

مسیر یادگیری

توسعه وب به کمک

PHP



7LEARN

تهیه شده توسط تیم تولید محتوای سون لرن

فهرست محتوا

۴مقدمه
۵ زبان PHP چگونه شکل گرفت؟
۶ برنامه نویسی Back End چیست؟
۷ تفاوت زبان برنامه نویسی و اسکریپت نویسی چیست؟
۸ چرا بهتر است در توسعه سایت از زبان PHP استفاده کنیم؟
۸ نسخه های PHP
۹ PHP 4
۹ PHP 5
۹ PHP 6
۱۰ PHP 7
۱۱ PHP 8
۱۲ آشنایی با فریم ورک های PHP
۱۳ سیمفونی (Symfony)
۱۴ کیک پی اچ پی (CakePHP)
۱۴ کدایگنایتر (CodeIgniter)
۱۵ فالکون (Phalcon)
۱۵ زند (Zend)
۱۶ پیش نیازهای یادگیری PHP چیست؟

- ۱۷..... مسیر یادگیری زبان PHP
- ۲۳..... پاسخ به چند سوال مهم و پرتکرار
- ۲۳..... در چه مدت زمانی به یک متخصص PHP تبدیل می‌شویم؟
- ۲۴..... افراد مبتدی برای یادگیری PHP روزانه چند ساعت زمان بگذارند؟
- ۲۵..... یادگیری PHP برای افراد آشنا به برنامه‌نویسی چقدر طول می‌کشد؟
- ۲۵..... آیا با دانستن PHP می‌توانم یک سایت اختصاصی ایجاد کنم؟
- ۲۶..... در سایت‌های مبتنی بر وردپرس، یادگیری PHP چه کمکی به ما می‌کند؟
- ۲۶..... چه مشاغلی در حوزه زبان PHP وجود دارد؟
- ۲۷..... آینده زبان PHP چگونه است؟
- ۲۷..... سخن پایانی
- ۲۸..... دوره متخصص PHP در آکادمی سون لرن

مقدمه

PHP یکی از زبان‌های محبوب برنامه‌نویسی است که افراد زیادی علاقمند به یادگیری آن هستند. اگر به دنبال آشنایی با زبان PHP و یک نقشه راه برای یادگیری این زبان مفسری هستید یا در نیمه راه سرگردان شده‌اید، این کتاب کوتاه شما را به سرمنزل مقصود می‌رساند و دید روشنی از مسیری که پیش‌رو دارید، به شما خواهد داد.

این کتاب هدیه به کسانی‌ست که با وجود تمام سختی‌های زندگی، جسارت رشد و تغییر دارند.



زبان PHP چگونه شکل گرفت؟

در دهه ۹۰ میلادی که تکنولوژی وب هنوز نوپا و ناشناخته بود، یک برنامه‌نویس روس به نام راسموس لردورف (Rasmus Lerdorf) توانست با تست یک ایده، قلمرو برنامه‌نویسی وب را گسترش دهد. او اسکریپتی برای مدیریت بهتر صفحه شخصی خود نوشت که تعداد و برخی اطلاعات کاربران آن صفحه را نشان می‌داد. اسکریپت او مورد توجه بسیاری از برنامه‌نویسان قرار گرفت و لردورف نام اولیه آن را عبارت کوتاه شده‌ی "Personal Home Page" یعنی PHP گذاشت.

پس از این استقبال، لردورف شروع به توسعه‌ی PHP کرد و با همراهی تعداد زیادی از برنامه‌نویسان، خیلی زود و در اواخر دهه ۹۰ نسخه‌های بعدی این زبان، یعنی PHP.۲ و PHP.۳

نیز به کمک زبان C در اختیار عموم قرار گرفتند. در این زمان تغییرات بنیادینی در هسته PHP شکل گرفت که باعث شکل‌گیری موتور متن‌باز اسکرپتی زِند (Zend Engine) شد و ارائه PHP.۳ و نسخه‌های بعدی زبان برپایه این موتور انجام شد. در این مدت هزاران کاربر در حال استفاده از PHP برای توسعه و ارتقای صفحات سایت خود بودند.

