

راهنمای جامع

ورود به دنیای برنامه نویسی



7LEARN

تهیه شده توسط تیم تولید محتوای سون لرن

فهرست محتوا

- 6 پیوند انسان و کامپیوتر چگونه شکل گرفت؟
- 9 برنامه‌نویسی چیست؟
- 10 انواع زبان‌های برنامه‌نویسی
- 11 - زبان‌های سطح پایین
- 11 - زبان‌های سطح بالا
- 13 یادگیری برنامه‌نویسی چه ضرورتی دارد؟
- 14 - درآمد نسبتاً خوب و یا عالی
- 14 - انعطاف‌پذیری
- 15 - تقاضا
- 16 - یادگیری ساده
- 16 رابطه برنامه‌نویسی با دانشگاه
- 17 رابطه برنامه‌نویسی و تکنولوژی
- 18 رابطه برنامه‌نویسی و مهاجرت
- 20 معرفی حوزه‌های برنامه‌نویسی
- 21 برنامه‌نویسی تحت وب
- 22 - برنامه‌نویسی فرانت‌اند (Front-End)
- 23 - معرفی زبان‌های Front-End
- 24 - برنامه‌نویسی بک‌اند (Back-End)

25	- معرفی زبان‌های Back-End
27	برنامه‌نویسی اپلیکیشن موبایل
27	- زبان‌های توسعه اندروید
28	- زبان‌های توسعه iOS
29	برنامه‌نویسی تحت دسکتاپ
32	هوش مصنوعی
34	داده کاوی
35	درباره UI و UX
40	شناخت بهترین زبان برنامه‌نویسی
44	زبان‌های برنامه‌نویسی برتر از نظر بازار کار
46	مهارت‌های برنامه‌نویس حرفه‌ای
51	سخن پایانی

راهنمای جامع

ورود به دنیای برنامه‌نویسی



"همه باید برنامه‌نویسی را یاد بگیرند، زیرا مهارتی است که به شما یاد می‌دهد چگونه **فکر** کنید."

استیو جابز؛ بنیانگذار کمپانی اپل

کتابی که پیش روی شماست چکیده مطالب معتبر حوزه برنامه‌نویسی و توصیه برنامه‌نویسانی است که تجربه‌ی زیادی در این زمینه دارند. این کتاب به همت تیم سون‌لرن گردآوری شده است و می‌تواند راهنمای شما در ورود به دنیای برنامه‌نویسی باشد.

اگر با برخی از موارد ذکر شده در کتاب آشنایی ندارید نگران نباشید. در بخش‌های مختلف لینک مقالات مرتبط به همان قسمت قرار داده شده تا بتوانید اطلاعات بیشتری درباره آن مبحث به دست بیاورید.

امیدواریم این کتاب همان شجاعتی باشد که برای ایجاد یک تحول بزرگ در زندگی، به آن نیاز دارید.

پیوند انسان و کامپیوتر چگونه شکل گرفت؟



بیش از پنجاه سال پیش و در خلال تحقیقات آزمایشگاهی و نظامی، ریاضی‌دانان متوجه شدند که می‌توان از ماشین‌ها (که امروزه آن‌ها را با نام سیستم‌های کامپیوتری و ربات می‌شناسیم)، برای حل بسیاری از مسائل پیچیده استفاده کرد. در آن زمان انجام برخی کارهای ریاضیاتی مانند رمزگشایی پیام‌های مخفیانه نظامی به زمان زیادی احتیاج داشت؛ بنابراین دانشمندان به این فکر افتادند که با ساخت یک ماشین هوشمند، توان محاسباتی و قدرت پردازشی قدرتمندتر و سریع‌تری ایجاد کنند. علاوه بر آن، اطلاعات ذخیره شده در یک

ماشین، بسیار امن هستند و شکنجه و تهدید بر آن‌ها اثری ندارد.

ماشین تورینگ (Turing Machine) یکی از قوی‌ترین ماشین‌هایی بود که در زمان جنگ جهانی دوم توانست با برنامه‌ریزی و کنترل الگوریتمی آلن تورینگ، دانشمند انگلیسی، پیام‌های مخفی ارتش آلمان را رمزگشایی کرده و در مقابل ماشین انیگما آلمانی‌ها پیروز شود.

اقدامات آلن تورینگ در دنیای کامپیوتر، بسیار تاثیرگذار بود و او توانست از اولین پایه‌گذاران نظریه محاسبات عددی و هوش مصنوعی باشد. او همچنین معیار ارزشمندی برای تعریف هوشمندی ماشین ارائه داد: "سزاوارترین معیار برای هوشمند شمردن ماشین، این است که بتواند با عملکرد خود انسان را به گونه‌ای بفریبد، که آن فرد متقاعد شود با یک انسان روبرو است."

پس از این، دانشمندان خیلی زود متوجه شدند که علاوه بر موارد بالا، برای انجام کارهای پیچیده‌تری که فعلاً قادر به انجام آن‌ها نیستند، می‌توانند از ماشین‌ها کمک بگیرند. به شرط آن که زبان ارتباط با ماشین را بدانند. بنابراین به فکر ایجاد

زبان‌هایی برای برنامه‌ریزی ماشین افتادند تا بتوانند از ماشین‌ها به‌طور هدفمند استفاده کنند.

با تکامل ماشین‌ها و ابداع رسمی کامپیوترهای اولیه، رشد و معرفی زبان‌های برنامه‌نویسی نیز آغاز شد. با گذشت زمان و رسیدن به عصر فعلی که استفاده از کامپیوترها، گوشی‌های هوشمند و سایر دستگاه‌های دیجیتال قابل برنامه‌ریزی بسیار آسان شده، این شرایط فراهم شده است تا علاقمندان به این حوزه بتوانند به سادگی زبان‌های برنامه‌نویسی را بیاموزند و به دنیای بیکرانی وارد شوند که آینده‌ای درخشان دارد.

برنامه‌نویسی چیست؟



به تدریج که رویای سپردن کارهای سخت به ماشین‌ها تحقق پیدا کرد، وقت آن رسید تا الگوریتم‌های ریاضیاتی و اصول حل مسئله به شیوه‌ای خاص و منطقی ارائه شوند تا بتوان به کمک آن‌ها، ماشین را برنامه‌ریزی (Program) کرد. این قواعد منطقی به شکل دستورهای اجرایی درآمدند و با کنار هم قرار دادن چند دستور، برنامه موردنظر در ماشین اجرا می‌شد.

به مرور و با پیشرفت الگوریتم‌ها، توان حل مسئله و اجرایی ماشین نیز افزایش پیدا کرد و زبان‌های مختلف برنامه‌نویسی توسعه پیدا کردند. استفاده از دستورات زبان‌ها برای دستیابی

به اهداف خاص و کسب نتیجه موردنظر توسط ماشین، با اصطلاح برنامه‌نویسی (Programming) شناخته می‌شود.

بیشتر بخوانید: آشنایی کامل تر با برنامه‌نویسی

انواع زبان‌های برنامه‌نویسی



برنامه‌نویسی با مجموعه‌ای از دستورات بسیار ساده انجام می‌شد که این دستورات به مرور زمان تکمیل‌تر شده و تحت عنوان "زبان برنامه‌نویسی" معرفی شدند. در ادامه این بخش درباره دو دسته مهم از زبان‌ها یعنی سطح پایین و سطح بالا صحبت می‌کنیم.

