

前端笔试题

1. 如右图所示，屏幕正中间有个元素A

随着屏幕宽度的增加，始终需要满足以下条件：

- A元素垂直居中于屏幕中央；
- A元素距离屏幕左右边距各10px；
- A元素里面的文字“A”的font-size:20px；水平垂直居中；
- A元素的高度始终是A元素宽度的50%；(如果搞不定可以实现为A元素的高度固定为200px;)

请用 html及css 实现

2. 函数中的 arguments 是数组吗？若不是，如何将它转化为真正的数组

3. 请说出以下代码打印的结果

```
if ({} == false) {console.log(1);};  
if ({} == false ) {console.log(2);};  
if ([]) {console.log(3);};  
if ([1] == [1]) {console.log(4);};
```

4. 请说出以下代码打印的结果

```

async function async1() {
  console.log('async1 start');
  await async2();
  console.log('async1 end');
}
async function async2() {
  console.log('async2');
}

console.log('script start');
setTimeout(function() {
  console.log('setTimeout');
}, 0);
async1();
new Promise(function(resolve) {
  console.log('promise1');
  resolve();
}).then(function() {
  console.log('promise2');
});
console.log('script end');

```

5. 以最小的改动解决以下代码的错误（可以使用es6）

```

const obj = {
  name: "jsCoder",
  skill: ["es6", "react", "angular"],
  say: function() {
    for(var i = 0, len = this.skill.length; i < len; i++){
      setTimeout(function () {
        console.log('No.' + i + this.name);
        console.log(this.skill[i]);
        console.log('-----');
      }, 0);
      console.log(i);
    }
  }
};
obj.say();

```

/*

期望得到下面的结果:

1

2

3

No.1 jsCoder

es6

No.2 jsCoder

react

No.3 jsCoder

angular

*/

6. 实现Function的bind方法，使得以下程序最后能输出'success'

```
function Animal(name, color) {  
  this.name = name;  
  this.color = color;  
}  
Animal.prototype.say = function () {  
  return `I'm a ${this.color} ${this.name}`;  
};  
const Cat = Animal.bind(null, 'cat');  
  
const cat = new Cat('white');  
  
if (cat.say() === 'I'm a white cat' && cat instanceof Cat && cat instanceof Animal){  
  console.log('success');  
}
```

7. 请用JS实现throttle（函数节流）函数

函数节流解释: 对函数执行增加一个控制层，保证一段时间内（可配置）内只执行一次。此函数的作用是对函数执行进行频率控制，常用于用户频繁触但可以以更低频率响应的场景

如上图，在一段时间内函数触发了9次，实际只执行了5次，且每次执行的时间间隔不小于100ms;

请注意函数接口的设计, 尽量合理

8. 请用算法实现，从给定的无序、不重复的数组data中，取出n个数，使其相加和为sum。并给出
算法的时间/空间复杂度。（不需要找到所有的解，找到一个解即可）

```
function getResult(data, n, sum) {  
  
}
```
