**Promise：Promise.all、Promise. allSettled、Promise.race、Promise.any的用法及区别**

**promise**

**Promise** 对象用于表示一个异步操作的最终完成 (或失败)及其结果值。

一个 **Promise** 必然处于以下几种状态之一：

* *待定（pending）* : 初始状态，既没有被兑现，也没有被拒绝。
* *已兑现（fulfilled）* : 意味着操作成功完成。
* *已拒绝（rejected）* : 意味着操作失败。

[**约定**](https://link.juejin.cn/?target=https%3A%2F%2Fdeveloper.mozilla.org%2Fzh-CN%2Fdocs%2FWeb%2FJavaScript%2FGuide%2FUsing_promises%23%25E7%25BA%25A6%25E5%25AE%259A)

不同于“老式”的传入回调，在使用 Promise 时，会有以下约定：

* 在本轮 [事件循环](https://link.juejin.cn/?target=https%3A%2F%2Fdeveloper.mozilla.org%2Fzh-CN%2Fdocs%2FWeb%2FJavaScript%2FEventLoop%23%25E6%2589%25A7%25E8%25A1%258C%25E8%2587%25B3%25E5%25AE%258C%25E6%2588%2590) 运行完成之前，回调函数是不会被调用的。
* 即使异步操作已经完成（成功或失败），在这之后通过 [**then()**](https://link.juejin.cn/?target=https%3A%2F%2Fdeveloper.mozilla.org%2Fen-US%2Fdocs%2FWeb%2FJavaScript%2FReference%2FGlobal_Objects%2FPromise%2Fthen)添加的回调函数也会被调用。
* 通过多次调用 **then()** 可以添加多个回调函数，它们会按照插入顺序进行执行。

Promise 特点就是**链式调用**（**chaining**）。

本篇重点主要是把用法与区别梳理出来。

**用法**

**Promise.all**

js

复制代码

**let** **wake** = (time) => {

**return** **new** **Promise**(**function** (resolve, reject) {

**resolve**(`${time / 1000}后醒来`);

});

};

**let** p1 = **wake1**(1000);

**let** p2 = **wake**(500);

**Promise**.**all**([p1,p2]).**then**(res=>{

**let** [res1,res2]=res;

console.**log**(res1,'res1');

console.**log**(res2,'res2');

}).**catch**(err=>{

console.**log**(err);

})

语法：

js

复制代码

**Promise**.**all**(iterable);

返回值

* 如果传入的参数是一个空的可迭代对象，则返回一个**已完成（already resolved）** 状态的 **Promise**
* 如果传入的参数不包含任何 **promise**，则返回一个**异步完成（asynchronously resolved）**  **Promise**。注意：Google Chrome 58 在这种情况下返回一个**已完成（already resolved）** 状态的 **Promise**。
* 其它情况下返回一个**处理中（pending）** 的**Promise**。这个返回的 **promise** 之后会在所有的 **promise** 都完成或有一个 **promise** 失败时**异步**地变为完成或失败。 见上方关于“Promise.all 的示例。返回值将会按照参数内的 **promise** 顺序排列，而不是由调用 **promise** 的完成顺序决定。

**Promise.allSettled**

与**promsie.all** 不同的是，**Promise.allSettled**等所有的的promise状态确认，返回一个promise（不管成功还是失败）

ini

复制代码

var resolvedPromisesArray = [Promise.resolve('success'), Promise.reject('error')]*;*

Promise.allSettled(resolvedPromisesArray).then(res => {

console.log(res)*;*

})*;*

返回值

* 一旦所指定的 promises 集合中每一个 promise 已经完成，无论是成功的达成或被拒绝，**未决议的** **Promise**将被**异步**完成。那时，所返回的 promise 的处理器将传入一个数组作为输入，该数组包含原始 promises 集中每个 promise 的结果。
* 对于每个结果对象，都有一个 **status** 字符串。如果它的值为 **fulfilled**，则结果对象上存在一个 **value** 。如果值为 **rejected**，则存在一个 **reason** 。value（或 reason ）反映了每个 promise 决议（或拒绝）的值。