این زبان شی‌گرا، در حوزه برنامه‌نویسی Back-End جای دارد و در پشت صحنه یک سایت نقش خود را ایفا می‌کند. اگر با حوزه Back-End آشنایی ندارید مطالعه مقاله زیر برای شما مفید است:

برنامه نویسی Back End چیست؟

PHP یک زبان تفسیرشونده است که اساس و بنیان آن طراحی و ایجاد صفحات پویای وب می‌باشد. به عبارت دیگر این زبان از طریق ارتباط با پایگاه داده، اطلاعات کاربران را ذخیره کرده و همچنین اطلاعات موردنیاز را به آن‌ها نشان می‌دهد.

ما قادر به مشاهده ظاهر یک سایت هستیم که با زبان‌هایی مانند HTML و CSS توسعه یافته است. اما تعاملی که با یک سایت داریم مانند ثبت‌نام در سایت، داشتن یک پنل کاربری، پر کردن فرم و... از طریق کدهایی صورت می‌گیرد که به زبان PHP نوشته شده‌اند و ما قادر به دیدن آن‌ها نیستیم.

بیشتر بخوانید: [آشنایی کامل تر با برنامه‌نویسی](#)

تفاوت زبان برنامه‌نویسی و اسکریپت‌نویسی چیست؟

وقتی با یک زبان برنامه‌نویسی مانند C کدنویسی کنیم، کدهای نوشته شده باید ابتدا کامپایل شده و سپس اجرا شوند. تفاوت یک زبان اسکریپت‌نویسی مانند PHP در این است که کدهای نوشته شده نیازی به مرحله کامپایل ندارند و مستقیم اجرا می‌شوند. البته با توجه به پیشرفت روزافزون زبان‌ها و سخت‌افزارهای کامپیوتری، تفاوت این زبان‌ها اهمیت خاصی ندارد.

چرا بهتر است در توسعه سایت از زبان PHP استفاده کنیم؟

این زبان از زمان پیدایش تاکنون، انتخاب بسیاری از برنامه‌نویسان حوزه وب بوده است. برخورداری از مزایای فوق‌العاده‌ای مانند:

- سرعت و امنیت بالا
- رایگان و متن‌باز بودن زبان
- قابلیت اجرایی در انواع سیستم‌عامل‌ها مانند: ویندوز و لینوکس
- استفاده بهینه از حافظه

و... منجر به برتری این زبان بر سایر رقبای خود شده است. همچنین سیستم مدیریت محتوای وردپرس برپایه این زبان گسترش یافته است. فریم‌ورک‌های قدرتمند PHP نیز یکی دیگر از عوامل محبوبیت این زبان هستند.

نسخه‌های PHP

توسعه‌دهندگان این زبان تاکنون ۸ نسخه از آن را به‌طور عمومی منتشر کرده‌اند. ما در تمام این سال‌ها شاهد رشد و پیشرفت مطلوب این زبان بودیم و بد نیست نگاه مختصری به نسخه‌های مختلف آن بیندازیم. از آنجا که قصد داریم به مسیر

یادگیری این زبان بپردازیم، ممکن است پرداختن به تمام به‌روزرسانی‌ها و ویژگی‌های هر نسخه از PHP از حوصله شما خارج بوده و دور از هدف این کتاب باشد، بنابراین در صورت علاقمندی به دانستن جزئیات پیشنهاد می‌کنیم به منابع باارزش موجود در اینترنت مراجعه کنید.

PHP 4

این نسخه که در سال ۲۰۰۰ عرضه شد، یک جهش بزرگ در روند توسعه PHP محسوب می‌شد. در این زمان تخمین زده شد تعداد کاربرانی که در سایت خود از این زبان استفاده می‌کنند از مرز ۲ میلیون نفر عبور کرده است. پشتیبانی از شی‌گرایی، پشتیبانی از زبان جاوا و بهبود مدیریت منابع از جمله مهم‌ترین تغییرات این نسخه بودند.

PHP 5

در سال ۲۰۰۴ شاهد عرضه نسخه 5 این زبان محبوب بودیم. توسعه‌دهندگان در این نسخه، پیش از معرفی ویژگی‌های جدید، به بهبود و ارتقاء توابع پیشین پرداختند.

