**1、不需要借助第三个临时变量，实现两个变量的交换**

function swap（a,b）{

b = b - a;

a = a + b;

b = a - b;

return [a,b];

}

**2、清除字符串前后的空格（兼容所有浏览器）**

function trim(str) {

if (str & typeof str === "string") {

return str.replace(/(^s\*)|(s\*)$/g, ''); //去除前后空白符

}

}

**3、去掉一组整型数组中重复的值**

let unique = function(arr){

let hash={};

let data=[];

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

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

hash[arr[i]] = true;

data.push(arr[i]);

}

}

return data

}

### 4、翻转字符串

function reverseString(str){

return str.split('').reverse().join('');

}

### 5、验证一个数是否是素数

function isPrime(num){

　　if (num === 2 || num === 3) {

　　return true;

　　};

　　if (num % 2 === 0) {

　　return false;

　　};

　　let divisor = 3,limit = Math.sqrt(num);

　　while(limit >= divisor){

　　if (num % divisor === 0) {

　　return false;

　　}

　　else {

　　divisor += 2;

　　}

　　}

　　return true;

　　}

### 6、选择排序实现

function selectionSort(arr) {

var len = arr.length;

var minIndex, temp;

for (var i = 0; i < len - 1; i++) {

minIndex = i;

for (var j = i + 1; j < len; j++) {

if (arr[j] < arr[minIndex]) { // 寻找最小的数

minIndex = j; // 将最小数的索引保存

}

}

temp = arr[i];

arr[i] = arr[minIndex];

arr[minIndex] = temp;

}

return arr;

}

### 7、插入排序

function insertionSort(arr) {

var len = arr.length;

var preIndex, current;

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

preIndex = i - 1;

current = arr[i];

while(preIndex >= 0 && arr[preIndex] > current) {

arr[preIndex+1] = arr[preIndex];

preIndex--;

}

arr[preIndex+1] = current;

}

return arr;

}

### 8、希尔排序

function shellSort(arr) {

var len = arr.length,

temp,

gap = 1;

while(gap < len/3) { //动态定义间隔序列

gap =gap\*3+1;

}

for (gap; gap > 0; gap = Math.floor(gap/3)) {

for (var i = gap; i < len; i++) {

temp = arr[i];

for (var j = i-gap; j >= 0 && arr[j] > temp; j-=gap) {

arr[j+gap] = arr[j];

}

arr[j+gap] = temp;

}

}

return arr;

}

### 9、归并排序

function mergeSort(arr) { // 采用自上而下的递归方法

var len = arr.length;

if(len < 2) {

return arr;

}

var middle = Math.floor(len / 2),

left = arr.slice(0, middle),

right = arr.slice(middle);

return merge(mergeSort(left), mergeSort(right));

}

function merge(left, right)

{

var result = [];

while (left.length && right.length) {

if (left[0] <= right[0]) {

result.push(left.shift());

} else {

result.push(right.shift());

}

}

while (left.length)

result.push(left.shift());

while (right.length)

result.push(right.shift());

return result;

}

### 10、快速排序

function quickSort(arr, left, right) {

var len = arr.length,

partitionIndex,

left = typeof left != 'number' ? 0 : left,

right = typeof right != 'number' ? len - 1 : right;

if (left < right) {

partitionIndex = partition(arr, left, right);

quickSort(arr, left, partitionIndex-1);

quickSort(arr, partitionIndex+1, right);

}

return arr;

}

function partition(arr, left ,right) { // 分区操作

var pivot = left, // 设定基准值（pivot）

index = pivot + 1;

for (var i = index; i <= right; i++) {

if (arr[i] < arr[pivot]) {

swap(arr, i, index);

index++;

}

}

swap(arr, pivot, index - 1);

return index-1;

}

function swap(arr, i, j) {

var temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

functiion paritition2(arr, low, high) {

let pivot = arr[low];

while (low < high) {

while (low < high && arr[high] > pivot) {

--high;

}

arr[low] = arr[high];

while (low < high && arr[low] <= pivot) {

++low;

}

arr[high] = arr[low];

}

arr[low] = pivot;

return low;

}

function quickSort2(arr, low, high) {

if (low < high) {

let pivot = paritition2(arr, low, high);

quickSort2(arr, low, pivot - 1);

quickSort2(arr, pivot + 1, high);

}

return arr;

