۵ مفهوم ضروری که علاقمندان برنامه نویسی باید یاد بگیرند

همه دوست دارند یک توسعه دهنده نرم افزار باشند اما کمتر کسی حاضر است سختی های این کار را به جان بخرد چرا که به خاطر سپردن همه عبارت های مربوط به این زمینه اصلا آسان نیست.

در ادامه برخی مفاهیم ضروری برنامه نویسی ذکر شده که به عنوان یک تازه کار قطعا به آنها بر خواهید خورد.

قبل از شروع آرامش خود را حفظ کنید

قرار نیست همه واژه هایی که با آن مواجه می شوید را به خاطر بسپارید. به لطف تمرین شما به صورت حسی معنی واژگان بسیاری را یاد می گیرید. پس به جای تلاش صرف برای حفظ واژه ها و معنی آنها سعی کنید منطق چگونگی و چرایی استفاده از این کلمات در محیط برنامه نویسی را یاد بگیرید.

با توجه اینکه بسیاری از واژه های گیج کننده برنامه نویسی از چند بخش دشوار تشکیل شده اند استفاده از این روش بهترین راه برای تسلط بر عبارت ها است. به خاطر داشته باشید که حفظ آرامش و پرهیز از استرس حین یادگیری برنامه نویسی از اهمیت بالایی برخوردار است.

ویرایشگر متن

ویرایشگر متن برنامه ای است که برای ویرایش متون از آن استفاده می شود. شاید این جمله ساده به نظر برسد اما بررسی ویژگی های یک ویرایشگر متن مناسب چندان ساده نیست. نوت پد ویندوز نمونه ای از یک ویرایشگر متن معمولی است اما بهترین برنامه باید چندین قابلیت و دستورات کیبورد را یکپارچه کرده و امکان کد نویسی سریع و واضح را میسر سازد.

یکی از این ویژگی ها برجسته سازی سینتکس است که پارامترهای مختلف کد را به رنگ های مختلف در می آورد. برنامه نویس با استفاده از این ویژگی می تواند دستورات، زبان ها و نشانه های متفاوت را مشخص کند. برجسته سازی سینتکس برای تعیین خطاها و جداسازی بخش های کد حیاتی بوده و به درک بهتر منطق برنامه نویسی کمک می کند. برای مثال یک پرانتز قرمز بدین معنی است که مورد دوم باید همان نزدیکی ها باشد.

Notepad++و **Sublime Text** دو نمونه از ویرایشگرهای متن قابل قبول و در دسترس هستند.

۲. محیط توسعه یکپارچه IDE

ویرایشگر متن تنها روی کدی که می نویسید تمرکز می کند اما یک IDE مجموعه ابزاری است که امکان نگارش، کامپایل (برگردان از متن به زبان کامپیوتری)، تست و عیب یابی (یا تصحیح) برنامه را برای شما میسر می سازد. واژه محیط در IDE معنای خاص خود را دارد، به عبارت دیگر استفاده از IDE بدین معنی است که شما از چندین ابزار برنامه نویسی در یک محیط واحد استفاده می کنید که هدف نهایی از آن تولید یک محصول است.

اگرچه IDE نسبت به ویرایشگر متنی لزوما محیط ساده تر یا دشوارتری را در اختیار کاربر قرار نمی دهد، برخی بر این باورند که IDE جنبه های کلیدی برنامه نویسی مثل لینک کردن فایل ها را ساده تر می سازد. شاید اصلاح یک فایل html ساده با استفاده از یک شیت CSS آسان باشد اما پروژه های پیچیده تر مستلزم برقراری ارتباط بین چندین زبان، فریمورک و غیره است.

اینجاست که IDE با یکپارچه سازی فایل ها، فولدرها و ابزارهای دیگر در یک پکیج واحد به داد برنامه نویس می رسد. IDE ها در زبان های برنامه نویسی شی گرا نظیر C++, C, Ruby و غیره هم کاربرد بسیاری دارند.

۳ .زبان های برنامه نویسی در برابر زبان های اسکریپت نویسی

جمله «این یک اسکریپت نیست بلکه یک زبان است»، یکی از قدیمی ترین مجادلات معمول بین برنامه نویسان است. اگرچه اغلب افراد تازه کار به تمامی کدها به عنوان زبان اشاره می کنند اما بین زبان های اسکریپت نویسی و برنامه نویسی مرز ظریفی وجود دارد.

زبان های برنامه نویسی کامپایل می شوند در حالی که زبان های اسکریپت نویسی تفسیر می شوند. فرض کنید روی کامپیوتر ۱ اپلیکیشنی ایجاد کرده و قصد انتقال آن به کامپیوتر ۲ را دارید. اگر فردی در کامپیوتر ۲ بخواهد از برنامه استفاده کند شما باید کدهایتان روی سیستم اول را کامپایل کنید، به عبارت دیگر باید آن کد منبع را به کد ماشینی تبدیل کنید که تنها برای کامپیوتر قابل خواندن است. در نتیجه فایل قابل اجرایی ایجاد می شود که روی سیستم ۲ دریافت و نصب می شود.

