# کامپوننت چیست؟

اصلی‌ترین واحد در انگولار است. هر اپلیکیشن انگولار از مجموعه‌ای از کامپوننت‌ها تشکیل می‌شود.

# اجزای کامپوننت

آنگولار از معماری MVC استفاده می‌کند و به همین خاطر هر کامپوننت شامل یک کلاس TypeScript برای Logic و فایل‌هایی برای HTML و CSS برای UI است.

# ارتباط UI و Logic

در آنگولار این ارتباط از طریق Binding انجام می‌شود.

# انواع Binding در آنگولار

* String Interpolation: برای نمایش خروجی از آن استفاده می‌شود، یعنی Binding به صورت یکطرفه است. {{}}
* Property Binding: از نوع Binding یکطرفه است و برای مقداردهی Property های HTML Tag مقصد استفاده می‌شود. []
* Event Binding: Bind کردن مقادیر یک HTML Tag به کلاس TypeScript مربوط به Logic. ()
* Two-way Binding: ترکیبی از Property Binding (ارتباط یکطرفه از Logic به UI) و Event Binding (ارتباط یکطرفه از UI به Logic). [()]

# انتقال داده بین Component ها

* ارسال از کامپوننت پدر به فرزند: در کامپوننت فرزند یک property تعریف می‌کنیم که با Decoratorی با نام @Input() مشخص می‌شود و از طریق selector آن کامپوننت در پدر قابل دسترس است.
* ارسال از کامپوننت فرزند به پدر: این کار باید از طریق event انجام شود، پس در کامپوننت فرزند یک event تعریف می‌کنیم که با Decoratorی با نام Output مشخص می‌شود و از طریق selector آن کامپوننت در پدر قابل دسترس است.

# Style دهی به کامپوننت‌ها

* ViewEncapsulation.Emulated (پیش فرض): Style ها از HTML اصلی به Component منتقل می شوند و همچنین Style های تعریف شده برای آن کامپوننت فقط برای آن کامپوننت است.
* ViewEncapsulation.Native: Style های HTML اصلی به کامپوننت منتقل نمی‌شوند و همچنین Style های تعریف شده برای آن کامپوننت فقط برای آن کامپوننت است.
* ViewEncapsulation.None: Styleهای کامپوننت به کل اپلیکیشن انتقال می‌یابد، دقت شود اپلیکیشن‌هایی که دارای اجزای None و Native در خود هستند. همه اجزای دارای کپسول‌سازی None در همه اجزای دارای کپسوله سازی Natvie، Style خود را منتشر می‌کنند.