

Đây là giải thích chi tiết về các lớp CSS mà bạn đã cung cấp:

**1. .layout-right**

.layout-right {

width: 694px;

height: 243px;

background-color: #c4c4c4;

border-radius: 12px;

position: absolute;

transform: translateX(30px) translateY(30px);

z-index: 1;

}

* **width: 694px;**: Chiều rộng của phần tử .layout-right là 694px.
* **height: 243px;**: Chiều cao của phần tử .layout-right là 243px.
* **background-color: #c4c4c4;**: Màu nền của phần tử là màu xám (#c4c4c4).
* **border-radius: 12px;**: Các góc của phần tử này được bo tròn với bán kính 12px.
* **position: absolute;**: Phần tử này được định vị tuyệt đối so với phần tử cha (hoặc với phần tử gần nhất có position: relative).
* **transform: translateX(30px) translateY(30px);**: Phần tử sẽ được dịch chuyển 30px sang phải (X) và 30px xuống dưới (Y) từ vị trí gốc của nó.
* **z-index: 1;**: Phần tử này sẽ nằm dưới các phần tử có z-index lớn hơn, nhưng trên các phần tử có z-index nhỏ hơn.

**2. .food-card**

.food-card {

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

width: 694px;

height: 243px;

background-color: #587330;

border-radius: 12px;

padding: 20px;

position: relative;

z-index: 2;

}

* **display: flex;**: Phần tử sử dụng mô hình Flexbox để định dạng các phần tử con bên trong.
* **flex-direction: column;**: Các phần tử con của .food-card sẽ được sắp xếp theo chiều dọc (column).
* **justify-content: center;**: Các phần tử con sẽ được căn giữa theo chiều dọc trong phần tử .food-card.
* **align-items: center;**: Các phần tử con sẽ được căn giữa theo chiều ngang trong phần tử .food-card.
* **width: 694px;**: Chiều rộng của phần tử .food-card là 694px.
* **height: 243px;**: Chiều cao của phần tử .food-card là 243px.
* **background-color: #587330;**: Màu nền của phần tử là màu xanh lá cây đậm (#587330).
* **border-radius: 12px;**: Các góc của phần tử này cũng được bo tròn với bán kính 12px.
* **padding: 20px;**: Phần tử có khoảng cách 20px giữa nội dung và các cạnh của phần tử .food-card.
* **position: relative;**: Phần tử này được định vị tương đối, có thể sử dụng position và top, left nếu cần thiết để di chuyển.
* **z-index: 2;**: Phần tử này sẽ nằm trên các phần tử có z-index nhỏ hơn, nhưng dưới các phần tử có z-index lớn hơn.

**Tóm tắt**

* .layout-right là một phần tử có nền xám, được đặt ở vị trí tuyệt đối và dịch chuyển ra ngoài so với vị trí gốc của nó. Nó nằm dưới .food-card vì có z-index nhỏ hơn.
* .food-card là một phần tử có nền xanh lá cây, sử dụng Flexbox để căn giữa các phần tử con. Nó được định vị tương đối và có z-index lớn hơn, nên nó sẽ hiển thị trên .layout-right.

Hy vọng giải thích này giúp bạn hiểu rõ hơn về cách hoạt động của các lớp CSS này!



Đây là giải thích chi tiết về các lớp CSS mà bạn đã cung cấp:

**1. .vertical-line**

.vertical-line {

position: absolute;

top: 20px;

bottom: 20px;

left: 50%;

height: 100%;

width: 1px;

background-color: #587330;

margin-top: -20px;

}

* **position: absolute;**: Phần tử .vertical-line được định vị tuyệt đối trong một phần tử chứa có position: relative.
* **top: 20px;**: Phần tử sẽ cách 20px từ cạnh trên của phần tử chứa.
* **bottom: 20px;**: Phần tử sẽ cách 20px từ cạnh dưới của phần tử chứa.
* **left: 50%;**: Phần tử sẽ được căn ở giữa chiều ngang của phần tử chứa, dựa trên vị trí 50% từ bên trái.
* **height: 100%;**: Chiều cao của đường thẳng này sẽ chiếm toàn bộ chiều cao của phần tử chứa.
* **width: 1px;**: Độ rộng của đường thẳng là 1px, tức là một đường kẻ thẳng đứng mảnh.
* **background-color: #587330;**: Màu nền của đường thẳng là màu xanh lá cây đậm (#587330).
* **margin-top: -20px;**: Đặt lề âm ở phía trên của phần tử, để đường thẳng có thể kéo dài ra ngoài một chút so với vị trí ban đầu.