یکی از بهترین راه ها برای به خاطر سپردن این مساله این است که کد کامپایل شده برای سیستم ویندوزی را نمی توان روی مک اجرا کرد.

از سوی دیگر اگر بخواهید کاربر سیستم ۲ از برنامه تفسیر شده شما استفاده کند، می توانید زبان اسکریپت نویسی جاوا اسکریپت را با یک سایت یکپارچه کرده و لینک سایت را به سیستم ۲ ارسال کنید. این لینک در سیستم ۲ بدون نیاز به دانلود یا نصب و از طریق مرورگر وب تفسیر می شود. از آنجا که کد تفسیر شده به کامپایل نیاز ندارد می توان آن را در پلتفرم های کامپیوتری مختلف اجرا کرد.

۴ .فریمورک های نرم افزاری در برابر کتابخانه های نرم افزاری

فریمورک نرم افزاری یک چارچوب برنامه نویسی نه چندان انعطاف پذیر است که دستورالعمل های پروژه کد نویسی را مشخص می سازد. مهم نیست از چه زبانی استفاده می کنید، در هر حال فریمورک جنبه های زبان مورد استفاده در یک پارامتر را مشخص می کند. برای مثال Bootstrap به عنوان یک فریمورک فرانت اند شامل دستورات HTML و CSS برای طراحی رابط کاربری سایت و اپ های تحت وب است. روبی ان ریلز هم فریمورک بک اند است که چگونگی ذخیره سازی داده ها در سرور را مشخص می سازد.

کتابخانه نرم افزاری یک قطعه کد با قابلیت استفاده مجدد است که می توان برای تسریع در روند پروژه آنها را به کد اصلی اضافه کرد. فریمورک به کاربر اجازه می دهد با استفاده از کتابخانه قابلیت های برنامه را توسعه دهد. برای مثال روبی جمز کتابخانه هایی هستند که با استفاده زبان برنامه نویسی روبی توسعه پیدا کرده اند و با یکپارچه سازی آنها در فریمورک روبی ان ریلز می توان کارایی کد را بهبود بخشید.

۵ .شبه کد

برنامه نویس ها برای کد نویسی معمولا از دو روش استفاده می کند که شامل کد آماده با سینتکس مناسب و شبه کد است. سینتکس ساختار یک زبان خاص را مشخص کرده و بیشتر به قوانین و دستورالعمل های زبان به منظور نگارش دستورات خاص اشاره دارد.

شبه کد اما به توضیح منطق سینتکس با استفاده از زبان معمولی اشاره می کند. هدف برنامه نویس از نوشتن شبه کد نه ایجاد دستورات قابل اجرا بلکه مشخص کردن منطق دستور پیش از پیاده سازی آن با استفاده از سینتکس است. در واقع شبه کد با ایجاد قالبی مشخص برای کد، اجرای پروژه را تسهیل کرده و فرمت های کد نویسی را در ذهن ترسیم می کند.

برای مثال به منظور رتبه بندی یک آزمون باید چه مراحلی صورت گیرد؟ می توانید با عبارات منطقی بسیار ساده زیر شروع کنید:

```
SEE Test
LOOK At Question
MARK Answer
IF Correct, Check
IF Incorrect, X
WRITE Grade
```

اگرچه شبه کد فوق به درستی مراحل لازم برای امتیازدهی به یک آزمون را لیست می کند اما مقدار قابل توجهی از پروژه را به تصور برنامه نویس واگذار می سازد. آیا روش مورد نظر را به روشنی بیان کرده ایم؟ آیا باید پاسخ ها را به صورت تصادفی یا به ترتیب عددی بررسی کنیم؟ آیا امتیاز به صورت درصدی هم ارائه می شوند؟اگر بله به چه شکلی سوالات صحیح، غلط و کل آنها را به صورت درصد نهایی ارائه می کنیم؟

کاربر حین کد نویسی نهایی باید پاسخ این سوالات را در آستین داشته باشد و بهترین مکان پیدا کردن روش مناسب برای انجام آنها نگارش شبه کد است. با تمرین و تکرار بیشتر شبه کدهای شما با منطق برنامه نویسی و سینتکس ها هماهنگی بیشتری پیدا خواهد کرد.

این قوانین تنها شروع کار هستند

به شما حق می دهیم اگر هنوز آمادگی کامل برای شروع کار به عنوان توسعه دهنده نرم افزار را نداشته باشید. دنیای توسعه نرم افزاری بسیار وسیع و بی پایان است، یا حداقل چنین به نظر می رسد.

اگر این مقاله نخستین آشنایی شما با دنیای واژگان برنامه نویسی است، بعدا متوجه می شوید که این عبارت ها تا چه حد در جنبه های مختلف کد نویسی و به ویژه در مقاطع اولیه آن کاربرد دارند. دلیل این مساله این است که کلمات مربوط به برنامه نویسی همدیگر را تکمیل می کنند. در واقع با یادگیری هر کلمه، دو کلمه جدید وجود دارد.

در این مقاله با برخی از ضروری‌ترین واژگان مربوط به برنامه نویسی آشنا شدید و حالا وقت آن رسیده که به سراغ اصول ضروری کد نویسی بروید.