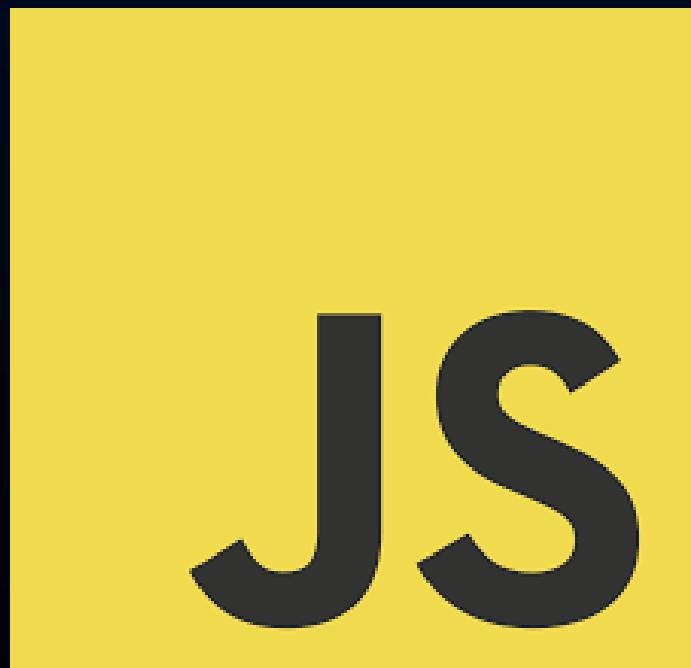




**Soheib Kiani**  
Front-end Developer

# Promise.all

## تابع پیشرفته در جاوا اسکریپت

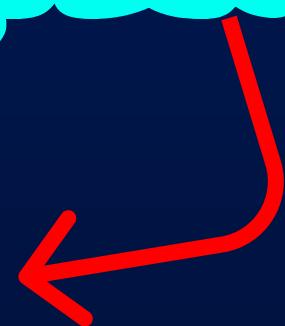




Soheib Kiani  
Front-end Developer

# معنی لغوی کلمه Promise

قول دادن



دیدی میگن فلانی قول داده که حتما انجامش بده

پس چاره ای نداره که انجامش بده

اگر انجامش نده کل کار رو هواست!

## کابردش کجاست؟





Soheib Kiani  
Front-end Developer

# کلی بگم

برای اجرای چند تا کار بصورت همزمان هستش

این کارها بصورت آرایه ای به `Promise.all` پاس داده میشن

این تابع اون عملیات رو بصورت همزمان اجرا میکنه

و

منتظر تکمیل همه عملیات هستش

خود `Promise.all` یک `Promise` جدید برمیگردونه که مقدار اون آرایه ای هستش از عملیاتی که قول داده بود انجامش بده





Soheib Kiani  
Front-end Developer

# کاربرد اصلی

در موقعی هستش که نیاز دارد چندین عملیات ناهمزمان را اجرا کنید و منتظر تکمیل همه اونها باشد

## به عنوان مثال

شما نیاز دارد از چندین منبع داده مختلف درخواست کنید و منتظر تکمیل درخواست های همه منابع باشد تا بتوانید نتایج

را تجمعی کنید

با استفاده از `Promise.all` میتوانید همه درخواست ها رو همزمان ارسال کرده و بعد از تکمیل همه درخواست ها، نتایج را بدست بیارید





Soheib Kiani  
Front-end Developer

# درگی پیشتر

شما نیاز دارید که از سه سرویس وب مختلف درخواست کنید

بعد از تکمیل همه درخواست‌ها، نتایج را نمایش بدهید  
اگر از روش سنتی استفاده کنید

باید هر درخواست را به صورت متوالی ارسال کنید و منتظر تکمیل هر درخواست باشید تا بتونید درخواست بعدی رو ارسال کنید.

اما با استفاده از `Promise.all`. می‌توانید همه درخواست‌ها رو همزمان ارسال کرده و بعد از تکمیل همه درخواست‌ها، نتایج رو دریافت و نمایش دهید.





# Soheib Kiani

## Front-end Developer

reject

در صورت خطا این فراخوانی میشے



```
JS
const getDataFromService1 = () => {
  return new Promise((resolve, reject) => {
    // کدی برای دریافت داده از سرویس اول
    // فراخوانی میشود و داده را بازمیگردانی میکند resolve، در صورت موفقیت
    // فراخوانی میشود و خطأ را بازمیگردانی میکند reject، در صورت خطأ
    // });
  });
}
```

resolve

در صورت موفقیت این فراخوانی میشے

```
const getDataFromService2 = () => {
  return new Promise((resolve, reject) => {
    // کدی برای دریافت داده از سرویس دوم
    // فراخوانی میشود و داده را بازمیگردانی میکند resolve، در صورت موفقیت
    // فراخوانی میشود و خطأ را بازمیگردانی میکند reject، در صورت خطأ
    // });
  });
}
```

getDataFromService1  
getDataFromService2  
getDataFromService3

سه تابع هستن که هر کدام توسط سرویس‌های مختلف داده را دریافت میکند و یک وعده (Promise) برمی‌گردونن. با استفاده از Promise.all و ارسال تابعهای مربوطه به آن، همه‌ی توابع همزمان اجرا می‌شوند و منتظر تکمیل همه وعده‌ها می‌باشند. پس از تکمیل همه وعده‌ها، نتایج به ترتیب در آرایه results قرار می‌گیرن و قابل استفاده هستن

```
const getDataFromService3 = () => {
  return new Promise((resolve, reject) => {
    // کدی برای دریافت داده از سرویس سوم
    // فراخوانی میشود و داده را بازمیگردانی میکند resolve، در صورت موفقیت
    // فراخوانی میشود و خطأ را بازمیگردانی میکند reject، در صورت خطأ
    // });
  });
}
```

```
// برای همزمان اجرای درخواست‌ها استفاده از
Promise.all([getDataFromService1(), getDataFromService2(), getDataFromService3()])
  .then(results => {
    // قرار دارد results نتایج درخواست‌ها در آرایه
    console.log('نتایج:', results);
  })
  .catch(error => {
    // در صورت بروز خطأ در هر کدام از درخواست‌ها، این قسمت اجرا می‌شود
    console.error('خطأ:', error);
  });
}
```

اینجا میدونی چرا  
new  
گذاشته؟!

زمانی که شما من خوايد  
یک نمونه جديد از کلاس  
درست بکنيد



اجباریه همیشه ???  
نه!  
بیا اسلاید بعدی  
بهت بگم

یادتکه گفتم آرایه





**Soheib Kiani**  
Front-end Developer

# new

از هر دو تاش میتوانی استفاده کنی

```
const promise = new Promise((resolve, reject) => {  
  // اجرایی کد Promise  
});
```

```
const promise = Promise.resolve(value);  
// با  
const promise = Promise.reject(reason);
```





**Soheib Kiani**  
**Front-end Developer**

اگر برات مفید بود

ممنون میشم به اشتراک بذاریش

