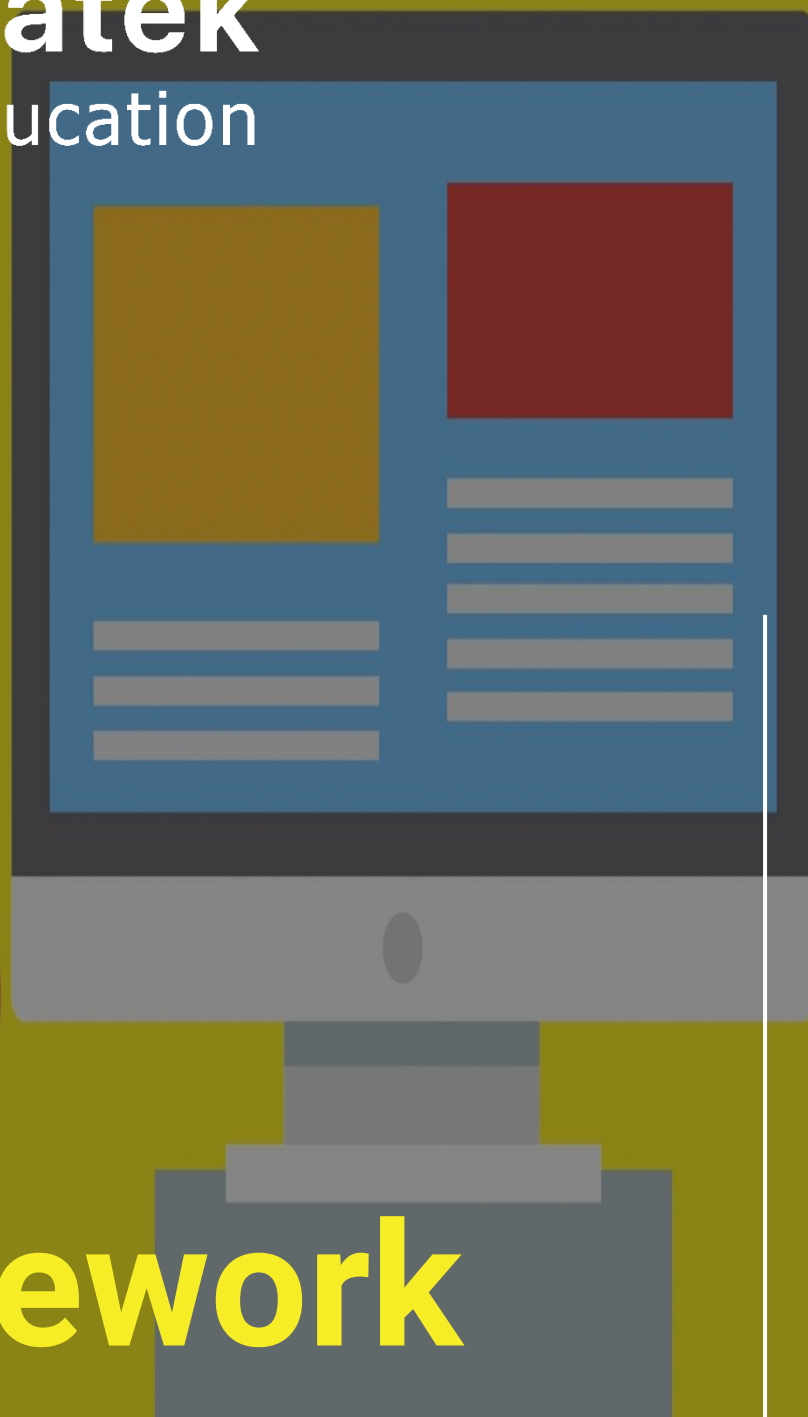
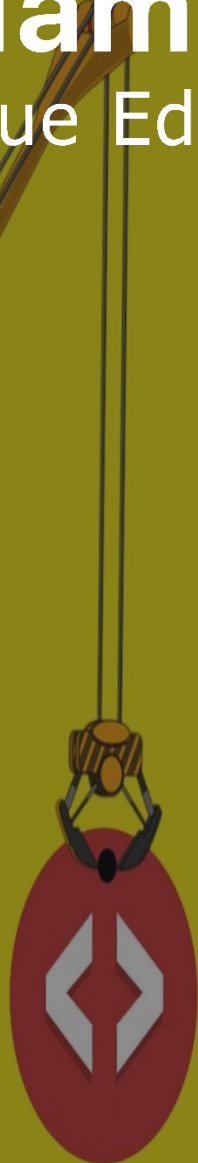




