**[Promise.allSettled](https://learn.javascript.ru/promise-api" \l "promise-allsettled)**

Метод Promise.allSettled всегда ждёт завершения всех промисов. В массиве результатов будет

{status:"fulfilled", value:результат} для успешных завершений,

{status:"rejected", reason:ошибка} для ошибок.

[**Promise.race**](https://learn.javascript.ru/promise-api#promise-race)

Метод очень похож на Promise.all, но ждёт только первый *выполненный* промис, из которого берёт результат (или ошибку).

**[Promise.any](https://learn.javascript.ru/promise-api" \l "promise-any)**

Метод очень похож на Promise.race, но ждёт только первый *успешно выполненный* промис, из которого берёт результат.

Если ни один из переданных промисов не завершится успешно, тогда возвращённый объект Promise будет отклонён с помощью AggregateError – специального объекта ошибок, который хранит все ошибки промисов в своём свойстве errors.

**[Promise.resolve/reject](https://learn.javascript.ru/promise-api" \l "promise-resolve-reject)**

Методы Promise.resolve и Promise.reject редко используются в современном коде, так как синтаксис async/await делает их, в общем-то, не нужными.

**[Promise.resolve](https://learn.javascript.ru/promise-api" \l "promise-resolve)**

Promise.resolve(value) создаёт успешно выполненный промис с результатом value.

То же самое, что:

let promise = new Promise(resolve => resolve(value));

**[Promise.reject](https://learn.javascript.ru/promise-api" \l "promise-reject)**

Promise.reject(error) создаёт промис, завершённый с ошибкой error.

То же самое, что:

let promise = new Promise((resolve, reject) => reject(error));