Trong framework Vue.js, **computed** và **methods** là hai khái niệm quan trọng được sử dụng để xử lý logic trong các đối tượng Vue. Dưới đây là sự khác nhau giữa chúng và khi nào bạn nên sử dụng mỗi loại:

**1. Computed Properties:**

* **Đặc điểm:**
  + Thường được sử dụng để tính toán giá trị dựa trên các thuộc tính khác trong đối tượng Vue.
  + Có bản chất giống như một thuộc tính, nhưng Vue sẽ tự động quản lý việc cập nhật khi các giá trị phụ thuộc thay đổi.
* **Khi nào sử dụng:**
  + Khi bạn muốn tính toán một giá trị dựa trên các thuộc tính khác và muốn tự động cập nhật nếu có sự thay đổi.
  + Thích hợp cho các tính toán động và tài nguyên tốn kém.
* **Ví dụ:**

javascriptCopy code

computed: { fullName() { return this.firstName + ' ' + this.lastName; } }

**2. Methods:**

* **Đặc điểm:**
  + Chứa các hàm được gọi khi có sự kiện hoặc hành động từ người dùng.
  + Không tự động cập nhật giống như **computed**.
* **Khi nào sử dụng:**
  + Khi bạn cần thực hiện một hành động cụ thể khi có sự kiện, chẳng hạn như click button hoặc submit form.
  + Thích hợp cho các hành động không liên quan đến tính toán giá trị hiển thị.
* **Ví dụ:**

javascriptCopy code

methods: { greet() { alert('Hello, ' + this.fullName); } }

**Lý do sử dụng Computed:**

1. **Hiệu suất tốt hơn:**
   * **computed** giữ các giá trị được tính toán và chỉ cập nhật khi các giá trị phụ thuộc thay đổi, giúp giảm thiểu lượng tính toán không cần thiết.
2. **Độ đồng bộ tự động:**
   * Vue tự động quản lý việc cập nhật các giá trị **computed**, đảm bảo rằng chúng sẽ luôn đồng bộ với các thuộc tính khác.
3. **Đọc dễ hiểu:**
   * **computed** thường được sử dụng cho các tính toán phức tạp, làm cho mã nguồn dễ đọc hơn so với việc đặt tất cả logic vào các phương thức.

**Lý do sử dụng Methods:**

1. **Hành động người dùng:**
   * **methods** thích hợp cho các hành động người dùng như click, submit form, nơi bạn cần thực hiện một hành động cụ thể.
2. **Không tính toán giá trị:**
   * Khi không cần tính toán giá trị nào và chỉ cần thực hiện một hành động, sử dụng **methods** để giữ mã nguồn sáng sủa hơn.

Tóm lại, khi bạn cần tính toán giá trị dựa trên các thuộc tính khác và muốn có sự đồng bộ tự động, hãy sử dụng **computed**. Ngược lại, khi bạn chỉ cần thực hiện một hành động cụ thể, hãy sử dụng **methods**. Có thể kết hợp cả hai để đạt được sự linh hoạt tốt nhất trong quá trình phát triển ứng dụng Vue.js của bạn.

Trong Vue, nếu bạn gọi một phương thức trong một biểu thức (**{{ ... }}**), nó sẽ thực thi mỗi khi có bất kỳ sự thay đổi nào trong dữ liệu. Trong trường hợp phương thức **{{addB()}}** và **{{addA()}}** đều đang sử dụng các phương thức để hiển thị kết quả.

Mỗi khi có sự thay đổi trong dữ liệu (bao gồm cả **a** và **b**), Vue sẽ thực thi lại tất cả các biểu thức, bao gồm cả cả **addA** và **addB**. Điều này có thể tạo ra sự nhầm lẫn khi bạn chỉ muốn thực hiện một phương thức cụ thể khi một điều kiện cụ thể được đáp ứng.