Namatek
True Education



www.namatek.com

Framework

فریم ورک چیست؟

فهرست مطالب

۱. فریم ورک چیست؟
۲. چرا از Framework ها استفاده می کنیم؟
۳. چه چیزهایی در یک فریم ورک وجود دارد؟
۴. تفاوت کتابخانه و فریم ورک چیست؟
۵. مزایای استفاده از فریم ورک چیست؟
۶. ویژگی های یک Framework خوب چیست؟
۷. چالش ها و معایب فریم ورک چیست؟

اگر با برنامه نویسی و توسعه نرم افزار سروکار داشته باشید، لازم است که بدانید فریم ورک چیست. کار با فریم ورک ها امروزه محبوبیت بسیاری میان برنامه نویسان پیدا کرده است و آشنایی با آن می تواند شما را در دسترسی سریع تر به اهدافتان کمک کند. این مقاله به بررسی کامل فریم ورک ها پرداخته است، با ما همراه باشید تا به سوالات شما درباره آن پاسخ دهیم.

#۱ فریم ورک چیست؟

فریم ورک (Framework) یا فریم ورک نرم افزاری (Software Framework) بستری است که امکانات پایه ای را برای توسعه برنامه های نرم افزاری فراهم می کند. در واقع فریم ورک را می توان، الگو یا چهارچوبی از یک برنامه کاری در نظر گرفت که با افزودن کد به صورت دلخواه قابل سفارشی سازی و اصلاح است. به بیان ساده فریم ورک به عنوان پشتیبان یا راهنمای ساخت چیزی است که ساختار آن را بهبود می بخشد. چیزی مشابه راهنمای ساخت ماکت سفینه فضایی که در تصویر زیر قابل مشاهده است.



فریم ورک ها از منابع مشترکی استفاده می کنند و آن ها را در یک بسته قرار می دهند. برخی از این منابع عبارتند از:

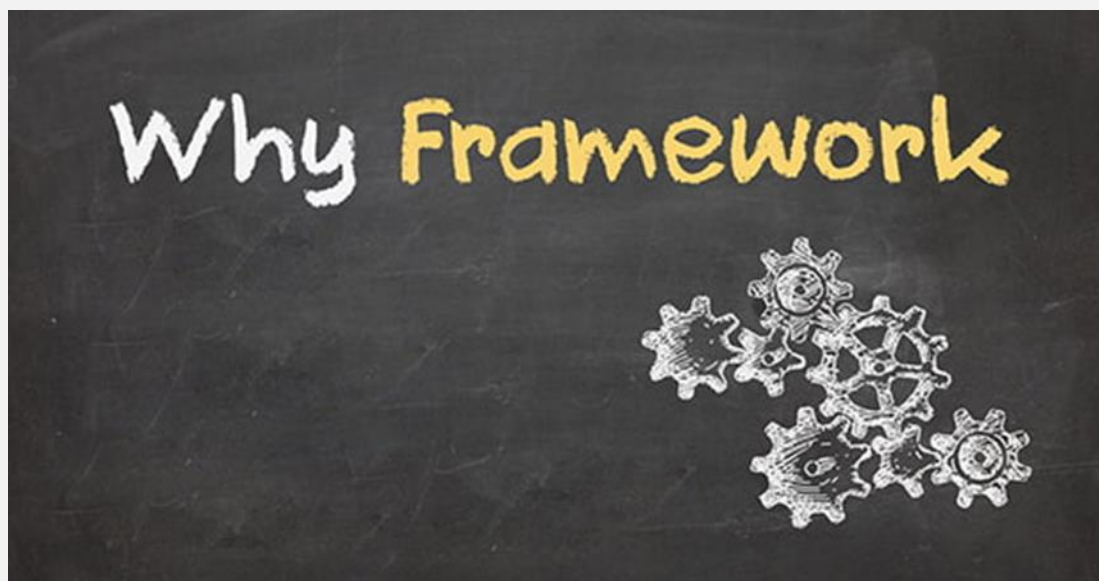
- کتابخانه ها
- پرونده های تصویری
- اسناد مرجع