PHP 6

یکی از مهم‌ترین تغییراتی که قرار بود در این نسخه اتفاق بیفتد، پشتیبانی از یونیکدها (Unicode) بود. به این معنی که

علاوه بر کاراکترهای تک‌بایتی، PHP بتواند از کاراکترهای چندبایتی نیز پشتیبانی کند. پس از معرفی نسخه پنجم، تیم توسعه به سراغ عملی کردن این ایده رفتند اما بنا بر دلایلی، توسعه نسخه ششم متوقف شد.

سایر ویژگی‌های مهم نسخه ششم نیز در نسخه‌های 5.3 و 5.4 از زبان ارائه شدند اما از آنجا که پشتیبانی از یونیکدها ویژگی بسیار مهمی بود و عده زیادی منتظر استفاده از آن در نسخه ششم بودند، PHP.6 که می‌توانست اصلی‌ترین نسخه‌ی این زبان باشد، منتشر نشد. دلیل عرضه نشدن PHP.6 این بود که کاربران در نسخه‌های بعدی منتظر پشتیبانی از یونیکدها نباشند و متوجه توقف این پروژه بشوند. در نتیجه بعد از عرضه نسخه 5.4، نسخه PHP.7 منتشر شد.

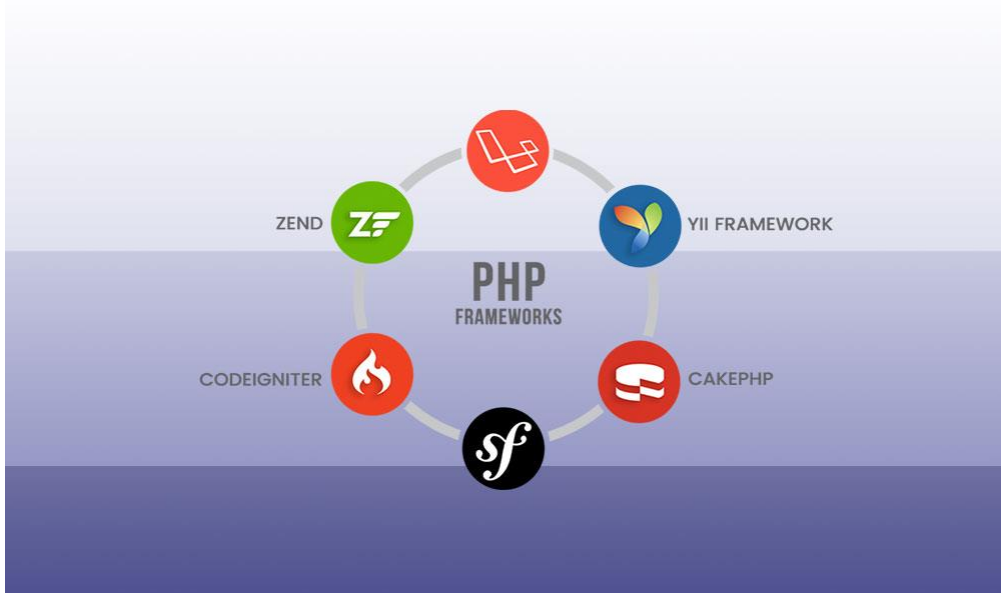
PHP 7

امروزه اغلب برنامه‌نویسان از این نسخه استفاده می‌کنند که ویژگی‌های کاربردی بسیاری دارد. سال ۲۰۱۵، با عرضه نسخه هفتم PHP، این زبان دوست داشتنی دوران جوانی خود را پشت سر گذاشت و به دوران بلوغ و پختگی خودش وارد شد. به تعبیری این نسخه از PHP را می‌توانیم به عنوان یکی از بزرگترین و بهترین به‌روزرسانی‌های یک زبان برنامه نویسی به حساب آوریم. از جمله تغییرات مهم این نسخه می‌توان به

بهبود عملکرد، کاهش استفاده از حافظه تا حدود ۵۰% و افزایش چشمگیر سرعت (حدود دو برابر اجرای آن) اشاره کرد. این افزایش سرعت و بهبود عملکرد را مدیون استفاده از نسخه سوم Zend Engine است.

