

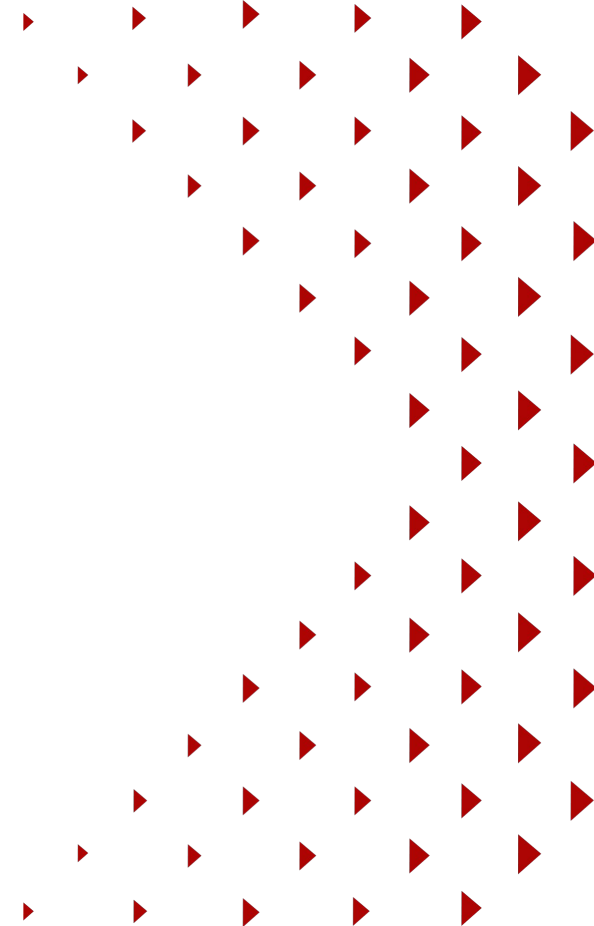


SESSION 04:

CSS Layout structure - RWD

Module 1: Web Application UI Design

Version: 1.0



- 1. Giới thiệu CSS Layout**
- 2. Dàn trang với floats, flexbox và grid**
- 3. Responsive Web Design**

GIỚI THIỆU CSS LAYOUT - 1

CSS Layout

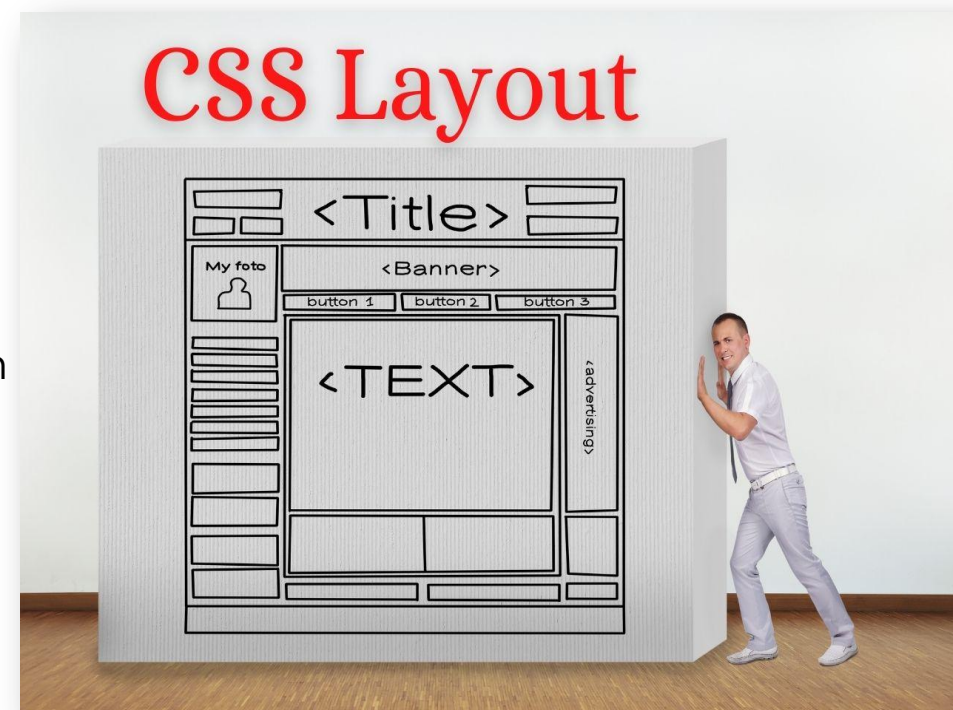
- Khung xác định cấu trúc của trang
- Cách bố trí các thành phần chính trên một trang web

Tại sao phải thiết kế CSS Layout

- Nâng cao giá trị thẩm mỹ
- Tạo sự liên kết giữa các thành phần trong thiết kế
- Gia tăng sự thu hút
- Hình dung và đánh giá được sản phẩm trước khi hình thành
- SEO

