



عنوان مقاله: آشنایی با فریم ورک های پایتون | بهترین فریمورک ها برای هر حوزه را بشناسید

نویسنده مقاله: تیم فنی نیک آموز

تاریخ انتشار: ۲۴ اسفند ۱۴۰۲

منبع: <https://nikamooz.com/python-frameworks/>

فریم ورک های پایتون هر یک با کاربردها و مزیت های مخصوص به خود توسعه داده شده اند و انتخاب یک چارچوب نرم افزاری مشخص، به نیازمندی ها پروژه و ترجیحات شما بستگی دارد. پس از [آشنایی با مفاهیم پایه پایتون](#) و [نصب و راه اندازی آن](#)، در این آموزش با انواع فریمورک های پایتون آشنا خواهید شد؛ درحقیقت، با مطالعه این مقاله متوجه می شوید که چرا فریمورک ها کاربردی هستند و بهترین آن ها کدامند.

### زبان برنامه نویسی پایتون چیست؟

از میان [انواع زبان های برنامه نویسی](#)، زبان پایتون (Python) به خوانایی و کاربرپسند بودن بالا شهرت پیدا کرده است. این زبان همه منظوره در امور مختلفی همچون توسعه وب (Web Development)، تجزیه و تحلیل داده ها (Data Analysis)، خودکارسازی (Automation) و محاسبات علمی (Scientific Computing) کاربرد دارد. پایتون [زبان سطح بالا](#) (High-Level)، [مفسری](#) (Interpreted) و قابل درک است؛ ضمن اینکه [آینده شغلی زبان برنامه نویسی پایتون](#) فرصت های شغلی متنوع و جذابی را شامل می شود. در [مقاله زبان برنامه نویسی پایتون چیست](#) به طور مفصل این زبان را معرفی کرده ایم که پیشنهاد می کنیم برای آشنایی بیشتر با این زبان، حتماً آن را مطالعه نمایید.

### فریمورک چیست؟

در جهان برنامه نویسی، فریمورک (Framework) یا همان چارچوب نرم افزاری، یک ساختار از قبل ایجاد شده تلقی می شود که توسعه دهندگان می توانند از آن برای ایجاد اپلیکیشن ها به صورت کارآمدتر و سریع تر استفاده کنند. به عنوان مثال، ساختن خانه را در نظر بگیرید؛ طبیعتاً اینکه هر بار از پایه و از ابتدا شروع کنید، ترجیح کسی نیست. فریمورک دقیقاً به همین دلیل در دنیای برنامه نویسی به کار می رود و با ارائه یک پایه منسجم به همراه اجزای ضروری، به توسعه دهندگان اجازه می دهد تا تمرکز خود را بر روی ویژگی ها و کارکردهای منحصر به فرد اپلیکیشن قرار دهند.

### مزایای چارچوب های نرم افزاری

مزایای استفاده از فریمورک در برنامه نویسی عبارتند از:

- بهبود کیفیت کد و افزایش قدرت و امنیت اپلیکیشن ها
- صرفه جویی در زمان و امکان استفاده از کدهایی با قابلیت استفاده مجدد
- افزایش سرعت فرآیند توسعه اپلیکیشن
- پشتیبانی و کامیونیتی گسترده تر

با این مقدمات از مفهوم فریمورک ، در بخش های بعدی به بررسی بیشتر فریمورک های پایتون می پردازیم.

## فریم ورک پایتون چیست؟

فریمورک های پایتون مجموعه ای از کدهای از قبل نوشته شده ای هستند که به هدف ساده سازی و تسریع توسعه اپلیکیشن های پایتون طراحی شده اند. این چارچوب ها عملکردهای رایج زیر را دارا هستند:

- **امنیت: احراز هویت کاربر (User Authentication)**، **مجوز کاربر (User Authorization)** و محافظت در برابر آسیب پذیری ها، مواردی هستند که در امنیت فریمورک ها لحاظ می شوند.
- **تعامل با بانک اطلاعاتی:** فریمورک ها، ابزارهایی برای اتصال به **انواع پایگاه داده**، ذخیره و بازیابی داده دارا هستند.
- **مسیریابی:** این موضوع رسیدگی به درخواست های کاربر و هدایت آن ها به بخش های مناسب از اپلیکیشن را دربرمی گیرد.
- **قالب بندی (Templating):** جداسازی ساختار صفحه از محتوای آن باعث می شود تا صفحات وب پویا قابل ایجاد باشند.
- **رسیدگی به فرم:** پردازش ورودی کاربر از فرم ها توسط این Functionality انجام خواهد شد.

