

生成器异步方案：

笔记本： 1.有关印象笔记

创建时间： 2020/6/3 14:43

更新时间： 2020/6/5 15:00

作者： 1639079350@qq.com

标签： 4.生成器异步方案

生成器异步方案：

1.完整的生成器函数调用：

(1) 定义（和普通函数的区别：*定义、实例化、函数调用）、生成生成器对象、调用对象的next方法执行生成器函数；

(2) 获取yield的值；（done属性表示生成器是否执行完毕）

(3) 调用next方法传参时，参数会作为yield的返回值；

(4) 生成器的异常处理：先捕获错误try...catch、再抛错throw；

```
// 调用
function * foo() {
  log('start');
}
const generate = foo();
generate.next(); // start

// 获取yield的值
function * foo() {
  yield 'foo';
}
const generate = foo();
let result = generate.next(); // 直接输出generate.next(): value: undefined
log(result); // {value: "foo", done: false}

// next传参
function * foo() {
  const res = yield 'foo';
  log(res);
}
const generate = foo();
const result = generate.next();
log(result); // {value: "foo", done: false}
generate.next('bar'); // bar

// 抛错
function * foo() {
  try {
    const res = yield 'foo';
    log(res);
  } catch (e) {
```

```

        log(e);
    }
}
const generate = foo();
const result = generate.next();
log(result); // {value: "foo", done: false}
generate.throw(new Error('err')); // Error: err

```

2.generate更优的异步编程：= 消除promise的异步，类似同步调用。

(1) 生成器next的方法返回结果 = promise对象，所以可以调用`.value.then()`方法；
【=promise.then()?】，返回done属性；

```

const { log } = console;

// 多异步，先捕获再抛错；
function* main() {
    try {
        const users = yield ajax('url1');
        log(users);

        const posts = yield ajax('url2');
        log(posts);

        const urls = yield ajax('url3');
        log(urls);
    } catch (e) {
        log(e);
    }
}

const g = main();
const result = g.next();

result.value.then(data => {
    g.next(data);
})

```

多异步封装： <https://github.com/tj.co>（社区的库）

```

function co(generator) { // 生成器函数名
    const g = generator();

    function handleResult(result) {
        if(result.done) return; // 生成器函数结束
        result.value.then( data => {

```

```
        handleResult( g.next(data))
      }, error => {
        g.throw(error);
      })
    }
    handleResult( g.next());
  }
}
```

async: generate函数的一种语法糖:

(1) 返回promise对象;

(2) 改生成器函数为普通函数, 去星号加async关键词, yield改为await。(await只出现在async内部;)

```
const { log } = console;
async function main () {
  try {
    const users = await ajax('url');
    const post = await ajax('ur2');
    log(users, post);
  } catch (e) {
    log(e);
  }
}
```