PHP 8

توسعه‌دهندگان در این نسخه ویژگی‌های بسیار مهمی را به PHP اضافه کرده‌اند و نسخه هشتم گام بلندیست در جهت اینکه این زبان محبوب، به سریع‌ترین زبان مفسری دنیا تبدیل شود. اضافه شدن ویژگی‌هایی مانند Union Types، JIT و اپراتور NullSafe تنها بخشی از به‌روزرسانی‌های این زبان در نسخه 8 هستند.



آشنایی با فریمورک‌های PHP

فریمورک مجموعه‌ای از کتابخانه‌ها، استانداردها و قوانین کدنویسی است که در بستر یک چارچوب نرم‌افزاری در اختیار برنامه‌نویسان قرار می‌گیرد. فریمورک‌های برنامه‌نویسی برای هدفی خاص ایجاد شده‌اند. فریمورک‌های برنامه‌نویسی، به برنامه‌نویسان کمک می‌کنند کدهای بهینه‌تری نوشته و سریع‌تر به نتیجه برسند. در ادامه با تعدادی از بهترین فریمورک‌های زبان PHP آشنا می‌شویم:

لاراول (Laravel)

لاراول محبوب‌ترین فریمورک برنامه‌نویسان PHP است که در سال ۲۰۱۱ منتشر شد. فریمورک لاراول برای مدیریت برنامه‌های

پیچیده تحت وب به اولین گزینه توسعه‌دهندگان تبدیل شده است چرا که لاراوِل در بین سایر فریم‌ورک‌های PHP از امنیت و سرعت توسعه بیشتری برخوردار است. لاراوِل فریم‌ورکی مبتنی بر معماری MVC است و به صورت رایگان و متن باز در اختیار شما قرار دارد. **آموزش لاراوِل** برای برنامه‌نویسانی که با زبان PHP آشنایی دارند، بسیار آسان خواهد بود.

سیمفونی (Symfony)

این فریم‌ورک در سال ۲۰۰۵ در اختیار توسعه‌دهندگان قرار گرفت. سیمفونی سابقه بیشتری نسبت به سایر فریم‌ورک‌های PHP دارد و بسیار قابل اطمینان و منطع است. سیمفونی به عنوان یکی از سریع‌ترین فریم‌ورک‌های PHP شناخته می‌شود و ساخت وب‌اپلیکیشن را برای برنامه‌نویسان بسیار ساده‌تر می‌کند همچنین پایه و اساس فریم‌ورک لاراوِل را تشکیل می‌دهد.

بسیاری از برنامه‌نویسان PHP بر سر دو راهی لاراوِل یا سیمفونی قرار می‌گیرند. انتخاب یکی از آن‌ها به عنوان فریم‌ورک برتر، کاری سخت خواهد بود زیرا هر دو بسیار کارآمد هستند و مزیت‌های مشترک و خاصی دارند. بنابراین برای برنامه‌نویسان و توسعه‌دهندگانی که علاقه‌مند به ایجاد

ساختارهای برنامه‌نویسی خاص خود هستند، شاید توسعه سایت با سیمفونی یک انتخاب مناسب باشد.

کیک پی‌اچ‌پی (CakePHP)

این فریم‌ورک در اوایل سال ۲۰۰۰ عرضه شد و پلاگین‌ها و کامپوننت‌هایی دارد که باعث شده بیشتر در پروژه‌های تجاری و تبلیغاتی از آن استفاده شود.

از مزیت‌های CakePHP می‌توان به بهره‌مندی از جامعه‌ای فعال، به‌روزرسانی‌های مرتب، پشتیبانی قوی، ساخت آسان عملیات چهارگانه اصلی یعنی CRUD (نوشتن، خواندن، حذف کردن و به‌روزرسانی) و... اشاره کرد. این فریم‌ورک با شعار «سریع‌تر، قوی‌تر و خوشمزه‌تر» می‌تواند گزینه بسیار مناسبی برای برنامه‌نویسان PHP باشد.

