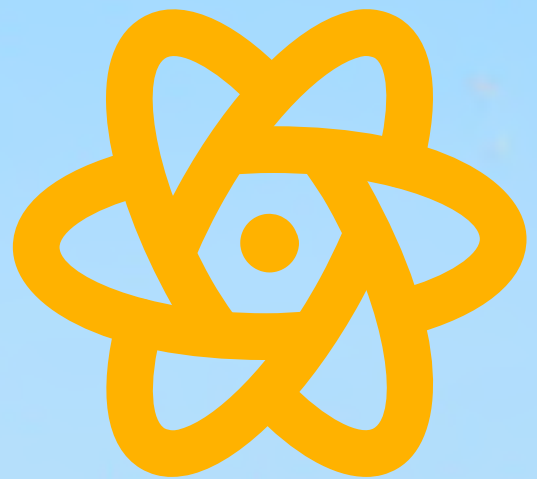


مقایسه فلاتر و ری اکت نیتیو



VS



مقایسه فلاتر و ری اکت نیتیو



<https://iracode.com/flutter-vs-react-native/>



۰۹۱۲۴۱۰۴۵۳۵

مقایسه فلاتر و ری اکت نیتیو



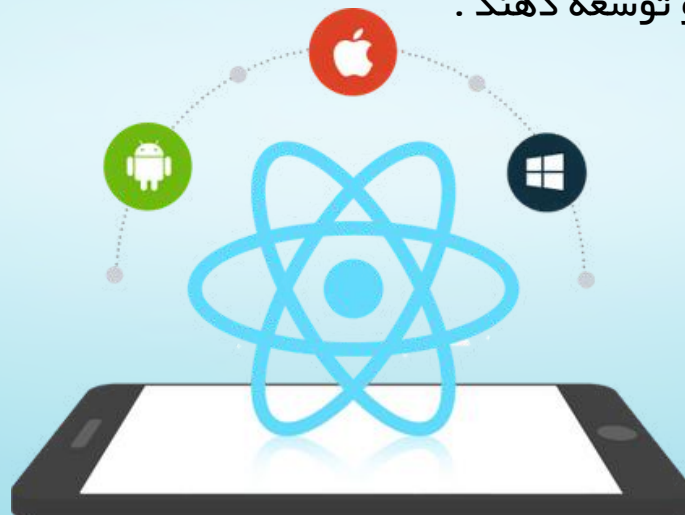
توسعه اپلیکیشن ها به صورت چندسکوپی یکی از موضوعات بسیار پرطرفدار است و برنامه نویسان می توانند به کمک کدبیس ها ، هم برای اندروید و هم برای iOS اپلیکیشن تولید کنند. این شیوه بسیار سریع تر و کم هزینه تر از حالت نیتیو خواهد بود . تا به امروز ابزار ها و پلتفرم های مختلفی ایجاد شده است که به ما در انجام و آموزش برنامه نویسی به این روش یاری میرسانند.

React Native یکی از گزینه های قدرتمند در این زمینه است که با معرفی رقیب جدی آن flutter ، بحث مقایسه بین این دو بسیار داغ شد و هر گروه از برنامه نویسان طرفدار یکی از این دو فریمورک شدند. در ادامه ی این مقاله به بررسی و مقایسه بین این دو فریمورک خواهیم پرداخت.



React Native چیست؟

ری اکت نیتیو یک کتابخانه متن باز به زبان جاوا اسکریپت ، برای نوشتن برنامه ها و اپلیکیشن های کاربردی تلفن همراه برای iOS و Android است . برنامه نویسان می توانند به کمک react برنامه های کاربردی را با استفاده از زبان جاوا که قبلا با آن کار کرده اند بنویسند و توسعه دهند .



مقایسه فلاتر و ری اکت نیتیو

ری اکت به جای هدف قرار دادن مرورگر، سیستم عامل های تلفن همراه را هدف قرار می دهد و از انجایی که اکثر کدهایش قابلیت اشتراک گذاری بین سیستم عامل ها را دارند، به کمک آن می توان هم برای اندروید و هم برای iOS به طور همزمان برنامه نویسی کرد و آنها را توسعه داد. React Native یک فریمورک متن باز و نسبتا جدید است که توسط فیس بوک منتشر شده و پشتیبانی می شود این فریمورک به برنامه نویس این امکان را می دهد که با استفاده از جاوا اسکریپت و کتابخانه React اپلیکیشن موبایل بسازند. تا به حال اپلیکیشن های مختلفی از جمله اینستاگرام و اسکایپ با استفاده از این فریمورک ساخته شده اند.

Flutter چیست؟

فلاتر یکی از فریم ورک های Cross-Platform است که در سال 2017 توسط گوگل ارائه شد و در سیستم عامل اندروید و iOS مورد استفاده قرار گرفت.



فلاتر با استفاده از زبان Dart که زمانی رقیب جاوا اسکریپت معرفی شده بود به برنامه نویسان امکان توسعه ی زیباترین و بهترین اپلیکیشن ها را به صورت Native می دهد و توسعه دهندگان می توانند از کد متن های فلاتر که به صورت متن باز و رایگان به راحتی در دسترس است استفاده کنند.