**2. .vertical-line::before**

.vertical-line::before {

position: absolute;

bottom: 0;

right: -10px;

content: "";

padding: 10px;

background-color: #587330;

}

* **position: absolute;**: Pseudo-element ::before được định vị tuyệt đối.
* **bottom: 0;**: Đặt phần tử ở cạnh dưới của đường thẳng .vertical-line.
* **right: -10px;**: Dịch chuyển phần tử này sang bên phải thêm 10px so với phần tử cha.
* **content: "";**: Cung cấp nội dung cho pseudo-element. Trong trường hợp này, không có nội dung cụ thể, chỉ cần một phần tử trống để tạo hiệu ứng.
* **padding: 10px;**: Phần tử này sẽ có khoảng đệm 10px xung quanh, tạo ra một khu vực có chiều rộng lớn hơn.
* **background-color: #587330;**: Màu nền của phần tử này là màu xanh lá cây đậm giống với màu đường thẳng.

**3. .vertical-line::after**

.vertical-line::after {

position: absolute;

top: 0;

right: -10px;

content: "";

padding: 10px;

background-color: #587330;

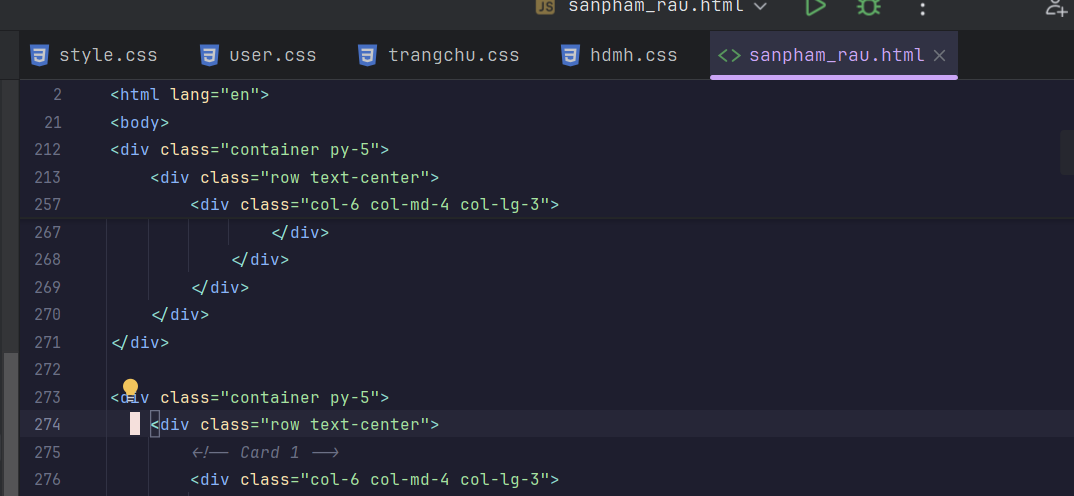
}

* **position: absolute;**: Pseudo-element ::after được định vị tuyệt đối.
* **top: 0;**: Đặt phần tử này ở cạnh trên của đường thẳng .vertical-line.
* **right: -10px;**: Dịch chuyển phần tử này sang bên phải thêm 10px so với phần tử cha.
* **content: "";**: Cung cấp nội dung cho pseudo-element. Trong trường hợp này, không có nội dung cụ thể, chỉ cần một phần tử trống để tạo hiệu ứng.
* **padding: 10px;**: Phần tử này sẽ có khoảng đệm 10px xung quanh, tạo ra một khu vực có chiều rộng lớn hơn.
* **background-color: #587330;**: Màu nền của phần tử này là màu xanh lá cây đậm giống với màu đường thẳng.

**Tóm tắt**

* .vertical-line tạo ra một đường thẳng dọc với chiều rộng rất nhỏ (1px), có màu xanh lá cây đậm, nằm ở giữa phần tử chứa và có chiều cao 100% của phần tử chứa.
* ::before và ::after tạo các mảng nhỏ ở đầu và cuối của đường thẳng, mỗi mảng có màu nền và khoảng đệm 10px. Các mảng này nằm ngoài đường thẳng và được dịch chuyển sang bên phải thêm 10px, tạo hiệu ứng "đầu" và "cuối" cho đường thẳng.

Hiệu ứng này có thể được sử dụng để tạo đường thẳng dọc có điểm bắt đầu và kết thúc rõ ràng, giúp tạo kiểu cho các phần tử giao diện người dùng như phân cách hoặc trang trí.