Công cụ và kỹ thuật thiết kế layout

- Flexbox
- Grid
- Floats
- Position
- Responsive



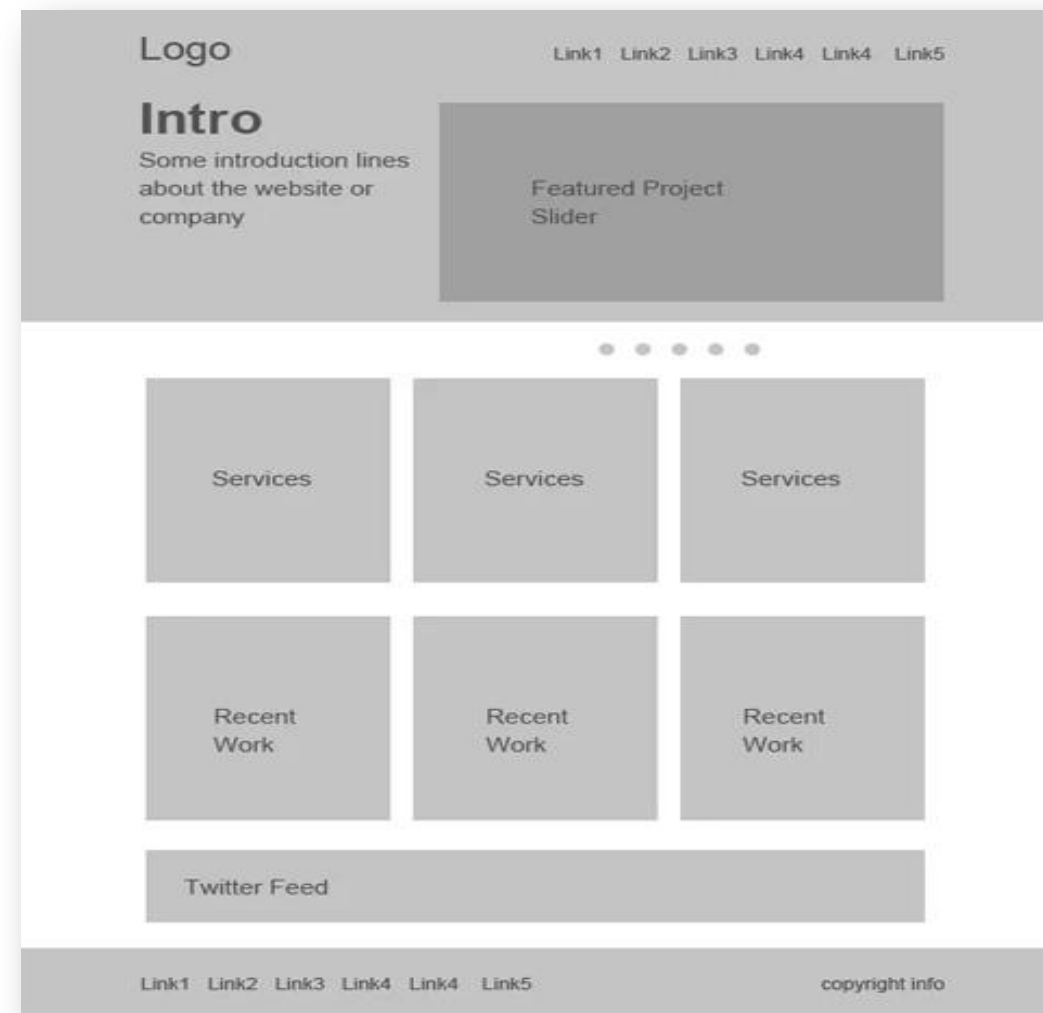
GIỚI THIỆU CSS LAYOUT - 2

Nguyên tắc thiết kế layout

- Grid View
- Sự cân bằng
- Tỷ lệ và điểm nhấn
- Quy tắc số lẻ
- Quy tắc 1/3

Công cụ dùng thiết kế layout

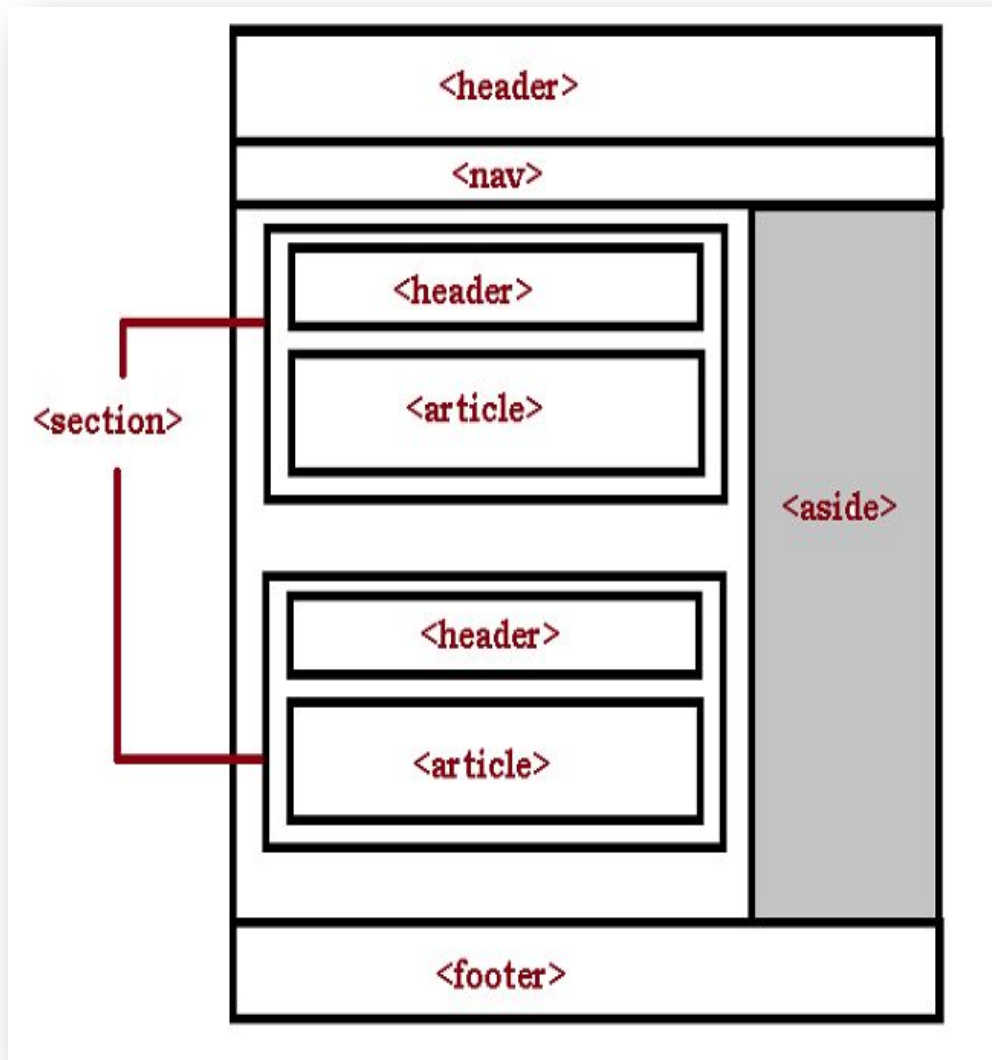
- Figma
- Framer X
- Adobe Photoshop
- MockFlow



GIỚI THIỆU CSS LAYOUT - 3

HTML Layout

- **<header>** đại diện cho nội dung giới thiệu. Bao gồm tiêu đề, logo, biểu mẫu tìm kiếm, tên tác giả, ...
- **<nav>** đại diện cho thành phần các liên kết điều hướng. Các ví dụ phổ biến về phần điều hướng là menu, mục lục và chỉ mục.
- **<main>** đại diện cho nội dung chính của <body>. Khu vực nội dung chính bao gồm nội dung liên quan trực tiếp đến hoặc mở rộng
- **<section>** đại diện cho một phần độc lập của website, không có phần tử ngữ nghĩa cụ thể hơn để biểu thị nó. Các phần phải luôn có tiêu đề, với rất ít ngoại lệ.
- **<aside>** thanh bên chứa các phần như menu phụ, các bài viết liên quan, quảng cáo...
- **<article>** đại diện cho một thành phần độc lập trong trang web nhằm mục đích phân phối hoặc tái sử dụng độc lập. Các ví dụ bao gồm: bài đăng trên diễn đàn, tạp chí hoặc bài báo hoặc bài viết trên blog, thẻ sản phẩm, nhận xét do người dùng gửi,...
- **<footer>** đại diện cho một chân trang cho nội dung gần nhất của nó hoặc phần tử gốc phân chia, thường chứa thông tin về tác giả của phần, liên kết đến các tài liệu liên quan,...



