

به نام خدای رنگین کمان

جنگو چیست و بررسی اجمالی بوت کمپ

جنگو یکی از فریم ورک های زبان پایتون برای ساختن و نگه داری (Maintaining) برنامه های تحت وب با کیفیت هست، کتابخانه جنگو از طریق حذف وظایف تکراری، فرایند توسعه برنامه های کاربردی تحت وب را ساده و از لحاظ زمانی مقرون به صرفه می کند.

برای درک بهتر مطالب بهتر از ابتدا با مفاهیم زیر آشنا شده باشید :

- « برنامه نویسی شیء گرا » (OOP Programing)
- « برنامه نویسی رویه ای » (Procedural Programming)
- « ساختارهای کنترلی » (Control Structures)
- « ساختارهای داده » (Data Structures)
- « متغیرها » (Variables)
- « کلاس ها » (Classes)
- « اشیاء » (Objects) و سایر موارد داشته باشند.

تاریخچه جنگو

- در سال 2003 :جنگو به عنوان یک پروژه داخلی در روزنامه Lawrence Journal-World ، توسط Adrian Holovaty و Simon Willison شکل گرفت.
- در سال 2005 :پدیدآوردگان کتابخانه جنگو، این پلتفرم برنامه نویسی را در جولای 2005 عرضه کردند و به احترام یاد و خاطره موسیقیدان و نوازنده ای به نام Django Reinhardt ، نام Django را برای آن انتخاب کردند.
- در سال 2005 :جنگو در عمل نشان داد که توانایی مدیریت سایت های پرتراфик را به راحتی دارد.
- در حال حاضر :جنگو هم اکنون یکی از پروژه های برنامه نویسی «منبع باز (Open Source)» محسوب می شود که برنامه نویسان و توسعه دهندگان زیادی از سراسر دنیا، برای افزودن قابلیت های نرم افزاری جدید به آن تلاش می کنند.

فلسفه طراحی جنگو

استقلال نسبی مولفه های این پلتفرم (ماژولار بودن)

کد نویسی کمتر ، که مزایا و معایب خاص خودش را دارد

اصل عدم تکرار (Don't Repeat Yourself | DRY) توصیه میشود ماژول های مختلف در یک بخش

نوشته شوند و تکرار نشوند.

توسعه سریع و طراحی تمیز

مزایای جنگو

پشتیبانی از ORM (Object-Relational Mapping) برای دیتابیس های SQL و NoSQL

پشتیبانی از چند زبانه بودن (Multilingual Support)

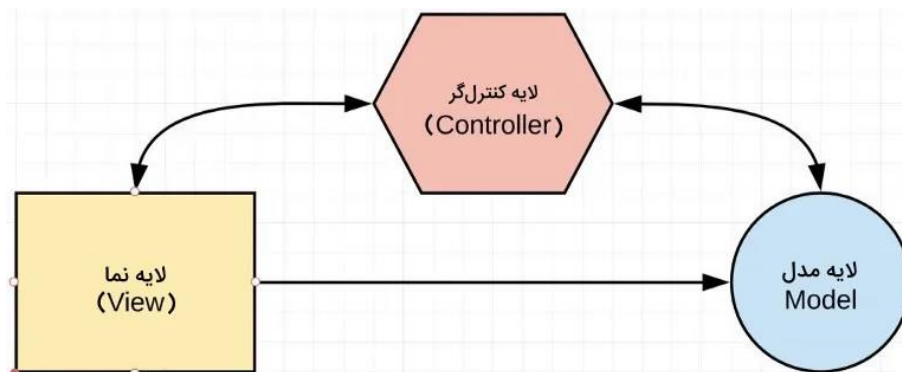
پشتیبانی از پلتفرم ها و چارچوب های مختلف توسعه وب مانند : Ajax، RSS، Caching و سایر موارد

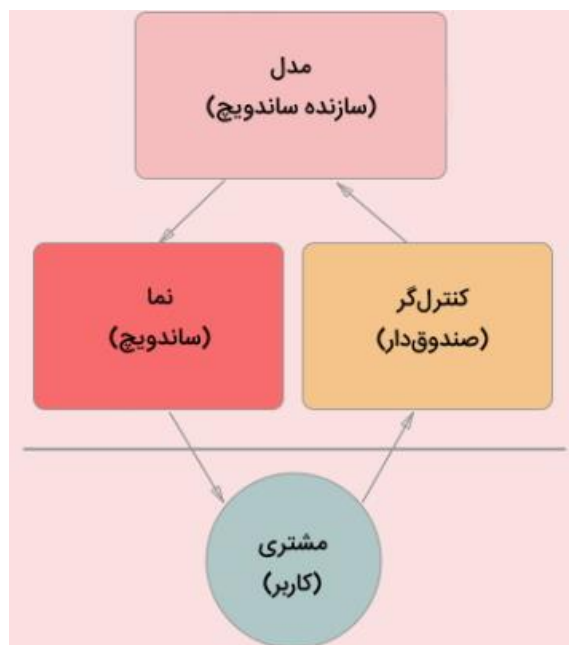
واسط کاربری و GUI برای فعالیتهای مدیریتی وبسایت و مؤلفه های آن (پنل ادمین)

محیط توسعه (Development Environment)

ساختار کلی جنگو

جنگو برای پیاده سازی مدل MVC از معماری MVT استفاده میکند. در الگوی معماری MVC لایه نما (View) تنها می داند که چه طور باید رابط کاربری را نمایش دهد. اگر برای نمایش رابط کاربری نیاز به هر نوع داده ای باشد، لایه View آن را از لایه مدل دریافت خواهد کرد. اما نما مستقیماً از مدل برای پیدا کردن داده ها سوال نخواهد کرد؛ لایه نما این کار را از طریق لایه Controller (لایه کنترلر) انجام می دهد.





مدل MVT :

معماری MVT، کمی متفاوت‌تر از معماری MVC است؛ در اصل، تفاوت بنیادی میان این دو معماری این است که در مدل MVT، کتابخانه Django مدیریت مؤلفه Controller را بر عهده می‌گیرد (مؤلفه Controller، کد نرم‌افزاری است که تعامل میان مؤلفه‌های Model و View را کنترل می‌کند). پس چون جنگو قسمت کنترل را به عهده دارد مؤلفه ی Template شکل می‌گیرد. مؤلفه Template، یک فایل HTML است که با (Django Template Language | DTL) ترکیب شده است.

