



پایتون یکی از معدود زبان‌های برنامه‌نویسی پرفروغ و محبوب جهان است. تا به امروز رقبای مختلفی برای این زبان پیدا شده‌اند که هنوز هیچ‌یک از آن‌ها موفق نشده‌اند پایتون را با چالش جدی روبه‌رو کنند. برنامه‌نویسان مسلط به پایتون می‌توانند در حوزه‌های هوش مصنوعی (یادگیری ماشین و شبکه‌های عمیق عصبی)، برنامه‌نویسی وب و علم داده‌ها، داده‌کاوی، برنامه‌نویسی دسکتاپ (زیاد توصیه نمی‌شود) و... برنامه‌های کاربردی قدرتمندی ایجاد کرده و بدون مشکل خاصی برای آگهی‌های شغلی مرتبط درخواست بدهند. در خوب بودن پایتون شکمی نیست و به‌عنوان یک توسعه‌دهنده برنامه‌های کاربردی می‌توانید برای کارهای مختلفی از پایتون استفاده کنید، اما برنامه‌نویسان و توسعه‌دهندگانی که تازه به دنیای برنامه‌نویسی پایتون وارد شده‌اند، این پرسش برایشان ایجاد شده که دقیقاً با پایتون چه کاری می‌توان انجام داد؟ پایتون از عهده انجام چه کارهایی بر می‌آید؟ چگونه می‌توان از پایتون استفاده کرد؟ آیا پایتون این ارزش را دارد که وقت خود را صرف یادگیری آن کنید؟ هرگاه تصمیم می‌گیرید یک‌زبان برنامه‌نویسی همچون پایتون را یاد بگیرید، پرسش‌هایی این‌چنینی به ذهن شما خطور می‌کند. توسعه‌دهندگان پایتون در پاسخ به این پرسش‌ها می‌گویند: «با یادگیری پایتون می‌توان برنامه‌های کاربردی متعددی ایجاد کرد. پایتون در سه حوزه توسعه برنامه‌های کاربردی تحت‌وب، علم داده‌ها (یادگیری ماشین، تحلیل داده‌ها و تجسم‌سازی داده‌ها) و اسکریپت‌نویسی عملکردی ایده‌آل دارد.» که در این مقاله درباره هر یک از این حوزه‌ها توضیح کوتاهی ارائه خواهیم کرد.

توسعه برنامه‌های کاربردی تحت وب

چند وقتی است، چهارچوب‌های وب مبتنی بر پایتون همچون Django و Flask با اقبال عمومی روبه‌رو شده‌اند. آگهی‌های شغلی مرتبط با برنامه‌نویسی وب و پایتون که تأکید خاصی روی Django و Flask دارند، گواه این مطلب هستند. چهارچوب‌های تحت وب به شما کمک می‌کنند در پایتون کدهای سمت سرور (کدهای بک‌اند) بنویسید. کدهایی که سمت سرور اجرا می‌شوند، با کدهایی که سمت کلاینت (کدهای فرانت‌اند) و روی دستگاه‌هایی شبیه دستگاه‌ها و مرورگرها اجرا می‌شوند، تفاوت دارند. اگر با تفاوت کدهای فرانت‌اند و بک‌اند آشنا نیستید، پی‌نوشت این مقاله را مطالعه کنید. (پرونده ویژه تکنیک‌های طراحی وب شماره 202 ماهنامه شبکه نیز اطلاعات جامع‌تری در اختیار شما قرار می‌دهد.)

چرا ما به یک چهارچوب وب نیاز داریم؟

یک چهارچوب تحت وب به ما اجازه می‌دهد منطقی را که در پس‌زمینه یک برنامه کاربردی قرار دارد و به آن بک‌اند می‌گوییم، به شکل ساده‌تری ایجاد کنیم. منطلق بک‌اند به نداشت آدرس‌های مختلف اینترنتی به قطعه کدهای پایتون، نحوه تعامل با بانک‌های اطلاعاتی و ایجاد فایل‌های HTML (فایل‌هایی که کاربران آن‌ها را روی مرورگرهایشان مشاهده می‌کنند) اشاره دارد.