زبان‌های سطح پایین

زبان‌های سطح پایین به صورت مستقیم با پردازنده رایانه ارتباط دارند و با استفاده از آن‌ها می‌توان دستورات پایه برنامه‌نویسی را اجرا کرد، مانند زبان ماشین و اسمبلی.

هر رایانه، به طور مستقیم فقط زبان ماشین خود را درک می‌کند. درک زبان ماشین برای انسان خسته کننده و بسیار مشکل است زیرا شباهت زیادی به نماد ریاضی معمولی یا زبان انسانی ندارد و کدهای آن از یک رایانه به رایانه دیگر متفاوت است. به همین دلیل کد ماشین تقریباً هیچ‌گاه برای نوشتن برنامه‌ها مورد استفاده قرار نمی‌گیرد اما زبان اسمبلی کاربرد بیشتری دارد.

زبان‌های سطح بالا

در مراحل پیشرفته‌تر زبان‌های سطح بالا توسعه پیدا کردند که امکان برنامه‌نویسی برنامه‌های کامپیوتری، بازی‌ها و نرم‌افزارهای موبایل را برای ما فراهم می‌کنند. امروزه زبان‌های سخت‌افزاری و نرم‌افزاری طرفداران زیادی دارند و در حوزه‌های متنوعی از جمله برق، مکانیک، رباتیک، توسعه نرم‌افزار، برنامه‌نویسی بُردهای صنعتی و غیره استفاده می‌شوند.

زبان‌های سطح بالا برای انسان قابل فهم‌تر هستند زیرا از دستوراتی شبیه به عبارات انگلیسی برای دستورالعمل‌های خود استفاده می‌کنند. منطق اصلی طراحی زبان‌های سطح بالا، راحت‌تر کردن فرآیند برنامه‌نویسی برای برنامه‌نویسان بوده است. زبان‌های محبوبی مانند پایتون، جاوا اسکریپت، جاوا، C و غیره که در توسعه نرم‌افزار کاربرد دارند، از زبان‌های سطح بالا محسوب می‌شوند.

بیشتر بخوانید: [آشنایی بیشتر با انواع زبان‌های برنامه‌نویسی](#)

پرطرفداترین نوع زبان‌های برنامه‌نویسی، زبان‌های سطح بالا هستند که در ادامه به زیرشاخه‌های مهم آن خواهیم پرداخت.

یادگیری برنامه‌نویسی چه ضرورتی دارد؟



امروزه اغلب افرادی که که علاقه‌ای به مشاغل سنتی و یا اداری ندارند به فعالیت در حوزه تکنولوژی ترغیب می‌شوند که البته امری طبیعی به نظر می‌رسد زیرا تحولات جهان سرعت گرفته و تقریباً همه چیز در حال پیوند با تکنولوژی است. هر چند نمی‌توان گفت که همه زمینه‌های شغلی مبتنی بر تکنولوژی به برنامه‌نویسی نیاز دارند اما اگر می‌خواهید حرفی برای گفتن داشته باشید و در تحولات جهان سهمیه باشید، به یادگیری برنامه‌نویسی نیاز دارید.

موفقیت در برنامه‌نویسی به عوامل مختلفی از جمله سطح مهارت برنامه‌نویس، نوع پروژه و یا شرایط همکاری بستگی

دارد اما با فرض وجود شرایطی معقول، شغلی بسیار جذاب و پرچالش است و می‌تواند مزایای چشم‌گیری نیز داشته باشد. مزایایی مانند:

درآمد نسبتاً خوب و یا عالی

یکی از مهم‌ترین عواملی که ما را به یادگیری برنامه‌نویسی تشویق می‌کند، این است که می‌توان با انجام پروژه‌های مختلف در حوزه تخصصی خود، درآمدی شناور داشت. برای مثال، اگر شما با انجام پروژه‌های 1 میلیون تومان درآمد کسب کنید این امید را دارید که پروژه بعدی بتواند درآمدی بیش از 5 میلیون تومان برای شما داشته باشد. به عبارت دیگر، شما با توجه به مهارتی که دارید می‌توانید در حوزه‌های مختلفی فعالیت کنید و طبیعتاً درآمد متفاوتی داشته باشید.

انعطاف‌پذیری

اگر برنامه‌نویس باشید به شرکت یا مکان خاصی برای کار کردن نیاز ندارید بلکه می‌توانید از هر جای دنیا پروژه‌های خود را انجام دهید. به همین دلیل برنامه‌نویسی یکی از مشاغل ایده‌آل برای فریلنسرها است و برنامه‌نویسان می‌توانند از

سراسر دنیا با بهترین کمپانی‌ها همکاری کرده و روی پروژه‌های بزرگ کار کنند.

این انعطاف‌پذیری، علاوه بر کاهش هزینه برای کارفرمایان، برنامه‌نویسی را به یک مهارت امن تبدیل می‌کند. زیرا در بحران‌های مختلفی که در طول سالیان گذشته اتفاق افتاده، برنامه‌نویسان کمتر از سایر صنف‌ها شغل خود را از دست داده‌اند.

تقاضا

برنامه‌نویسی پیوندی عمیق با تکنولوژی دارد و از آنجا که فناوری همواره رو به جلو حرکت می‌کند، تقاضا برای برنامه‌نویسان حرفه‌ای نیز همواره در حال افزایش است. این تقاضا امکان مهاجرت در دنیای برنامه‌نویسی را نیز فراهم می‌کند. به این معنی که اگر با توسعه زبان‌های جدید حوزه فعالیت شما با رکود مواجه شد، می‌توانید به راحتی با استفاده از دانش پایه خود در علم برنامه‌نویسی به حوزه دیگری وارد شوید، مهارت کسب کنید و شغلی جدید را بر عهده بگیرید.

بنابراین برنامه‌نویس شدن خیال شما را راحت می‌کند و به شما اجازه می‌دهد در دنیای تکنولوژی پیش بروید و به عمیق‌ترین

حوزه‌هایی که امروزه بر زندگی بشر بسیار تاثیرگذار هستند، وارد شوید.

یادگیری ساده

یادگیری برنامه‌نویسی به مهارت اولیه خاصی نیاز ندارد و همه افراد به شرط علاقه و تلاش می‌توانند به این حوزه وارد شوند. شما برای این که تبدیل به یک برنامه‌نویس شوید نیازی به مدرک دانشگاهی و تحصیلات آکادمیک ندارید و همین دلیل باعث شده است تا کودکان و نوجوانان زیادی قبل از ورود به دانشگاه و حتی در دوره دبیرستان به این حوزه علاقه‌مند شوند.

رابطه برنامه‌نویسی با دانشگاه

برای برنامه‌نویس شدن لازم نیست حتما به دانشگاه بروید بلکه می‌توانید با استفاده از دوره آموزشی معتبر و یا به صورت خودآموز، مهارت‌های لازم را کسب کنید. اما اگر هدف شما این است که یک مهندس نرم افزار باشید و به حوزه‌های تحقیقاتی و مطالعاتی وارد شوید، پس تحصیل در رشته علوم و مهندسی کامپیوتر ضروری به نظر می‌رسد. تحصیل در این رشته امکان فعالیت‌های آکادمیک و یا مهاجرت را برای شما تسهیل می‌کند و به شما بینش همه جانبه‌ای می‌بخشد در حالی که طی کردن

این مراحل، بدون دانشگاه کمی سخت‌تر و زمان‌بر تر خواهد بود.