**Promise.any**

与以上两个不同的是： **Promise.any**接受一个对象集合，当其中一个promise成功，就返回成功的那个。本质上这个方法**Promise.all()**是相反的。

**Promise.any()** 方法依然是实验性的，尚未被所有的浏览器完全支持。它当前处于 [TC39 第四阶段草案（Stage 4）](https://link.juejin.cn/?target=https%3A%2F%2Fgithub.com%2Ftc39%2Fproposal-promise-any)

返回值

* 如果传入的参数是一个空的可迭代对象，则返回一个 **已失败（already rejected）** 状态的 **Promise**。
* 如果传入的参数不包含任何 **promise``，则返回**一个 **异步完成** （**asynchronously resolved**）的 **Promise**。
* 其他情况下都会返回一个**处理中（pending）** 的 **Promise**。 只要传入的迭代对象中的任何一个 **promise** 变成成功（resolve）状态，或者其中的所有的 **promises** 都失败，那么返回的 **promise** 就会 **异步地**（当调用栈为空时） \*\*\*\*变成成功/失败（resolved/reject）状态。

**Promise.race**

js

复制代码

**let** **wake1** = (time) => {

**return** **new** **Promise**(**function** (resolve, reject) {

**reject**(`${time / 1000}后失败`);

});

};

**let** **wake2** = (time) => {

**return** **new** **Promise**(**function** (resolve, reject) {

**resolve**(`${time / 1000}后醒来`);

});

};

**let** p1 = **wake1**(500);

**let** p2 = **wake2**(10000);

**Promise**.**race**([p1, p2])

.**then**((res) => {

console.**log**(res);

})

.**catch**((err) => {

console.**log**(err);

});

返回值：

* 只要给定的迭代中的一个promise解决或拒绝，就采用第一个promise的值作为它的值，从而**异步**地解析或拒绝（一旦堆栈为空）。

总结：

* promise.all: 只有参数里面的promise全部成功，才会触发成功（返回结果按照参数顺序），一旦有一个异常，就返回失败
* promise.allSettled: 等待参数里面的promise执行结束，一起返回，不管成功还是失败
* promise.any:谁先成功就先返回
* promise.race:谁先结束就先返回

*[Promise](https://juejin.cn/post/6965596525388890142?from=search-suggest" \t "_blank)*[.](https://juejin.cn/post/6965596525388890142?from=search-suggest" \t "_blank)*[any](https://juejin.cn/post/6965596525388890142?from=search-suggest" \t "_blank)*[的作用，如何自己实现一个](https://juejin.cn/post/6965596525388890142?from=search-suggest" \t "_blank)*[Promise](https://juejin.cn/post/6965596525388890142?from=search-suggest" \t "_blank)*[.](https://juejin.cn/post/6965596525388890142?from=search-suggest" \t "_blank)*[any](https://juejin.cn/post/6965596525388890142?from=search-suggest" \t "_blank)*

*[Promise](https://juejin.cn/post/7194766561956069433?from=search-suggest" \t "_blank)*[API简介 all allSettled](https://juejin.cn/post/7194766561956069433?from=search-suggest" \t "_blank)*[any](https://juejin.cn/post/7194766561956069433?from=search-suggest" \t "_blank)**[race](https://juejin.cn/post/7194766561956069433?from=search-suggest" \t "_blank)*

[手写](https://juejin.cn/post/7147291888423272484?from=search-suggest" \t "_blank)*[Promise](https://juejin.cn/post/7147291888423272484?from=search-suggest" \t "_blank)*[静态方法 all allSettled](https://juejin.cn/post/7147291888423272484?from=search-suggest" \t "_blank)*[race](https://juejin.cn/post/7147291888423272484?from=search-suggest" \t "_blank)**[any](https://juejin.cn/post/7147291888423272484?from=search-suggest" \t "_blank)*