### }

### 11.连字符转成驼峰

### 12.加油站问题-贪心算法

### 13.用正则实现trim() 清除字符串两端空格

### 14.岛问题：判断有几个岛

### 15.将数字12345678转化成RMB形式：12,345,678

### 16.删除相邻相同的字符串

### 17.宣讲会安排

### 18.汉诺塔问题

### 19.母牛生母牛问题

### 20、阐述下 JavaScript 中的变量提升

### 21、Doctype作用？严格模式与混杂模式如何区分？它们有何意义？

### 22、行内元素有哪些？块级元素有哪些？空(void)元素有那些？

### 23、页面导入样式时，使用link和@import有什么区别？

### 24、介绍一下你对浏览器内核的理解？

### 25、常见的浏览器内核有哪些？

### 26、html5有哪些新特性、移除了那些元素？如何处理HTML5新标签的浏览器兼容问题？

### 27、如何区分HTML和HTML5？

### 28、简述一下你对HTML语义化的理解？

### 29、HTML5的离线储存怎么使用，工作原理能不能解释一下？

### 30、常见排序算法的时间复杂度,空间复杂度

### 31、[前端需要注意哪些SEO](http://yehe.isd.com/column/support-plan/article-edit/5705766" \l "%E5%89%8D%E7%AB%AF%E9%9C%80%E8%A6%81%E6%B3%A8%E6%84%8F%E5%93%AA%E4%BA%9Bseo" \t "_blank)

### 32、[web开发中会话跟踪的方法有哪些](http://yehe.isd.com/column/support-plan/article-edit/5705766" \l "web%E5%BC%80%E5%8F%91%E4%B8%AD%E4%BC%9A%E8%AF%9D%E8%B7%9F%E8%B8%AA%E7%9A%84%E6%96%B9%E6%B3%95%E6%9C%89%E5%93%AA%E4%BA%9B" \t "_blank)

### 33、<img>的title和alt有什么区别

### 34、doctype是什么,举例常见doctype及特点

### 35、HTML全局属性(global attribute)有哪些

### 36、什么是web语义化,有什么好处

### 37、HTTP method

### 38、从浏览器地址栏输入url到显示页面的步骤(以HTTP为例)

### 39、HTTP request报文结构是怎样的

### 40、[HTTP response报文结构是怎样的](http://yehe.isd.com/column/support-plan/article-edit/5705766" \l "http-response%E6%8A%A5%E6%96%87%E7%BB%93%E6%9E%84%E6%98%AF%E6%80%8E%E6%A0%B7%E7%9A%84" \t "_blank)

### 41、如何进行网站性能优化

### 42、什么是渐进增强

### 43、HTTP状态码及其含义

### [CSS部分](http://yehe.isd.com/column/support-plan/article-edit/5705766#$css%E9%83%A8%E5%88%86)

### 44、css sprite是什么,有什么优缺点

### 45、display: none;与visibility: hidden;的区别

### 46、css hack原理及常用hack

### 47、specified value,computed value,used value计算方法

### 48、display: block;和display: inline;的区别

### 49、[PNG,GIF,JPG的区别及如何选](http://yehe.isd.com/column/support-plan/article-edit/5705766" \l "pnggifjpg%E7%9A%84%E5%8C%BA%E5%88%AB%E5%8F%8A%E5%A6%82%E4%BD%95%E9%80%89" \t "_blank)

### 50、CSS有哪些继承属性

### 51、IE6浏览器有哪些常见的bug,缺陷或者与标准不一致的地方,如何解决

### 52、容器包含若干浮动元素时如何清理(包含)浮动

### 53、[什么是FOUC?如何避免](http://yehe.isd.com/column/support-plan/article-edit/5705766" \l "%E4%BB%80%E4%B9%88%E6%98%AFfouc%E5%A6%82%E4%BD%95%E9%81%BF%E5%85%8D" \t "_blank)

### 54、如何创建块级格式化上下文(block formatting context),BFC有什么用

### 55、display,float,position的关系

### 56、外边距折叠(collapsing margins)

### 57、如何确定一个元素的包含块(containing block)

### 58、stacking context,布局规则

### 59、如何水平居中一个元素

### 60、如何竖直居中一个元素

### javascript概念部分

### 61、DOM元素e的e.getAttribute(propName)和e.propName有什么区别和联系

### 62、offsetWidth/offsetHeight,clientWidth/clientHeight与scrollWidth/scrollHeight的区别

### 63、XMLHttpRequest通用属性和方法

### 64、focus/blur与focusin/focusout的区别与联系

### 65、mouseover/mouseout与mouseenter/mouseleave的区别与联系

### 66、sessionStorage,localStorage,cookie区别

### 67、javascript跨域通信

### 68、什么闭包,闭包有什么用

### 69、javascript有哪几种方法定义函数

### 70、应用程序存储和离线web应用

### 71、客户端存储localStorage和sessionStorage

### 72、cookie及其操作

### 73、javascript有哪些方法定义对象

### 74、[===运算符判断相等的流程是怎样的](http://yehe.isd.com/column/support-plan/article-edit/5705766" \l "===%E8%BF%90%E7%AE%97%E7%AC%A6%E5%88%A4%E6%96%AD%E7%9B%B8%E7%AD%89%E7%9A%84%E6%B5%81%E7%A8%8B%E6%98%AF%E6%80%8E%E6%A0%B7%E7%9A%84" \t "_blank)