کدایگنایتر (CodeIgniter)

کدایگنایتر در سال ۲۰۰۶ با هدف ساخت وب‌اپلیکیشن‌های مبتنی بر الگوی MVC توسعه داده شد. این فریم‌ورک به واسطه حجم کم و سرعت بالای خود شناخته می‌شود؛ چرا که نیاز به نصب و پیکربندی خاصی ندارد. فریم‌ورک کدایگنایتر از کتابخانه‌های قدرتمندی بهره‌مند است و بعد از فالتون به‌عنوان دومین فریم‌ورک سریع زبان PHP شناخته می‌شود. این فریم‌ورک بهترین انتخاب برای برنامه‌نویسانی خواهد بود که

به دنبال سادگی، سرعت و در عین حال قدرت در برنامه‌نویسی هستند.

فالكون (Phalcon)

فالكون سریع‌ترین فریم‌ورک زبان PHP محسوب می‌شود و بر پایه زبان C نوشته شده که زبانی محبوب و پرسرعت است. فالكون علاوه بر سرعت بالا، توانایی مدیریت تعداد زیادی از درخواست‌ها را در مدت زمان محدودی دارد، مبتنی بر MVC است و به شکل بهینه از منابع سرور استفاده می‌کند.

زند (Zend)

فریم‌ورک Zend گزینه‌ای بسیار مناسب برای توسعه پروژه‌های سازمانی است. این فریم‌ورک از معماری MVC پشتیبانی می‌کند و بر اساس متد چابک (Agile) طراحی شده است. همچنین Zend کاملاً از اصول شی‌گرایی (SOLID) در ساختار خود پیروی می‌کند و در به‌روزرسانی‌های اخیر با تغییرات و بهبودهای بسیاری همراه بوده است. آموزش فریم‌ورک Zend آسان است و منابع آموزشی فراوانی در سطح وب دارد.

پیش‌نیازهای یادگیری PHP چیست؟

PHP برای همه است و یادگیری آن پیش‌نیاز خاصی ندارد. باید بدانید برای چه می‌خواهید این زبان را یاد بگیرید و به کلنچار رفتن با منابع آموزشی و پروژه‌های تمرینی علاقه‌مند باشید. آشنایی با نحوه جستجو در اینترنت و زبان انگلیسی نیز از ملزومات یادگیری هر زبان برنامه‌نویسی است. البته این به معنی تسلط کامل بر زبان انگلیسی نیست بلکه کفایت توانایی خواندن در سطح متوسط را داشته باشید تا از منابع غنی و مهمی که به زبان انگلیسی هستند محروم نمانید.



مسیر یادگیری زبان PHP

ما می‌توانیم به کمک PHP یک وبسایت قدرتمند داشته باشیم اما ایجاد چنین سایتی به دو اصل مهم نیاز دارد.

اصل اول در هر کاری این است که سخت‌کوش باشید. کلمه‌ی "سخت" همان چیزی‌ست که معمولا ما را می‌ترساند و از خود دور می‌کند اما به کسانی که آن را می‌پذیرند و بر آن غلبه می‌کنند فرصت‌ها و موفقیت‌های بی‌نظیری هدیه می‌کند.

اصل دوم این است که استفاده از زبان PHP را یاد بگیرید و و تسلط نسبتا خوبی بر آن پیدا کنید. پس لازم است بدانیم یادگیری چه مواردی برای تسلط بر زبان برنامه‌نویسی PHP لازم است:

قدم اول: آشنایی با مبانی وب و پروتکل‌های مهم دنیای وب ابتدا بهتر است با اصول اولیه دنیای وب آشنا شوید. دانستن برخی استانداردها، آشنایی با معماری کاربر-سرور (Client-Server) و زبان‌های سمت سرور و سمت کاربر از جمله موارد پایه و مهم در یادگیری PHP هستند.

قدم دوم: یادگیری HTML و CSS

بعید است کسی در حوزه وب فعالیت کند و با این دو زبان آشنایی نداشته باشد. بنابراین لازم است در همین ابتدای کار به این دو زبان کاربردی مسلط شویم. البته یادگیری آن‌ها بسیار راحت است.

قدم سوم: آشنایی با IDE ها و راه‌اندازی محیط برنامه‌نویسی PHP

برای شروع یادگیری نیاز به یک محیط کدنویسی (IDE) دارید که بتوانید کدهای PHP را در آنجا نوشته و اجرا کنید. نرم‌افزارهایی مانند PHPStorm و VSCode از بهترین محیط‌های کدنویسی برای این زبان هستند.

