



به‌تازگی باوری در جامعه فناوری شکل‌گرفته که می‌گوید برای پیشرفت بیشتر همه باید برنامه‌نویسی یاد بگیرند؛ اما چنین ایده‌ای کمی افراط‌گونه به نظر می‌رسد، زیرا کدنویسی نوع جدید سواد خواندن و نوشتن محسوب نمی‌شود. تب تبدیل‌شدن به ابر ثروتمندان دره سیلیکون جنبشی به نام «یادگیری برنامه‌نویسی» را به راه انداخته است. جوامع فناوری، سازمان‌های غیرانتفاعی مثل Code.org و حتی سیاستمداران این را به‌عنوان یک مهارت ضروری برای نیروی کار آینده تلقی می‌کنند. ممکن است چنین دیدگاهی از برخی جهات حقیقت داشته باشد، به‌ویژه آن‌که نیاز کشورهای توسعه‌یافته به این مهندسان هیچ نشانه‌ای از کند شدن را نشان نمی‌دهد؛ اما موضوع به همین‌جا ختم نمی‌شود و پیچیده‌تر از این‌ها است.

ما در یک دنیای فوق‌العاده رقابتی زندگی می‌کنیم، با مردمانی که از هر شیوه‌ای به‌ویژه به‌کارگیری فناوری برای رسیدن به هدف و کسب موفقیت استفاده می‌کنند. فروش کد به یک راهکار برای پیشرفت اقتصادی تبدیل‌شده است. با نگاه به پدیده جدید اردوهای توسعه نرم‌افزاری (coding bootcamps) و خبرهای مربوط به موفقیت مهندسان نرم‌افزار مقیم دره سیلیکون، همه یا می‌خواهند یک استارت‌آپ راه‌اندازی کنند یا مهندس شوند. در سریال «سیلیکون ولی» ساخته‌شده توسط شرکت فیلم‌سازی HBO، نمایی از اواخر قرن بیست و چندم به تصویر کشیده می‌شود که در آن قهرمان‌های داستان به شکل جنون‌آمیزی روز و شب مشغول **کدنویسی** یا مواد کشیدن هستند و در این بین پای میلیون‌ها دلار پول نیز در میان است. مردم دنیا با چهره‌هایی مثل ایلان ماسک و مارک زاکربرگ آشنا می‌شوند که یک شبه میلیون‌ها دلار پول می‌سازند. بعد از تصویب قوانین استفاده از دانش کامپیوتری در برنامه درسی مدارس توسط رئیس‌جمهور اسبق ایالات‌متحده تب **کدنویسی** در این کشور داغ شد و نشان داد مفهوم **برنامه‌نویسی** تا چه اندازه به مرکز قدرت این کشور نزدیک شده است.

علاوه بر کمپ‌های آموزش و توسعه نرم‌افزارها و سیاستمداران که مردم را به یادگیری برنامه‌نویسی تشویق می‌کنند، بخش‌های مستقل دیگر از هالیوود گرفته تا روشنفکران دائم در حال تشویق افکار عمومی هستند تا کدنویسی را یاد بگیرند. صرف‌نظر از تبلیغاتی که در گوشه و کنار شنیده می‌شود، بوت کمپ‌ها هم به این شک و تردیدها دامن می‌زنند. هر چند افکار عمومی مراکزی مثل دره سیلیکون را جذاب می‌بیند و بروشورهای پر زرق‌وبرق بوت کمپ‌ها شغل‌های پردرآمد را وعده می‌دهند، اما واقعیت این است که خیلی از موسسات ارائه‌دهنده چنین ارقامی معتبر نیستند، آمار کار درستی را منتشر نمی‌کنند و اصلاً مشخص نیست که دانشجویان آن‌ها تا چه میزان در کار خود موفق خواهند بود. باوجود این‌که خیلی از این اردوهای توسعه نرم‌افزاری قانونی هستند و به شاگردان خود رسیدگی می‌کنند، اما در مقابل تعداد دیگری از آن‌ها توسط سوداگران جعلی سرمایه اداره می‌شوند.

یادگیری برنامه‌نویسی مهم است اما...

اشتباه برداشت نکنید؛ همه ما بر این عقیده‌ایم که مهندسی و **برنامه‌نویسی** مهارت‌های پراهمیتی هستند؛ اما در

جایگاه درست خود و تنها برای افرادی که حاضرند برای موفقیت در این حرفه تمام دشواری‌ها را پشت سر بگذارند. همین موضوع در مورد سایر حرفه‌های دیگر نیز صدق می‌کند. یک جامعه همان اندازه که به مهارت **برنامه‌نویسی** نیاز دارد، به مهارت لوله‌کشی هم نیاز خواهد داشت. قبل از این‌که ما به دنبال یافتن یک راه‌حل برای یک مسئله کدنویسی را باشیم باید تصمیم بگیریم با چه مسئله‌ای سر و کار داریم و آیا پیشامدی که پیش روی ما قرار گرفته، اصلاً برای ما یک مسئله به شمار می‌رود یا خیر؟ یک عضو سابق انجمن بین‌المللی محاسبات ماشینی و برنده مسابقات بین‌المللی برنامه‌نویسی دانشگاهی از استنفورد، رمز موفقیت خود را در روزهای مسابقات اهمیت درک درست مسئله‌ای که شما قصد حل کردن آن را دارید، می‌داند.