FLOATS, FLEXBOX VÀ GRID - 1

Floats

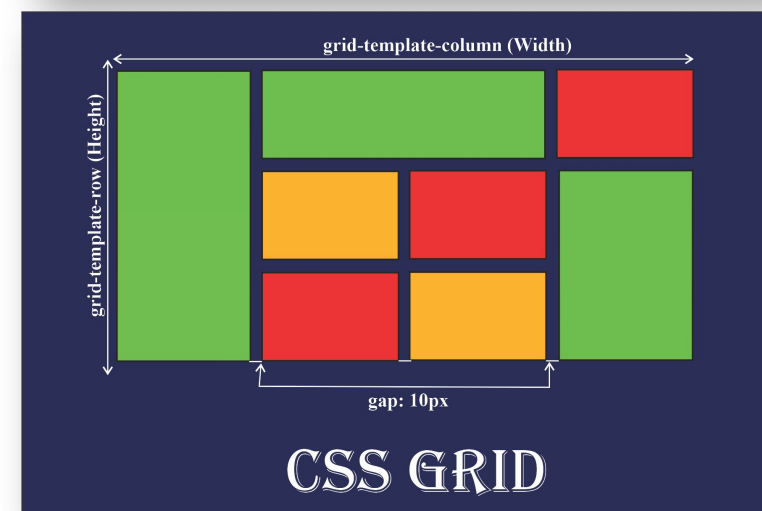
- Thuộc tính của CSS được sử dụng để chuyển một phần tử sang góc trái hoặc góc bên phải của không gian bao quanh nó
- Phải thiết lập thuộc tính width và height cụ thể cho từng phần tử, thiết lập dạng hiển thị inline hay block

Flexbox

- Kiểu dàn trang (layout mode) tự cân đối kích thước của các phần tử bên trong để hiển thị trên mọi thiết bị, không cần thiết lập kích thước của phần tử, không cần float, chỉ cần thiết lập hiển thị chiều ngang hay chiều dọc, các phần tử bên trong có thể hiển thị theo ý muốn
- Nên sử dụng để thiết lập bố cục trong phạm vi nhỏ (khung thành phần trong một trang)
- Sắp xếp theo một chiều (row hoặc column)
- Tài liệu tham khảo: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

Grid

- Hệ thống bố cục dựa trên lưới hai chiều
- Chia các phần tử thành hàng và cột nhằm tạo ra bố cục nhất quán và liên mạch cho các ứng dụng web
- Tài liệu tham khảo: <https://css-tricks.com/snippets/css/complete-guide-grid/>

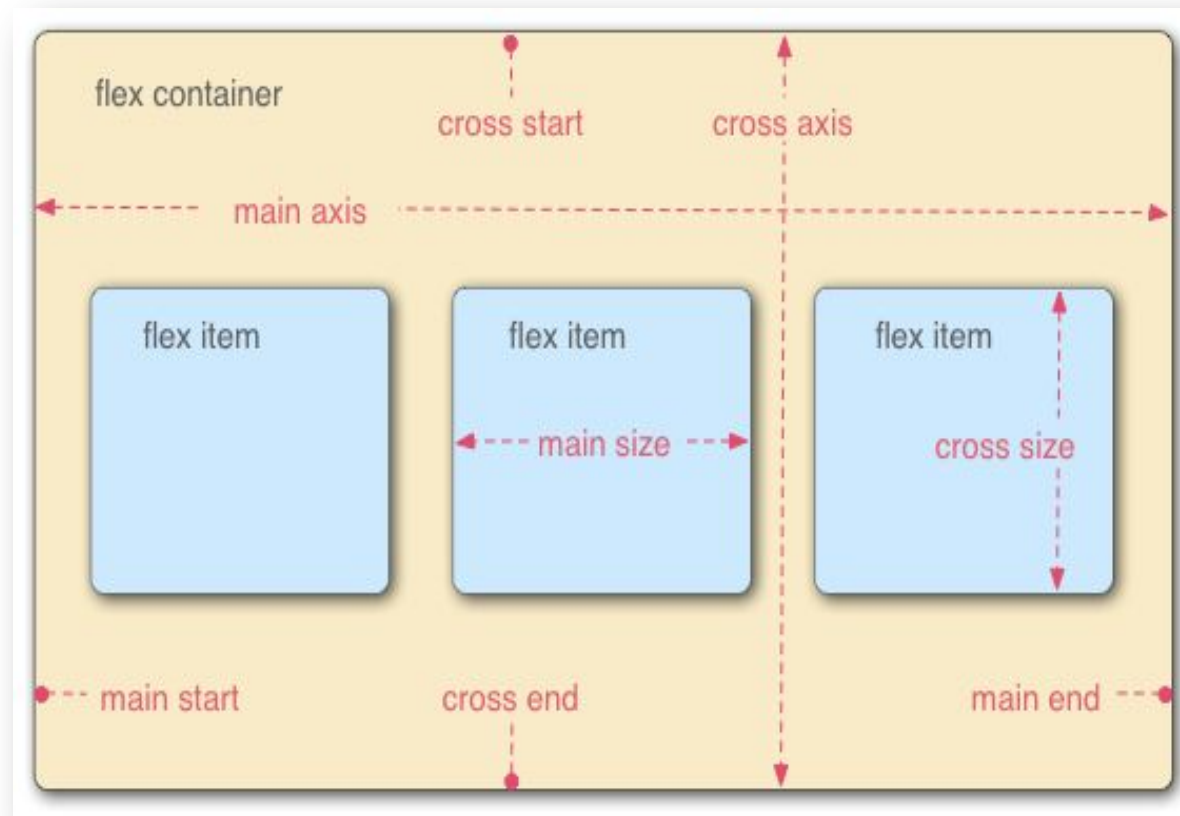


FLOATS, FLEXBOX VÀ GRID – 2

Các thành phần Flexbox

Flex Container: thành phần bao quanh các phần tử

- *flex-direction*: hướng sắp xếp của trục chính (*main axis*)
- *flex-wrap*: xác định item có xuống dòng khi tràn hay không
- *justify-content*: căn chỉnh các khoảng trống của item theo trục chính (*main axis*)
- *align-items*: sắp xếp căn chỉnh vị trí đặt theo trục chéo (*cross axis*)
- *align-content*: dùng để căn chỉnh các khoảng trống item theo trục chéo (*cross axis*)
- *gap*: thiết lập khoảng trống của các item