پس از آن باید با سرورهای محلی (Local Servers) آشنا شوید زیرا PHP یک زبان سمت سرور است، به این معنی که کدهای آن روی سرور اجرا می‌شوند. لازم است نحوه نصب و راه‌اندازی

یک سرور محلی مانند زَمپ (XAMPP) یا وَمپ (WAMP) را یاد بگیرید و یک دامنه مجازی راه‌اندازی کنید.

قدم چهارم: یادگیری Git، سیستم کنترل ورژن سورس کد (نیاز عمومی)

یکی از نیازهای هر برنامه‌نویس حرفه‌ای آشنایی با سیستم‌های کنترل نسخه (VCS) است. به کمک این سیستم‌ها می‌توانید کدهای هر پروژه را با تیم خود به اشتراک بگذارید و تغییرات اعمال شده در کدها را راحت‌تر پیگیری کنید. Git یکی از معروف‌ترین سیستم‌های کنترل نسخه محسوب می‌شود.

قدم پنجم: یادگیری سینتکس و مبانی زبان PHP

برای شروع برنامه‌نویسی با زبان PHP به یادگیری قواعد لغوی این زبان (سینتکس) و مفاهیم رایج برنامه‌نویسی مانند ساختمان داده‌ها، انواع داده و کار با فایل نیاز دارید.

قدم ششم: یادگیری مفاهیم پایگاه داده و SQL

زبان PHP در برنامه‌نویسی سمت سرور کاربرد دارد بنابراین لازم است زبان SQL را بشناسیم و به‌خوبی بر نحوه کار با پایگاه داده مسلط شویم. تسلط بر این مورد برای یادگیری بسیار ضروری است پس تا جای ممکن مهارت خود را با انجام تمرین و پروژه‌های عملی ارزیابی کنید.

قدم هفتم: یادگیری Composer، مدیریت پکیج در PHP

Composer ابزاریست که با استفاده از دستورات از پیش تعریف شده آن می‌توانیم به مدیریت پکیج‌ها بپردازیم. کامپوزر، فرایند مدیریت وابستگی (Dependency Management) را برای توسعه‌دهندگان PHP تسهیل می‌کند.

قدم هشتم: یادگیری شی‌گرایی و مباحث پیشرفته

شی‌گرایی یکی از مهم‌ترین روش‌های برنامه‌نویسی است که در آن با مفاهیمی همچون کلاس، اشیا، وراثت و... آشنا می‌شویم. یادگیری این مبحث، بسیار جذاب و البته برای حرفه‌ای شدن بسیار مهم است و در سایر زبان‌های برنامه‌نویسی مطرح نیز کاربرد دارد.

قدم نهم: یادگیری مباحث بهبود امنیت و عملکرد در PHP

نیازی به گفتن نیست که تامین امنیت نرم‌افزار نیاز به آموزش تخصصی دارد و باید به خوبی بر این مبحث مسلط شوید. بهینه‌سازی عملکرد و سرعت سایت نیز یکی از چالش‌ها و حتی مزایای رقابتی کسب‌وکارها محسوب می‌شود که باید در آن حرفه‌ای شوید.

قدم دهم: یادگیری توسعه وب سرویس و API در PHP

در این مرحله باید بتوانید وب سرویس‌هایی را برای اتصال از طریق برنامه‌های موبایل و یا وب‌اپلیکیشن‌های دیگر پیاده‌سازی کنید. وب سرویس‌ها یکی از مهم‌ترین مباحث در کار با زبان‌های سمت سرور هستند و بهتر است با انجام پروژه‌های متنوع و کاربردی، از یادگیری و تسلط خود مطمئن شوید.

قدم یازدهم: یادگیری اصول کدنویسی تمیز و الگوهای طراحی یکی از مهارت‌هایی که برنامه‌نویسان حرفه‌ای را از بقیه متمایز می‌کند، توانایی آن‌ها در به کارگیری اصول کدنویسی تمیز است. توانایی به کارگیری صحیح الگوهای طراحی و حفظ استانداردهای کدنویسی نیز یکی از نکاتیست که در کدنویسی تمیز به آن توجه می‌شود.

