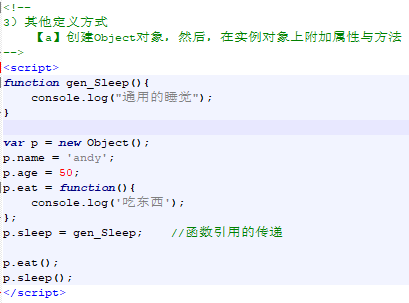
对象



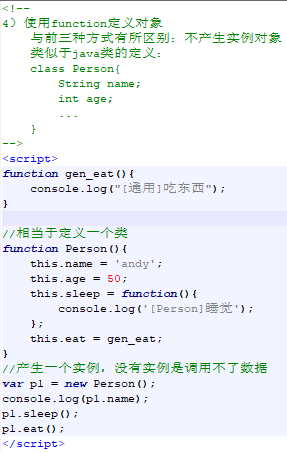




？

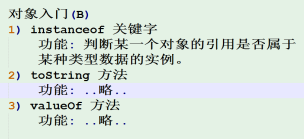






后面两种方法可以通过键值法来访问对象的属性和方法吗？

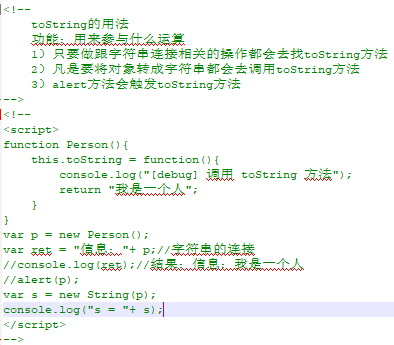
一共三种方法 两种访问方式



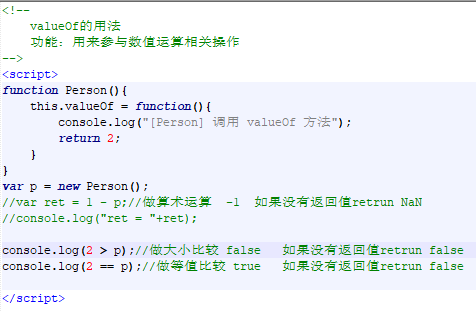


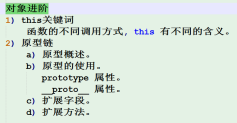
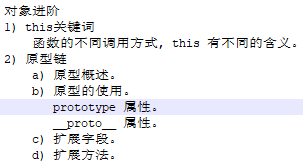


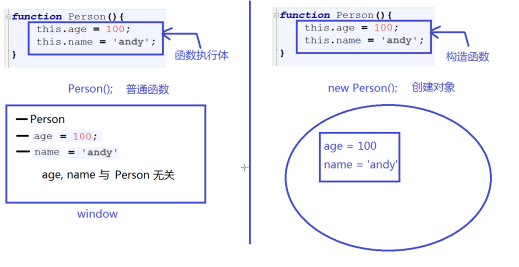






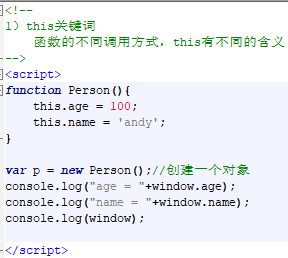




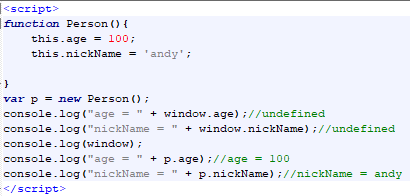


this.指向window









原型链