این بسته می تواند متناسب با نیازهای خاص پروژه اصلاح شود. به عنوان مثال در فریم ورک های مربوط به طراحی سایت از منابع مورد نیاز برنامه نویس می توان به موارد زیر اشاره کرد:

- فایل های بهبود سازگاری در مرورگرهای مختلف مانند Modernizr و CSS reset
- کتابخانه های پایه ای برای طراحی بخش های مختلف سایت نظیر jQuery و غیره

بنابراین توسعه دهنده به وسیله فریم ورک می تواند ویژگی هایی را به پروژه اضافه یا جایگزین آن کند تا عملکرد جدیدی به آن بدهد.

#۲ چرا از Framework ها استفاده می کنیم؟



توسعه نرم افزار یک فرآیند پیچیده است. این کار به مجموعه ای از وظایف از جمله کدنویسی، طراحی و آزمایش نیاز دارد. فقط برای قسمت کدنویسی، برنامه نویسان در حین کدنویسی باید مراقب بسیاری از بخش ها باشند که برخی از آن ها شامل موارد زیر هستند:

- سینتکس (syntax)
- تعریف ها (declarations)
- زباله رویی (garbage collection)
- دستورالعمل ها (statements)
- موارد استثنا (exceptions)

و موارد دیگر

اما در این میان نقش فریم ورک چیست؟

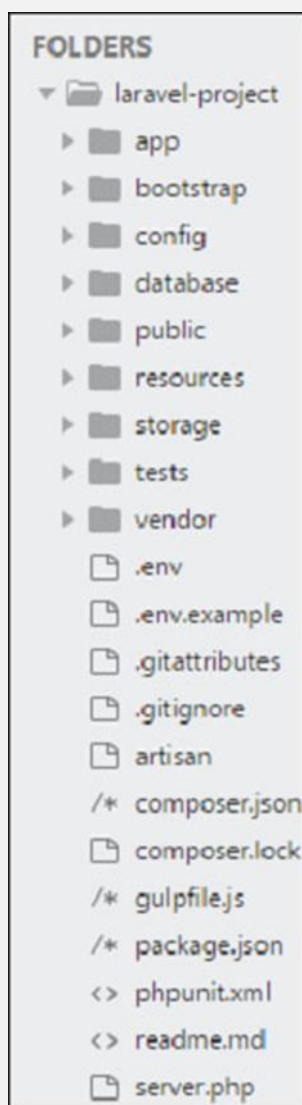
فریم ورک ها با کنترل کل فرآیند توسعه نرم افزار یا بیشتر از آن، با یک پلتفرم (platform) یا بستر واحد، کار با مواردی نظیر بخش های بالا و حتی موارد بیشتر را برای توسعه دهندگان آسان می کنند.

#۳ چه چیزهایی در یک فریم ورک وجود دارد؟

هنگام نصب یک فریم ورک، اولین چیزی که باید به آن توجه کنید نیازهای سیستم است. چرا که برخی از فریم ورک ها برای نصب روی

سیستم به شرایط و برنامه های خاصی وابسته هستند. پس از نصب و پیکربندی یک فریم ورک، یک ساختار دایرکتوری ایجاد می شود. دایرکتوری در فریم ورک چیست؟

دایرکتوری به ساختاری از پوشه ها و فایل ها گفته می شود که به طور منظم و دقیق طبقه بندی شده اند. به عنوان مثال، تصویر زیر ساختار دایرکتوری را در فریم ورک لاراوِل (Laravel) نشان می دهد.



هر یک از پوشه ها در هر فریم ورک می توانند دایرکتوری های اضافی داشته باشند.