با این حال همه چیز به خواسته شما بستگی دارد. اگر به برنامه‌نویسی فقط به دلیل کسب درآمد بیشتر و یا ورود به حوزه‌های تخصصی‌تر علاقه دارید و به دنبال فعالیت‌های پژوهشی و آکادمیک نیستید، بدون تحصیلات دانشگاهی مرتبط و فقط با استفاده از منابع معتبر می‌توانید به آن دست پیدا کنید.

بیشتر بخوانید: معرفی کامل رشته مهندسی کامپیوتر

رابطه برنامه‌نویسی و تکنولوژی

می‌توان گفت تکنولوژی از پیوند انسان، ماشین و ریاضیات شروع شد که نتیجه آن، ایجاد یک زبان مشترک بین این عناصر یا همان زبان‌های برنامه‌نویسی بود. از آنجا که با گذشت زمان سیستم‌های کامپیوتری و پردازشی پیچیده‌تر شده و حوزه‌های بیشتری را پوشش می‌دهند، یادگیری برنامه‌نویسی می‌تواند پا گذاشتن به دنیایی بدون مرز باشد. وقتی از تکنولوژی حرف می‌زنیم یعنی دنیایی متشکل از گوشی‌های هوشمند، خودروهای مدرن، کامپیوترها و پردازنده‌های قوی، ویدیو

گیم‌ها، ابزارهای پزشکی، انواع نرم‌افزارها و سیستم‌عامل‌ها و همه چیزهایی که به‌طور مستقیم یا غیر مستقیم بخش‌های مختلف زندگی ما را ساده کرده است. این سادگی به معنی تحول در سبک زندگی و دست یافتن به فناوری‌های بیشتر است. با این وجود فکر می‌کنید ساده‌سازی فرآیندهای انسانی پایان داشته باشد؟ احتمالا نه. پس برنامه‌نویسی را می‌توان فعالیتی بدون مرز و محدودیت دانست.

رابطه برنامه‌نویسی و مهاجرت



شاید بتوان گفت عمده مهاجرت برنامه‌نویسان در سال‌های اخیر، مربوط به دانشجویان رشته مهندسی کامپیوتر است که از طریق فرآیندهای آکادمیک و در مقاطع کارشناسی ارشد و

دکتری، اقدام به گرفتن پذیرش و بورسیه تحصیلی کرده‌اند. در این فرآیند شما به یک مدرک دانشگاهی معتبر و ریزنمرات، نمرات آزمون‌های بین المللی مانند TOEFL و GRE و سایر مواردی که توسط دانشگاه مقصد اعلام می‌شوند، احتیاج دارید تا شرایط لازم برای پذیرش فراهم شود. اغلب دانشجویان در سال‌های گذشته، کانادا، آمریکا و آلمان را برای مهاجرت انتخاب کرده‌اند.

حال اگر تحصیلات دانشگاهی لازم را نداشته باشید در برخی موارد امکان **مهاجرت شغلی** نیز وجود دارد. در این فرآیند شما به رزومه‌ای بسیار قوی نیاز دارید تا کمپانی مقصد به استخدام و انجام سایر کارهای اقامتی شما ترغیب شود. البته لازم به ذکر است که متأسفانه در مهاجرت شغلی، محدودیت‌هایی برای برنامه‌نویسان ساکن خاورمیانه وجود دارد.

فرآیند دیگری از مهاجرت در بین افراد زیر 18 سال رایج است که تمایل دارند پس از اتمام دوره دبیرستان و برای مقطع لیسانس در حوزه علوم کامپیوتر اقدام به مهاجرت تحصیلی کنند. در این حالت امکان دریافت بورسیه تحصیلی بسیار کم است و اغلب افراد پس از گرفتن پذیرش، با هزینه شخصی اقدام به تحصیل در دانشگاه‌های خارج از کشور می‌کنند.

اگر علاقه‌مند به مهاجرت در زمینه برنامه‌نویسی به هر طریقی هستید لازم است تا با افراد متخصص و یا باتجربه در این زمینه مشورت کنید و از تمامی شرایط اطلاع پیدا کنید.

معرفی حوزه‌های برنامه‌نویسی

به‌عنوان برنامه‌نویس امکان فعالیت در سه حوزه زیر برای شما فراهم است:

- تحت وب
- تحت دسکتاپ
- اپلیکیشن موبایل

برنامه‌نویسی تحت وب



برنامه نویسی وب، نوشتن برنامه‌هایی است که در وب سرور اجرا شده و توسط کاربران اینترنت قابل استفاده هستند. به این معنی که شما با اتصال به اینترنت و جستجو قادر به استفاده از آن‌ها هستید. این برنامه‌ها توسط زبان‌های برنامه نویسی سمت سرور نوشته می‌شوند و در هر زمانی قابل دسترسی هستند. سایت‌ها و سرویس‌هایی چون Google، Gmail، Yahoo و غیره را می‌توان به عنوان نمونه‌هایی از برنامه‌های تحت وب قلمداد کرد. برنامه‌نویسی تحت وب خود به دو بخش تقسیم می‌شود:

آشنایی با فرانت‌اند

برنامه‌نویسی فرانت‌اند (Front-End)



برنامه‌نویسی Front-End که با نام طراحی وب نیز شناخته می‌شود، شامل برنامه‌نویسی ظاهر یک سایت و همه آن چیزی است که کاربر می‌بیند و با آن ارتباط برقرار می‌کند.

برنامه‌نویسی که تنها به مهارت‌های طراحی وب تسلط دارد به تنهایی نمی‌تواند یک سایت راه‌اندازی کند و باید از یک توسعه‌دهنده بک‌اند کمک بگیرد.

معرفی زبان‌های Front-End

HTML

html اولین زبانی است که باید یاد بگیرید. HTML برای طراحی ساختار اساسی و اصلی یک صفحه وب استفاده می‌شود و چارچوب یک صفحه از وب را مشخص می‌کند.

