

# Tài liệu: CSS Flexbox Layout

## Mục tiêu

- Hiểu rõ **CSS Flexbox**, cách hoạt động, và vai trò trong việc tạo bố cục 1D linh hoạt (hàng hoặc cột).
- Nắm vững các thuộc tính chính, cách sử dụng để xây dựng giao diện responsive và dễ bảo trì.
- Cung cấp ví dụ minh họa và bài tập thực hành để áp dụng Flexbox vào các dự án thực tế.

## Đối tượng

- Sinh viên hoặc lập trình viên đã biết CSS cơ bản (selector, box model, media queries).
- Người muốn tạo các bố cục tuyến tính (hàng/cột) hoặc giao diện responsive mà không cần framework.
- Lập trình viên tìm hiểu cách áp dụng Flexbox trong các dự án web hiện đại (2025).

## 1. Định nghĩa

**CSS Flexbox** (Flexible Box Layout) là một mô hình bố cục 1D trong CSS, được thiết kế để sắp xếp các phần tử trong một hàng hoặc cột một cách linh hoạt. Flexbox cho phép dễ dàng kiểm soát vị trí, kích thước, thứ tự, và căn chỉnh của các phần tử, đặc biệt phù hợp cho các giao diện responsive và các bố cục tuyến tính.

## 2. Cách hoạt động

### Cơ chế

- Flex Container:** Phần tử cha được đặt `display: flex` hoặc `display: inline-flex`, tạo ra một ngữ cảnh flex để quản lý các phần tử con (flex items).
- Flex Items:** Các phần tử con trực tiếp của container, được sắp xếp theo một trục chính (main axis) và trục phụ (cross axis).
- Trục chính (Main Axis):** Hướng chính của bố cục, xác định bởi `flex-direction` (row hoặc column).
- Trục phụ (Cross Axis):** Hướng vuông góc với trục chính, dùng để căn chỉnh dọc hoặc ngang.
- Cơ chế trình duyệt:** Trình duyệt tính toán kích thước và vị trí của các flex items dựa trên các thuộc tính như `flex-grow`, `flex-shrink`, `flex-basis`, và căn chỉnh chúng theo `justify-content` (trục chính) và `align-items` (trục phụ).

### Các thuộc tính chính

- Container:**
  - `display: flex | inline-flex`: Kích hoạt Flexbox. `flex` tạo block-level container, `inline-flex` tạo inline-level container.
  - `flex-direction: row | row-reverse | column | column-reverse`: Xác định hướng trục chính.
  - `flex-wrap: nowrap | wrap | wrap-reverse`: Quyết định các item có xuống dòng hay không.
  - `justify-content: flex-start | flex-end | center | space-between | space-around | space-evenly`: Căn chỉnh items theo trục chính.

- `align-items: stretch | flex-start | flex-end | center | baseline;` Căn chỉnh items theo trục phụ.
- `align-content: stretch | flex-start | flex-end | center | space-between | space-around | space-evenly;` Căn chỉnh các dòng (khi `flex-wrap` bật).
- `gap` (hoặc `row-gap`, `column-gap`): Khoảng cách giữa các flex items.
- **Flex Items:**
  - `flex-grow`: Xác định mức độ item mở rộng để lấp đầy không gian (mặc định: 0).
  - `flex-shrink`: Xác định mức độ item co lại khi thiếu không gian (mặc định: 1).
  - `flex-basis`: Xác định kích thước ban đầu của item trước khi mở rộng/co lại (mặc định: `auto`).
  - `flex`: Shorthand cho `flex-grow`, `flex-shrink`, `flex-basis` (ví dụ: `flex: 1 1 auto`).
  - `order`: Thay đổi thứ tự hiển thị của item (mặc định: 0).
  - `align-self`: Ghi đè `align-items` cho item cụ thể (ví dụ: `center`, `flex-start`).

### Lưu ý

- **Responsive:** Kết hợp với media queries hoặc `flex-wrap` để tạo bố cục thích ứng với kích thước màn hình.
- **Hiệu suất:** Flexbox được tối ưu hóa bởi trình duyệt, nhẹ hơn so với các kỹ thuật cũ như float.
- **Tương thích:** Hỗ trợ tốt trên các trình duyệt hiện đại (Chrome, Firefox, Edge, Safari từ 2013 trở đi). Kiểm tra trên **CanIUse.com** nếu cần hỗ trợ trình duyệt cũ.
- **So sánh với Grid:** Flexbox phù hợp cho bố cục 1D (hàng hoặc cột), trong khi CSS Grid lý tưởng cho bố cục 2D.