فایل ها در دایرکتوری ها می توانند شامل موارد دیگری نیز باشند که برخی از آن ها عبارتند از:

- کلاس ها
 - کتابخانه ها
 - قالب ها
- و غیره

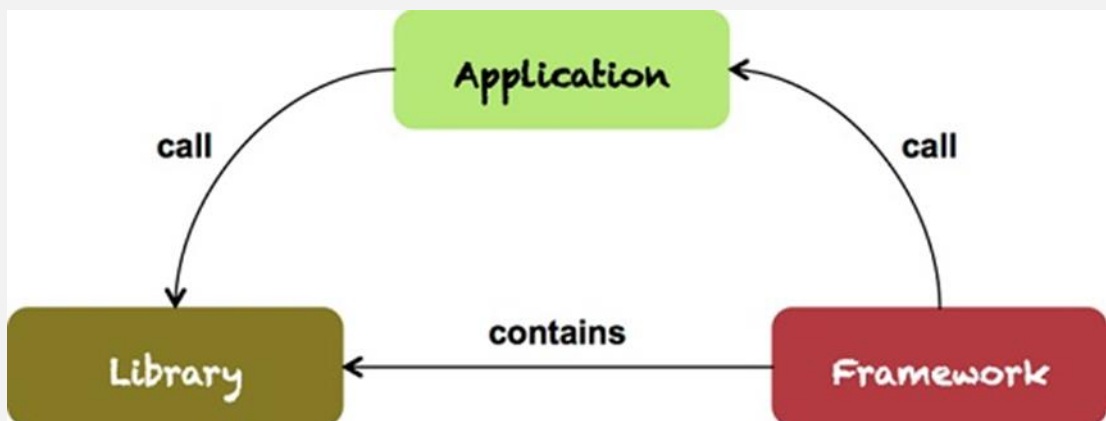
#۴ تفاوت کتابخانه و فریم ورک چیست؟

هیچ چیز جادویی در مورد فریم ورک ها یا کتابخانه ها وجود ندارد. همان طور که گفته شد هم کتابخانه ها و هم Framework ها کدی قابل استفاده مجدد هستند که توسط شخص دیگری نوشته شده اند و هدف آن ها کمک به شما در حل مشکلات رایج از راه های آسان تر است. به عنوان مثال، فرض کنید شما برنامه ای دارید که قصد کار با رشته ها را داشته و می خواهید از تکرار کد خودداری کنید. کافیست کد مورد نظر را ایجاد کرده و در بخش های مورد نیاز آن را فراخوانی کنید. تبریک می گوئیم، با این کار شما یک کتابخانه ایجاد کرده اید.

اما تفاوت میان کتابخانه و فریم ورک چیست؟

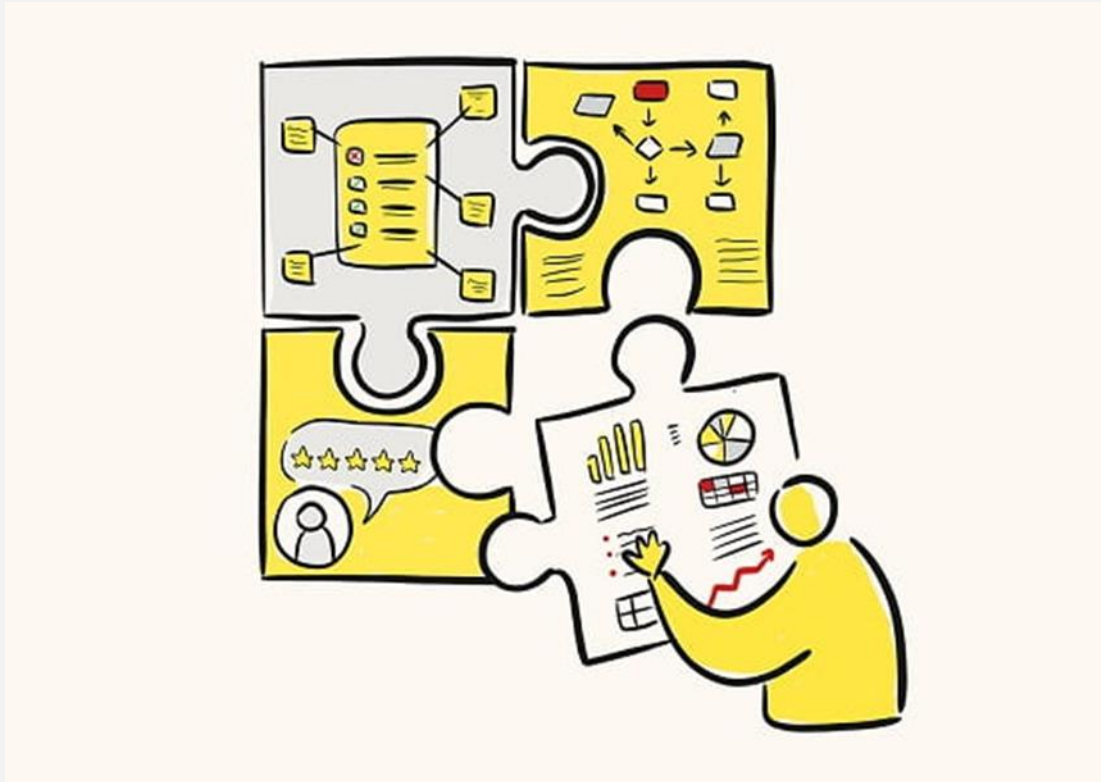
برای درک بهتر لطفاً به این مثال توجه کنید. کتابخانه مانند رفتن به فروشگاه مبلمان است. شما در حال حاضر خانه ای دارید که برای مبلمان آن نیازمند کمک هستید؛ اما شما نمی‌خواهید مبلمان خود را از ابتدا درست کنید. فروشگاه ایکیا به شما امکان می‌دهد چیزهای مختلفی را برای خانه انتخاب کرده و بردارید؛ بنابراین شما در این خرید و انتخاب کنترل کامل دارید. از طرف دیگر، یک Framework مانند ساخت یک خانه نمونه است. در ساختار فریم ورک‌ها، شما در مورد معماری و طراحی مجموعه ای از نقشه‌های اصلی چند انتخاب محدود دارید. در نهایت، پیمانکار و طراح طرح اصلی پروژه را تحت کنترل دارند.