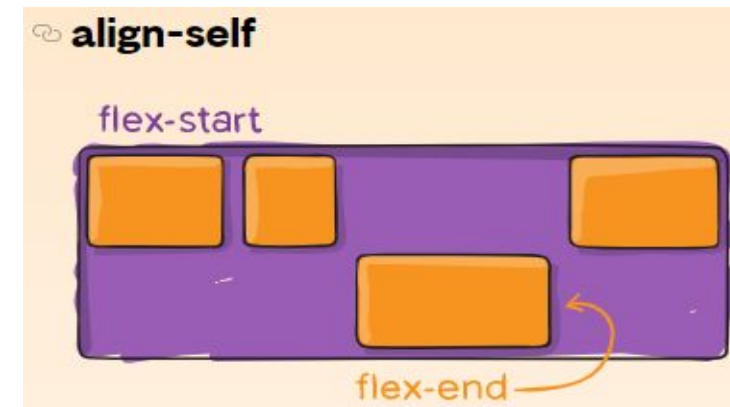
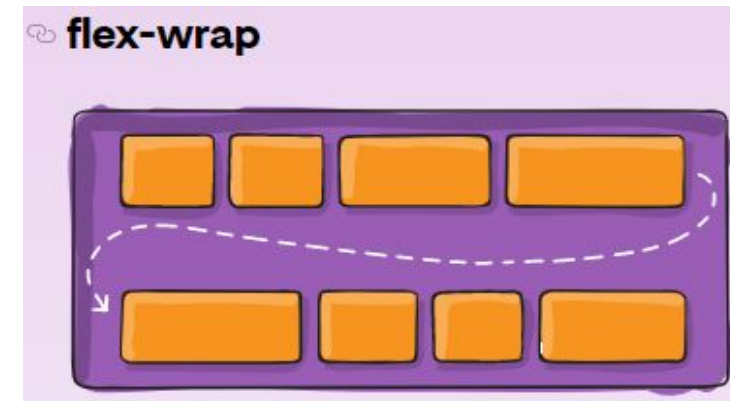
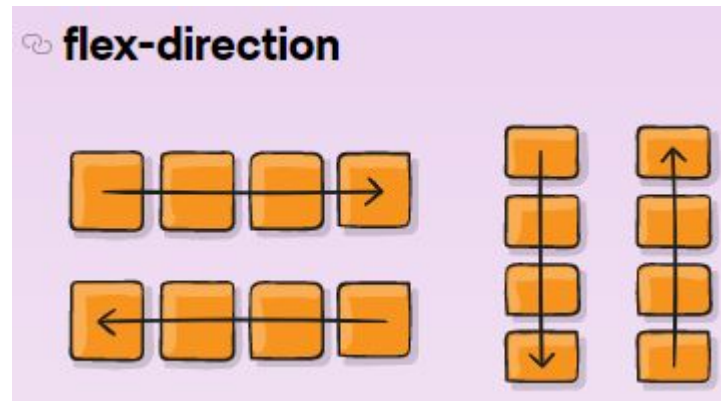
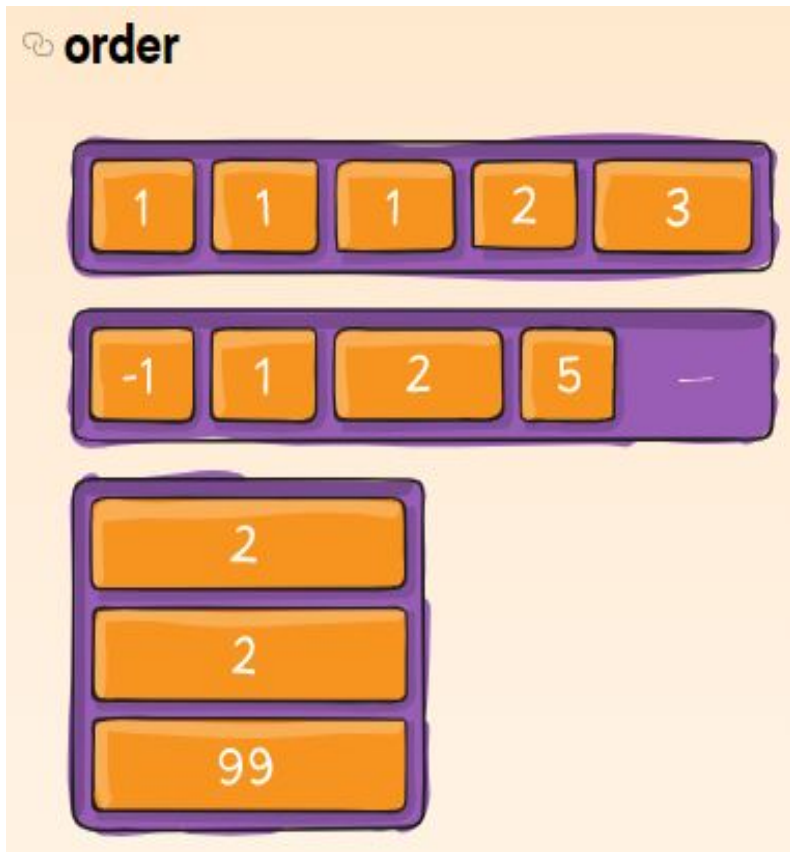
FLOATS, FLEXBOX VÀ GRID – 3

Các thành phần Flexbox

- **Flex Item:** phần tử con của container
 - *flex-grow*: xác định tỷ lệ chiều dài item so với toàn bộ các item của flex (khi tổng kích thước của item < kích thước container)
 - *flex-basis*: xác định chiều dài ban đầu của item
 - *flex-shrink*: xác định tỷ lệ co lại của item so với các item khác (dùng khi tổng chiều dài của item > chiều dài của container)
 - *align-self*: căn chỉnh vị trí theo trục chéo của một item
 - *Order*: xác định thứ tự sắp xếp của item
- **Main start, main end, cross start, cross end**
 - **Main start, cross start**: điểm bắt đầu của container
 - **Main end, cross end**: điểm kết thúc của container
- **Main axis**: trục chính để điều khiển hướng mà các item sẽ hiển thị
- **Main size**: kích thước của mỗi item theo trục main axis
- **Cross size**: kích thước của mỗi item theo trục cross axis

