aysnc/await & Object.values

ES8

Async/Await

async 和 await 是 Promise 的拓展，如果对 Promise 还不了解的话，请移步 Promise 章节进行学习。

我们知道 JavaScript 是单线程的，使用 Promise 之后可以让我们书写异步操作更加简单，而 async 是让我们写起 Promise 像同步操作。看下示例

```
async function firstAsync () {
  return 27
}
firstAsync().then(console.log) // 27
```

这块代码成功的输出了 27，仔细观察发现 `firstAsync()` 的返回值竟然有 then 方法，在 JavaScript 世界里只有原生的 Promise 对象拥有 then 方法，来验证下我们的猜测?

```
console.log(firstAsync() instanceof Promise) // true
```

所以，所以，所以想获得一个 Promise 对象，可以不再 `new Promise` 了。

> [!TIP]
> async 函数显式返回的不是 Promise 的话，会自动包装成 Promise 对象

也就是说上面的代码等同于:

```
async function firstAsync () {
  return Promise.resolve(27)
}
```

async 的作用我们清楚了，await呢？话不多说:

```
async function firstAsync () {
  let promise = new Promise((resolve, reject) => {
    setTimeout(() => resolve("Now it's done!"), 1000)
  })

  // wait until the promise returns us a value
  let result = await promise

  // "Now it's done!"
  return result
}

firstAsync().then(console.log)
```

这段代码使用了 await，从这个字面的意思看就是“等待”，它等什么呢？很简单，它等 Promise 返回结果。上面代码的意思是 async 开启了一个 Promise 对象，这个函数内部嵌套了一个 Promise 操作，这个操作需要等 1 秒才返回 “Now it’s done!” 这个结果。也就是说 await 在拿到这个结果前不会继续执行，一直等到结果才往后继续执行，也就是 async function 声明的 Promise 才会响应（console.log才执行）。

思考

1. 聪明的同学一定会反问：await 后面必须是 Promise 对象吗，如果不是会怎样?
2. await 可以脱离 async 单独使用吗?

答案

1. await 后面一定是 Promise 对象，如果不是会自动包装成 Promise 对象