### 75、==运算符判断相等的流程是怎样的

### 76、对象到字符串的转换步骤

### 77、对象到数字的转换步骤

### 78、<,>,<=,>=的比较规则

### 79、函数内部arguments变量有哪些特性,有哪些属性,如何将它转换为数组

### 80、DOM事件模型是如何的,编写一个EventUtil工具类实现事件管理兼容

### 81、评价一下三种方法实现继承的优缺点,并改进

### [javascript编程部分](http://yehe.isd.com/column/support-plan/article-edit/5705766#$javascript%E7%BC%96%E7%A8%8B%E9%83%A8%E5%88%86)

### 82、请用原生js实现一个函数,给页面制定的任意一个元素添加一个透明遮罩(透明度可变,默认0.2),使这个区域点击无效,要求兼容IE8+及各主流浏览器,遮罩层效果如下图所示:

### 83、请用代码写出(今天是星期x)其中x表示当天是星期几,如果当天是星期一,输出应该是"今天是星期一"

### 84、[下面这段代码想要循环延时输出结果0 1 2 3 4,请问输出结果是否正确,如果不正确,请说明为什么,并修改循环内的代码使其输出正确结果](http://yehe.isd.com/column/support-plan/article-edit/5705766" \l "%E4%B8%8B%E9%9D%A2%E8%BF%99%E6%AE%B5%E4%BB%A3%E7%A0%81%E6%83%B3%E8%A6%81%E5%BE%AA%E7%8E%AF%E5%BB%B6%E6%97%B6%E8%BE%93%E5%87%BA%E7%BB%93%E6%9E%9C0-1-2-3-4%E8%AF%B7%E9%97%AE%E8%BE%93%E5%87%BA%E7%BB%93%E6%9E%9C%E6%98%AF%E5%90%A6%E6%AD%A3%E7%A1%AE%E5%A6%82%E6%9E%9C%E4%B8%8D%E6%AD%A3%E7%A1%AE%E8%AF%B7%E8%AF%B4%E6%98%8E%E4%B8%BA%E4%BB%80%E4%B9%88%E5%B9%B6%E4%BF%AE%E6%94%B9%E5%BE%AA%E7%8E%AF%E5%86%85%E7%9A%84%E4%BB%A3%E7%A0%81%E4%BD%BF%E5%85%B6%E8%BE%93%E5%87%BA%E6%AD%A3%E7%A1%AE%E7%BB%93%E6%9E%9C" \t "_blank)

### 85、现有一个Page类,其原型对象上有许多以post开头的方法(如postMsg);另有一拦截函数chekc,只返回ture或false.请设计一个函数,该函数应批量改造原Page的postXXX方法,在保留其原有功能的同时,为每个postXXX方法增加拦截验证功能,当chekc返回true时继续执行原postXXX方法,返回false时不再执行原postXXX方法

### 86、编写javascript深度克隆函数deepClone

### 87、补充代码,鼠标单击Button1后将Button1移动到Button2的后面

### 88、网页中实现一个计算当年还剩多少时间的倒数计时程序,要求网页上实时动态显示"××年还剩××天××时××分××秒"

### 89、完成一个函数,接受数组作为参数,数组元素为整数或者数组,数组元素包含整数或数组,函数返回扁平化后的数组

### 90、如何判断一个对象是否为数组

### 91、如何判断一个对象是否为函数

### 92、编写一个函数接受url中query string为参数,返回解析后的Object,query string使用application/x-www-form-urlencoded编码

### 93、解析一个完整的url,返回Object包含域与window.location相同

### 94、[完成函数getViewportSize返回指定窗口的视口尺寸](http://yehe.isd.com/column/support-plan/article-edit/5705766" \l "%E5%AE%8C%E6%88%90%E5%87%BD%E6%95%B0getviewportsize%E8%BF%94%E5%9B%9E%E6%8C%87%E5%AE%9A%E7%AA%97%E5%8F%A3%E7%9A%84%E8%A7%86%E5%8F%A3%E5%B0%BA%E5%AF%B8" \t "_blank)

### 95、现有一个字符串richText,是一段富文本,需要显示在页面上.有个要求,需要给其中只包含一个img元素的p标签增加一个叫pic的class.请编写代码实现.可以使用jQuery或KISSY.

### 96、请实现一个Event类,继承自此类的对象都会拥有两个方法on,off,once和trigger

### 97、编写一个函数将列表子元素顺序反转

### 98、编写一个函数实现form的序列化(即将一个表单中的键值序列化为可提交的字符串)

### 99、使用原生javascript给下面列表中的li节点绑定点击事件,点击时创建一个Object对象,兼容IE和标准浏览器

### 100、有一个大数组,var a = ['1', '2', '3', ...];a的长度是100,内容填充随机整数的字符串.请先构造此数组a,然后设计一个算法将其内容去重