CSS

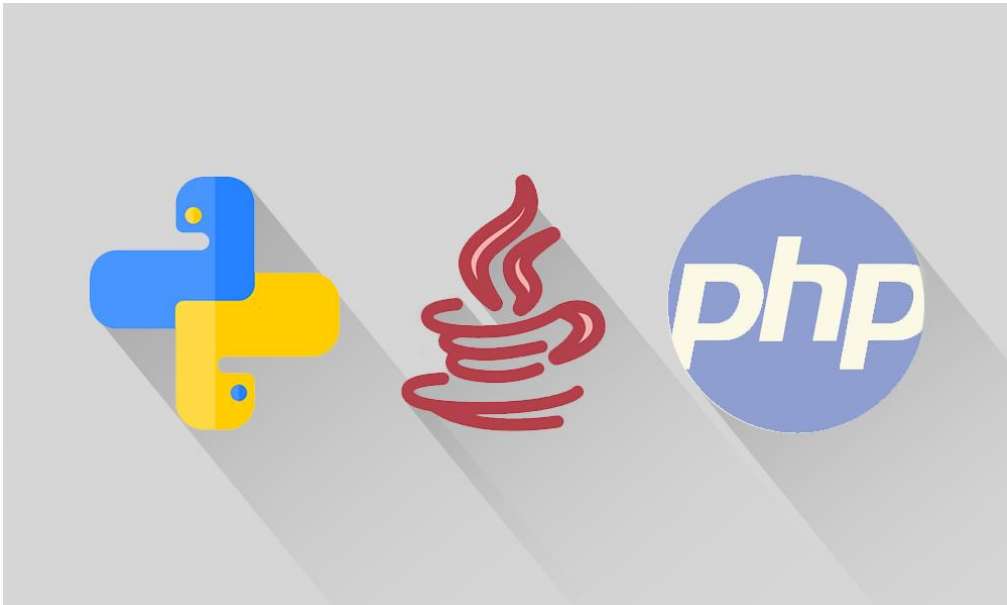
برنامه‌نویسان وب از CSS برای قالب‌بندی عناصر صفحه در html استفاده می‌کنند. CSS می‌تواند به تگ‌های HTML زیبایی ببخشد. به عنوان مثال شما با HTML یک صفحه ایجاد کرده و در آن یک کادر برای فرم و یک دکمه ارسال قرار می‌دهید. CSS با قابلیت‌هایی مانند رنگ، سایه، تغییر فونت و غیره این ساختار را زیباتر می‌کند.

JavaScript

به کمک جاوا اسکریپت می‌توانیم به صفحات طراحی شده جان ببخشیم و کاری کنیم که با کلیک بر دکمه ارسال، فرم به سرور ارسال شود. اگر HTML بدنه و CSS اجزا داخلی یک خودرو باشند، این جاوا اسکریپت است که کار کردن پدال گاز و حرکت خودرو را برای شما فراهم می‌کند.

بیشتر بخوانید: آشنایی بیشتر با برنامه‌نویسی Front-End

برنامه‌نویسی بک‌اند (Back-End)



پشت ظاهر زیبای عناصر موجود در یک وب‌سایت و عملکرد آن‌ها، کدهایی نوشته شده که باعث برقراری ارتباط با سرور و انجام درخواست‌های کاربران می‌شوند. این کدهای پنهان شده از چشم کاربر را برنامه‌نویسان بک‌اند (Back-End) توسعه می‌دهند. برنامه‌نویسی که تنها به مهارت‌های بک‌اند تسلط دارد به تنهایی نمی‌تواند یک سایت راه‌اندازی کند و باید از یک توسعه‌دهنده فرانت‌اند کمک بگیرد.

پس تا اینجا متوجه شدیم که طراحی وب، همان برنامه‌نویسی فرانت‌اند است؛ در حالی که توسعه وب، شامل برنامه‌نویسی در هر دو بخش Front-End و Back-End است.

معرفی زبان‌های Back-End

پایتون

پایتون یک زبان برنامه‌نویسی شگفت‌انگیز است. یادگیری پایتون برای مبتدیان ساده است و فعالیت‌هایی مانند برنامه‌نویسی اپلیکیشن و وب، مدیریت سرورهای شبکه، توسعه‌ی هوش مصنوعی، نرم‌فزارهای تحت دسکتاپ و غیره را با استفاده از آن می‌توان انجام داد. در ادامه درباره کاربرد پایتون در هوش مصنوعی نیز صحبت خواهیم کرد.

PHP

سینتکس این زبان بسیار شبیه به زبان سی (C) است و امکان استفاده از انواع پایگاه داده‌ها از جمله MySQL، اوراکل، Microsoft SQL Server، PostgreSQL و SQLite را با دستورهای ساده فراهم می‌کند. همچنین یکی از بهترین نمونه‌های برنامه‌نویسی با زبان php در سطح وسیع، وب‌سایت

فیس‌بوک و ویکی پدیا است. این زبان از فریم‌ورک‌های قدرتمندی همچون لاراول و سیمفونی نیز بهره‌مند است.

ASP.NET

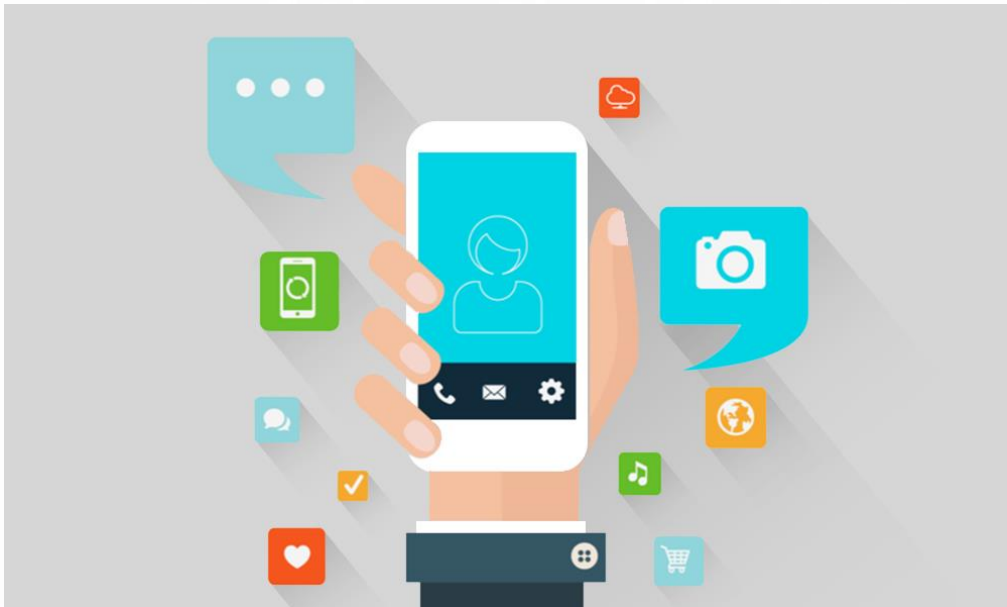
ASP.NET نیز از جمله چارچوب‌های محبوبی است که به‌طور گسترده توسط برنامه‌نویسان استفاده می‌شود.

شما می‌توانید در یکی از دو بخش فرانت‌اند یا بک‌اند فعالیت کنید و یا با یادگیری زبان‌های هر دو بخش به یک برنامه‌نویس فول‌استک (Full-Stack) تبدیل شوید. چنین شخصی مهارت توسعه وب را دارد و می‌تواند به تنهایی یک سایت را راه‌اندازی کند.

برای افرادی که آشنایی قبلی با برنامه‌نویسی ندارند یادگیری برنامه‌نویسی Front-End می‌تواند ساده‌تر باشد.

بیشتر بخوانید: آشنایی بیشتر با برنامه‌نویسی [Back-End](#)

برنامه‌نویسی اپلیکیشن موبایل



از مهم‌ترین سیستم عامل‌های حال حاضر برای تلفن‌های همراه، می‌توان به iOS کمپانی اپل و Android که بانی آن کمپانی گوگل است، اشاره کرد. طبق آمار سایت گوگل، در حال حاضر حدود 2 میلیارد دستگاه اندرویدی فعال موجود است. این امر موجب می‌شود تا توسعه‌دهندگان علاقه بیشتری به یادگیری برنامه‌نویسی در سیستم عامل اندروید داشته باشند.