در واقع آن‌ها به شما می‌گویند که چه زمانی و از کجا می‌توانید ورودی خود را ارائه دهید؛ یعنی کنترل در موارد اصلی در دست فریم ورک است. این موضوع تنوع را تا حد زیادی کاهش می‌دهد. تصویر زیر نحوه عملکرد فریم ورک و کتابخانه را نمایش می‌دهد.



#۵ مزایای استفاده از فریم ورک چیست؟

Framework ها کمک بسیاری به توسعه دهندگان در طراحی محصولات می کنند. آن ها زیرساختی را ارائه می دهند که قبلاً جزئیات سطح پایین را کشف کرده است و به توسعه دهنده اجازه می دهد تا روی جزئیات منحصر به فرد پروژه خاص خود تمرکز کند. توسعه دهنده همچنین باید کد کمتری بنویسد و کد کمتر به معنای زمان مورد نیاز کمتری برای رفع اشکالات در کدهای نوشته شده است. علاوه بر این کدنویسی کمتر زمان توسعه را نیز کاهش می دهد که این موارد سبب صرفه جویی در هزینه می شوند. حتی اگر یادگیری فریم ورک ها در ابتدا خسته کننده باشد؛ اما به مرور و با آشنایی بیشتر، ارزش آن ها بیشتر می شود. چرا که تکمیل پروژه های آینده سریع تر و آسان تر می شود و کار بیشتر در زمان کمتر باعث ایجاد یک توسعه بهتر می گردد.



علاوه بر این استفاده از فریم ورک مزایای دیگری نیز به همراه دارد که برخی از آن ها عبارتند از:

