

珠峰培训就业班-第一次月考题目

- 每一页答题纸上写上自己的：姓名、小组、电话
- 每一页答题纸上标清：页码、题目序号（无需抄题）
- 认真作答，把字写清楚，看不清按错误处理
- 不能抄袭，一旦发现，开除学籍，学费不退

一、问答与计算题

01~10题，每题2分

11~20题，每题3分

简答，回答到重点上即可；如遇代码报错，可继续执行下面代码；遇到无输出结果或者报错的要标明；

1. JS中的数据类型都有哪些？它们之间有什么区别？该如何检测？
2. 常用浏览器的内核都有哪些？
3. 数组中常用的迭代方法有哪些？都是什么意思？（至少四种）

```
1 //=>迭代方法：指的是能够对数组中的每一项进行遍历，例如forEach
2 arr.forEach((item, index)=>{
3     ...
4 });
```

4. 说一下你对闭包的理解，以及工作中什么地方你用到了闭包？
5. 阐述一下let/var/const三者之间的区别？
6. 阐述一下call/apply/bind三者之间的区别，以及应用场景？
7. 怎么让一个 div 水平垂直居中？（不少于三种解决方案）
8. 写出下面代码的输出结果

```
1 let result = 10+false+undefined+[]+'北京珠峰培训'+null+true+{};
2 console.log(result);
```

9. 写出下面代码的输出结果

```
1 let res = parseFloat('left:200px');
2 if(res===200){
3     alert(200);
4 }else if(res===NaN){
5     alert(NaN);
6 }else if(typeof res==='number'){
7     alert('number');
8 }else{
9     alert('Invalid Number');
10 }
```

10. 写出下面代码输出结果

```
1 var a = 10,
2     b = 20;
3 function func() {
4     console.log(a, b);
5     var a = b = 13;
6     console.log(a, b);
7 }
8 a = func();
9 console.log(a, b);
```

11. 写出下面代码输出的结果

```
1 var x = 4;
2 function func() {
3     return function(y) {
4         console.log(y + (--x));
5     }
6 }
7 var f = func(5);
8 f(6);
9 func(7)(8);
10 f(9);
11 console.log(x);
```

12. 写出下面代码输出结果

```

1  var x=5,
2      y=6;
3  function func(){
4      x+=y;
5      func=function(y){
6          console.log(y + (--x));
7      };
8      console.log(x, y);
9  }
10 func(4);
11 func(3);
12 console.log(x, y);

```

13. 写出下面代码输出结果

```

1  let obj = {
2      0: 3,
3      1: 4,
4      length: 0,
5      push: Array.prototype.push
6  }
7  obj.push(10);
8  obj.push(20);
9  console.log(obj);

```

14. 写出下面代码输出结果

```

1  var x = 3,
2      obj = {x: 5};
3  obj.fn = (function () {
4      this.x *= ++x;
5      return function (y) {
6          this.x *= (++x)+y;
7          console.log(x);
8      }
9  })();
10 var fn = obj.fn;
11 obj.fn(6);
12 fn(4);
13 console.log(obj.x, x);

```

15. 写出下面代码的运行结果

```

1  //example 1

```

```

2  let a={}, b='0', c=0;
3  a[b]='珠峰';
4  a[c]='培训';
5  console.log(a[b]);
6
7  -----
8  //example 2
9  let a={}, b=Symbol('1'), c=Symbol('1');
10 a[b]='珠峰';
11 a[c]='培训';
12 console.log(a[b]);
13
14 -----
15 //example 3
16 let a={}, b={n:'1'}, c={m:'2'};
17 a[b]='珠峰';
18 a[c]='培训';
19 console.log(a[b]);

