task20

问题

1.什么是闭包? 有什么作用

闭包是定义在一个函数内部的函数，这个函数能够读取其他函数的内部变量。其作用有二：
1.读取函数内部的变量；
2.让这些变量的值始终保持在内存中。

关于闭包的谣言

闭包会造成内存泄露？

错。

说这话的人根本不知道什么是内存泄露。内存泄露是指你用不到（访问不到）的变量，依然占居着内存空间，不能被再次利用起来。

闭包里面的变量明明就是我们需要的变量（lives），凭什么说是内存泄露？

这个谣言是如何来的？

因为 IE。IE 有 bug，IE 在我们使用完闭包之后，依然回收不了闭包里面引用的变量。

这是 IE 的问题，不是闭包的问题。参见司徒正美的这篇文章。

一个小经验

编程界崇尚以简洁优雅唯美，很多时候

如果你觉得一个概念很复杂，那么很可能是你理解错了。

作者：方应杭
链接：https://zhuanlan.zhihu.com/p/22486908
来源：知乎

2.setTimeout 0 有什么作用

答：如果设置为`setTimeout 0`，会先执行后面的语句，再执行设置了此语句的代码。

代码题

1.下面的代码输出多少？修改代码让`fnArr[i]()`输出 i。使用两种以上的方法

```
//1
var fnArr = [];
for (var i = 0; i < 10; i ++) {
    fnArr[i] = (function(i){
         function a(){
         return i;
       }
       return a;
   })(i);
}
console.log( fnArr[3]() );

//2
var fnArr = [];
for (var i = 0; i < 10; i ++) {
    (function(){
        var n = i;//为什么这里需要一个中介？因为下面的fnArr是在外面的呀，它不认得i
        fnArr[i] = function(){
            return n;
        };
    })();

}
console.log( fnArr[3]() );
```

2.使用闭包封装一个汽车对象，可以通过如下方式获取汽车状态

```
var Car = (function(){


   var speed=0;
   function setSpeed(n){
    speed = n;
   }
   function accelerate(){
     speed += 10;
   }
   function decelerate(){
     speed -= 10;
   }
   function getSpeed(){
    console.log(speed);
   }

  function getStatus(){
```

```
        if (speed !== 0){
            console.log("running");
        }else {
            console.log("stop");
        }
    }
   return{
        setSpeed: setSpeed,
        getSpeed:getSpeed,
        accelerate:accelerate,
        decelerate:decelerate,
        getStatus:getStatus
  };
  })();
Car.setSpeed(30);
Car.getSpeed(); //30
Car.accelerate();
Car.getSpeed(); //40;
Car.decelerate();
Car.decelerate();
Car.getSpeed(); //20
Car.getStatus(); // 'running';
Car.decelerate();
Car.decelerate();
Car.getStatus();  //'stop';
//Car.speed;  //error
```

3.写一个函数使用setTimeout模拟setInterval的功能

```
var i=0;
function fn(){
  setTimeout(function(){
  console.log(i++);
  fn();
 },1000);
}
fn();
//与真正的setInterval时间间隔并不等同；后者是在时间点自动运行，而前者是执行完后再算时间间隔；
//1-----2-----3-----4(时间点)
//1+++--2+++--3+++--4(setInterval)
//1+++++++2+++++++3+++++++4(setInterval,执行时间超过间隔的情况)
//1++++-----2++++-----3++++-----4(setTimeout)
```

4.写一个函数，计算setTimeout平均[备注：新加]最小时间粒度

```
function getMini(){
  var i=0;
  var start=Date.now();
  var clock=setTimeout(function(){
      i++;
      if(i==1000){
        clearTimeout(clock);
        var end=Date.now();
        console.log((end-start)/i);
      }
      clock=setTimeout(arguments.callee,0);
  },0);
  }
getMini();
```

5.下面这段代码输出结果是? 为什么?

```
var a = 1;
setTimeout(function(){
    a = 2;
    console.log(a);
}, 0);//4.这样设置就是最后一个结算,a=2;
var a ;
console.log(a); //1.先结算这个，此时a=1;
a = 3;  //2.然后a=3;
console.log(a);   //3.a=3;
//最后就是1 3 2
```

6.下面这段代码输出结果是? 为什么?

```
var flag = true;
setTimeout(function(){
    flag = false;
},0);  //这句被放到后面，然后while()永续循环，使得此句永不被结算
while(flag){}  //永远循环
console.log(flag);  //因此一直为true
```

7.下面这段代码输出？如何输出delayer: 0, delayer:1...（使用闭包来实现）

```
for(var i=0;i<5;i++){
  (function(){
    var a=i;
  setTimeout(function(){
```

```
          console.log('delayer:' + a );
    }, 0);
  }());
    console.log(i);
}
```