[手写](https://juejin.cn/post/7046285295943155748?from=search-suggest" \t "_blank)*[Promise](https://juejin.cn/post/7046285295943155748?from=search-suggest" \t "_blank)*[中 all、](https://juejin.cn/post/7046285295943155748?from=search-suggest" \t "_blank)*[race](https://juejin.cn/post/7046285295943155748?from=search-suggest" \t "_blank)*[、](https://juejin.cn/post/7046285295943155748?from=search-suggest" \t "_blank)*[any](https://juejin.cn/post/7046285295943155748?from=search-suggest" \t "_blank)*[方法](https://juejin.cn/post/7046285295943155748?from=search-suggest" \t "_blank)

[js -](https://juejin.cn/post/6958416579130294303?from=search-suggest" \t "_blank)*[Promise](https://juejin.cn/post/6958416579130294303?from=search-suggest" \t "_blank)*[用法](https://juejin.cn/post/6958416579130294303?from=search-suggest" \t "_blank)

[搞懂](https://juejin.cn/post/7214164344292048933?from=search-suggest" \t "_blank)*[Promise](https://juejin.cn/post/7214164344292048933?from=search-suggest" \t "_blank)*[中四个静态方法，清晰明了！](https://juejin.cn/post/7214164344292048933?from=search-suggest" \t "_blank)

[同一个套路手撕](https://juejin.cn/post/6948277034304929799?from=search-suggest" \t "_blank)*[Promise](https://juejin.cn/post/6948277034304929799?from=search-suggest" \t "_blank)*[的 all、allSettled、](https://juejin.cn/post/6948277034304929799?from=search-suggest" \t "_blank)*[any](https://juejin.cn/post/6948277034304929799?from=search-suggest" \t "_blank)*[、](https://juejin.cn/post/6948277034304929799?from=search-suggest" \t "_blank)*[race](https://juejin.cn/post/6948277034304929799?from=search-suggest" \t "_blank)*[方法](https://juejin.cn/post/6948277034304929799?from=search-suggest" \t "_blank)

[我把 JS](https://juejin.cn/post/7242493101088555064?from=search-suggest" \t "_blank)*[Promise](https://juejin.cn/post/7242493101088555064?from=search-suggest" \t "_blank)*[翻了个底朝天，以后再也不怕](https://juejin.cn/post/7242493101088555064?from=search-suggest" \t "_blank)*[Promise](https://juejin.cn/post/7242493101088555064?from=search-suggest" \t "_blank)*[面试题了](https://juejin.cn/post/7242493101088555064?from=search-suggest" \t "_blank)

[JS 实现](https://juejin.cn/post/6844904074614161421?from=search-suggest" \t "_blank)*[Promise](https://juejin.cn/post/6844904074614161421?from=search-suggest" \t "_blank)*[、](https://juejin.cn/post/6844904074614161421?from=search-suggest" \t "_blank)*[Promise](https://juejin.cn/post/6844904074614161421?from=search-suggest" \t "_blank)*[.all 和](https://juejin.cn/post/6844904074614161421?from=search-suggest" \t "_blank)*[Promise](https://juejin.cn/post/6844904074614161421?from=search-suggest" \t "_blank)*[.](https://juejin.cn/post/6844904074614161421?from=search-suggest" \t "_blank)*[race](https://juejin.cn/post/6844904074614161421?from=search-suggest" \t "_blank)*

[JS](https://juejin.cn/post/7026358178090582029?from=search-suggest" \t "_blank)*[Promise](https://juejin.cn/post/7026358178090582029?from=search-suggest" \t "_blank)*[扫盲兼快速上手「组合期约」](https://juejin.cn/post/7026358178090582029?from=search-suggest" \t "_blank)

关注