

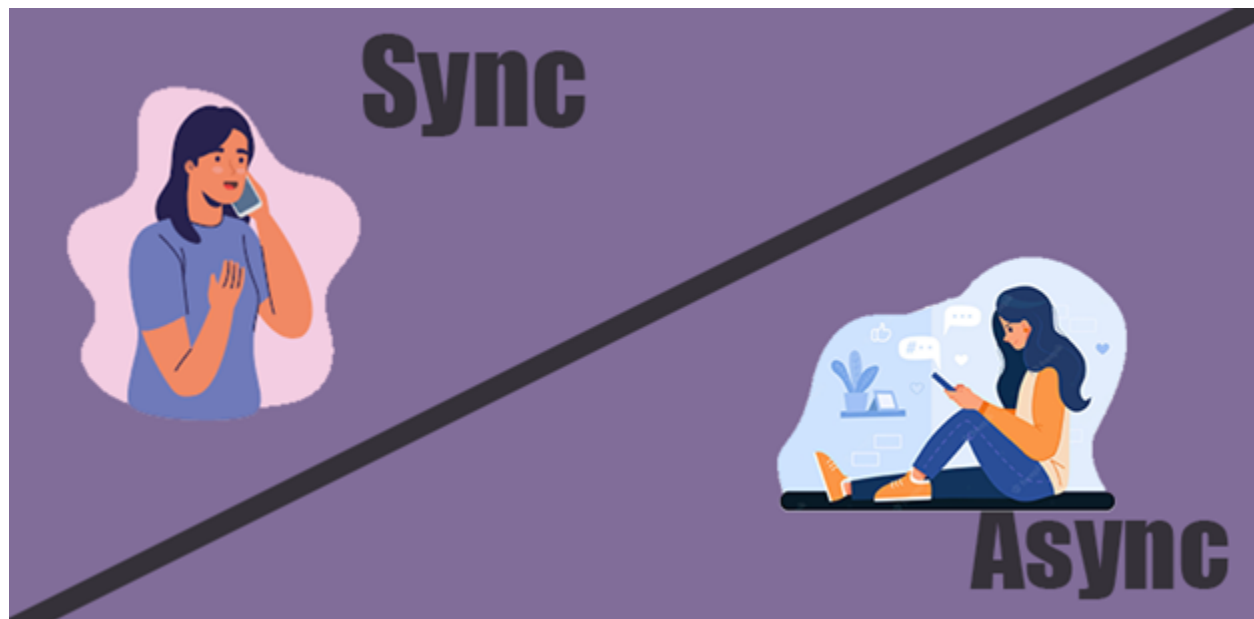
به نام خدا

تهیه کننده : علی صحرایی

موضوع : برنامه نویسی ASYNC

برای درک مفهوم برنامه نویسی غیرهمزمان بهتر است اول برنامه نویسی همگام یا Synchronous را بشناسیم. در این روش کدهای برنامه نویس پشت سر هم و به صورت خطی اجرا می‌شوند. یعنی اگر برنامه ما 400 خط کد داشته باشد، دستورات و توابع از خط 1 و به نوبت اجرا می‌شوند تا زمانی که به خط 400 برسیم و اجرای برنامه تمام شود. در این روش از برنامه نویسی دستورات باید به ترتیب اجرا شوند و تا زمانی که تابع "الف" اجرا نشده نمی‌توان سراغ تابع "ب" رفت.

اگر پیامک را یک نوع سیستم Async یا چندرشته‌ای در نظر بگیریم، در آن صورت می‌توان تماس تلفنی را یک سیستم با پایه Sync در نظر گرفت. در هنگام برقراری تماس تلفنی، شما باید منتظر باشید تا فردی که با شما در حال مکالمه است، صحبت خود را تمام کرده و سپس شما صحبت کنید.



برنامه نویسی **Asynchronous** (ناهمگام یا نامتقارن) یک مدل و مفهوم در برنامه نویسی است. در این مدل برخلاف روش **Synchronous** کدهای ما پشت سر هم اجرا نمی‌شوند و به اصطلاح ترتیب کدها غیرخطی است. حالا دیگر اجباری نیست که برنامه 400 خطی ما به ترتیب از خط 1 تا 400 اجرا شود و این روال می‌تواند تغییر کند.

Async می‌تواند بسیاری از مشکلات ما را حل کند. این روش از لحاظ ترتیب اجرا کاملاً در نقطه مقابل **Synchronous** قرار دارد و ما در یک برنامه می‌توانیم چندین جریان اجرایی یا **Control Flow** داشته باشیم. **Asynchronous** کمی شبیه به حل پازل است. همانطور که ما هنگام چیدن پازل از ترتیب خطی استفاده نکرده و با توجه به شرایط مختلف قطعات پازل را کنار هم می‌چینیم، در روش **Asynchronous** هم پردازش‌ها با ترتیب مشخصی اجرا نمی‌شوند.