زبان‌های توسعه اندروید

نخستین گام برای شروع برنامه‌نویسی اندروید، یادگیری زبان جاوا یا کاتلین (Kotlin) است که به فراخور نیازتان می‌توانید یکی از آن‌ها را انتخاب کرده و روی آن مانور دهید؛ البته در حال

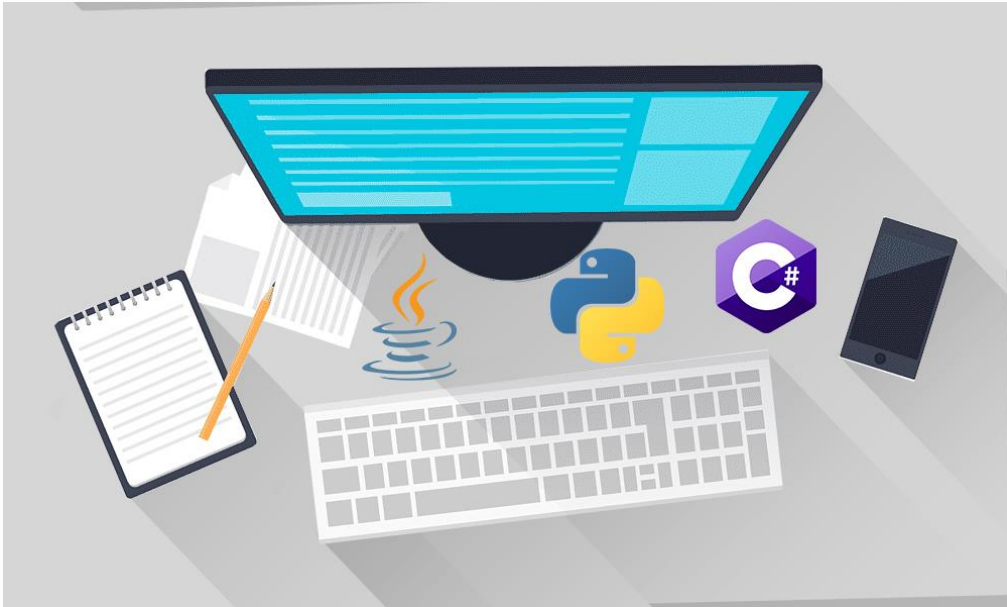
حاضر و در سال 2020 تمامی سورس کدهای گوگل به صورت پیش فرض بر روی Kotlin است.

پس از تسلط بر زبان موردنظر، به آشنایی خوب با مواردی چون مفاهیم اولیه اندروید و طراحی لایه، پایگاه داده، کامپوننت‌های اصلی اندروید، برنامه‌نویسی شی گرا، طراحی متریکال و غیره نیز نیاز دارید.

زبان‌های توسعه iOS

سوئیفت (Swift) یک زبان برنامه‌نویسی است که برای توسعه iOS, WatchOS, MacOS, tvOS استفاده می‌شود. برای توسعه اپلیکیشن در سیستم عامل مکینتاش و iOS می‌توانید از زبان سوئیفت استفاده کنید که مورد استفاده و حمایت مستقیم کمپانی اپل است.

برنامه‌نویسی تحت دسکتاپ



نرم‌افزار تحت دسکتاپ یعنی نرم‌افزارهایی که روی سیستم‌های شخصی و سیستم‌عامل‌های مختلفی مانند ویندوز، لینوکس و مکینتاش نصب و استفاده می‌شوند. برنامه‌هایی نظیر فتوشاپ، مجموعه آفیس و ابزارهای فشرده‌سازی، نمونه‌هایی از نرم‌افزارهای پرکاربرد تحت دسکتاپ هستند.

با این تعریف، برنامه‌نویسی دسکتاپ یعنی تولید نرم‌افزارهایی که روی کامپیوترهای شخصی نصب و استفاده می‌شوند. در ادامه چند زبان برنامه‌نویسی محبوب در این حوزه را معرفی می‌کنیم:

زبان سی‌شارپ

می‌توان گفت که زبان سی‌شارپ یکی از محبوب‌ترین زبان‌های برنامه‌نویسی تحت دسکتاپ در ایران است. در واقع سی‌شارپ یک زبان چند منظوره است که با استفاده از آن می‌توان علاوه بر نوشتن برنامه‌های دسکتاپ، برنامه‌های تحت وب را نیز پیاده‌سازی کرد.

این زبان دارای ساختاری شیء‌گرا مشابه C++ است و به صورت کاملاً اختصاصی برای استفاده از امکانات فریم‌ورک NET. مایکروسافت نوشته شده است.

فریم‌ورک NET. مجموعه‌ای متشکل از یک محیط اجرایی که تحت عنوان (CLR (Common Language Runtime شناخته می‌شود و همچنین مجموعه کاملی از توابع و کلاس‌های آماده، می‌باشد که می‌توان از آن در نوشتن برنامه‌هایی به زبان‌های مختلف استفاده نمود.

زبان برنامه‌نویسی جاوا

جاوا شباهت‌هایی به C++ دارد، ولی قابلیت انتقال آن بهتر است و استفاده از آن ساده‌تر از C++ است. همچنین مدیریت حافظه نیز توسط خود ماشین مجازی جاوا انجام می‌شود. این زبان به گونه‌ای بسیار امن طراحی شده است و وابسته به

سیستم عامل خاصی نیست زیرا جاوا یک ماشین مجازی در سیستم شما راه‌اندازی می‌کند و برنامه‌ها در آن ماشین مجازی اجرا می‌شوند.

زبان برنامه‌نویسی جاوا اسکریپت

جاوا اسکریپت یک زبان برنامه‌نویسی پویا و مبتنی بر شی (Object Based) است که در هر دو سمت کاربر و سمت سرور، برای ایجاد تعامل با صفحات وب به کار می‌رود. این زبان به دلیل داشتن فریم‌ورک‌های زیاد، هم در برنامه‌نویسی تحت وب و هم در توسعه اپلیکیشن‌های تحت دسکتاپ استفاده می‌شود.

لازم به ذکر است که هر فریم‌ورک شامل مجموعه قواعد و استانداردهایی است که به بهینه‌سازی کدها و بهبود فرآیند برنامه‌نویسی کمک می‌کنند. هر زبان ممکن است شامل چندین فریم‌ورک باشد که کاربردهای مختلفی دارند.

زبان پایتون

خوب است کمی هم درباره زبان محبوب پایتون صحبت کنیم. پایتون طرفداران زیادی دارد و یادگیری آن ساده است. این زبان علاوه بر حوزه برنامه‌نویسی تحت وب، در زمینه هوش

مصنوعی و زمینه‌های مرتبط با آن مانند داده کاوی (Data Mining)، یادگیری ماشین (Machine Learning) و یادگیری عمیق (Deep Learning) نیز به کار می‌رود.