کدامیک از چهارچوب‌های وب پایتون را باید استفاده کرد؟

Django و Flask دو مورد از محبوب‌ترین چهارچوب‌های تحت وب برای پایتون هستند. بهتر است برای شروع کار یکی از این دو چهارچوب را انتخاب کنید.

چه تفاوتی میان Django و Flask وجود دارد؟

Flask انعطاف‌پذیری، سادگی و کنترل دقیق بر همه جوانب برنامه‌نویسی وب در پایتون را ارائه می‌کند. این چهارچوب به توسعه‌دهندگان اجازه می‌دهد، تصمیم بگیرند، چگونه می‌خواهند اشیا را پیاده‌سازی کنند. Django تجربه‌ای فراتر از کار کردن با یک چهارچوب ساده را ارائه می‌کند. به عبارت دقیق‌تر، شما یک پانل مدیریتی، رابط‌های بانک اطلاعاتی، نگاشت رابطه به شی (Object-Relational Mapping) و یک ساختار دایرکتوری برای برنامه و پروژه‌های خود خواهید داشت. یک توسعه‌دهنده وب ممکن است Flask را به دلایل زیر انتخاب کند: - اگر روی تجربه و فرصت‌های جدید یادگیری متمرکز هستید یا به دنبال آن هستید که کنترل بیشتری روی مولفه‌هایی که استفاده می‌کنید، داشته باشید (چه نوع بانک‌های اطلاعاتی را استفاده کرده و چگونه با بانک‌های اطلاعاتی به تعامل بپردازید) باید Flask انتخاب اول شما باشد. Django زمانی مناسب است که روی یک محصول (نهایی) متمرکز شده‌اید. پروژه‌هایی شبیه یک سایت خبری، فروشگاه الکترونیکی یا وبلاگ به روشی مستقیم برای برنامه‌نویسی و سازمان‌دهی وظایف نیاز دارند. در نتیجه Django گزینه مناسبی برای این‌گونه پروژه‌ها است. به عبارت دیگر، اگر یک مبتدی هستید، شاید Flask انتخاب خوبی است، زیرا مولفه‌های کمتری دارد. همچنین، اگر به دنبال سفارشی‌سازی بیشتری هستید، Flask گزینه مناسب‌تری است. از سوی دیگر، اگر به دنبال آن هستید تا کارها را به شکل مستقیم انجام داده و سرعت عمل بیشتری در انجام پروژه‌ها داشته باشید، Django گزینه بهتری است.

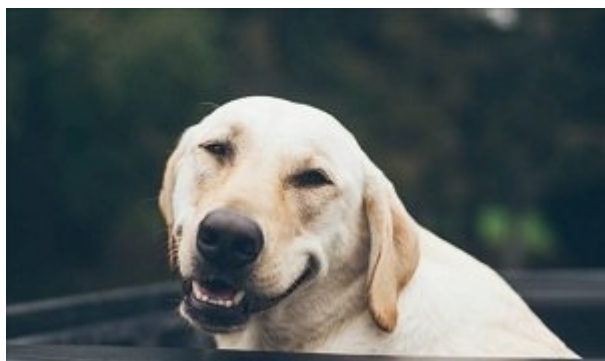
مطلب پیشنهادی



منطق با جدیدترین نگارش
آموزش رایگان زبان برنامه‌نویسی پایتون (python) - مقدمه

علم داده‌ها (یادگیری ماشین، تحلیل داده‌ها و تجسم‌سازی داده‌ها)

در ابتدا بهتر است کمی در مورد یادگیری ماشین بدانیم. برای توضیح این‌که یادگیری ماشین چیست، مثالی ساده می‌آوریم: برنامه‌ای را تصور کنید که باید اشیا و موجودات درون یک تصویر را تشخیص دهد. برای مثال، تصویری شبیه شکل 1 باید در اختیار برنامه قرار بگیرد و برنامه تشخیص دهد سگی درون تصویر است.



شکل 1. تصویر یک سگ که در اختیار برنامه قرار می‌گیرد تا تشخیص دهد که درون تصویر چه چیزی قرار دارد.