```

16. 写出下面代码的运行结果

```

1  function Fn() {
2      this.x = 10;
3      this.y = 20;
4      this.getX = function () {
5          console.log(this.x);
6      }
7  }
8  Fn.prototype.y=40;
9  Fn.prototype={
10     x:30,
11     getY:function(){
12         this.y+=1;
13         console.log(this.y);
14     },
15     sum:function(){
16         console.log(this.x+this.y);
17     }
18 };
19 let f1 = new Fn;
20 let f2 = new Fn;
21 console.log(f1.getX === f2.getX);
22 console.log(f1.getY === f2.getY);
23 console.log(f1.constructor);
24 f1.getX();

```

```
25 f2.getY();
26 f1.sum();
```

17. 写出下面代码的运行结果

```
1  var name = '珠峰培训';
2  function A(x,y){
3      var res=x+y;
4      console.log(res,this.name);
5  }
6  function B(x,y){
7      var res=x-y;
8      console.log(res,this.name);
9  }
10 B.call(A,40,30);
11 B.call.call.call(A,20,10);
12 Function.prototype.call(A,60,50);
13 Function.prototype.call.call.call(A,80,70);
```

18. 写出下面代码的运行结果

```
1  function A(){
2      alert(1);
3  }
4  function Func() {
5      A=function(){
6          alert(2);
7      };
8      return this;
9  }
10 Func.A=A;
11 Func.prototype={
12     A:()=>{
13         alert(3);
14     }
15 };
16 A();
17 Fn.A();
18 Fn().A();
19 new Fn.A();
20 new Fn().A();
21 new new Fn().A();
```

19. 写出下面代码的运行结果

```

1  var name = "WINDOW",
2      obj = {
3          name: "OBJ",
4          fn: (()=>{
5              console.log(this.name);
6              return function(){
7                  console.log(this.name);
8              }
9          })()
10     },
11     fn=()=>{
12         console.log(this.name);
13     };
14 fn();
15 obj.fn();
16 fn.call(obj);

```

20. 写出下面代码的运行结果

```

1  function A1(name) {
2      this.name = name;
3  }
4  function A2(name) {
5      this.name = name || 'Jerry';
6  }
7  function A3(name) {
8      name && this.name = name;
9  }
10 A1.prototype.name = 'Tom';
11 A2.prototype.name = 'Tom';
12 A3.prototype.name = 'Tom';
13 alert((new A1().name) + (new A2().name) + (new A3().name));

```

二、编程题

21~30题，每题5分

每一种实现方案前都要标清楚是第几种，代码要书写正确和清晰（代码书写有误扣一分）

21. 下面代码能否实现预期效果？如不能请写出正确答案！

预期效果：点击每一个按钮让#box的背景颜色改变为数组中对应的某种颜色

要求解决办法不得低于三种，其中有一种必须是用JQUERY代码重新实现这个效果

```
1 <body>
2   <div id='box' style='background:red'></div>
3   <button>红</button>
4   <button>绿</button>
5   <button>蓝</button>
6 </body>
7 <script>
8   var box=document.querySelector('#box'),
9       btnList=document.querySelectorAll('button'),
10      colorAry=['red','green','blue'];
11   for(var i=0;i<btnList.length;i++){
12       btnList[i].onclick=function(){
13           box.style.background=colorAry[i];
14       }
15   }
16 </script>
```

22. 完成如下需求

```
1  /*
2   * 1.编写一个方法“flatten”，将数组扁平化 （至少两种办法）
3   * 2.然后实现“unique”数组去重方法，把数组进行去重 （至少两种办法）
4   */
5  let arr = [[1, 2, 2], [3, 4, 5, 5], [6, 7, 8, 9, [11, 12, [12,
6    13, [14]]]], 10];
7  ary.flatten().unique().sort((a,b)=>a-b); //=>[1, 2, 3, 4, 5,
8    6, 7, 8, 9....]
```