بیشتر بخوانید: [آشنایی با زبان پایتون](#)

هوش مصنوعی



همان‌طور که در ابتدای کتاب درباره ارتباط ماشین و انسان صحبت کردیم، هوش مصنوعی در تلاش است تا ماشین‌ها را به هوشمندی انسان برساند، به گونه‌ای که بتوانند قدرت تفکر و خودبرنامه‌ریزی پیدا کنند. اهداف و آرزوهای انسان‌ها همیشه بدیع و دور از دسترس بوده است اما به نظر می‌رسد

که تا اینجای کار، هوش مصنوعی توانسته انسان را پشت سر بگذارد. روند رو به رشد هوش مصنوعی به قدری سریع است که دغدغه بسیاری از محققان این حوزه، از تعریف عملکرد جدید برای ربات‌ها به تعریف اخلاقیات و مسائل انسانی تبدیل شده است.

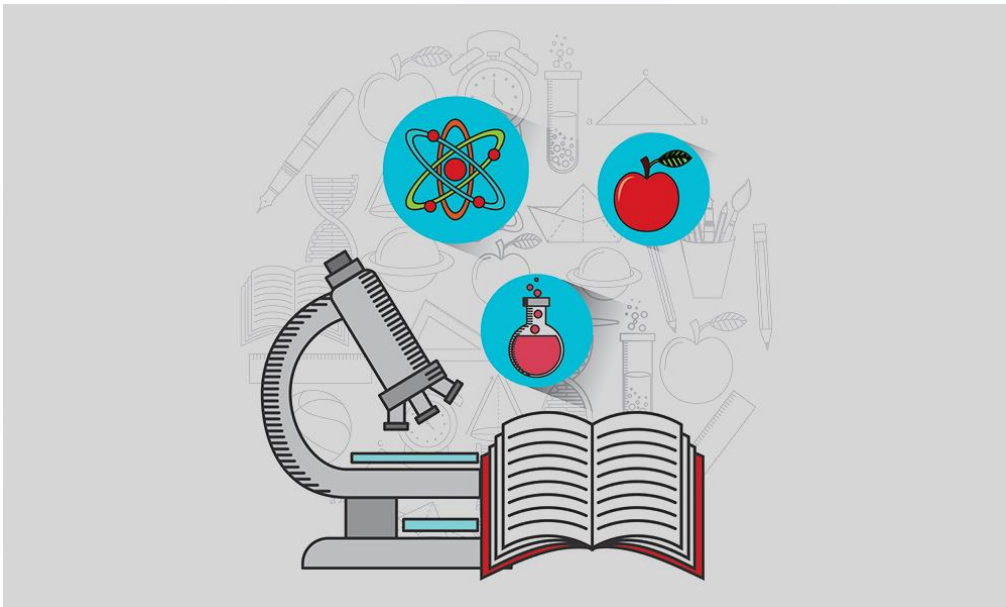
بیشتر بخوانید: [هوش مصنوعی چیست؟](#)

ماشین لرنینگ یا یادگیری ماشین، حوزه تخصصی‌تری از هوش مصنوعی است که تلاش می‌کند تا ماشین‌ها بتوانند از طریق تجربه و دریافت داده‌های محیطی و البته با کمترین برنامه‌ریزی از طرف انسان، یاد بگیرند و رفتارهای خود را بهبود دهند. به زبان ساده‌تر، محققان با استفاده از تحلیل مجموعه داده‌ها و تعریف الگوهای رفتاری اقدام به آموزش ماشین‌ها می‌کنند. سپس این ماشین‌ها در شرایط جدیدی قرار می‌گیرند تا با یادگیری قبلی و دانش اکتسابی خود، شرایط جدید را پیش‌بینی کنند. یک مثال عمومی ساده در این حوزه، مسئله تشخیص گربه و سگ است. یک ماشین پس از چند مرحله آموزش و یادگیری تفاوت عکس سگ و عکس گربه، با دیدن یک عکس جدید باید بتواند نوع حیوان را تشخیص دهد.

یادگیری عمیق یا Deep Learning نیز شاخه تخصصی‌تری از ماشین لرنینگ در جهت تحلیل ویژگی داده‌ها و استخراج مدل‌های رفتاری است.

بیشتر بخوانید: ماشین لرنینگ چیست؟

داده کاوی



حوزه داده کاوی نیز با استفاده از برخی الگوریتم‌های یادگیری ماشین و تحلیل مجموعه‌های بزرگ داده (DataSet) تلاش می‌کند تا رفتار کاربران را پیش‌بینی کرده و الگوهای رفتاری را ارائه دهد. به عنوان مثال همه رفتارهای ما در شبکه‌های اجتماعی و گوگل، داده‌هایی هستند که شرکت‌ها از آن‌ها برای

تحت تاثیر قرار دادن کاربران استفاده می‌کنند. تصور کنید که شما در اینستاگرام خود، بیشترین بازدید را از عکس غذا و حیوانات دارید. بنابراین وقتی وارد قسمت اکسپلورر (Explorer) اینستاگرام بشوید با عکس‌های جذاب و متنوعی از غذا و حیوانات مواجه خواهید شد. اینستاگرام با توجه به رفتار قبلی شما علاقه‌مندی‌تان را پیش‌بینی کرده و با نمایش موضوعات مرتبط سعی می‌کند شما را کاربری فعال نگه دارد. امروزه چنین الگوی رفتاری در بسیاری از سایت‌ها و شبکه‌های اجتماعی مشاهده می‌شود که تنها بخش کوچکی از کاربرد علم داده کاوی است.

یادگیری ماشین و داده کاوی از پرکاربردترین علوم امروزه هستند و زیربنای بسیاری از دستاوردهای جدید تکنولوژی محسوب می‌شوند. بنابراین یادگیری پایتون به صورت کاربردی می‌تواند حوزه‌های فعالیت جذاب و متنوعی را برای علاقه‌مندان فراهم کند.

درباره UI و UX

خوب است نگاهی هم به مفاهیم پرطرفدار UI و UX داشته باشیم:

هنگام فعالیت در حوزه طراحی و توسعه محصول، تجارت و بازاریابی دیجیتال و غیره بسیاری از اصطلاحات را می‌شنویم که برای توصیف تجربه‌ی کار با محصول و چگونگی ارتباط آن با کاربران یا مشتریان استفاده می‌شوند. از پرکاربردترین این مفاهیم می‌توان به رابط کاربری و تجربه‌ی کاربری اشاره کرد.

ال چیست؟



رابط کاربری (User Interface) که به اختصار ال نامیده می‌شود، مجموعه اصولی برای طراحی‌های گرافیکی است تا بتوان با در نظر گرفتن نیازمندی‌های کاربران، سایت‌ها و اپلیکیشن‌ها را با حفظ کیفیت و زیبایی طراحی کرد.

بسیاری از طراحان ممکن است رابط کاربری را چالشی برای ارائه یک طراحی زیبا و چشم‌گیر بدانند و آن را در زیبایی خلاصه کنند اما همان‌طور که گفتیم، این نیاز کاربران است که حرف اول را می‌زند. اگر کاربران سادگی را ترجیح دهند، پس داشتن اپلیکیشنی با طراحی پیچیده و خوش رنگ و لعاب ممکن است یک شکست محسوب شود. همه‌ی اجزای یک صفحه مانند

عکس، متن، دکمه‌ها، حتی رنگ و به‌طور کلی هر چیزی که در طراحی شما قرار دارد، نیاز به هماهنگی کامل با اصول طراحی کاربری دارد.

