

به زبان ساده، برنامه نویسی فانکشنال یک سبک توسعه نرم افزار است که تأکید عمده‌ای بر استفاده از توابع دارد.

در برنامه نویسی تابعی، توابع به عنوان اجزای اولیه مورد استفاده قرار می گیرند، به این معنی که **می توانند به متغیرها اختصاص داده شوند، به عنوان آرگومان به سایر توابع منتقل شوند و به عنوان نتیجه از توابع بازگردانده شوند.** توابع به طور معمول خالص هستند، به این معنی که برای هر ورودی، همان خروجی را تولید می کنند و تغییری در حالت خارجی ایجاد نمی کنند.

اصول کلیدی برنامه نویسی فانکشنال :

- (۱) عدم تغییر پذیری (Immutability)
- (۲) توابع خالص (Pure Functions)
- (۳) توابع مرتبه بالا (Higher-Order Functions)
- (۴) بازگشت (Recursion)
- (۵) شفافیت ارجاعی (Referential Transparency)

۱) عدم تغییر پذیری (Immutability) : داده‌ها به طور معمول غیرقابل تغییر هستند، به این معنی که پس از ایجاد، قابل اصلاح نیستند. به جای اینکه داده تغییر یابد، از طریق تبدیلات، داده‌های جدید ایجاد می‌شوند.

۲) توابع خالص (Pure Functions) : توابع خالص بدون اثرات جانبی هستند و برای هر ورودی، همان خروجی را تولید می‌کنند. آنها تنها به پارامترهای ورودی خود وابسته هستند و حالت خارجی را تغییر نمی‌دهند.

۳) توابع مرتبه بالا (Higher-Order Functions) : توابع می‌توانند توابع دیگر را به عنوان آرگومان دریافت یا توابع را به عنوان نتیجه بازگردانند. این امکان را برای ترکیب و انتزاع رفتار فراهم می‌کند.

۴) بازگشت (Recursion) : حلقه‌ها به طور معمول با بازگشت جایگزین می‌شوند، به این معنی که توابع خود را فراخوانی می‌کنند تا وظایف تکراری را انجام دهند.

۵) شفافیت ارجاعی (Referential Transparency) : عبارات را می‌توان با مقادیر متناظر آنها جایگزین کرد بدون اینکه رفتار برنامه تحت تأثیر قرار گیرد. این ویژگی به تفکر درباره کد به شکل آسان‌تر کمک می‌کند.

دکوراتورها به عنوان توابع High Order در پایتون در نظر گرفته می‌شوند.

دکوراتور تابع مرتبه بالا است که یک تابع را به عنوان آرگومان دریافت و یک تابع جدید را با رفتار اصلاح شده برمی‌گرداند. دکوراتورها برای افزودن قابلیت به تابع یا عملکردی موجود یا تغییر رفتار آن استفاده می‌شوند.