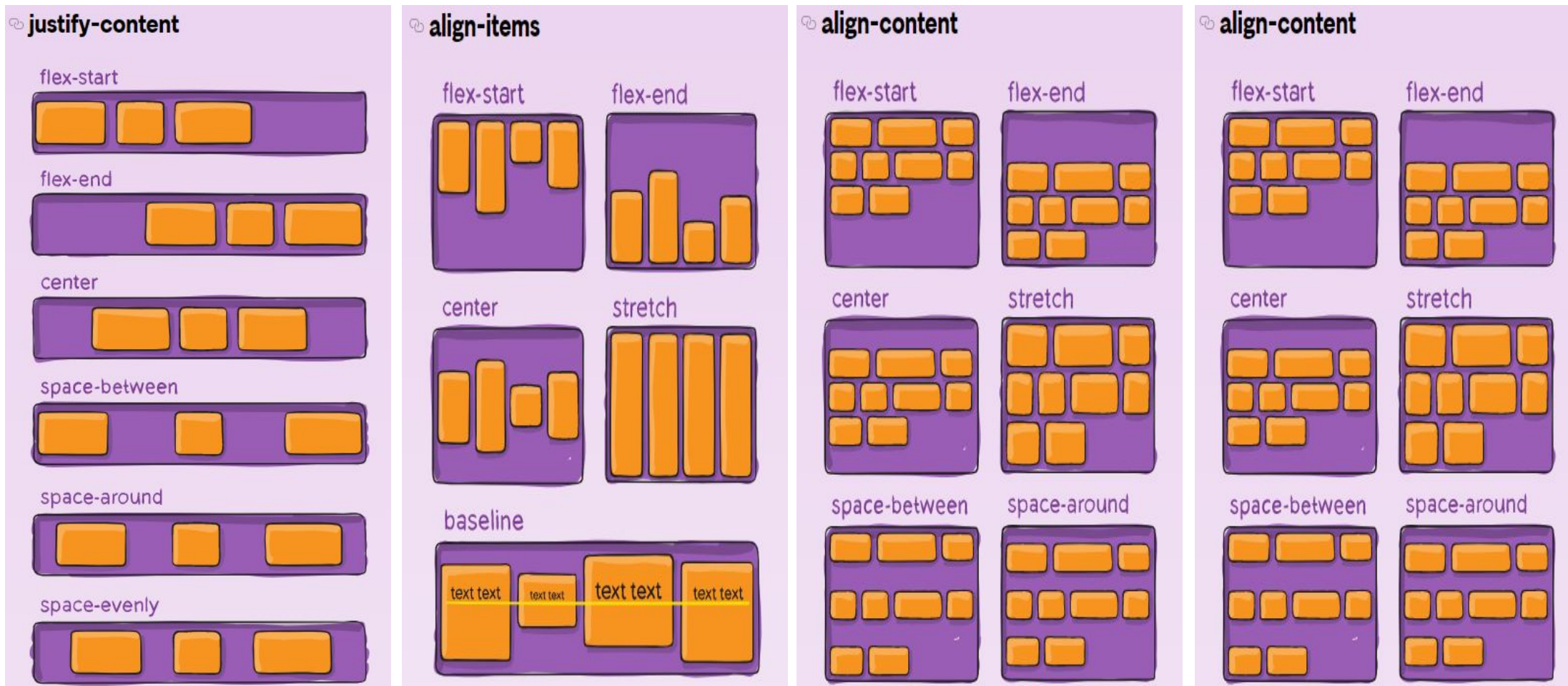
FLOATS, FLEXBOX VÀ GRID – 4

Giá trị thuộc tính Flex



FLOATS, FLEXBOX VÀ GRID – 5

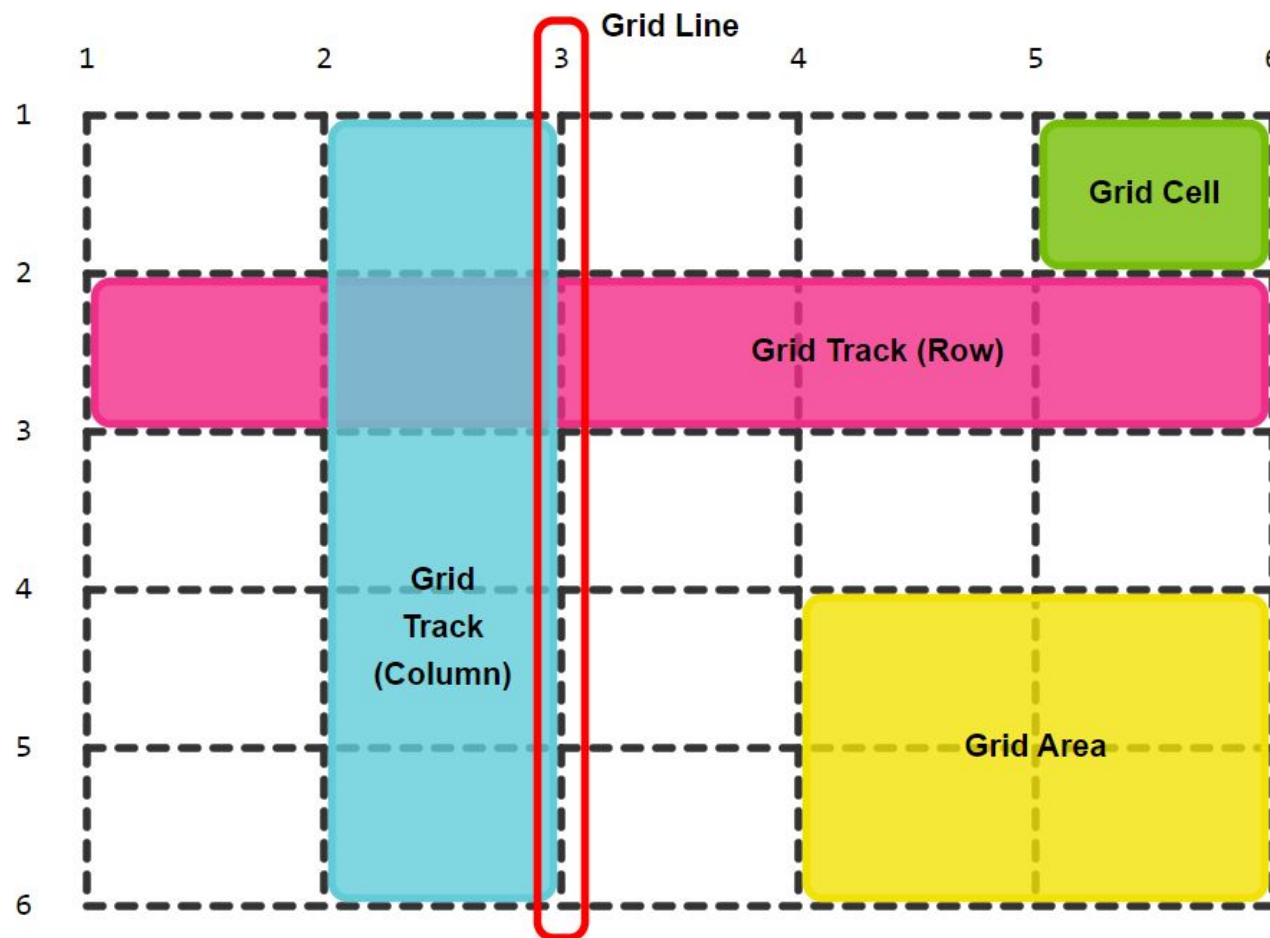
Giá trị thuộc tính Flex



FLOATS, FLEXBOX VÀ GRID – 6

Các thành phần grid

- **Grid container:** hộp chứa lưới, khối xây dựng chính của Grid
- **Grid line:** dòng kẻ ngang, dọc phân chia layout thành các hàng, cột
- **Grid track:** khoảng cách giữa 2 grid line (row, column)
- **Grid cell:** vùng diện tích giữa 2 grid line dọc và ngang liên tiếp (unit trên lưới)
- **Grid area:** vùng diện tích bao quanh bởi 4 đường grid line, có thể chứa 1 hoặc nhiều grid cell
- **Grid gaps:** khoảng cách giữa các row hoặc column



FLOATS, FLEXBOX VÀ GRID – 7

Các thuộc tính container

Thuộc tính	Giá trị
display	grid inline-grid
grid-template-columns grid-template-rows	<track-size> <line-name>
grid-template-areas	<grid-area-name> . none
grid-template	none <grid-template-rows> / <grid-template-columns>
column-gap row-gap grid-column-gap grid-row-gap	<line-size>
gap grid-gap	<grid-row-gap> <grid-column-gap>
justify-items	start end center stretch
align-items	start end center stretch baseline
place-items	<align-items> / <justify-items>
justify-content align-content	start end center stretch space-around space-between space-evenly
place-content	<align-content> / <justify-content>
grid-auto-columns grid-auto-rows	<track-size>
grid-auto-flow	row column dense
grid	none <grid-template-row> <grid-template-column> <grid-template-areas> <grid-auto-rows> <grid-auto-columns> <grid-auto-flow>