قدم دوازدهم: یادگیری تست نویسی برای نرم افزار در PHP یکی از مهم‌ترین روش‌های صرفه‌جویی در زمان انجام پروژه، تست نویسی است. برای جلوگیری از هدر رفتن منابع باید با انواع روش‌های تست‌نویسی و ابزارهای آن آشنا باشید. این کار باعث کاهش خطاهای موجود در کد شده و فرایند اشکال‌زدایی را سریع‌تر می‌کند.

قدم سیزدهم: آشنایی با معماری MVC و فریم‌ورک‌های PHP

معماری MVC در توسعه نرم‌افزارها اهمیت زیادی دارد و به کمک آن می‌توانیم بخش‌های مختلف یک پروژه را مستقل از هم توسعه داده و در نهایت بین این بخش‌ها ارتباط ایجاد کنیم. برای انجام پروژه‌های واقعی، تسلط بر این مدل معماری ضروری به نظر می‌رسد.



پاسخ به چند سوال مهم و پرتکرار

در چه مدت زمانی به یک متخصص PHP تبدیل می‌شویم؟ مدت زمان یادگیری بستگی به مواردی چون هدف از یادگیری، زمان آزاد و همچنین میزان آشنایی قبلی با برنامه‌نویسی دارد. اگر به طور کلی با حوزه برنامه‌نویسی آشنایی ندارید بهتر است ابتدا با برخی مفاهیم مرتبط با این حوزه مانند الگوریتم‌ها، ساختمان داده‌های مهم و غیره آشنا شده و گام‌به‌گام در مسیر یادگیری PHP مقدماتی تا پیشرفته پیش بروید.

بیشتر بخوانید: [کتاب راهنمای شروع یادگیری برنامه‌نویسی](#)

افراد مبتدی برای یادگیری PHP روزانه چند ساعت زمان بگذارند؟

اگر از سطح مقدماتی شروع کرده و روزی حداقل ۳ ساعت برای یادگیری وقت بگذارید، احتمالاً در پایان ماه هفتم به دانش خوبی دست پیدا خواهید کرد. مسئله مهم این است که آموزش را با تکیه بر تمرین زیاد و پروژه پیش ببرید. به این معنی که علاوه بر انجام تمرین‌های مربوط به سرفصلی که در حال یادگیری آن هستید، باید گاهی پروژه‌های کوچک برای خود تعریف کرده و همه آنچه که تا کنون یاد گرفته‌اید را در انجام آن به کار ببرید. شما زمانی به یک متخصص PHP تبدیل می‌شوید که علاوه بر دانش تئوری، تجربه انجام پروژه را داشته و به شکل تئوری و عملی بر مباحث این زبان مسلط باشید.

پس بهتر است در مسیر یادگیری از بروز این دو حالت پیشگیری کنید:

۱. پس از گذشت چند ماه تمامی سرفصل‌ها را یاد گرفته‌اید اما مهارت لازم را برای انجام پروژه ندارید.
۲. یادگیری را کاملاً پروژه‌محور پیش می‌برید اما مدت زمان یادگیری افزایش پیدا کرده است.

یادگیری PHP برای افراد آشنا به برنامه‌نویسی چقدر طول می‌کشد؟

اگر پیش از این برنامه‌نویسی را تجربه کرده‌اید، فرایند یادگیری برای شما کوتاه و ساده‌تر خواهد بود. فراموش نکنید که در مسیر یادگیری باید دو عامل بسیار مهم یعنی علاقه و زمان را در نظر بگیرید. علاقه به یادگیری باعث می‌شود در شرایط سخت هم تاب بیاورید، درجا نزنید و در پایان شیرینی نتیجه‌ای که کسب می‌کنید دوچندان به جانتان بنشیند.

برنامه‌نویسی دنیای فرصت‌هاست. بنابراین شاید بتوان گفت سنگین‌ترین هزینه‌ای که در این مسیر می‌پردازید، زمان است. بهتر است از دوره‌های معتبر استفاده کنید و یا اگر قصد خودآموزی دارید، با توجه به منابع موجود، برنامه‌ریزی کرده و به انجام آن متعهد بمانید تا مدت زمان یادگیری بیش از حد (مثلا ۱ سال یا حتی بیشتر) طول نکشد.