در تصویر دیگری برنامه باید میزی را که درون شکل 2 قرار دارد، تشخیص دهد.



2. اگر پیکسل‌های قهوه‌ای رنگ روشن زیادی در تصویر وجود داشته باشند، ممکن است قطعه کد شما به اشتباه یک شی را یک سگ تشخیص دهد. شاید برای حل این مشکل راهکاری این‌چنینی پیشنهاد دهید: اگر قطعه کد من بتواند لبه‌های مستقیمی را که درون یک تصویر قرار دارد، شناسایی کند، در نتیجه می‌تواند فرق یک میز با یک سگ را تشخیص دهد. در سناریو دیگری، ممکن است سگ درون یک تصویر کاملاً سفید بوده و موهای قهوه‌ای نداشته باشد. اگر شی درون تصویر لبه‌های تیزی نداشته باشد و یک میز لبه‌های گردی داشته باشد، برای این مشکل چه راهکاری پیشنهاد می‌کنید؟ این درست همان نقطه‌ای است که یادگیری ماشین به میدان وارد می‌شود. یادگیری ماشین الگوریتمی را پیاده‌سازی می‌کند که می‌تواند الگوی درون یک داده ورودی را تشخیص دهد.

اگر به یک الگوریتم یادگیری ماشین 1000 تصویر مرتبط با یک سگ و 1000 تصویر مرتبط با یک میز را بدهید، الگوریتم خواهد توانست تفاوت‌های میان یک سگ و میز را تشخیص دهد. زمانی‌که این آموزش به پایان رسید، در مرحله بعد اگر تصویر جدیدی از یک سگ یا میز به الگوریتم بدهید، الگوریتم به‌خوبی خواهد توانست یک سگ یا میزی را که درون یک تصویر قرار دارد، تشخیص دهد. فکر می‌کنم این شیوه آموزش تا حدودی شبیه زمانی است که سعی داریم مفاهیم جدیدی را به یک کودک آموزش بدهیم. این کار ممکن است از طریق به‌کارگیری چند نمونه انجام شود: شما با صراحت به یک کودک نمی‌گویید هر چیز خردار با موهای قهوه‌ای ممکن است یک سگ باشد، بلکه با نشان دادن تصویر می‌گویید این یک سگ است، این هم یک سگ است، این یک میز است و این هم میزی دیگری است. الگوریتم‌های یادگیری ماشین تا حد زیادی به همین شیوه کار می‌کنند. سامانه‌های تشخیص چهره، صدا، سامانه‌های پیشنهاددهنده‌ای که آمازون، یوتیوب و نت‌فلیکس از آن استفاده می‌کنند در کنار سایر برنامه‌های مشابه بر مبنای این تکنیک کار می‌کنند. از معروف‌ترین الگوریتم‌های یادگیری ماشین می‌توان به شبکه‌های عصبی (Neural Networks)، یادگیری ژرف (Deep learning)، ماشین‌های بردار پشتیبان (Support Vector Machines) و الگوریتم‌های جنگل تصادفی (Random Forest) اشاره کرد. هر یک از این الگوریتم‌ها می‌تواند برای حل مشکل تشخیص چهره (همانند مثالی که به آن اشاره شد) کاربرد داشته باشد.

مطلب پیشنهادی



یادگیری ماشینی در خدمت شبکه‌های مبتنی بر تصمیم
آغاز عصری جدید در شبکه‌ها و مراکز داده با **Based NetworkingIntent**

پایتون در خدمت یادگیری ماشین

کتابخانه‌ها و چهارچوب‌های محبوب یادگیری ماشین ویژه **پایتون** کار برنامه‌نویسان فعال در حوزه هوش مصنوعی را ساده کرده‌اند. تانسورفلو (Tensorflow) و سایکیت لرن (Scikit-learn) از محبوب‌ترین کتابخانه‌ها هستند. سایکیت لرن، یک کتابخانه متن‌باز است که بر پایه Scipy, NumPy و matplotlib طراحی شده است. این کتابخانه قابلیت‌ها و ابزارهای کاربردی برای داده‌کاوی، تجزیه و تحلیل داده‌ها و... در اختیار توسعه‌دهندگان قرار می‌دهد. کتابخانه‌ای که به شکل پیش‌فرض برخی از الگوریتم‌های یادگیری ماشین درون