FLOATS, FLEXBOX VÀ GRID – 8

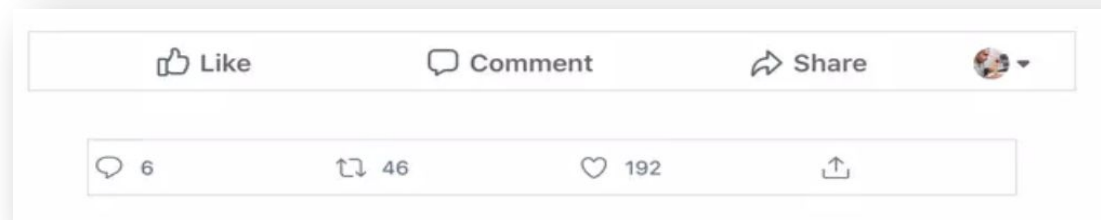
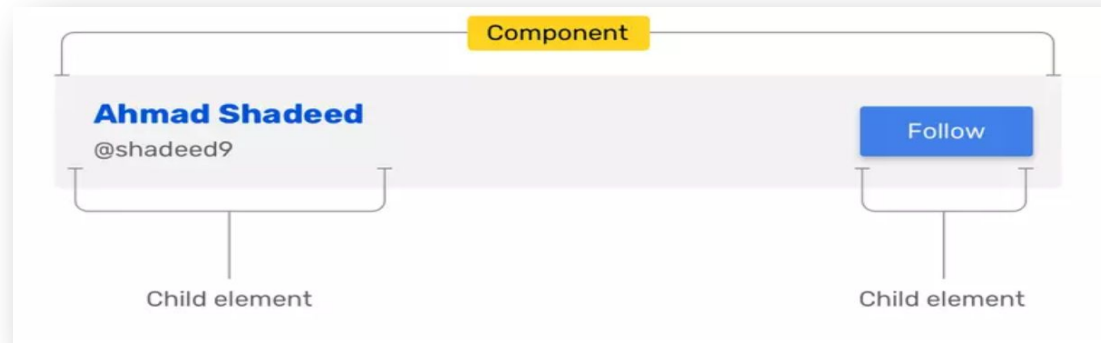
Các thuộc tính grid items

Thuộc tính	Giá trị
grid-column-start grid-column-end grid-row-start grid-row-end	<line> span <number> span <name> auto
grid-column grid-row	<start-line> / <end-line>
grid-area	<name> <row-start> / <column-start> / <row-end> / <column-end>
justify-seft	start end center stretch
align-seft	start end center stretch
place-seft	auto <align-seft> / <justify-seft>

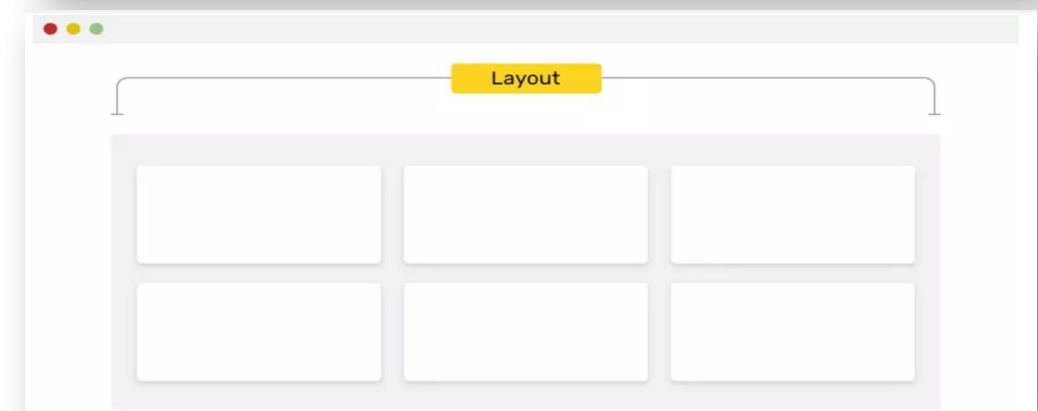
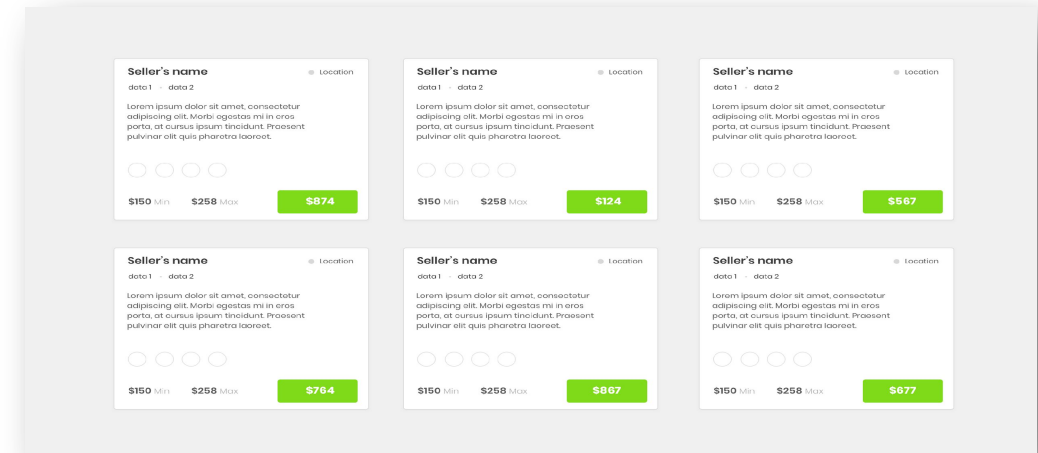
Đơn vị và chức năng đặc biệt

- Đơn vị fr
- Sizing keywords: min-content, max-content, auto, fit-content
- Sizing functions: minmax
- Repeat function, auto-fit, auto-fill