مقایسه فلاتر و ری اکت نیتیو



بزرگترین اختلاف بین این دو فریمورک زمانی است که می خواهیم یک بلوک برای رابط کاربری ایجاد کنیم.

React Native مبتنی بر اجزای بومی است و با استفاده از کامپوننت های native بسیار خوب ، با مجموعه ویجت های اختصاصی کار می کند ولی در Flutter جریان کاملا متفاوت است و از مجموعه اختصاصی widget ها استفاده می کند . این ویجت ها به برنامه نویسان کمک می کند تا طرح های سفارشی با قابلیت همکاری با کامپوننت های native ایجاد کنند.

در مقایسه بین این دو در پارمتر رابط کاربری ، React Native پیروز است زیرا به کاربران حس نیتیو بیشتری را منتقل میکند.

زمان توسعه :

یکی از مزیت های رقابتی شرکت های تولید کننده و توسعه دهنده نرم افزار های تلفن همراه تحویل به موقع پروژه هاست.

در این بخش React Native با استفاده از کامپوننت های آماده برای توسعه برنامه ها میتواند زمان توسعه اپلیکیشن ها رو کاهش دهد.

و از آنجا که Flutter جدید تر است زمان زیادی برای توسعه پروژه ها نیاز دارد و در این مقایسه هم React Native موفق تر است چون با کامپوننت های آماده با سرعت بسیار بیشتری توسعه میابد.

عملکرد:

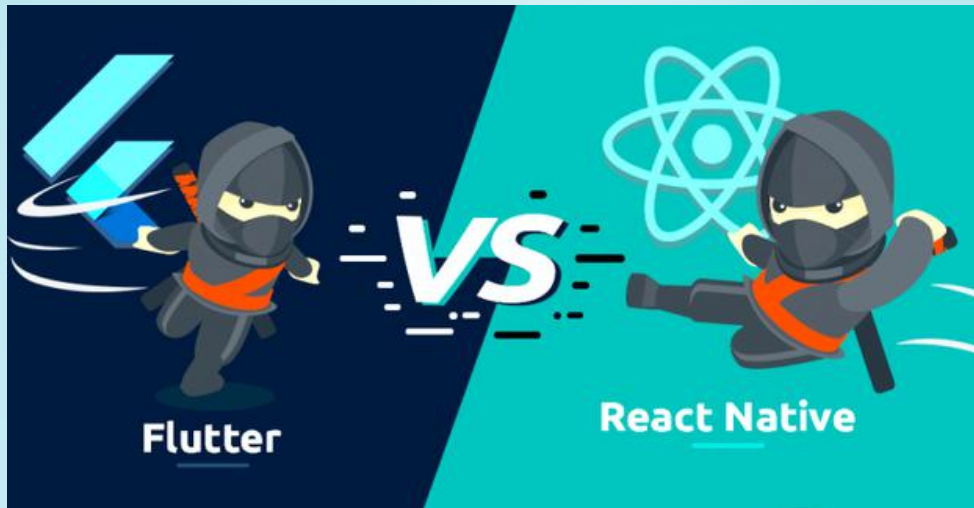
یکی از پارامتر های مهم برای مقایسه فریمورک ها performance است که به ما نتایج آشکاری می دهد. حال که به موضوع کارایی رسیدیم ممکن است نتایج برایتان تعجب آور باشد اما توسعه دهندگانی که این زبان ها را به خوبی می دانند مطمئنا با این نظر موافقند که در این پارامتر Flutter امتیاز بیشتری بدست می آورد .

از آنجایی که در React Native از جاوا اسکریپت استفاده می شود برای رندر کردن نهایی پروژه از یک پل برای تبدیل کد جاوا اسکریپت به نیتیو استفاده می شود که فلاتر به چنین پلی نیاز ندارد و تمام کامپوننت ها به صورت مستقیم آدرس دهی می شوند.

هنگامی که بخواهید یک اپلیکیشن بسیار بزرگ را توسعه دهید این موضوع بیش از پیش خود را نشان می دهد و در این مورد فلاتر بسیار سریع تر عمل می کند و در مجموع می توان Flutter را از لحاظ عملکرد بالاتر از React Native دانست.

ثبات :

از آنجا که امروزه بسیاری از توسعه دهندگان از React Native استفاده میکنند این فریمورک به حالت پایدار نزدیک تر است اما در مورد Flutter ، با توجه به جدید بودن زمان بسیار بیشتری برای رسیدن به ثبات و پایداری نیاز دارد.



نتیجه گیری :

این که توسعه دهندگان از کدام فریمورک برای توسعه اپلیکیشن های خود در نظر بگیرند به تجربه شخصی توسعه دهنده و نتیجه و آنالیزی که از مطالعه این مطالب و مقایسه ها بدست می آورد بستگی دارد .

هرکدام از این فریمورک ها نقاط ضعف و قوت خود را دارند و به طور کلی تصمیم گیری برای مشخص کردن برتری ها بسیار دشوار است.