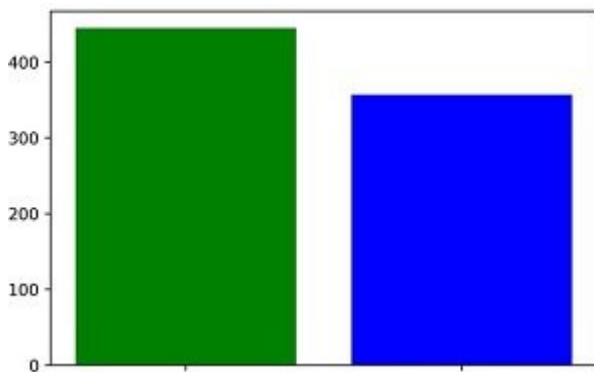
آن قرار دارد. تانسورفلو، یک کتابخانه سطح پایین است که به شما امکان ساخت الگوریتم‌های یادگیری ماشین سفارشی را می‌دهد. بهتر است برای شروع کار و آشنایی با پروژه‌های یادگیری ماشین کار را با ساینکیت لرن آغاز کنید و در صورت کار روی مسائلی که مرتبط با کارایی است، تانسورفلو را انتخاب کنید.

چگونه باید یادگیری ماشین را فرا گرفت؟

برای یادگیری اصول اولیه یادگیری ماشین، آموزش‌های دانشگاه استنفورد یا Caltech را دنبال کنید. این آموزش‌ها روی یوتیوب وجود دارند. البته روی آپارات نیز آموزش‌های یادگیری ماشین خوبی پیدا می‌شوند، اما در صورت مسلط بودن به زبان انگلیسی آموزش‌های استنفورد اصولی و منطبق بر سرفصل‌های استاندارد است. در نظر داشته باشید که برای دانستن برخی از مباحث مطرح‌شده در این دوره‌ها باید در حوزه ریاضیات و جبر خطی دانش اولیه داشته باشید. پس از یادگیری اصول اولیه و مباحث پایه باید کار عملی و تمرین کردن را آغاز کنید. وبسایت Kaggle نقطه شروع خوبی برای افرادی است که دوست دارند برای یک مشکل در محیطی چالشی و رقابتی راه‌حلی ارائه کنند. این سایت برای افراد مبتدی آموزش‌های خوبی دارد.

علم داده‌ها و تجسم‌سازی/مصورسازی داده‌ها چیست؟

برای آن‌که تصویر درستی از این دو مفهوم به دست آورید، اجازه دهید این دو فناوری پر تقاضا را با ذکر مثالی بررسی کنیم. تصور کنید برای شرکتی کار می‌کنید که برخی از محصولات را به شکل آنلاین به فروش می‌رساند. به‌عنوان یک تحلیل‌گر داده‌ها ممکن است نموداری شبیه شکل 3 رسم کنید.