## 3. Cú pháp

### Container

```
.flex-container {
  display: flex;
  flex-direction: row; /* Hoặc column */
  flex-wrap: wrap; /* Cho phép xuống dòng */
  justify-content: space-between; /* Căn chỉnh trục chính */
  align-items: center; /* Căn chỉnh trục phụ */
  gap: 20px; /* Khoảng cách giữa các item */
}
```

### Flex Items

```
.flex-item {
  flex: 1 1 200px; /* grow: 1, shrink: 1, basis: 200px */
  order: 2; /* Thay đổi thứ tự hiển thị */
  align-self: flex-end; /* Căn chỉnh riêng lẻ */
}
```

## 4. Ví dụ minh họa

### Ví dụ cơ bản: Hàng flex với căn chỉnh đều

Tạo một hàng flex với 3 phần tử, căn chỉnh đều nhau với khoảng cách.

```
<!DOCTYPE html>
<html lang="vi">
<head>
  <meta charset="UTF-8">
  <title>Flexbox Basic Example</title>
  <style>
    .flex-container {
      display: flex;
      justify-content: space-between;
      align-items: center;
      gap: 15px;
      padding: 20px;
      background-color: #f0f0f0;
    }
    .flex-item {
      flex: 1;
      background-color: #b2ebf2;
      padding: 10px;
      text-align: center;
      border-radius: 5px;
    }
  </style>
</head>
<body>
  <div class="flex-container">
    <div class="flex-item">Item 1</div>
    <div class="flex-item">Item 2</div>
    <div class="flex-item">Item 3</div>
  </div>
</body>
</html>
```

#### Giải thích:

- `display: flex` kích hoạt Flexbox cho container.
- `justify-content: space-between` căn các item đều nhau, với khoảng cách tối đa.
- `flex: 1` khiến các item mở rộng để lấp đầy không gian, chia đều chiều rộng.

#### Ví dụ nâng cao 1: Navigation bar responsive

Tạo một menu điều hướng responsive, chuyển từ hàng sang cột trên màn hình nhỏ.

```
<!DOCTYPE html>
<html lang="vi">
<head>
  <meta charset="UTF-8">
  <title>Flexbox Advanced Example 1</title>
  <style>
    .nav {
      display: flex;
      justify-content: space-around;
```

```

    align-items: center;
    background-color: #333;
    padding: 10px;
    gap: 10px;
}
.nav__item {
    color: white;
    text-decoration: none;
    padding: 10px 20px;
    border-radius: 5px;
}
.nav__item:hover {
    background-color: #555;
}
.nav__item--active {
    background-color: #007bff;
    font-weight: bold;
}
@media (max-width: 600px) {
    .nav {
        flex-direction: column;
        align-items: stretch;
    }
    .nav__item {
        text-align: center;
    }
}
</style>
</head>
<body>
    <nav class="nav">
        <a href="#" class="nav__item">Trang chủ</a>
        <a href="#" class="nav__item nav__item--active">Giới thiệu</a>
        <a href="#" class="nav__item">Liên hệ</a>
    </nav>
</body>
</html>

```

### Giải thích:

- `flex-direction: row` (mặc định) sắp xếp các item theo hàng ngang.
- `justify-content: space-around` tạo khoảng cách đều giữa các item.
- Media query chuyển `flex-direction: column` trên màn hình nhỏ, khiến menu xếp dọc.

### Ví dụ nâng cao 2: Card layout với tỷ lệ linh hoạt

Tạo một hàng card với các item có tỷ lệ co giãn khác nhau, sử dụng `flex-grow` và `order`.

```

<!DOCTYPE html>
<html lang="vi">
<head>
    <meta charset="UTF-8">
    <title>Flexbox Advanced Example 2</title>

```

```

<style>
  .card-container {
    display: flex;
    flex-wrap: wrap;
    gap: 15px;
    padding: 20px;
    background-color: #f0f0f0;
  }
  .card {
    background-color: #e0f7fa;
    padding: 15px;
    border-radius: 5px;
    text-align: center;
  }
  .card--large {
    flex: 2 1 300px; /* Mở rộng gấp đôi, basis 300px */
    order: 1; /* Xuất hiện thứ hai */
  }
  .card--small {
    flex: 1 1 150px; /* Mở rộng bình thường, basis 150px */
    order: 2;
  }
  .card--highlight {
    flex: 1 1 150px;
    order: 0; /* Xuất hiện đầu tiên */
    background-color: #ffccbc;
  }
</style>
</head>
<body>
  <div class="card-container">
    <div class="card card--small">Card nhỏ 1</div>
    <div class="card card--large">Card lớn</div>
    <div class="card card--small">Card nhỏ 2</div>
    <div class="card card--highlight">Card nổi bật</div>
  </div>
</body>
</html>

```

### Giải thích:

- `flex-wrap: wrap` cho phép các card xuống dòng nếu không đủ không gian.
- `flex: 2 1 300px` khiến card lớn mở rộng gấp đôi so với các card nhỏ (`flex: 1 1 150px`).
- `order` thay đổi thứ tự hiển thị, đưa card nổi bật lên đầu.