آیا با دانستن PHP می‌توانم یک سایت اختصاصی ایجاد کنم؟

زبان PHP یکی از زبان‌های سمت سرور است و در توسعه بخش Back-End در برنامه‌نویسی وب استفاده می‌شود. شما برای توسعه یک سایت به آشنایی با زبان‌های حوزه Front-

End هم نیاز دارید که حوزه‌های جداگانه است. بنابراین لازم است از برنامه‌نویسان حوزه طراحی وب نیز کمک بگیرید.

بیشتر بخوانید: برنامه نویسی Front-End چیست؟

در سایت‌های مبتنی بر وردپرس، یادگیری PHP چه کمکی به ما می‌کند؟

سیستم رایگان وردپرس بر مبنای زبان PHP نوشته شده است و امکانات متنوعی را به طور پیش‌فرض برای ایجاد یک سایت در اختیار شما قرار می‌دهد. با این حال امکان تغییر عملکردهای پیش‌فرض، تغییر قالب‌ها و افزونه‌های وردپرس برای شما وجود دارد و به کمک اسکریپت‌های PHP انجام می‌شود. به عبارت دیگر، آشنایی با PHP به شما کمک می‌کند که یک سایت مبتنی بر وردپرس بسازید و مطابق میل خود آن را شخصی‌سازی کنید.

چه مشاغلی در حوزه زبان PHP وجود دارد؟

کاربرد عمده‌ی این زبان در بخش Back-End سایت است. با توجه به افزایش تعداد کسب‌وکارهای آنلاینی که نیاز به حداقل یک وب‌سایت دارند بازار کار برنامه‌نویسان PHP همچنان رونق خوبی خواهد داشت. همچنین راه‌اندازی سایت شخصی خود

با اهداف تجاری از جمله برندینگ فردی نیز از دیگر دستاوردهای یادگیری PHP است.

آینده زبان PHP چگونه است؟

دلایل زیادی وجود دارد که نشان می‌دهد PHP حداقل تا ۵ الی ۱۰ سال آینده عقب‌نشینی نخواهد کرد. فعالیت مستمر توسعه‌دهندگان این زبان در به‌روزرسانی نسخه‌های جدید، اضافه کردن ویژگی‌های کاربردی، استفاده گسترده از این زبان در سراسر جهان و کارایی PHP در سایت‌های بزرگی مانند فیسبوک و پینترست تنها بخشی از عواملی هستند که در سال‌های پیش‌رو زبان PHP را زنده نگه خواهند داشت.

بنابراین بهتر است نگران افول PHP نباشید و به‌خوبی برای مسلط شدن بر مفاهیم برنامه‌نویسی تلاش کنید.

سخن پایانی

در این کتاب به نحوه شکل‌گیری PHP، نسخه‌ها و فریم‌ورک‌های این زبان و مسیر یادگیری آن پرداختیم و به تعدادی از سوالات پرتکرار در این زمینه پاسخ دادیم. فراموش نکنید که یادگیری برنامه‌نویسی می‌تواند برای آینده شما بسیار مفید باشد به شرط آنکه کنجکاو و سخت‌کوش باشید تا بتوانید

علاوه بر کسب درآمد در این زمینه، به سراغ عملی کردن ایده‌های بزرگ بروید.

در مسیر یادگیری در کنار شما هستیم. دوره‌های متخصص مرجع خوبی برای یادگیری PHP هستند اما در پایگاه دانش سون‌لرن نیز مطالب آموزشی رایگانی برای شما آماده کرده‌ایم تا بتوانید مسیر خودآموزی را شروع کنید.

دوره متخصص PHP در آکادمی سون‌لرن

اگر به دنبال یک روش جامع و تضمینی برای یادگیری زبان PHP و دست یافتن به شغل دلخواه‌تان هستید، پیشنهاد می‌کنیم از دوره متخصص PHP در آکادمی سون‌لرن غافل نشوید.

بیشتر بخوانید: [دوره متخصص PHP در آکادمی سون‌لرن](#)

در پایان از اینکه برای خواندن این کتاب وقت گذاشته‌اید، از شما سپاس‌گزاریم و در شبکه‌های اجتماعی سون‌لرن آماده دریافت نظرات و پیشنهادات شما هستیم.

از یادگیری تا ورود به بازار کار

www.7Learn.com