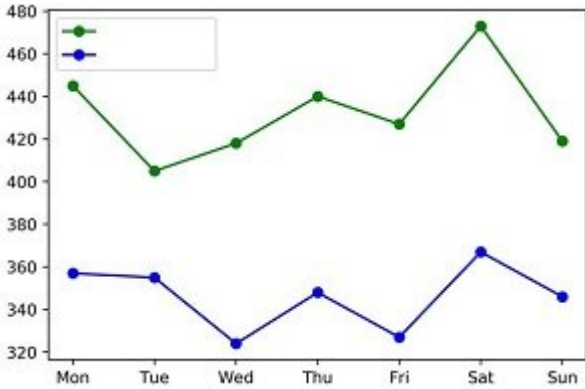


شکل 3. نمودار می‌سازد

براساس نمودار ترسیم‌شده متوجه می‌شویم در این یکشنبه خاص مردان بیش از 400 عدد از یک محصول را خریداری کرده‌اند، درحالی‌که این رقم برای زنان 350 عدد بوده است. به‌عنوان یک تحلیل‌گر داده‌ها ممکن است چند دلیل احتمالی برای این تفاوت ارائه کنید: 1. مردان نسبت به زنان علاقه بیشتری به این محصول داشته‌اند؛ 2. اندازه نمونه خیلی کوچک است و این تفاوت ممکن است به شکل تصادفی به وجود آمده باشد؛ 3. مردان به دلایلی این محصول را تنها در یکشنبه‌ها خریداری می‌کنند. برای درک این‌که کدام‌یک از این دلایل ممکن است درست باشد، ممکن است گراف دیگری همانند شکل 4 ترسیم کنید.

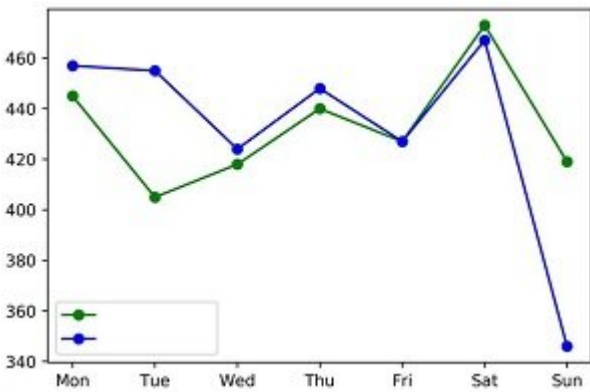
در گراف چهار به‌جای آن‌که تنها داده‌های یکشنبه را نشان دهیم، داده‌های یک هفته کامل را ترسیم کردیم. شکل 4 به‌وضوح نشان می‌دهد در وضعیت تقریباً پایداری در روزهای مختلف تفاوت‌هایی وجود دارد. با نگاه کردن به این گراف و تحلیلی مختصر، یک دلیل قانع‌کننده ارائه می‌کنید که مردان نسبت به زنان علاقه بیشتری به این محصول دارند.

شکل 4. نموداری شبیه به شکل 5 را مشاهده کنید، چه توضیحی ارائه می‌کنید؟ شما ممکن است این‌گونه استدلال کنید:



حال اگر نموداری شبیه شکل 5 را مشاهده کنید، چه توضیحی ارائه می‌کنید؟ شما ممکن است این‌گونه استدلال کنید: «مردان بنا به دلایلی تمایل داشته‌اند تنها روز یکشنبه این محصول را خریداری کنند یا شاید تنها یک تصادف بوده که مردان روز یکشنبه این محصول را بیشتر از زنان خریداری کرده‌اند.»

شکل 5. نموداری شبیه به شکل 4 را مشاهده کنید، تنها یک مثال ساده از یک تجزیه و تحلیلی است که ممکن است در دنیای واقعی نظاره‌گر آن باشیم. وای کی سوگی (YK Sugi)، تحلیل‌گر اسبق شرکت‌های گوگل و مایکروسافت نیز یک چنین تحلیل‌هایی برای این دو شرکت انجام داده است. او می‌گوید: «مایکروسافت و گوگل چنین تحلیل‌هایی را انجام می‌دهند، منتها فرآیند تحلیل در این دو شرکت پیچیده‌تر است. زمانی که در شرکت گوگل مشغول به کار بودم، از پایتون برای انجام چنین کاری استفاده می‌کردم، در حالی که در شرکت مایکروسافت از جاوا اسکریپت استفاده می‌کردم. در هر دو شرکت از SQL برای واکنشی داده‌ها از درون بانک‌های اطلاعاتی استفاده می‌کردم. در ادامه پایتون یا Matplotlib (در گوگل) یا جاوا اسکریپت و D3.js (در مایکروسافت) را برای تجسم‌سازی تجزیه و تحلیل داده‌ها استفاده می‌کردم.»



مایکروسافت از جاوا اسکریپت استفاده می‌کردم. در هر دو شرکت از SQL برای واکنشی داده‌ها از درون بانک‌های اطلاعاتی استفاده می‌کردم. در ادامه پایتون یا Matplotlib (در گوگل) یا جاوا اسکریپت و D3.js (در مایکروسافت) را برای تجسم‌سازی تجزیه و تحلیل داده‌ها استفاده می‌کردم.»