مطلب پیشنهادی



دو مفهوم در ظاهر یکسان، اما در باطن متفاوت مهندسی نرم‌افزار و برنامه‌نویسی چه تفاوت‌هایی با یکدیگر دارند؟

شما باید از خود پرسید: آیا اصلاً مسئله‌ای وجود دارد؟ آیا شما می‌توانید اصل فاینمن (Feynman principle) را اعمال کنید؟ آیا می‌توانید آن را به نحوی توضیح دهید که دیگران هم بتوانند آن را درک کنند؟ این عضو سابق انجمن می‌گوید حتی در مدارس تیزهوشان نیز دانشجویان یک‌بار به مشکل کدنویسی می‌پردازند و پس از آن بلافاصله شروع به کدنویسی می‌کنند. او می‌گوید در دورانی که برنده این مسابقات شده یک چیز را یاد گرفته، این‌که **برنامه‌نویسی** پیش از آن‌که یک تخصص فنی باشد، هنر تحلیل مسائل است.

آماده‌اید در کمتر از یک یا دو سال از یک‌زبان به زبان دیگری مهاجرت کنید؟

چند سال پیش برنامه‌نویسان پلتفرم اپل از Objective-C استفاده می‌کردند و بعد به سراغ Swift رفتند. این افراد به‌خوبی می‌دانستند که چه دشواری‌ها و کاستی‌هایی را در **برنامه‌نویسی** طی کرده‌اند و به چه موقعیت بهتری رسیده‌اند. در مقطع کنونی ما با توسعه‌دهندگان iOS جویای کاری روبه‌رو می‌شویم که حتی یک‌بار هم با Objective-C کدنویسی نکرده‌اند. درست است که یادگیری Swift ساده‌تر است، از امنیت بیشتری برخوردار است، از مدل‌های توسعه پیشرفته‌تری استفاده می‌کند، از ظرافت به‌مراتب بیشتری نسبت به Objective-C برخوردار است و این‌که توسعه‌دهنده جدید هرگز به کاستی‌های Objective-C واقف نخواهد شد، اما این مسئله باعث می‌شود تا یک توسعه‌دهنده تازه‌کار واقعیت‌های حرفه خود را نادیده گرفته و تصور کند همه‌چیز به همین سادگی است. همین توسعه‌دهنده چند سال دیگر با موضوعات پیچیده‌ای در **برنامه‌نویسی** روبه‌رو خواهد شد که برای حل آن‌ها باید به سراغ راهکارهای جدید برود. در آن زمان است که با واقعیت‌ها و مصائب شغل خود روبه‌رو خواهد شد. توسعه‌دهندگان امروزی انتظار دارند با کمترین صرف وقت و دریافت راهنمایی به‌سرعت آموزش ببینند. انتظاری که قابل انجام نیست. می‌توان استدلال کرد که این هم بخشی از هزینه‌های انتخاب چنین حرفه‌ای است. اما اگر توسعه‌دهندگان فعلی ناامید شده یا کنار گذاشته شوند (و شواهدی وجود دارد که نشان می‌دهد این‌گونه است) چرا افراد را تشویق به وارد شدن به چنین قلمرو نامشخصی می‌کنند؟

چه بلایی سر افرادی خواهد آمد که روزها و شب‌های طولانی از وقت خود را صرف یادگیری Objective-C کرده‌اند و حالا می‌بینند یک‌زبان جدید تازه از راه رسیده (Swift) و جایگزین آن شده است. آیا آن‌ها به کدنویسی با این زبان در حال انقراض ادامه می‌دهند یا از نو شروع می‌کنند؟ اگر یک جوان بیست‌وچندساله باشید ممکن است چنین کاری امکان‌پذیر باشد، اما اگر سرپرست خانوار هستید که سر ماه باید هزینه‌ها را پرداخت کند، این کار بسیار طاقت‌فرسا خواهد بود. افراد زیادی هستند که بدون درک درست از واقعیت **برنامه‌نویسی** و مهندسی به‌یک‌باره با چنین شرایطی مواجه می‌شوند.