در حقیقت، با بهره مندی از ویژگی های توکار (Built-In Features) چارچوب های نرم افزاری، برنامه نویسی و توسعه دهنده می تواند به سایر تسک های پراهمیت بپردازد. برای آشنایی بیشتر با وظایف این افراد در چرخه توسعه نرم افزار، پیشنهاد می کنیم **مقاله تفاوت بین برنامه نویسی ، کدنویسی ، توسعه دهنده و مهندس نرم افزار** را مطالعه کنید.

## چرا باید از چارچوب های پایتون استفاده کرد؟

دلایل قانع کننده ای برای استفاده از فریم ورک های پایتون وجود دارد که در اینجا به آن ها اشاره می کنیم:

- **افزایش سرعت توسعه:** فریمورک ها اجزای از پیش ساخته و کارکردهای مختلفی را ارائه می دهند که به کمک آن ها دیگر به نوشتن کد از پایه برای وظایف رایج وجود نخواهد داشت. این امر به صرفه جویی قابل توجهی در زمان، به خصوص برای وظایف تکراری همچون احراز هویت کاربر، تعامل با پایگاه داده و مسیریابی، منجر خواهد شد.
- **کیفیت کد بهبود یافته:** غالباً چارچوب ها با کدهای تست شده و امن ارائه می شوند که این موضوع به برنامه های قوی تر و قابل اعتمادتر منجر می شود. این کدها از قواعد و بهترین روش های برنامه نویسی پیروی می کنند و تمیزی و قابل نگهداری تر شدن کدها را به همراه دارند.
- **تمرکز بر منطق برنامه:** با انجام کارهای پایه ای، فریمورک ها به توسعه دهندگان اجازه می دهند تا بر کارکردهای اصلی و فیچرهای اپلیکیشن تمرکز کنند. این امر آن ها را از غرق شدن در جزئیات سطح پایین رها خواهد کرد.
- **ساختار و قابلیت نگهداری:** فریم ورک ها رویکردی ساختاریافته برای توسعه ارائه می دهند که به واسطه آن، درک کدبیس ، پیمایش و اصلاح برای ارتقاها آتی تسهیل می یابد. این ویژگی جذاب از چارچوب ها، به طور ویژه برای پروژه های بزرگ و دارای چند توسعه دهنده مفید است.

- **جامعه و پشتیبانی قوی:** اغلب فریم ورک های محبوب، کامیونیتی غنی و فعالی از توسعه دهندگان دارا هستند. به همین دلیل، مزیت‌هایی مانند دسترسی آسان‌تر به منابع آموزشی، کمک به رفع اشکال و کتابخانه‌های Readily Available برای کارکردهای اضافی در آن‌ها فراهم شده است.
- **ویژگی‌های امنیتی:** بسیاری از فریمورک‌ها اقدامات امنیتی built-in خاصی مانند احراز هویت کاربر، اعطای مجوز و محافظت در برابر آسیب‌پذیری‌های رایج را در خود جای داده‌اند. این ویژگی به محافظت از اپلیکیشن شما در برابر تهدیدات امنیتی احتمالی کمک خواهد کرد.
- **نمونه‌سازی سریع:** چارچوب‌هایی مانند Flask به دلیل سادگی و انعطاف‌پذیری مناسب، در Prototyping سریع پیشرو هستند. شما می‌توانید به سرعت یک نسخه اولیه از اپلیکیشن خود را برای تست و بررسی مفاهیم و جمع‌آوری بازخورد ایجاد کنید.