تجزیه و تحلیل داده‌ها و تجسم‌سازی با پایتون

یکی از محبوب‌ترین کتابخانه‌ها برای تجسم‌سازی است. یادگیری کتابخانه فوق به معنای این است که شما در ادامه می‌توانید از کتابخانه‌های دیگری شبیه به seaborn که بر پایه این کتابخانه کار می‌کنند بدون مشکل استفاده کنید.

چگونه باید تجزیه و تحلیل داده‌ها و تجسم‌سازی با پایتون را یاد بگیریم؟

در اولین گام باید اصول اولیه تجزیه و تحلیل داده‌ها و تجسم‌سازی را یاد بگیرید. درس‌هایی که سایت Pluralsight ارائه کرده نقطه شروع مناسبی هستند.

<https://goo.gl/C9NoHt>

پس از یادگیری اصول پایه تحلیل داده‌ها و تجسم‌سازی، باید اصول اولیه مباحث آماری را یاد بگیرید. Coursera و Khan Academy اطلاعات مفیدی در اختیاران قرار می‌دهند. با کمی جست‌وجو سایت‌های آموزشی فارسی خوبی می‌توانید پیدا کنید. اما دو منبع ذکر شده اطلاعات جامعی در اختیاران قرار می‌دهند.

اسکرپت‌نویسی اسکرپت‌نویسی چیست؟

اسکرپت‌نویسی به فرآیند نوشتن برنامه‌های کوچکی اشاره دارد که برای خودکارسازی انجام کارهای ساده استفاده می‌شوند. تصور کنید در شرکتی کار می‌کنید یک سامانه پشتیبان ایمیلی برای پاسخ‌گویی به پرسش‌های مشتریان وجود

دارد. در این شرکت وظیفه شما شمارش تعداد ایمیل‌ها بر مبنای کلیدواژه‌ها است. برای انجام این کار باید ایمیل‌های دریافتی را تحلیل کنید. می‌توانید این کار را به شیوه دستی انجام دهید، اما به‌جای این کار یک برنامه ساده اسکریپتی می‌نویسید که به شکل خودکار این کار را انجام دهد. ممکن است از رویی برای انجام این کار استفاده کنید، اما بدون تردید **پایتون** زبان مناسبی برای انجام این کار است. **پایتون** به دلیل ترکیب نحوی ساده‌ای که دارد، گزینه ایده‌آلی برای انجام این کار است. این زبان به شما اجازه می‌دهد با کمترین کد ممکن برنامه‌های این‌چنینی را نوشته و به‌سرعت آن‌ها را آزمایش کنید.



در ارتباط با برنامه‌های توکار وضعیت چگونه است؟

پایتون با بردهایی همچون رزبری پای به‌خوبی کار می‌کند. در نتیجه برای علاقه‌مندان به مباحث سخت‌افزاری و به‌ویژه کامپیوترهای مینیاتوری (رزبری پای) **پایتون** گزینه مناسبی است.

درباره ساخت بازی‌ها وضعیت چگونه است؟

برای طراحی و توسعه بازی‌ها از کتابخانه‌ای که PyGame نامیده می‌شود، استفاده کنید، اما در این حوزه نمی‌توانیم به محبوب‌ترین موتور ساخت بازی اشاره کنیم. می‌توانید از این کتابخانه برای پروژه‌هایی که جنبه سرگرمی دارند استفاده کنید، اما برای توسعه بازی‌های جدی نباید روی آن حساب باز کنید. بهتر است با Unity در سی شارپ کار کنید. Unity یکی از محبوب‌ترین موتورهای ساخت بازی است. این موتور به شما اجازه می‌دهد بازی خود را برای پلتفرم‌های مختلفی همچون ویندوز، اندروید، مک و iOS بسازید.