UX چیست؟



تجربه‌ی کاربری (User Experience) ، فرآیند کلی تولید یک محصول محسوب می‌شود و ال می‌تواند زیرشاخه‌ای از آن باشد. به طور خلاصه تجربه کاربری علاوه بر رفع نیازهای کاربران، تاکید زیادی بر راحتی سیستم‌ها دارد به گونه‌ای که کاربر به‌سختی ترجیح بدهد سیستم فعلی را با نرم‌افزار دیگری جایگزین کند.

تجربه‌ی کاربری نقشه‌ی راهی است که کمک می‌کند تا محصول نرم‌افزاری طبق خواسته‌ی کاربر طراحی شود. در تجربه کاربری مهم‌ترین مرحله، شناخت نیاز کاربر است و هماهنگی طراحی با این نیازها، باید به گونه‌ای باشد که کاربر به هدف خود برسد و از کار کردن با سیستم لذت ببرد.

به‌طور خلاصه متخصص UX باید از راه‌های مختلفی مانند بررسی تجربیات قبلی کاربر، نظرسنجی، پرسش و پاسخ و سایر روش‌های تحقیقاتی، نیاز و خواسته کاربران را هنگام استفاده از یک اپلیکیشن یا سایت پیدا کند و در اختیار تیم و متخصص UI قرار دهد. سپس متخصص UI با استفاده از دانش گرافیکی برای پیاده‌سازی و طراحی این موارد اقدام کند.

مثلاً شما در طی فرآیند UX متوجه می‌شوید که کاربران هنگام پر کردن یک فرم 5 قسمتی، فقط دو بخش را پر می‌کنند. پس سه بخش دیگر بدون استفاده است و متخصص UI باید با طراحی مجدد فرم فقط دو بخش را برای آن در نظر بگیرد تا کاربر احساس راحتی بیشتری کند.

شما برای فعالیت در این حوزه‌ها نیاز به دانش برنامه‌نویسی ندارید اما آشنایی با کدنویسی برای طراحان رابط کاربری، یک

مزیت محسوب می‌شود و تسلط آن‌ها بر کار را افزایش می‌دهد.

بیشتر بخوانید: [نگاهی به مفاهیم UI و UX و CX](#)

شناخت بهترین زبان برنامه‌نویسی



در برنامه‌نویسی همواره حوزه‌های ترندی وجود دارد که ممکن است از نظر میزان درآمد و تقاضای کار، شرایط بهتری داشته باشند اما به‌طور قطعی چیزی به‌عنوان بهترین زبان برنامه‌نویسی وجود ندارد. در واقع بهترین زبان برنامه‌نویسی را خود شما با توجه به فاکتورهایی که مد نظر دارید تعیین می‌کنید.

بیایید تعدادی از فاکتورهای مهم را بررسی کنیم:

علاقه و شناخت مسیر

مهم‌ترین فاکتور این است که از انجام کارتان لذت ببرید. یادگیری برنامه‌نویسی چندان ساده نیست و به تلاش زیاد و هدفمندی نیاز دارد. علاوه بر این، پس از ورود به دنیای حرفه‌ای و قبول پروژه نیز با چالش‌های زیادی مواجه می‌شوید که صبر و تمرکز زیادی را می‌طلبد. بنابراین کسی می‌تواند پای این سختی‌ها بماند و به نتیجه برسد که اگر از او بپرسیم: "اگر برنامه‌نویس نمی‌شدی چه کاره می‌شدی؟" آگاهانه بگوید: "نمی‌دانم. من به چیزی غیر از برنامه‌نویسی فکر نمی‌کنم."

پس اگر می‌خواهید در برنامه‌نویسی به موفقیت دست پیدا کنید، باید سختی‌هایش را بپذیرید و اگر این شجاعت را دارید، چه بهتر که مسیر را بشناسید و سختی‌هایتان را هدفمند کنید.

میزان تقاضا

بررسی آگهی‌های شغلی می‌تواند به شما کمک کند که از حوزه‌های پرتقاضا و به‌روز باخبر شوید. گاهی هم ممکن است شغلی مناسب با توانایی شما وجود داشته باشد اما اگر کمی

مهارت‌های خود را ارتقا دهید، فرصت‌های شغلی و مالی بیشتری در انتظارتان است.

میزان مهارت

بهتر است با خود روراست باشید و ببینید در حال حاضر چه مهارت‌هایی دارید که مرتبط با برنامه‌نویسی است. اگر آشنایی قبلی با این حوزه ندارید شاید بهتر باشد از یادگیری زبان‌های ساده‌تر شروع کنید و سطح دشواری زبان‌های مختلف را بسنجید. اگر هم تجربه برنامه‌نویسی دارید بهتر است زبانی را انتخاب کنید که یا مشترکات زیادی با دانش قبلی شما داشته باشد و یا بتواند سطح شغلی شما را تا حد زیادی ارتقا بدهد.

آینده زبان

اگر دوست دارید برنامه‌نویس شوید باید همیشه چشمتان به آینده باشد. به این معنی که همواره برای ارتقا مهارت‌های خود تلاش کنید تا به عرصه‌های بزرگتری وارد شوید و البته خودتان نیز برای ایجاد نوآوری دست به کار شوید. برنامه‌نویسی دنیای حل مسئله است و اگر خواهان پیشرفت و یک جایگاه متعالی هستید، باید دغدغه‌ای را با کدها گره زده و خلاق باشید.

میزان درآمد

بیل گیتس می‌گوید: " تلاش نکنید تا میلیارد شوید، چون این رویا قدیمی شده. " واقعیت این است که برنامه‌نویسی وابسته به میزان مهارت شما و زمانی که به آن اختصاص می‌دهید ممکن است درآمد بسیار خوبی داشته باشد اما یادگیری و موفق بودن در این مسیر ساده نیست. اگر تنها به امید درآمد بهتر برنامه‌نویس شوید، احتمال دارد خیلی زود ناامید شده و جا بزنید. برنامه‌نویسی به هر اندازه که سخت و پیچیده شود، به همان اندازه نیز فرصت یادگیری، تجربه و شرایط مالی بهتر برای شما ایجاد می‌کند و اگر خود را بخشی از این مسیر رو به رشد ندانید، خیلی زود به کارمندی ناراضی تبدیل می‌شوید که از درآمد خود راضی نیست اما علاقه و حوصله لازم برای پیشرفت را نیز ندارد.

آموزش زبان‌های برنامه‌نویسی

زبان‌های برنامه‌نویسی برتر از نظر بازار کار



طبق آخرین بررسی‌ها در سال‌های اخیر و در بازار بین المللی حوزه‌هایی مانند:

- پایتون
- جاوا اسکریپت
- سوئیفت
- PHP
- R
- برنامه‌نویسی اندروید
- محبوبیت بالایی دارند.

در ایران نیز با توجه به نیاز شرکت‌ها، حوزه‌های:

- پایتون
 - PHP و فریم‌ورک‌های آن به ویژه لاراول
 - جاوا اسکریپت و فریم‌ورک‌های محبوب آن
 - برنامه نویسی اندروید
 - طراحی وب
- محبوبیت بیشتری دارند.