کدام بهتر است؟ Async یا Sync

در برنامه نویسی پروژه های مختلف متداول است. استفاده از **Sync** همچنان استفاده از **Async** با وجود سریع‌تر بودن. هرکدام بسته به عملیات موردنظر و ترجیحات برنامه‌نویس در نظر گرفته می‌شود.

توابع به یکدیگر **Sync** در همین وابستگی توابع به یکدیگر است. در برنامه‌نویسی با مدل **Async** و **Sync** تفاوت بین وابسته‌اند و هر تابعی قبل از شروع عملیات خود، به پاسخی از تابع پیشین نیاز دارد. این در حالی است که در توابع بدون وابستگی به هم عملکرد خود را دارند **Async** برنامه‌نویسی با

تفاوت های sync & async

یک مدل **Sync** یک مدل چندرشته ای است که به برنامه‌نویسان قابلیت اجرای چند عملیات در موازات هم را می‌دهد. اما **Async** تک‌رشته‌ای است و در بازه مشخصی از زمان، تنها یک عملیات در آن قابل اجرا است.

در برنامه‌نویسی وب، **Async** مدل برنامه نویسی غیر مسدودکننده است که می‌تواند درخواست‌های متعددی را به یک سرور ارسال کند. در سمت دیگر، **Sync** مدلی بازدارنده است که تنها یک جواب به سرور ارسال کرده و تا رسیدن جواب آن، درخواست دیگری ارسال نمی‌کند..

به دلیل امکان اجرای چند عملیات هم زمان، کار با **Async** سریع‌تر پیش می‌رود. درحالی‌که **Sync** آهسته‌تر و با نظم بیشتری کارها را پیش می‌برد.



Asynchronous vs Synchronous Handling Concurrency in JavaScript

چه زمان از `async` استفاده کنیم؟؟

به صورت کلی می توان از برنامه نویسی `Async` در پروژه هایی با عملکردهای مستقل استفاده کرد. این شیوه برنامه نویسی مناسب پروژه هایی با تعداد عملیات و توابع زیاد است. به طور مثال در یک نرم افزار خرید آنلاین وقتی مشتری روی آیتم خریداری شده کلیک می کند، باید هم صفحه باز شده و هم فونت اسم محصول بزرگتر شود.

با سیستم برنامه نویسی `Async` می توان در یک زمان هر دوی این دستورات را اجرا کرد. تفاوت بین `Sync` و `Async` در اینجا این گونه است که اگر می خواهیم با `Sync` برنامه نویسی کنیم، باید صبر می کردیم اول صفحه لود شده و سپس فونت بزرگتر شود.

چه زمان از `sync` استفاده کنیم؟؟

برخلاف مدل پیچیده `Async`، برنامه نویسی `Sync` به راحتی قابل اجرا است. شروع هر عملیات در این برنامه نویسی به پایان مورد دیگر وابسته است. کار با این شیوه برنامه نویسی برای برنامه نویسان نسبت به `Async` راحت تر است چرا که نیازی به بررسی برای عدم وجود تداخل در عملکردها نیست.

اکنون فرض کنید در یک نرم افزار خرید آنلاین کاربر قصد سفارش چندین آیتم را داشته باشد. در این صورت باید اول همه آیتم های مورد نیاز را وارد سبد خرید کرده و سپس وارد پروسه پرداخت شود. در این صورت با برنامه نویسی `Sync` می توان مطمئن شد که پس از هر بار اضافه کردن یک آیتم، درگاه پرداخت بالا نمی آید. بلکه پس از تمام شدن پروسه انتخاب، کاربر می تواند تمام آیتم ها را به صورت یکجا پرداخت کند.

سوالات:

۱. مدل برنامه نویسی `async` در وب چگونه است؟ مسدود کننده که میتواند درخواست های متعدد را به سرور ارسال کند
۲. مدل برنامه نویسی `sync` در وب چگونه است؟ بازدارنده که تنها یک جواب به سرور ارسال کرده و تا رسیدن جواب آن، درخواست دیگری ارسال نمی کند..
۳. مثال برای `sync` و `async` ؟ تماس تلفنی `sync` و ارسال پیامک `async`
۴. چه زمان از `async` استفاده کنیم؟ به صورت کلی می توان از برنامه نویسی `Async` در پروژه هایی با عملکردهای مستقل استفاده کرد. این شیوه برنامه نویسی مناسب پروژه هایی با تعداد عملیات و توابع زیاد است.
۵. چرا `async` از `sync` سریع تر است؟ به دلیل اجرای همزمان عملیات ها