---

## 5. Ứng dụng thực tế

- **Navigation bar:** Tạo menu ngang hoặc dọc, responsive trên các thiết bị khác nhau (ví dụ: menu website hoặc mobile).
- **Card layout:** Sắp xếp danh sách sản phẩm, bài viết, hoặc hình ảnh trong các trang e-commerce, blog, hoặc portfolio.

- **Form UI:** Căn chỉnh các trường input và nút trong form, đảm bảo bố cục gọn gàng và linh hoạt.
  - **Responsive layouts:** Kết hợp với media queries để tạo giao diện thích ứng, ví dụ chuyển từ hàng ngang sang cột dọc trên mobile.
  - **Dashboard components:** Sắp xếp các widget nhỏ trong dashboard, như biểu đồ hoặc bảng dữ liệu, với khoảng cách và căn chỉnh chính xác.
- 

## 6. Ưu điểm và nhược điểm

### Ưu điểm

- **Linh hoạt:** Dễ dàng kiểm soát căn chỉnh, thứ tự, và kích thước của các item.
- **Responsive:** Hỗ trợ bố cục responsive với `flex-wrap` và media queries.
- **Dễ sử dụng:** Cú pháp đơn giản, dễ học hơn CSS Grid cho các bố cục tuyến tính.
- **Hiệu suất cao:** Được tối ưu hóa bởi trình duyệt, thay thế các kỹ thuật cũ như float hoặc inline-block.
- **Tương thích tốt:** Hỗ trợ rộng rãi trên các trình duyệt hiện đại (từ 2013).

### Nhược điểm

- **Hạn chế cho bố cục 2D:** Không lý tưởng cho các lưới phức tạp (CSS Grid phù hợp hơn).
  - **Khúc học tập:** Yêu cầu hiểu cách các thuộc tính như `flex-grow`, `flex-shrink` tương tác.
  - **Hành vi không nhất quán:** Một số trình duyệt cũ (như IE10) có hỗ trợ hạn chế hoặc lỗi.
- 

## 7. Bài tập thực hành

1. **Hàng flex cơ bản:**
    - Tạo một hàng flex với 4 phần tử, căn giữa và có khoảng cách 10px bằng gap.
    - Thêm media query để chuyển sang cột trên màn hình nhỏ (<600px).
  2. **Menu điều hướng:**
    - Tạo một menu ngang với 5 mục, sử dụng `justify-content: space-evenly` và hiệu ứng hover.
    - Sử dụng `flex-wrap: wrap` để đảm bảo menu xuống dòng nếu thiếu không gian.
  3. **Card layout linh hoạt:**
    - Tạo một bố cục với 3 card, trong đó 1 card có `flex-grow: 2` và các card còn lại có `flex-grow: 1`.
    - Sử dụng `order` để thay đổi thứ tự hiển thị của một card.
  4. **Form responsive:**
    - Tạo một form với 2 trường input và 1 nút submit, sử dụng Flexbox để căn chỉnh ngang trên desktop và dọc trên mobile.
    - Sử dụng `align-self` để căn chỉnh nút submit khác với các input.
- 

## 8. Tài nguyên bổ sung

- **MDN Web Docs:** Tài liệu về CSS Flexbox ([developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_Flexible\\_Box\\_Layout](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout)).
- **CSS Tricks:** Hướng dẫn chi tiết và ví dụ về Flexbox ([css-tricks.com/snippets/css/a-guide-to-flexbox](https://css-tricks.com/snippets/css/a-guide-to-flexbox)).
- **Flexbox Froggy:** Trò chơi học Flexbox ([flexboxfroggy.com](https://flexboxfroggy.com)).

- **CanIUse:** Kiểm tra tương thích trình duyệt ([caniuse.com/flexbox](https://caniuse.com/flexbox)).
  - **CodePen:** Thực hành và chia sẻ mã ([codepen.io](https://codepen.io)).
- 

## 9. Lưu ý

- **Tương thích trình duyệt:** Flexbox được hỗ trợ đầy đủ trên Chrome, Firefox, Edge, Safari từ 2013. Với IE10/11, cần thêm tiền tố `-webkit-` hoặc kiểm tra trên CanIUse.
- **Kết hợp với Grid:** Sử dụng Flexbox cho bố cục tuyến tính (menu, form) và CSS Grid cho bố cục 2D (toàn trang, dashboard).
- **Debug:** Sử dụng DevTools (Chrome/Firefox) để kiểm tra bố cục Flexbox, bật chế độ "Flexbox Overlay" để thấy trực chính và phụ.
- **Hiệu suất:** Tránh lạm dụng `flex-wrap` hoặc các thuộc tính phức tạp trong container có nhiều item để đảm bảo hiệu suất.