در مجموع، فریم ورک های پایتون ترکیب مناسبی از سرعت، امنیت، ساختار و پشتیبانی جامعه را ارائه می‌دهند که آن‌ها را به یک ابزار ارزشمند و کاربردی برای توسعه‌دهندگان با سطوح تجربه مختلف تبدیل می‌کند.

## انواع فریمورک های پایتون

انواع فریمورک های پایتون در ادامه مورد بررسی قرار می‌گیرند.

### فریم ورک های توسعه وب پایتون

فریمورک های وب پایتون عبارتند از:

- **جنگو (Django):** یک فریم ورک فول استک (Fullstack) و سطح بالا است که برای توسعه سریع طراحی تمیز وب‌اپلیکیشن‌ها، پروژه‌ها و CMS به کار می‌رود.
- **Flask:** این فریم ورک مینیمال و سبک‌وزن برای ساخت وب‌اپلیکیشن‌های ساده و منعطف، مانند SPA ها، RESTful API ها و پروژه‌های کوچک‌تر مناسب است.
- **Pyramid:** یکی از فریمورک های پایتون که به‌عنوان یک چارچوب نرم افزاری فول‌استک سازگار و قدرتمند شناخته شده است. به دلیل امکان سفارشی‌سازی مناسب آن، برای وب‌اپلیکیشن‌های پیچیده و بزرگ قابل استفاده است.
- **Bottle:** میکروفریمورک ساده و سریعی به‌شمار می‌رود و به‌منظور ایجاد API و وب‌اپلیکیشن‌ها با کد مینیمال مورد استفاده قرار می‌گیرد. می‌توان آن را برای اموری مانند پروتوتایپینگ سریع و وب‌سرویس‌های ساده به‌کار برد.
- **FastAPI:** این فریمورک وب کارآمد، به دلیل سرعت و تمرکز آن روی توسعه API به شهرت رسیده است. در FastAPI، انواع ویژگی‌های مدرن پایتون وجود دارند و قابلیت یکپارچه‌سازی با برنامه‌نویسی ناهمگام برای رسیدگی به درخواست‌های همروند نیز فراهم شده است.

## فریم ورک های پایتون برای هوش مصنوعی

کاربردترین فریمورک های پایتون برای هوش مصنوعی در ادامه لیست شده‌اند:

- **تانسرفلو (Tensorflow):** این کتابخانه متن‌باز (Open-Source) برای محاسبات عددی و یادگیری ماشین در مقیاس‌های گسترده به کار می‌رود. تانسرفلو در استقرار مدل‌ها، تحقیقات و یادگیری عمیق به وفور مورد استفاده قرار می‌گیرد و ابزارهای متعددی برای ساخت و آموزش شبکه‌های عصبی پیچیده در آن وجود دارند.
- **پایتورچ (Pytorch):** یکی دیگر از کتابخانه‌های متن‌باز در [یادگیری ماشین](#) است که به صورت عمده در یادگیری عمیق کاربرد دارد. شما می‌توانید با کمک این فریمورک پایتون، اموری همچون Prototyping، استقرار مدل‌ها و تحقیقات خود را به خوبی انجام دهید. البته Pytorch به سادگی استفاده و گراف محاسباتی پویا آن مشهور شده و در مقایسه با تانسرفلو، API های قابل‌درک‌تری دارا است.
- **Scikit-learn:** این فریمورک به شما امکانات رایگانی در زمینه تسک‌ها و الگوریتم‌های سنتی یادگیری ماشین، شامل طبقه‌بندی (Classification)، خوشه‌بندی (Clustering)، رگرسیون (Regression) و همچنین، تجزیه و تحلیل دیتا ارائه می‌دهد.
- **Keras:** هرچند Keras یک فریمورک Standalone محسوب نمی‌شود، اما یک API شبکه عصبی سطح بالا است و می‌توان آن را روی Tensorflow یا سایر فریمورک های پایتون مانند Pytorch اجرا کرد. Keras با ارائه رابط کاربرپسند، فرآیند ساخت و آموزش شبکه‌های عصبی را تسهیل می‌دهد.