23. 完成如下需求

```
1  function Dog(name) {
2      this.name = name;
3  }
4  Dog.prototype={
5      bark(){
6          console.log('wangwang');
7      },
8      sayName(){
9          console.log('my name is ' + this.name);
10     }
11 };
12 function _new() {
```

```
13      //=>完成你的代码
14
15  }
16  let sanmao = _new(Dog, '三毛');
17  sanmao.bark(); //=>"wangwang"
18  sanmao.sayName(); //=>"my name is 三毛"
19  console.log(sanmao instanceof Dog); //=>true
```

24. 完成如下需求

```
1  /* 实现一个$attr(name,value)遍历
2     * 属性为name
3     * 值为value的元素集合
4     * 例如下面示例：
5     */
6  let ary = $attr('class','box'); //=>获取页面中所有class为box的元素
```

25. 完成如下需求

```
1  ~function(){
2      function change(){
3          //=>实现你的代码
4      };
5      Function.prototype.change=change;
6  }();
7  let obj = {name:'zhufeng'};
8  function func(x,y){
9      this.total=x+y;
10     return this;
11 }
12 let res = func.change(obj,100,200);
13 //res => {name:'Alibaba',total:300}
```

26. 完成如下需求


```

1 ~function(){
2     //=>bind方法在IE6~8中不兼容，接下来我们自己基于原生JS实现这个方法
3     function bind(){
4
5     };
6     Function.prototype.bind=bind;
7 }();
8 var obj = {name:'zhufeng'};
9 function func(){
10     console.log(this,arguments);
11     //=>当点击BODY的时候，执行func方法，输出: obj
12     [100,200,MouseEvent事件对象]
13 }
14 document.body.onclick = func.bind(obj,100,200);

```

27. 完成如下需求

```

1 <body>
2     <div class="imgBox">
3         <img src="" data-img="img/1.jpg">
4     </div>
5     <div class="imgBox">
6         <img src="" data-img="img/1.jpg">
7     </div>
8     ...
9 </body>
10 <!-- IMPORT JS -->
11 <script src="js/jquery.min.js"></script>
12 <script>
13     /*
14         * 已知页面中有很多.imgBox盒子，盒子中的图片在开始加载页面的时候都
15         * 没有加载，现在期望你写一个程序实现如下的需求：
16         * 1. 第一次加载完成代码后，等待500ms开始加载真实图片
17         * 2. 页面滚动中把符合条件的图片也加载出来
18         * 3. 图片加载的条件是：当前图片所在的盒子.imgBox，半个盒子高度出
19         * 现在屏幕中即可
20         * 可使用JQ类库完成
21     */
22 </script>

```

28. 实现工具类方法toArray函数（至少两种解决方案）

```

1 let utils = (function(){
2     /*

```

```

3      * toArray: 转换为数组的方法
4      *   @params
5      *       不固定数量, 不固定类型
6      *   @return
7      *       [Array] 返回的处理后的新数组
8      * by zhufengpeixun on 2019/08/25
9      */
10     function toArray(){
11         //=>实现你的代码
12     }
13
14     return {
15         toArray
16     };
17 })();
18 let ary = utils.toArray(10,20,30); //=>[10,20,30]
19 ary = utils.toArray('A',10,20,30); //=>['A',10,20,30]

```

29. 完成如下需求

```

1  /*
2   * 编写queryURLParams方法实现如下的效果
3   */
4  let url="http://www.zhufengpeixun.cn/?lx=1&from=wx#video";
5  console.log(url.queryURLParams("from")); //=>"wx"
6  console.log(url.queryURLParams("_HASH")); //=>"video"

```

30. 基于ES6中的class重构下面的代码

```

1  function Modal(x,y){
2      this.x=x;
3      this.y=y;
4  }
5  Modal.prototype.z=10;
6  Modal.prototype.getX=function(){
7      console.log(this.x);
8  }
9  Modal.prototype.getY=function(){
10     console.log(this.y);
11 }
12 Modal.n=200;
13 Modal.setNumber=function(n){
14     this.n=n;
15 };
16 let m = new Modal(10,20);

```

