* **DOM**: Là thể hiện các dữ liệu và cấu trúc của một trang html.
* **DOM ảo**: Là phần code trong thẻ <template>. DOM ảo sẽ được deploy thành DOM thật khi ta thấy đã thấy các component trên màn hình trình duyệt.
* **ở stage created không dùng được JQuery**: Vì ở created chỉ mới có DOM ảo (đến lúc mounted mới có DOM thật để JQuery thao tác) nên nên dùng JQuery ở stage mount.
* **Props**: Có thể hiểu nôm na là một hàm sẽ cần có những tham số truyền vào, thì tương tự, một component con sẽ cần được truyền các "tham số cần thiết" (nếu có) từ component cha khi thằng cha gọi đến thằng con.
* **this.$refs.(tên biến sử dụng trên phần template):** Để lấy và thao tác một phần tử trên web (vd như các inputs của người dùng) để thực hiện cho một yêu cầu cụ thể nào đó từ component hiện tại.
* **Custom Event**: Dùng this.$emit({tên event cần gửi}, {tham số cần thay đổi}, {một string ví dụ như updated}) để gửi những thay đổi của các thuộc tính từ component con đến component cha, các component cha sẽ bắt tín hiệu từ phần {tên event cần gửi} và thực hiện các methods như thông thường kèm với tham số đã được thay đổi.
* **Slot**: Cho phép chèn phần html khác vào các phần slot mà được dặt trong đặt trong 1 component (Giống với cách thức yield - extends - section của Laravel).
* **Named slot**: Khi có nhiều chỗ cần dùng slot, ngoài default slot (không cần đặt tên) ra, ta sẽ đặt tên cho những chỗ cần dùng thêm slot ví dụ như footer hay header.

Ví dụ: Khung Ban đầu (Model.vue):



Khi sử dụng, ta sẽ gọi component Model ra như sau:



Phần v-slot: có thể được viết ngắn gọn bằng kí tự #, ví dụ:



* **Scoped** **slot**: Sử dụng phương thức này khi muốn bind cả những thuộc tính hay phương thức ở component con đi kèm với slot về component cha

Ví dụ: Ở component con



Và ở component cha khi gọi đến:



Hoặc có thể viết theo cách như sau:



**Mixin**: Dùng khi mong muốn tách biệt các logic thành các file riêng để tái sử dụng ở nhiều nơi, và nơi sử dụng nó chỉ cần chứa phần template UI và gọi đến các file logic đó.  
Khi nhiều files được import vào, nếu trùng lặp về tên thì bên trong script sẽ diễn ra như sau:

* Đối với phần mounted: Sẽ diễn ra tuần tự từ trên xuống trong script.
* Đối với các phần data hay methods: Các phần bị trùng sẽ bị ghi đè lên, và nội dung ưu tiên nhất là của thằng cuối cùng của script.
* Đối với các phần không bị trùng thì được giữ nguyên và gộp vào.