یک برنامه‌نویس تازه‌کار حداقل باید یک سال از وقت خود را صرف یادگیری کند تا بتواند یک شغل آزاد را در این حرفه آغاز کند. حتی بعد از آن هم دستمزد قابل‌ملاحظه‌ای دریافت نخواهد کرد. تازه این فرد به دلیل این‌که مدرک دانشگاهی در زمینه علوم کامپیوتر ندارد نمی‌تواند در خیلی از آزمون‌های استخدامی شرکت کند. همین دلایل باعث می‌شود تا خیلی از مواقع فرد از ادامه این حرفه ناامید شده و انگیزه خود را از دست دهد.

در حقیقت تبدیل شدن به یک توسعه‌دهنده، حتی در سطح کارآموزی کار راحتی نیست. شما نیاز دارید با دیگر متخصصان این حرفه ارتباط برقرار کنید؛ افرادی که کار شما را ضمانت کنند و نیاز به یک حساب معتبر GitHub دارید تا طی زمان پروژه‌های خود را به دیگران معرفی کنید. حتی اگر فرض بر این باشد که از فرصت‌های برابری

نسبت به دیگران برای پیشرفت برخوردار باشید، اگر از اقلیت کمتر شناخته شده باشید، باید دو برابر دیگران خوب عمل کنید و این حداقل کاری است که برای نشان دادن شایستگی خود باید انجام دهید.

مطلب پیشنهادی



برنامه نویسان از شنیدن چه جملاتی بیزار هستند؟
کدنویسان با شنیدن این هفت جمله به سرعت عصبانی می‌شوند

شرایط و ضوابط استخدامی و آزمون‌های حرفه برنامه نویسی نیز بر اساس نوع کار و کارفرما می‌تواند بسیار متفاوت و گاهی سخت‌گیرانه باشد. در یک مصاحبه استخدامی شما ممکن است مجبور باشید روی یک وایت‌برد طی چند دقیقه یک راهکار **برنامه نویسی** برای یک مشکل ارائه کنید. همچنین ممکن است ناآگاهی کارفرما در یک فرم استخدامی صلاحیت‌های خاص و عجیبی از جمله پنج سال سابقه کار را در زبان برنامه نویسی Swift (درحالی‌که Swift از سال 2014 منتشر شده است) قید کند. همه این‌ها موانعی است که شما را از رسیدن به حرفه موردنظر خود بازمی‌دارد.

کلام پایانی

اگر تبدیل شدن به یک مهندس نرم‌افزار تمام آن چیزی است که به دنبال آن هستید، نباید اجازه دهید هیچ‌کس و هیچ چیزی مانع رسیدن شما به هدفتان شود. اجازه ندهید محدودیت‌های سنتی مثل سیستم آموزشی ناکارآمد سرعت شما را کم کند. هیچ راه درست و نادرست مشخصی برای رسیدن شما به اهدافتان وجود ندارد. به زرق و برق‌ها و جذابیت‌های پیرامون این حرفه توجه نکنید و واقعیت‌ها را در نظر داشته باشید. باید برای درک و آشنایی کامل با این حرفه زمان بگذارید. شما باید این واقعیت را بپذیرید که قرار است حل‌کننده مسئله باشید نه یک توسعه‌دهنده که قرار است برای هر پروژه‌ای به سراغ فریم ورک‌های از پیش نوشته شده برود. همچنین باید به این موضوع عادت کنید که هر لحظه ممکن است لازم باشد یک فریم ورک یا زبان جدید را یاد بگیرید و این‌که اگر مدرک دانشگاهی و رسمی نداشته باشید برای پیدا کردن یک شغل باید بیشتر تلاش کنید. مهندسی نرم‌افزار یک حرفه پردرآمد است، اما تبدیل شدن از یک کدنویس به یک مهندس کاری دشوار و پرفراز و نشیب است. اگر در این راه حساب شده و درست گام بردارید، آینده درخشانی را پیش‌رو خواهید داشت.

تاریخ انتشار:

30 آبان 1397

نشانی منبع:

<https://www.shabakeh-mag.com/workshop/13952/%D8%AD%D9%82%D8%A7%DB%8C%D9%82-%D9%86%D8%A7%D9%BE%DB%8C%D8%AF%D8%A7-%D9%88-%D8%AA%D9%84%D8%AE-%D8%A8%D8%B1%D9%86%D8%A7%D9%85%D9%87%E2%80%8C%D9%86%D9%88%DB%8C%D8%B3%DB%8C-%DA%A9%D9%87-%D8%A8%DB%8C%D8%B4%D8%AA%D8%B1->

%D8%A8%D9%87-%D9%88%D8%A7%D9%82%D8%B9%DB%8C%D8%AA-
%D9%86%D8%B2%D8%AF%DB%8C%DA%A9-%D9%87%D8%B3%D8%AA%D9%86%D8%AF