### 单选题

1. JavaScript的对象有三大类，内部对象（本地对象和内置对象）、宿主对象和自定义对象，则以下选项中，属于内置对象的是（ ）

A. DOM

B. Global

C. Function

D. Object

E. Error

答案：B

知识点：JavaScript；对象分类

难度：1

1. 如下JavaScript代码运行后的结果正确的是（ ）

var a = ["monkey", "elephant", "horse"];

var b = a;

var c = a.slice();

a.push("panda");"

A. b = ["monkey", "elephant", "horse", "panda"], c = ["monkey", "elephant", "panda", "horse"];

B. b = ["monkey", "elephant", "horse", "panda"], c = ["monkey", "elephant", "horse"];

C. b = ["monkey", "elephant", "horse"], c = ["monkey", "elephant", "horse"];

D. b = ["monkey", "elephant", "horse"], c = ["monkey", "elephant", "panda", "horse"];

答案：B

知识点：JavaScript；数组

难度：1

1. 下列代码的实际执行顺序正确的是（ ）

console.log(f1());

console.log(f2);

function f1(){console.log('aa')};

var f2 = function() {};

1. 执行顺序与代码顺序一致

function f1(){console.log('aa')};

var f2 = function() {};

console.log(f1());

console.log(f2);

var f2 = function() {};

function f1(){console.log('aa')};

console.log(f1());

console.log(f2);

function f1(){console.log('aa')};

var f2;

console.log(f1());

console.log(f2);

f2 = function() {};

var f2;

function f1(){console.log('aa')};

console.log(f1());

console.log(f2);

f2 = function() {};

答案：D

知识点：JavaScript；函数；变量提升；函数提升

难度：2

1. 下列选项中，能够在代码层面实现性能优化的是（ ）

A. 用DOM操作替代innerHTML

B. 多用全局变量

C. 多用float定位

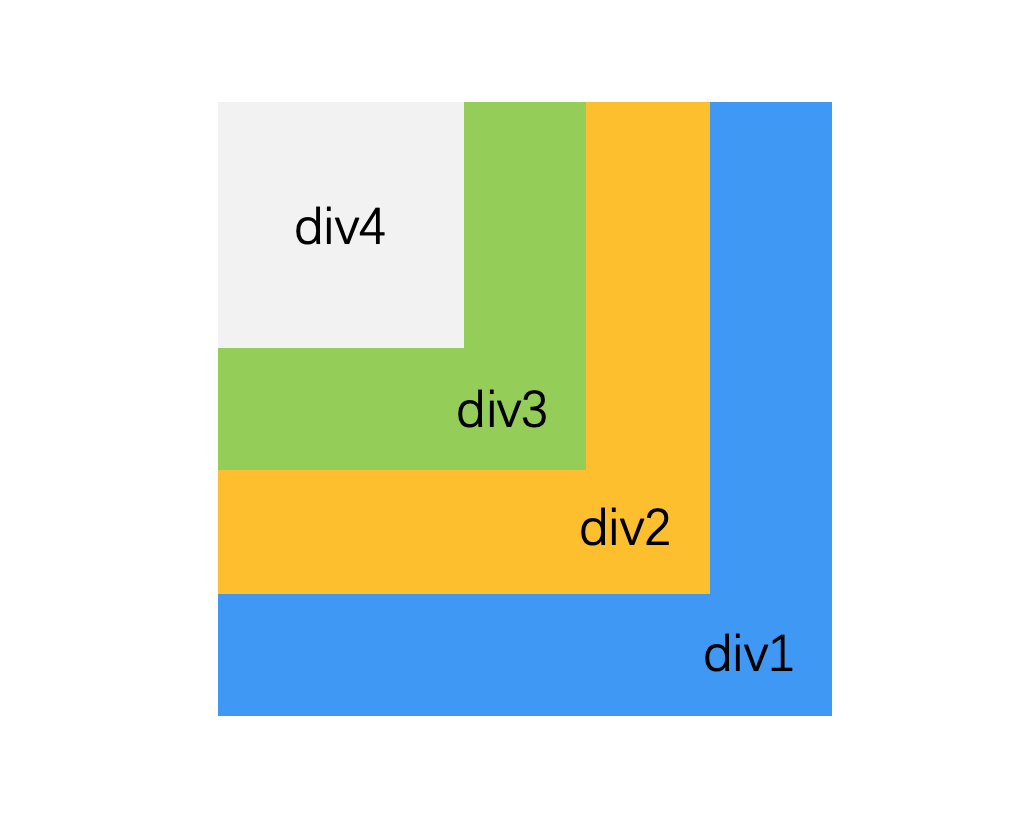
D. 缓存DOM节点查找的结果

答案：D

知识点：性能优化

难度：2

1. 4个div标签如图所示，执行下列代码为四个标签添加点击事件，则点击div4时，弹出数字顺序正确的是（ ）



div1.addEventListener("click",function(){alert("1");},false);

div4.addEventListener("click",function(){alert("4");},false);

div2.addEventListener("click",function(){alert("2");},true);

div3.addEventListener("click",function(){alert("3");},false);

div4.addEventListener("click",function(){alert("4-2");},true);

1. 2 4 4-2 3 1
2. 2 4-2 4 3 1
3. 2 4-2 1 4 3
4. 2 4-2 3 4 1
5. 1 4 3 4-2 2
6. 1 3 4 4-2 2

答案：A

知识点：事件捕获；事件冒泡

难度：3

### 二、判断题

1. 可替换元素的内容不受当前文档的样式的影响，但其本身具有的尺寸和基线（baseline）会被一些 CSS 属性用到，加入计算之中，例如 vertical-align。（ X ）