FLOATS, FLEXBOX VÀ GRID – 9



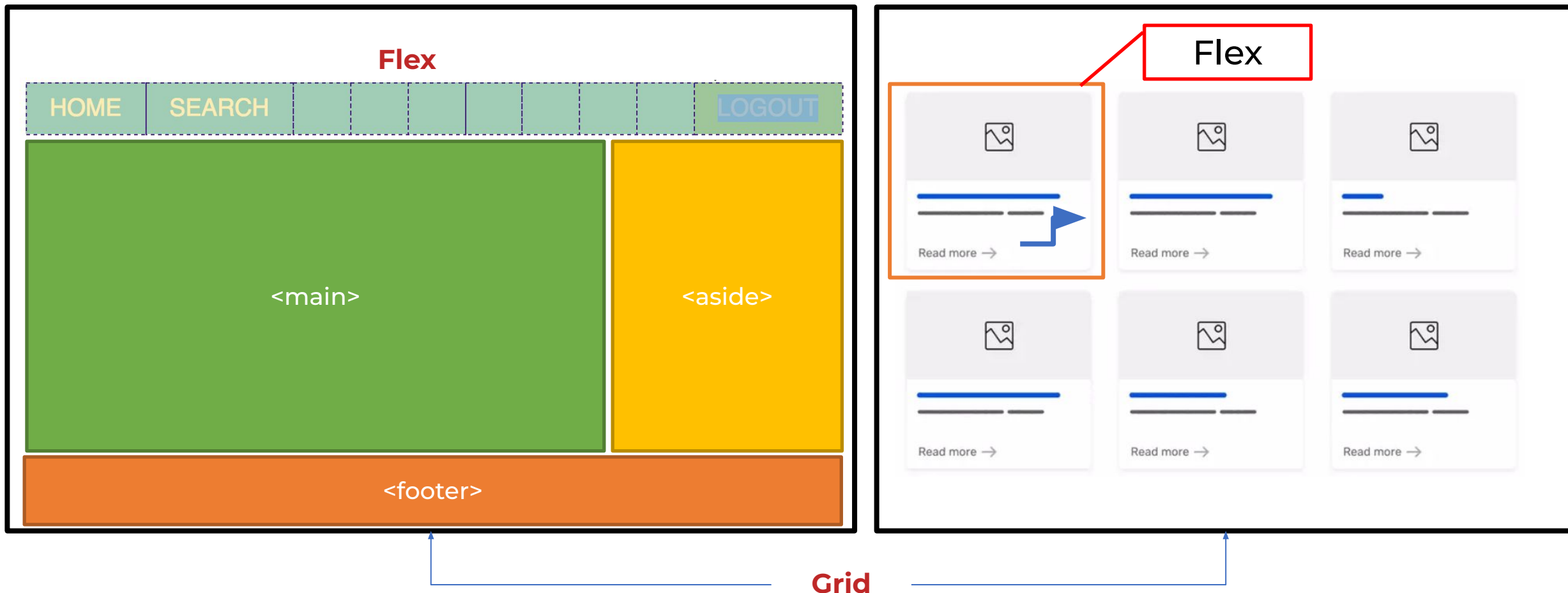
Trường hợp dùng Flexbox



Trường hợp dùng Grid

FLOATS, FLEXBOX VÀ GRID – 10

Grid kết hợp với flex



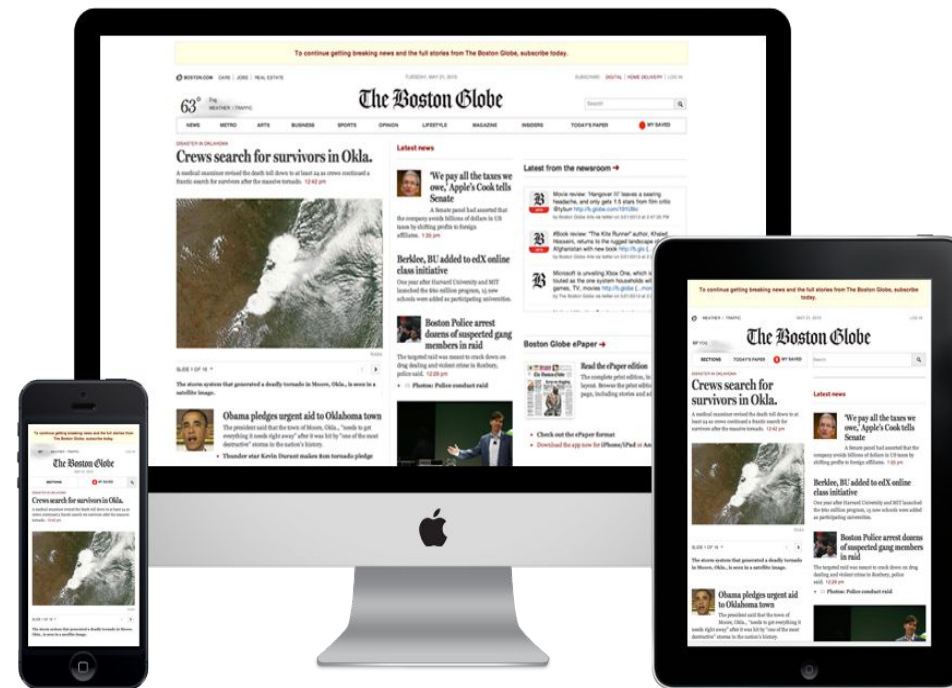
RESPONSIVE WEB DESIGN – RWD - 1

Khái niệm

- RWD (Responsive Web Design) là một phương pháp thiết kế web để làm cho các trang web hiển thị tốt trên mọi kích thước và độ phân giải màn hình trong khi vẫn đảm bảo khả năng sử dụng tốt. Đó là cách để thiết kế cho một trang web đa thiết bị (laptop, mobile, tablets,...)
- Một trang web được gọi là thiết kế responsive khi bạn sử dụng CSS và HTML để thay đổi kích cỡ, ẩn, thu nhỏ, phóng to, hoặc di chuyển nội dung để làm cho nó có giao diện đẹp trên bất kỳ màn hình nào

Lợi ích

- Việc chia sẻ dữ liệu của bạn dễ dàng hơn rất nhiều nếu bạn chỉ có một URL duy nhất.
- Nó cũng dễ dàng hơn cho các công cụ tìm kiếm như Google để lập chỉ mục trang hoặc đánh dấu để biết rằng bạn đã từng xem trang đó và để xếp hạng trang đó.
- Ít tập tin hơn có nghĩa là ít bảo trì hơn.
- Nếu bạn có một nơi duy nhất mà bạn đang đi đến một URL duy nhất, sau đó bạn sẽ ít phải chuyển hướng hơn. Chuyển hướng (redirection) là khi bạn truy cập một trang web và bạn thấy rằng URL đang được thay đổi khi server cố gắng tìm ra loại thiết bị bạn đang sử dụng và loại trang hoặc URL bạn nên được chuyển hướng đến.



