时间控制

模拟

```
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
//1+++--2+++--3+++--4
//1+++++++2+++++++3+++++++4
//1+++-----2+++-----3+++-----4

//最小时间粒度，标准各不相同；
```

推迟执行

```
var a=1;
setTimeout(function(){
   console.log('before',a);
   a=2;
   console.log('after',a);
},0);  //就被延迟执行了
a=100;
console.log(a);
```

触发a

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>JS Bin</title>
</head>
<body>
 <input type="text" id="ipt">
  <script>
```

```
    var ipt = document.getElementById('ipt');

ipt.addEventListener('keyup', function(){
  this.value = this.value.toUpperCase();
});
  </script>
</body>
</html>
```

触发b

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>JS Bin</title>
</head>
<body>
 <input id="ipt" type="text">
  <script>
    var ipt = document.getElementById('ipt');

ipt.addEventListener('keydown', function(){
    var me = this;
    setTimeout(function(){
        me.value = me.value.toUpperCase();
    }, 0);

});
  </script>
</body>
</html>
```

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();
```