知识点：HTML；可替换元素

难度：2

1. 箭头函数中的this与箭头函数所在作用域中的this相同。（ √ ）

知识点：箭头函数；作用域；this

难度：2

1. cookies，sessionStorage和localStorage都有存储大小的限制，且localStorage和sessionStorage的存储大小大于cookies。（ √ ）

知识点：数据存储

难度：2

1. vue的热更新可以实现局部内容的刷新，更改数据后，不会刷新整个页面。（ √ ）

知识点：vue；热更新

难度：1

1. ref不是响应式，应该避免做数据绑定。（ √ ）

知识点：Vue；ref引用属性；数据绑定

难度：2

### 三、编程题

1. 表格排序

知识点：冒泡排序；数组；call；sort；数据绑定；DOM映射

难度：3

<!doctype html>

<html lang="en">

<head>

<meta charset="UTF-8">

<link rel="stylesheet" href="index.css">

</head>

<body>

<table cellspacing="0" cellpadding="0" id="tab">

<caption align="top">王者荣耀英雄数据列表</caption>

<thead>

<tr>

<th class="sort zh">姓名</th>

<th class="sort zh">性别</th>

<th class="sort">最大生命</th>

<th class="sort">最大法力</th>

<th class="sort">物理攻击</th>

<th class="sort">身价</th>

</tr>

</thead>

<tbody>

</tbody>

</table>

<script>

//获取表格内元素

var table = document.getElementById('tab');

var thead = table.tHead;

var headTr = thead.rows[0];

var thArray = headTr.cells;

var tbody = table.tBodies[0];

var bodyTrArray = tbody.rows;

//数据源

var dataSource = [

{"name":"马超", "gender":"男", "HP":3307, "MP":0, "pyhsical-attack":168, "price":18888},

{"name":"刘禅", "gender":"男", "HP":3364, "MP":420, "pyhsical-attack":178, "price":2888},

{"name":"公孙离", "gender":"女", "HP":3050, "MP":440, "pyhsical-attack":173, "price":13888},

{"name":"张良", "gender":"男", "HP":3027, "MP":490, "pyhsical-attack":170, "price":5888},

{"name":"露娜", "gender":"女", "HP":3021, "MP":450, "pyhsical-attack":163, "price":8888}

]

//绑定数据

var fragment = document.createDocumentFragment();

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

var tr = document.createElement("tr");

for (key in dataSource[i]) {

var td = document.createElement("td");

td.innerHTML = dataSource[i][key];

tr.appendChild(td);

}

fragment.appendChild(tr);

}

tbody.appendChild(fragment);

fragment = null;

//添加点击事件

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

let th = thArray[i];

th.sortFlag = -1; //用于对列进行升降序排列（1：代表升序，-1：代表降序）

th.index = i; //记录当前点击列的索引

th.classArray = listToArray(th.classList);

//No.1

//开始写代码，为className为sort的列添加点击事件，点击后对数据进行排序

if (th.classArray.indexOf('sort') > -1) {

th.onclick = function() {

sort.call(this);

}

}

//end\_code

}

//排序

function sort() {

let target = this;

this.sortFlag \*= -1;

let dataArray = listToArray(bodyTrArray);

let targetTh = thArray[target.index]

//No.2

//开始写代码，按要求完善排序函数，并把拍好的数组重新写入页面

//要求1：使用数组sort()方法对数据进行排序

/\*数组排序法

dataArray = dataArray.sort(function(a, b) {

let textA = a.cells[target.index].innerHTML;

let textB = b.cells[target.index].innerHTML;

if (targetTh.classArray.indexOf('zh') > -1) {

return textA.localeCompare(textB) \* target.sortFlag;

}else{

return (parseFloat(textA) - parseFloat(textB)) \* target.sortFlag;

}

});

\*/

//要求2：列表中的文字使用数组sort()方法完成排序；数字使用冒泡排序完成排序

if (targetTh.classArray.indexOf('zh') > -1) {

dataArray = dataArray.sort(function(a, b) {

let textA = a.cells[target.index].innerHTML;

let textB = b.cells[target.index].innerHTML;

return textA.localeCompare(textB) \* target.sortFlag;

});

}else{

dataArray = bubbleSort(dataArray, target)

}

let fragment = document.createDocumentFragment();

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

fragment.appendChild(dataArray[i]);

}

tbody.appendChild(fragment);

fragment = null;

//end\_code

//点击某列时，要将其他列的排序标志恢复为-1，让下次再点击任意一个标签时都是默认是升序排列

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

if (thArray[i] != this) {

thArray[i].sortFlag = -1;

}

}

}

function bubbleSort(array, target) {

let low = 0;

let high = array.length - 1;

//No.3

//开始写代码，使用冒泡排序使数据升序排列；并根据target中的排序标记，适时颠倒数组中的元素顺序完成代码

while (low < high) {

for (let j = low; j < high; j++) {

let text1 = array[j].cells[target.index].innerHTML;

let text2 = array[j+1].cells[target.index].innerHTML;

if (parseFloat(text1) > parseFloat(text2)) {

const temp = array[j];

array[j] = array[j + 1];

array[j + 1] = temp;

}

}

--high;

}

if (target.sortFlag == -1) {

array.reverse()

}

//end\_code

return array;

}

//转数组

function listToArray(likeAry) {

var dataArray = [];

try {

dataArray = Array.prototype.slice.call(likeAry);

} catch (e) {

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

dataArray[i] = likeAry[i];

}

}

return dataArray;

}

</script>

</body>

</html>