وضعیت در ارتباط با برنامه‌های دسکتاپی چگونه است؟

از Tkinter و **پایتون** برای برنامه‌های دسکتاپی استفاده کنید، اما این ترکیب انتخاب مطلوبی نیست و بهتر است از جاوا، سی شارپ و سی پلاس پلاس که از محبوب‌ترین گزینه‌های روز هستند، استفاده کنید. برخی شرکت‌ها از جاوااسکریپت برای ساخت برنامه‌های دسکتاپی استفاده می‌کنند. برای مثال، نسخه دسکتاپی اسلک با Electron نوشته‌شده و Electron به شما اجازه می‌دهد از جاوااسکریپت برای ساخت برنامه‌های دسکتاپی استفاده کنید. برخی از کدهای وب قابلیت استفاده دوباره (Reuse) دارند و جاوااسکریپت به شما اجازه می‌دهد از این کدها در برنامه کاربردی خود استفاده کنید.

پایتون 2 یا 3

پایتون 3 به دلیل مدرن‌تر بودن قابلیت‌های کاربردی و به‌روزتری در اختیاران قرار می‌دهد.

پی‌نوشت:

فرض کنید در نظر دارید برنامه‌ای شبیه اینستاگرام بسازید. شما باید کدهای فرانت‌اندی بنویسید که روی دستگاه‌هایی که می‌خواهید از آن‌ها پشتیبانی کنید، اجرا شود. برای این کار ممکن است از سوئیفت برای iOS، جاوا برای اندروید و جاوااسکریپت برای مرورگرهای وب استفاده کنید. هر مجموعه کد روی دستگاه/مرورگر خاص خود اجرا می‌شود. این مجموعه کدها لایه‌بندی برنامه یا همان ظاهر برنامه را تعریف کرده، مشخص می‌کنند دکمه‌ها باید چگونه به نظر برسند، زمانی که روی دکمه‌ها کلیک می‌کنید، چه کاری انجام شود و

در مرحله بعد باید راهکاری برای ذخیره‌سازی اطلاعات و تصاویر کاربران بیابیم. به‌جای آن‌که اطلاعات را روی دستگاه‌های کاربران ذخیره کنید، روی سرور خودتان ذخیره کنید تا دنبال‌کنندگان یک کاربر بتوانند عکس‌ها را مشاهده کنند. این درست همان نقطه‌ای است که کدهای بک‌اند/ کدهای سمت سرور به میدان وارد می‌شوند. به کدنویسی بک‌اند برای انجام کارهای زیر نیاز دارید:

- مشخص کردن این موضوع که چه فردی، فرد دیگری را دنبال کرده است.
 - فشرده‌سازی عکس‌ها برای آن‌که فضای ذخیره‌سازی بهبود یافته‌اشغال نشود.
 - پیشنهاد حساب‌های کاربری و تصاویر مختلف به کاربرانی که تازه عضو شده‌اند.
- همان‌گونه که مشاهده کردید، کدهای بک‌اند و فرانت‌اند عملکردی متفاوت دارند. علاوه بر **پایتون**، گزینه‌های خوب دیگری نیز برای کدنویسی سمت سرور/بک‌اند وجود دارند. Node.JS بر پایه جاوااسکریپت، یکی از این گزینه‌های مطلوب است

تاریخ انتشار:

17 اسفند 1397

نشانی منبع:

<https://www.shabakeh-mag.com/workshop/programming/14704/%D9%BE%D8%A7%DB%8C%D8%AA%D9%88%D9%86-%D8%A8%D8%B1%D8%A7%DB%8C-%D8%A7%D9%86%D8%AC%D8%A7%D9%85-%DA%86%D9%87-%DA%A9%D8%A7%D8%B1%D9%87%D8%A7%DB%8C%DB%8C-%D8%B9%D8%A7%D9%84%DB%8C%D8%8C-%DA%86%D9%87-%DA%A9%D8%A7%D8%B1%D9%87%D8%A7%DB%8C%DB%8C-%D8%AE%D9%88%D8%A8-%D9%88-%DA%86%D9%87-%DA%A9%D8%A7%D8%B1%D9%87%D8%A7%DB%8C%DB%8C-%D8%A8%D8%AF-%D8%A7%D8%B3%D8%AA%D8%9F>