Đoạn mã bạn đưa ra là một phần của layout sử dụng Bootstrap để tạo cấu trúc lưới (grid) cho các thẻ card. Cụ thể:

<div class="row text-center">

<!-- Card 1 -->

<div class="col-6 col-md-4 col-lg-3">

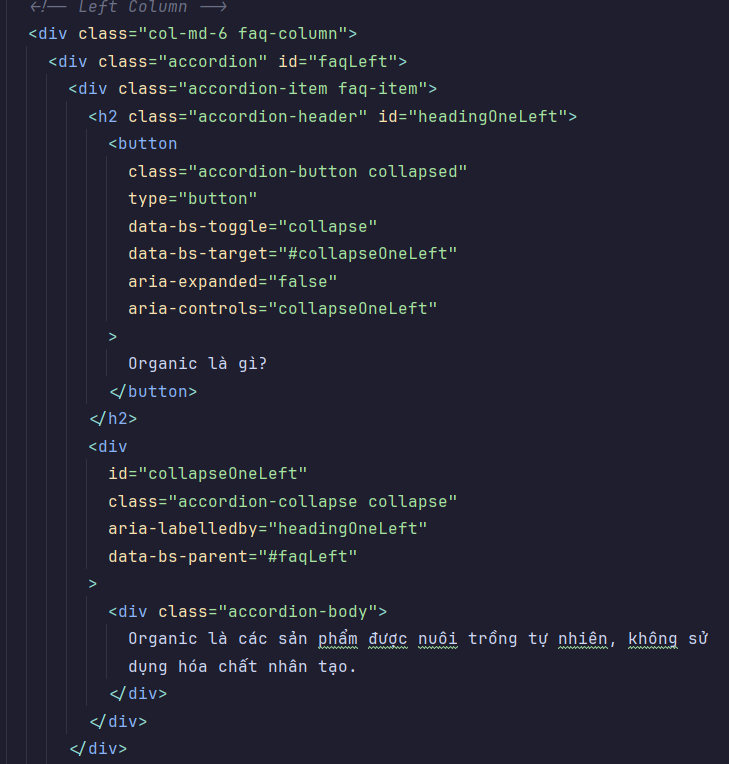
### Giải thích các lớp (classes) trong div:

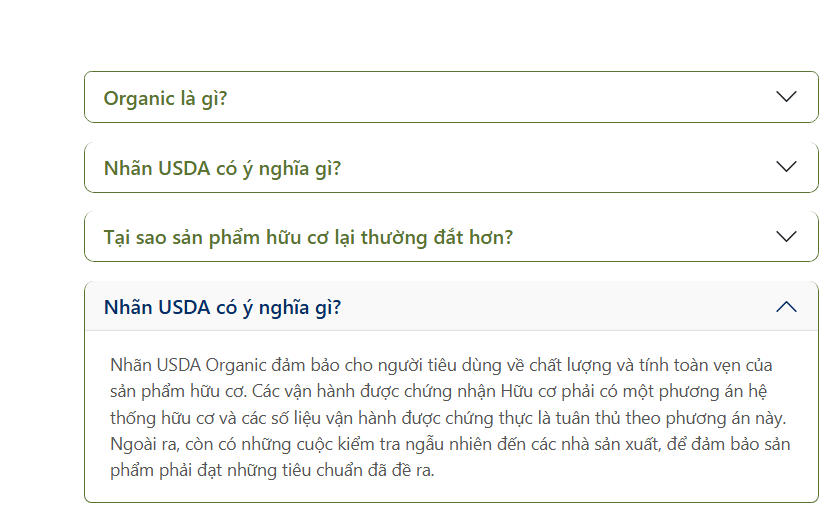
1. **row**:
   * Lớp row trong Bootstrap tạo một hàng trong hệ thống lưới (grid) chia 12 cột. Mọi cột (col-...) bên trong row sẽ tự động được căn chỉnh theo chiều ngang trong cùng một hàng này.
2. **text-center**:
   * Lớp text-center căn giữa nội dung trong div theo chiều ngang. Trong trường hợp này, nó sẽ căn giữa các thẻ card hoặc các phần tử khác bên trong div.
3. **col-6**:
   * Lớp col-6 chỉ định rằng phần tử này sẽ chiếm 6 cột trong hệ thống lưới 12 cột của Bootstrap khi màn hình có kích thước nhỏ (dưới 576px, thường là trên điện thoại di động). Tức là phần tử này chiếm 50% chiều rộng của hàng.
4. **col-md-4**:
   * Lớp col-md-4 chỉ định rằng khi màn hình có kích thước trung bình (từ 768px trở lên, thường là trên máy tính bảng), phần tử này sẽ chiếm 4 cột, tức là chiếm 33.33% chiều rộng của hàng.
5. **col-lg-3**:
   * Lớp col-lg-3 chỉ định rằng khi màn hình có kích thước lớn (từ 992px trở lên, thường là trên máy tính để bàn), phần tử này sẽ chiếm 3 cột, tức là chiếm 25% chiều rộng của hàng.