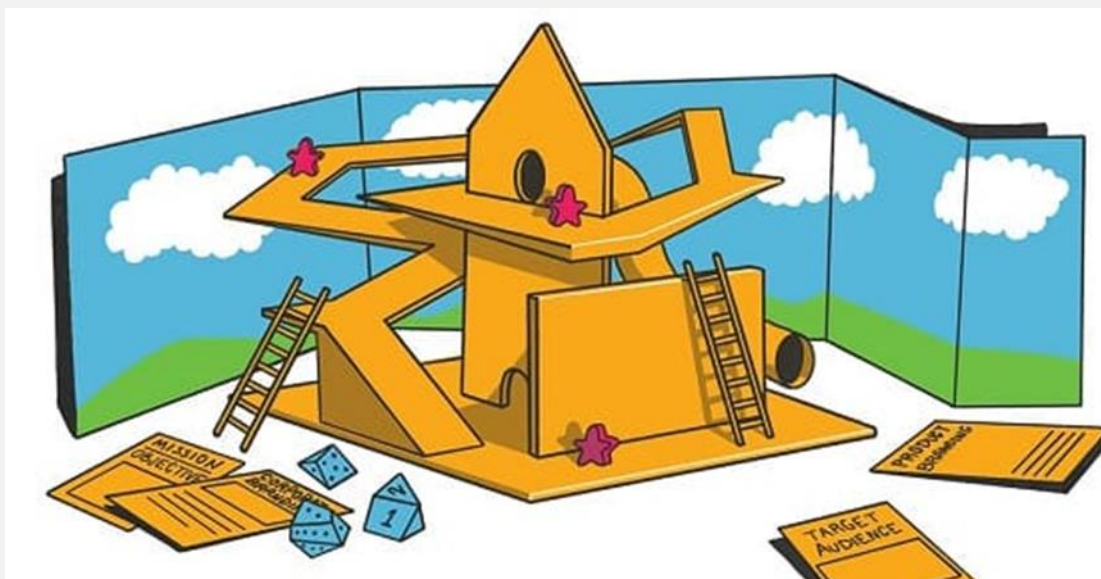
- به استفاده از الگوهای مناسب طراحی و ایجاد شیوه ها و فرآیندهای بهتر برنامه نویسی کمک می کنند.
- به دلیل استاندارد بودن ساختار به امنیت بیشتر کد کمک می کنند.
- از تکرار و ایجاد کد اضافی جلوگیری می کنند.
- از طریق کدنویسی در چهارچوب و با اشکالات کمتر توسط توسعه دهندگان، به توسعه سازگار کد کمک می کنند.
- از طریق فریم ورک ها به راحتی می توان با فناوری های پیشرفته کار کرد.

- تمام بخش‌ها در ساختار فریم‌ورک‌ها بارها آزمایش شده‌اند. به همین دلیل پروژه‌های تولید شده توسط فریم‌ورک‌ها قابل اعتمادتر هستند.

#۶ ویژگی‌های یک Framework خوب چیست؟

ویژگی‌های مهم یک فریم‌ورک چیست؟

در ادامه به برخی از مهم‌ترین موارد در انتخاب فریم‌ورک‌ها اشاره شده است:



- فهم آن ساده و اجرای آن آسان است.
- حاوی کد اضافی که عملکردی را انجام نمی‌دهد، نیست.
- با بسترهای مختلف سازگار است.
- دارای به روز رسانی‌های متعدد و تکامل در آینده است.

- به توسعه دهنده اجازه می دهد تا اجزای قابل استفاده مجدد را بسازد.
- می توان آن را با افزودن کد اضافی برای ارائه عملکرد متفاوت گسترش داد (البته این شامل کد پیش فرض فریم ورک است که قابل تغییر نیست و فقط می توان آن را بازنویسی کرد).
- همچنین استقبال توسعه دهندگان نشان دهنده کارایی و سازگاری یک فریم ورک است.

#۷ چالش ها و معایب فریم ورک چیست؟



بیشتر تکنولوژی ها در کنار مزایای خود چالش ها و معایبی را نیز به همراه دارند. فریم ورک ها نیز از این قاعده مستثنی نیستند. اما چالش های موجود بر سر راه توسعه دهندگان فریم ورک چیست؟ فریم ورک ها یک راه

حل مناسب برای همه هستند؛ اما با این وجود محدودیت هایی نیز دارند که برخی از آن ها عبارتند از:

- با یک Framework نمی توانید هر کاری که مورد نظرتان است و یا همه کارها را انجام دهید؛ بنابراین برخی از فریم ورک ها برای برخی از برنامه ها یا کارها مناسب نیستند.
- فریم ورک ها برای پروژه های کوچک مناسب نیستند؛ زیرا کدگذاری و تنظیم فریم ورک بیشتر از کدگذاری سفارشی برای پروژه طول می کشد.
- یادگیری فریم ورک ها برای یک کاربر جدید زمان بر است و پیش از شروع کار، برنامه نویس باید وقت بگذارد تا بفهمد چگونه از آن به درستی استفاده کند.
- هرگونه مشکلات امنیتی در یک فریم ورک می تواند بر روی هر برنامه ساخته شده توسط آن تأثیر بگذارد.