در صورتی که علاقه‌مند به شروع کار با هوش مصنوعی در پایتون هستید، مقاله [پایتون و هوش مصنوعی: آشنایی با یادگیری ماشین و شبکه های عصبی](#) می‌تواند برای شما مفید باشد. علاوه بر این، اگر می‌خواهید شناخت شما از هوش مصنوعی و ابزارهای جذاب آن عمیق‌تر شود، می‌توانید [مقاله جامع هوش مصنوعی](#) را نیز مطالعه کنید.

## فریم ورک های پایتون برای هک

می‌توان فریمورک های پایتون را برای [هک اخلاقی](#) (Ethical Hacking) و [تست نفوذ](#) (Peneration Testing) استفاده کرد. تعدادی از این چارچوب‌های نرم‌افزاری عبارتند از:

- **Scapy:** این کتابخانه قدرتمند به شما اجازه می‌دهد بسته‌های شبکه را ایجاد، ارسال، Sniff و تجزیه و تحلیل کنید و از این ابزار برای وظایفی مانند [شناسایی شبکه](#) (Network Reconnaissance)، اسکن آسیب‌پذیری و آزمایش کنترل‌های امنیتی شبکه بهره‌مند شوید.
- **Requests:** این کتابخانه، ارسال درخواست‌های HTTP و رسیدگی به پاسخ‌ها را ساده‌سازی می‌کند و به طور گسترده، برای تعامل با وب‌اپلیکیشن‌ها در طول تست نفوذ، مانند تست فرم‌های ورود یا API ها، استفاده می‌شود.
- **Paramiko:** با استفاده از [پروتکل Secure Shell](#)، ابزارهایی برای ارتباط امن با سرورهای از راه دور ارائه می‌دهد. این کتابخانه به هکرهای اخلاقی اجازه می‌دهد تا در طول تست نفوذ، به طور امن به سیستم‌های هدف متصل شوند، دستورات را اجرا کرده و فایل‌ها را منتقل کنند.

- **BeautifulSoup**: این کتابخانه، ابزاری مناسب برای تجزیه و تحلیل داده های HTML و XML است و به منظور تجزیه و تحلیل ساختار Web Application ها و شناسایی آسیب پذیری های بالقوه در طول تست نفوذ آن ها، ارزشمند خواهد بود.
- **Cryptography**: به واسطه الگوریتم ها و ابزارهای رمزنگاری مختلف آن، اموری مانند رمزگذاری، رمزگشایی، هش کردن (Hashing) و امضای دیجیتال میسر می شود. هکرهای اخلاقی می توانند از این کتابخانه برای تجزیه و تحلیل پروتکل های رمزنگاری، آزمایش نقاط ضعف یا رمزگشایی داده های رمزنگاری شده در طول تست نفوذ استفاده کنند.
- **Nmap**: یک اسکنر شبکه محبوب و python-nmap است که رابطی برای تعامل ارائه می دهد. این کتابخانه به هکرهای اخلاقی اجازه می دهد تا کشف شبکه را خودکارسازی کنند، هاست ها و سرویس ها را شناسایی کرده و به دنبال آسیب پذیری در شبکه های هدف بگردند.
- **Faker**: فیکر امکان تولید داده های جعلی مانند نام، آدرس، ایمیل و شماره تلفن را دارد. این ابزار برای ایجاد داده های تست واقعی در فعالیتهای مربوط به تست نفوذ، مانند پرکردن فرم ها یا تست مکانیزم های احراز هویت مفید است.
- **Twisted**: این موتور شبکه ناهمگام (Asynchronous Networking Engine)، امکان ساخت اپلیکیشن های شبکه پیچیده را فراهم می کند. اگرچه Twisted اصولاً برای تست نفوذ مستقیم استفاده نمی شود، اما می توان از آن برای توسعه ابزارهای سفارشی یا خودکارسازی وظایف مربوط به ارتباطات شبکه استفاده کرد.

