**1. Hiệu quả và hiệu suất**

* **Sinh viên A** sử dụng các thẻ <div> và display: inline-block, cộng với margin tĩnh và chiều rộng cố định. Cách này không tối ưu về hiệu suất hiển thị vì dễ gây dư thừa không gian, thiếu linh hoạt khi màn hình thay đổi kích thước.
* **Sinh viên B** dùng display: flex kết hợp với flex-wrap, giúp trình duyệt phân bố phần tử một cách hiệu quả hơn, tận dụng không gian tốt hơn. Mã CSS ngắn gọn và gọn gàng hơn so với sinh viên C.
* **Sinh viên C** tuy cũng sử dụng flex nhưng có thêm nhiều lớp lồng nhau như .article-details, làm tăng độ sâu của cây DOM, có thể ảnh hưởng nhẹ đến hiệu suất khi danh sách bài viết dài.

**Kết luận:** Sinh viên B tối ưu hơn về hiệu suất, kế tiếp là sinh viên C, và sau cùng là A.

**2. Độ dễ hiểu và bảo trì**

* **Sinh viên A** viết mã rất đơn giản, dễ hiểu, ít lớp, dễ đọc. Tuy nhiên, cấu trúc này sẽ khó bảo trì nếu cần mở rộng thêm thông tin cho bài viết (ví dụ: thêm ảnh, nút chia sẻ...).
* **Sinh viên B** tổ chức mã gọn gàng, có cấu trúc rõ ràng nhờ sử dụng danh sách (<ul>, <li>), dễ duy trì và mở rộng sau này.
* **Sinh viên C** tổ chức cấu trúc tốt, tách riêng nội dung mô tả vào .article-details, thuận tiện khi cần thay đổi hoặc thêm định dạng. Tuy nhiên, việc phân lớp nhiều hơn khiến mã dài và có thể hơi rối với người mới.

**Kết luận:** Sinh viên B dễ hiểu và dễ bảo trì nhất, tiếp theo là C, cuối cùng là A.

**3. Tính linh hoạt và khả năng tái sử dụng**

* **Sinh viên A** dùng thẻ <div> lặp lại theo cùng một cấu trúc, không tách biệt rõ ràng giữa các phần tử, do đó khó tái sử dụng hoặc áp dụng vào các thành phần khác của trang web.
* **Sinh viên B** dùng <ul> và <li> – cấu trúc danh sách rõ ràng, dễ lặp, dễ tái sử dụng và áp dụng trong các ngữ cảnh tương tự như blog, tin tức...
* **Sinh viên C** có cấu trúc lồng ghép rõ ràng, chia nhỏ các khối như .article-details giúp tái sử dụng tốt cho các giao diện phức tạp hơn, ví dụ thêm hình ảnh, phân loại, nút chia sẻ...

**Kết luận:** Sinh viên C có tính linh hoạt cao nhất, sau đó đến B và A.

**4. Tuân thủ nguyên tắc thiết kế web**

* **Sinh viên A** không sử dụng thẻ HTML ngữ nghĩa (semantic HTML) như <ul>, <li>, dẫn đến kém về khả năng truy cập (accessibility) và SEO. Cũng không có dấu hiệu hỗ trợ responsive rõ ràng.
* **Sinh viên B** sử dụng thẻ danh sách, có flex-wrap, nên hỗ trợ tốt cho thiết kế đáp ứng (responsive) và cũng tuân thủ tốt nguyên tắc thiết kế web ngữ nghĩa.
* **Sinh viên C** không dùng thẻ danh sách nhưng phân chia lớp rõ ràng, kết hợp flex và bố cục hợp lý nên cũng hỗ trợ tốt responsive và dễ tiếp cận.

**Kết luận:** Sinh viên B tuân thủ tốt nhất các nguyên tắc thiết kế web, tiếp đến là C và cuối cùng là A.