### Tóm lại:

* **Trên màn hình nhỏ (mobile)**: Mỗi phần tử div (thẻ card) sẽ chiếm 50% chiều rộng của hàng (col-6), tức là sẽ có 2 phần tử trên mỗi hàng.
* **Trên màn hình vừa (tablet)**: Mỗi phần tử div sẽ chiếm 33.33% chiều rộng của hàng (col-md-4), tức là sẽ có 3 phần tử trên mỗi hàng.
* **Trên màn hình lớn (desktop)**: Mỗi phần tử div sẽ chiếm 25% chiều rộng của hàng (col-lg-3), tức là sẽ có 4 phần tử trên mỗi hàng.

Bootstrap tự động điều chỉnh kích thước và cách hiển thị của các cột tùy thuộc vào kích thước màn hình để tạo ra một thiết kế phản hồi (responsive design).





Đoạn mã HTML trên đang sử dụng cấu trúc Accordion từ Bootstrap, được dùng để tạo các câu hỏi thường gặp (FAQ) với tính năng mở/đóng. Cụ thể, đoạn mã này tạo ra một cột chứa một câu hỏi, và khi người dùng nhấn vào câu hỏi, câu trả lời sẽ hiện ra.

### Giải thích chi tiết:

1. **Cấu trúc của Accordion:**
   * **<div class="col-md-6 faq-column">**: Đây là phần tử chứa các câu hỏi, với lớp col-md-6 nghĩa là cột này chiếm 6 phần trong tổng số 12 phần trên màn hình có độ rộng trung bình (ví dụ: trên máy tính bảng hoặc desktop).
2. **<div class="accordion" id="faqLeft">**: Tạo một nhóm Accordion với id là faqLeft. Cả nhóm Accordion này sẽ là nơi chứa các câu hỏi và câu trả lời.
3. **<div class="accordion-item faq-item">**: Đây là một phần tử trong Accordion, đại diện cho một câu hỏi và câu trả lời. Trong trường hợp này, câu hỏi là "Organic là gì?"
4. **<h2 class="accordion-header" id="headingOneLeft">**: Đây là phần tiêu đề (header) của câu hỏi. Trong đoạn mã này, tiêu đề là "Organic là gì?"
5. **<button class="accordion-button collapsed" type="button" data-bs-toggle="collapse" data-bs-target="#collapseOneLeft" aria-expanded="false" aria-controls="collapseOneLeft">**:
   * Lớp accordion-button collapsed cho biết nút này bắt đầu ở trạng thái "đóng" (collapsed).
   * data-bs-toggle="collapse": Kích hoạt tính năng đóng/mở.
   * data-bs-target="#collapseOneLeft": Chỉ định phần tử sẽ mở ra khi nút được nhấn (phần tử có id là collapseOneLeft).
   * aria-expanded="false": Đảm bảo rằng nút sẽ bắt đầu ở trạng thái không mở.
6. **<div id="collapseOneLeft" class="accordion-collapse collapse" aria-labelledby="headingOneLeft" data-bs-parent="#faqLeft">**:
   * Phần tử này chứa nội dung câu trả lời. Khi người dùng nhấn vào câu hỏi, phần tử này sẽ mở ra.
   * Lớp accordion-collapse collapse biểu thị rằng phần tử này sẽ được ẩn hoặc hiển thị dựa trên hành động của người dùng.
   * data-bs-parent="#faqLeft" đảm bảo rằng chỉ một câu trả lời có thể mở ra tại một thời điểm.
7. **<div class="accordion-body">**: Đây là phần thân của câu trả lời, trong trường hợp này là giải thích về "organic". Nội dung là "Organic là các sản phẩm được nuôi trồng tự nhiên, không sử dụng hóa chất nhân tạo."

### Tóm lại:

Đoạn mã này tạo ra một Accordion cho một câu hỏi trong phần FAQ. Khi người dùng nhấn vào câu hỏi "Organic là gì?", câu trả lời sẽ hiện ra ngay bên dưới câu hỏi đó.