**Basic Async**

* setTimeout(callback,time)
  + 只运行一次，达到触发时间后自动触发callback
* setInterval(callback,time)
  + 按照间隔不间断运行callback
* clearTimeout(timeId)
  + 清除timeout超时运行callback,timeId 由setTimeout返回获得
* clearInterval(timeId)
  + 清除interval超时运行callback,timeId 由setInterval返回获得
* Queue(setTimeout,setInterval) 必需在当前的Stack(普通函数)结束才会再运行
* Promise 约定模式(Promise对象接收两个回调，成功与失败，可以利用这种机制来避开Queue Stack运行先后的问题)
  + var p1 = new Promise(function(resolve,reject){

//成功

Resolve(“data”); 运行p1.then(function(data){})

//失败

Reject(“data”); 运行p1.catch(function(data){})

});

* 使用Promise Chain构建多级Promise(在then中return new Promise再进行控制从而实现多级Promise)

**Ajax**

* setTimeout(callback,time)