**نکته مهم:** به یاد داشته باشید که این کتابخانه ها ابزارهای قدرتمندی هستند و فقط باید برای فعالیتهای مربوط به تست نفوذ مجاز و با دارا بودن مجوز مناسب استفاده شوند. درک تبعات قانونی و اخلاقی استفاده از این فریم ورک های پایتون حائز اهمیت است.

### فریم ورک های پایتون برای اندروید

هرچند **زبان برنامه نویسی جاوا** در درجه اول انتخاب مناسب تری برای توسعه اندروید محسوب می شود، اما پایتون دارای چارچوب هایی برای آن است. برخی از محبوب ترین فریم ورک های پایتون برای اندروید عبارتند از:

- **Kivy**: این فریم ورک برای ساخت برنامه های موبایل چندسکویی (Cross-Platform) با رابط کاربری گرافیکی (GUI) استفاده می شود. در Kivy، از زبان kv برای طراحی رابط کاربری و از OpenGL برای رندرینگ استفاده می شود.
- **Beeware**: این چارچوب برای ایجاد برنامه های اندروید ساده و سبک به کار می رود. Beeware از زبان Python به هدف نوشتن رابط کاربری و منطق برنامه بهره می برد.
- **Chaquopy**: این فریم ورک به شما امکان تعبیه کردن کدهای پایتون در برنامه های اندروید را می دهد. Chaquopy برای توسعه افزونه ها و توابع سفارشی برای اپ های اندروید مفید است.
- **DroidScript**: با استفاده از DroidScript می توان برنامه های اندروید را از طریق زبان JavaScript و HTML5 ایجاد کرد. DroidScript برای توسعه اپلیکیشن های ساده و سریع ایده آل است.

## بهترین فریم ورک های پایتون

پاسخ به این سؤال که بهترین فریمورک های پایتون کدام موارد هستند، مستقیماً به ترجیحات و نیازمندی‌های دقیق پروژه شما وابسته است. با این وجود، برخی از بهترین گزینه‌های پیش رو را به شما معرفی می‌کنیم تا روند تصمیم‌گیری کمی برایتان ساده‌سازی شود. برای انتخاب بهترین فریمورک های پایتون، موارد زیر را در نظر داشته باشید.

- **پیچیدگی پروژه:** برای وب‌اپلیکیشن‌های پیچیده‌تر، جنگو می‌تواند یکی از انتخاب‌های مناسب در توسعه وب باشد. اگر نیاز به کنترل بیشتری روی پروژه دارید و مقیاس پروژه شما در ابعاد گسترده نیست، Flask گزینه معقول‌تر محسوب می‌شود.
- **تجربه توسعه‌دهنده:** در شرایطی که تیم فنی شما با ساختار و اصول کار با جنگو آشنا باشد، ممکن است انتخاب آن برای وب‌اپلیکیشن از نظر زمانی مناسب‌تر از Flask باشد.
- **نیازمندی‌های پروژه:** به این موضوع توجه کنید که پروژه شما دقیقاً به چه مواردی نیاز دارد؟ به‌عنوان مثال، آیا پروژه شما یادگیری عمیق را شامل می‌شود؟ در چنین شرایطی، تنسرفلو یا پایتورچ برایتان کاربردی خواهند بود.

درنهایت، به این نکته توجه کنید که معیارهای مختلفی برای تعیین بهترین فریمورک های پایتون وجود دارند و نمی‌توان یک پاسخ یکتا و قابل‌تعمیم به تمامی شرایط برای این پرسش در نظر گرفت.

## کلام آخر: بررسی انواع فریمورک های پایتون

در این مطلب به معرفی انواع فریمورک های پایتون پرداختیم و مزیت استفاده از آن‌ها را تشریح کردیم. استفاده از چارچوب‌ها مزیت‌هایی مانند افزایش کارایی، توسعه استانداردسازی شده، کاهش خطا، بهبود امنیت و تسریع فرآیند پروتوتایپینگ را به‌دنبال دارد و به توسعه‌دهندگان این امکان را می‌دهد که زمان و انرژی خود را روی توسعه فیچرهای جذاب برای اپلیکیشن اختصاص دهند.