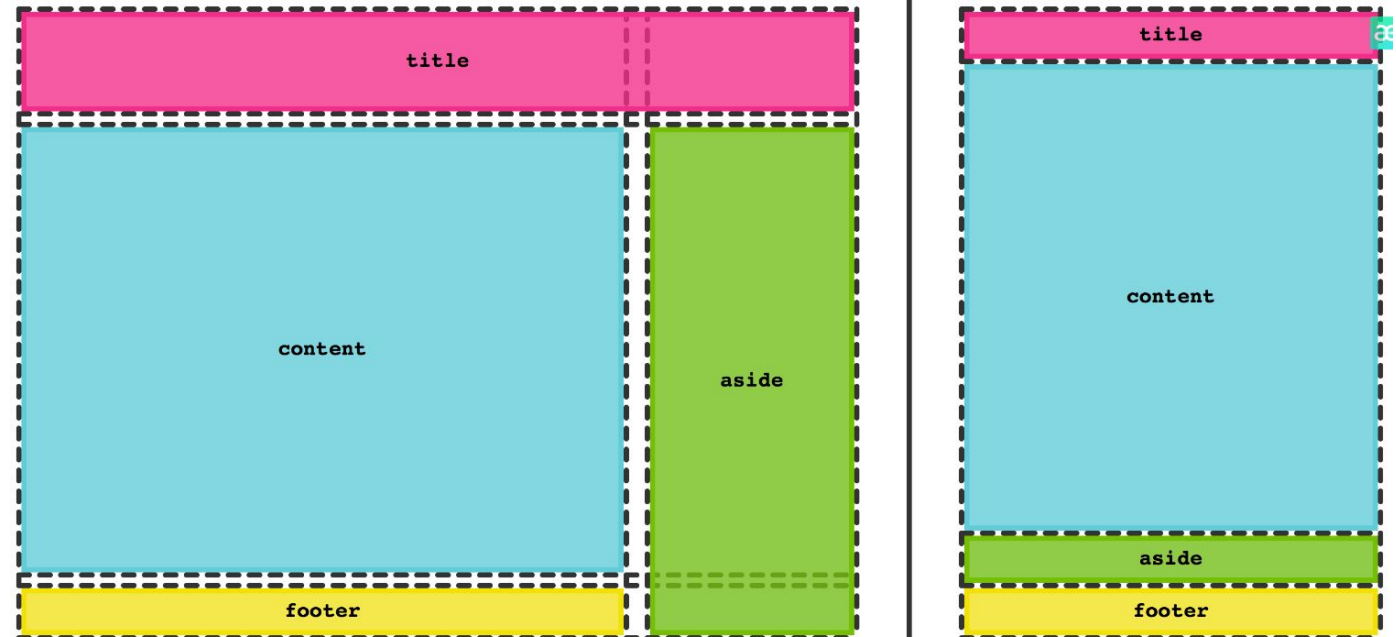
RESPONSIVE WEB DESIGN – RWD - 2

Các kỹ thuật dùng RWD

- RWD Viewport
- RWD Grid View
- RWD Media queries
- RWD Images, Videos
- RWD Framework, Templates

Viewport:

- Thay đổi theo thiết bị và sẽ nhỏ hơn trên diện thoại di động so với trên màn hình máy tính.
- Cài đặt

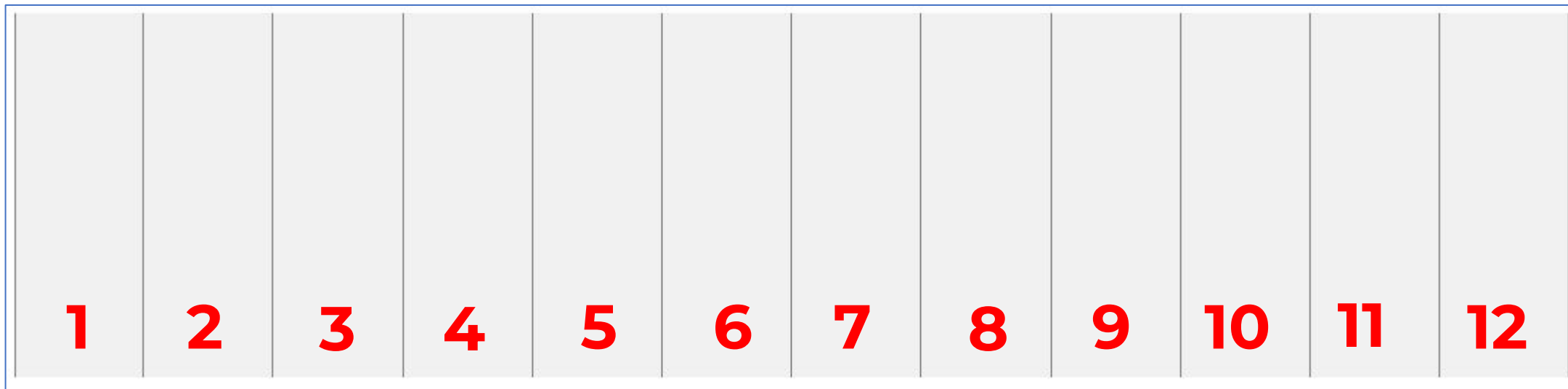


```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

RESPONSIVE WEB DESIGN – RWD - 3

Grid View

- Trang web được phân bố nội dung theo dạng lưới
- Nhờ sử dụng Grid View mà việc thiết kế trở nên dễ dàng hơn, giúp cho chúng ta dễ đặt các nội dung lên các vùng phù hợp của trang web
- Grid View thường có 12 cột. Tổng chiều rộng của 12 cột này sẽ chiếm 100% chiều rộng của trang web. Các cột sẽ được thu nhỏ hoặc mở rộng theo kích thước của cửa sổ trình duyệt



RESPONSIVE WEB DESIGN – RWD - 4

Media Queries

- Xử lý thay đổi giao diện giữa các màn hình khác nhau (chiều rộng màn hình)

Cấu trúc

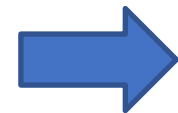
```
@media media-type and (media-feature-rule) { /* CSS rules go here */ }  
  
@media screen and (min-width: 600px) {  
    body {  
        color: red;  
    }  
}
```

media-type

- all
- print
- screen

media-feature-rule

- min-width
- max-width
- width
- orientation



Xác định **Breakpoint**

RESPONSIVE WEB DESIGN – RWD - 4

Media Queries

Cách sử dụng:

- Đưa trực tiếp media queries vào CSS

```
@media media-type and (media-feature-rule) { /* CSS rules go here */ }
```

- Import url vào đầu file CSS

```
@import url('mobile.css') screen and (min-width: 600px)
```

- Dùng thẻ link

```
<link rel="stylesheet" href="css/custom.css" media="screen and (min-width: 400px)">
```

RESPONSIVE WEB DESIGN – RWD - 5

Design For Mobile First

- Thiết kế giao diện cho Mobile trước, sau đó mới đến các thiết bị khác
- Lưu ý về thứ tự của media queries và điều kiện => Có thể dẫn đến bị chồng lặp các thuộc tính CSS

```
/* Small devices (portrait tablets and large phones, 600px  
and up) */  
@media only screen and (min-width: 600px) {...}
```

```
/* Medium devices (landscape tablets, 768px and up) */  
@media only screen and (min-width: 768px) {...}
```

```
/* Large devices (laptops/desktops, 992px and up) */  
@media only screen and (min-width: 992px) {...}
```

```
/* Extra large devices (large laptops and desktops, 1200px  
and up) */  
@media only screen and (min-width: 1200px) {...}
```

```
/* Small devices (portrait tablets and large phones, 600px  
and up) */  
@media only screen and (min-width: 600px) {...}
```

```
/* Medium devices (landscape tablets, 768px and up) */  
@media only screen and (min-width: 768px) {...}
```

```
/* Large devices (laptops/desktops, 992px and up) */  
@media only screen and (min-width: 992px) {...}
```

```
/* Extra large devices (large laptops and desktops, 1200px  
and up) */  
@media only screen and (min-width: 1200px) {...}
```

- ❑ **Bố cục (layout) của website rất quan trọng, nó quyết định đến việc bố trí các thành phần trên trang web có hợp lý, chuyên nghiệp hay không**
- ❑ **RWD (Responsive Design Web) là kỹ thuật được sử dụng để trang web có thể hiển thị tốt trên nhiều loại thiết bị với kích thước và độ phân giải khác nhau**
- ❑ **Media Queries (Truy vấn phương tiện) là một kỹ thuật của CSS, là công nghệ nền tảng của thiết kế web đáp ứng RWD**
- ❑ **Sử dụng Flexbox, Grid Layout thiết lập phần hiển thị theo chiều ngang hoặc chiều dọc.**
- ❑ **CSS Unit quan trọng trong việc RWD**



KẾT THÚC

HỌC VIỆN ĐÀO TẠO LẬP TRÌNH CHẤT LƯỢNG NHẬT BẢN