**$event:** هرگاه برای استفاده از تابع در تگ html از یک آرگومان استفاده کنیم مانند:‌func(arg) و به event داخلی آن تگ نیاز داشته باشیم می­توانیم با $event به آن دسترسی داشته باشیم: func($event, arg) .

**نکته:** فراخوانی تابع در کد html مانند {{ func() }} باعث کاهش performance می­شود زیرا با تغییر هر بخش از صفحه، vue آن تابع را نیز دوباره اجرا می­کند، راه حل بهتر برای اینکار استفاده ازcomputed است.

**عبارات شرطی:** می­توانیم از v-if=’’ و v-else-if=’’ و v-else استفاده کنیم. همچنین می­توانیم به جای v-if از v-show استفاده کنیم اما v-show از v-else پشتیبانی نمی­کند، بهتر است برای تگ­هایی که خیلی visible و hidden میشن از v-show استفاده کنیم چون v-show فقط css تگ را تغییر می­دهد اما v-if کل تگ را از DOM حذف و یا به آن اضافه می­کند و ممکن است برای تگ­هایی که خیلی مخفی و آشکار می­شوند مناسب نباشد و کارایی را کاهش دهد.

دیتا را میتوان به وسیله prop و emit بین کامپوننت فرزند و والد رد و بدل کرد اما نمیتوان بین دو کامپوننت فرزند رد و بدل کرد بنابراین برای انتقال دیتا از یک کامپوننت فرزند به دیگر کامپوننت فرزند باید ابتدا دیتا را به کامپوننت والد فرستاده و از آنجا به کامپوننت فرزند دیگر منتقل کرد.

برای رد و بدل کردن دیتا بین دو کامپوننت فرزند میتوان در فایل main، یک شی دیگر از ویو ساخته و در کامپوننت های فرزند از متدهای آن شی استفاده کنیم، مانند متدهای .$emit & $on

Slot: برای انتقال کدهای html به کامپوننت فرزند از سوی والد استفاده می­شود. برای style دادن به کدهای انتقال یافته باید style مورد نظر در کامپوننت فرزند اعمال شود، زیرا در صورتی که style کامپوننت والد در حالت scoped باشد استایلی برای کدهای در حال انتقال اعمال نمی­شود. همچنین غیر از style، کدهای دیگر ویو مانند دیتای متغیر یا v-if & v-for و ...، باید در کامپوننت والد و بین تگ­های کامپوننت فرزندی که در والد فراخوانی شده است نوشته شود.

@click.native: هنگامی استفاده می‌شود که بخواهیم روی یک کامپوننت رخداد کلیک را تعریف کنیم، زیرا کامپوننت فراخوانی شده به Dom اضافه نشده است چون فقط کدهای html موجود در کامپوننت به Dom اضافه می‌شوند.