مهارت‌های برنامه‌نویس حرفه‌ای



توانایی حل مسئله

توانایی حل مسئله مهم‌ترین ویژگی یک برنامه‌نویس است. شما باید بتوانید مشکل را تشخیص داده و با توجه به زمان و منابعی که دارید، به بهترین شکل ممکن آن را حل کنید. چنین مهارتی تنها از دل تجربه و تمرین فراوان به دست می‌آید و قرار نیست پس از حل نکردن مسئله دهم به کلی نا امید شوید. شاید بتوان گفت حل مسئله مهم‌ترین چالش علاقه‌مندان به برنامه‌نویسی است و در همین مرحله است که بسیاری از افراد تغییر حوزه داده و از برنامه‌نویسی دست می‌کشند.

زبان انگلیسی سطح متوسط

شما به زبان انگلیسی در حد متوسطی نیاز دارید تا بتوانید از منابع مفید خارجی بهره‌مند شوید. اگر واقعا به برنامه‌نویسی علاقه داشته باشید از اینکه قرار است در کنار توانایی فنی، سطح زبان خود را ارتقا بدهید خوشحال خواهید شد.

توانایی جستجو در اینترنت

نباید دست خالی از گوگل برگردید. برنامه‌نویسان هنگام مواجه با یک مشکل قبل از هر چیزی از گوگل کمک می‌گیرند. گوگل اگر به شما پاسخ قطعی هم ندهد، دست‌کم صورت سوال را ساده‌تر کرده و ذهن شما را در حل مشکل جهت‌دهی می‌کند.

مهارت رفع اشکال

هر باگ، امتحانی برای سنجش صبری برنامه‌نویس است. اگر برای حل یک مسئله شاخ غول را هم بشکنید و الگوریتم سوال را پیدا کنید، باز هم ممکن نیست خیالتان راحت باشد زیرا امکان دارد باگ‌ها در هر قسمت از کد گریبان شما را بگیرند. باتجربه‌ها می‌گویند در این موقعیت، اولین قدم پذیرش باگ است. یعنی دست از گفتن این جمله که " همه چی درسته

نمی‌دونم چرا کار نمی‌کنه " بردارید و مطمئن باشید یک چیزی یک جایی می‌لنگد و باید آن را پیدا کنید.

کنجکاوی

علاقه داشته باشید که از همه چیز سردر بیاورید. معمولاً برنامه‌نویس‌ها اخبار روز تکنولوژی، ایده‌های ایلان ماسک، ویدیو گیم‌ها، دستگاه‌های دیجیتالی جدید، نرم‌افزارها و سایر اخبار فناوری را با اشتیاق دنبال می‌کنند. این کار علاوه بر ارتقاء بینش شما، به بهبود وجهه اجتماعی شما نیز کمک می‌کند. امروزه ماهیت شرکت‌ها به‌گونه‌ای است که افراد تیم در کنار حوزه تخصصی خود، آشنایی مختصری با شغل دیگران نیز دارند. بنابراین شاید یک برنامه‌نویس همه فن حریف اما منزوی و کم‌حرف، نسبت به برنامه‌نویس متوسطی که جنب و جوش دارد و از سایر حوزه‌ها چندان بی‌خبر نیست، شانس کمتری برای استخدام و همکاری داشته باشد.

روحیه منعطف با محیط

محیط، نقش کلیدی در تمرکز انسان دارد. یک برنامه‌نویس ممکن است به دلایل مختلفی ناچار به کار کردن در محیطی

باشد که با روحیه او سازگاری کمی دارد. شما باید بتوانید ذهن خود را اولویت‌بندی کرده و شرایط را کنترل کنید.

روحیه کار تیمی

نحوه کد نویسی، پایبندی به زمان تحویل، ارائه توضیح درباره پروژه و برخورد و بیان خوب از جمله ویژگی‌هایی هستند که می‌توانند یک شغل خوب را برای شما تضمین کنند.

مدیریت زمان

وقتی پروژه‌ای را به عهده می‌گیرید هیچ چیز به اندازه رعایت زمان‌بندی، برای مدیر یا صاحب پروژه اهمیت ندارد. اگر توانایی مدیریت زمان نداشته باشید فرقی نمی‌کند که کارآموز هستید و یا یک برنامه‌نویس حرفه‌ای، در هر صورت اعتبار شما نزد سایر افراد درگیر در پروژه زیر سوال می‌رود.

توسعه فردی

تمایل به یادگیری باعث می‌شود مدتی طولانی در یک سطح نمانید و روند پیشرفت خود را تسریع کنید. این به معنای از این شاخه به آن شاخه پریدن نیست بلکه شما باید در راستای

هدف و فعالیت‌های فعلی خود مهارت‌های جدیدی را یاد بگیرید که به بهبود شرایط فردی و شغلی‌تان کمک کند.

ویژگی‌های ذکر شده در بالا، مهارت نرم نامیده می‌شوند. یک برنامه‌نویس ممکن است در ابتدا با مهارت‌های سخت یا همان دانش فنی استخدام شود، اما با گذشت زمان مشخص می‌شود که این مهارت‌های نرم و توانایی تعامل با دیگران است که باعث رشد و یافتن او خواهند شد.

در صورت تمایل برای آشنایی بیشتر با این مهارت‌ها می‌توانید به سایت سون‌لرن مراجعه کنید و کتاب دیگری تحت عنوان **آشنایی با مهارت‌های نرم برنامه‌نویسان** را مطالعه کنید.

سخن پایانی

در این کتاب تلاش کردیم تا مهم‌ترین دغدغه‌ها و سوالات علاقه‌مندان به حوزه برنامه‌نویسی را پاسخ دهیم و به شما در تصمیم‌گیری بهتر کمک کنیم. همه ما برای پیدا کردن شغل مورد علاقه خود باید حداقل چهار سوال را مدنظر قرار بدهیم:

- من چقدر به این کار علاقه دارم؟
- این کار تا چه مقدار نیازهای من را برآورده می‌کند؟
- این کار چه تاثیر مثبتی بر جهان دارد؟
- امکانات، میزان تقاضا و زمینه‌های فعالیت در آن چقدر فراهم است؟

اگر برنامه‌نویسی پاسخ مشترک شما به این چهار سوال باشد پس بهتر است تمامی شرایط خود را بسنجید و با تحقیق و مطالعه بیشتر، حوزه مناسب خود را انتخاب کنید.

برای موفق شدن در برنامه‌نویسی باید با علاقه و سخت تلاش کنید و از بهترین منابع آموزشی بهره بگیرید تا شیرینی یادگیری و کسب نتیجه به جانتان بنشیند. در غیر این صورت خیلی زود از مسیر خسته و نا امید می‌شوید و موفقیتی هم به دست نمی‌آورید. به چشم یک مهارت و مسئولیت جهانی به

برنامه‌نویسی نگاه کنید تا به جمع‌های حرفه‌ای راه پیدا کرده و
در اتفاقات و تحولات بزرگ این دنیا سهیم باشید.

از یادگیری تا ورود به بازار کار

www.7Learn.com