

Theory 02

CSS Basics

WEEK 01

Nội dung chính

- ❑ TỔNG QUAN VỀ CSS 03
- ❑ CẤU TẠO MỘT KHAI BÁO CSS 04
- ❑ CÁCH CHÈN CSS VÀO HTML 05
- ❑ SELECTOR TRONG CSS 09
- ❑ CÁC CSS THƯỜNG DÙNG 12
- ❑ BỐ CỤC FLEXBOX 27



Tổng quan về CSS

❑ Giới thiệu CSS:

- **CSS** (Cascading Style Sheets), là một **ngôn ngữ định kiểu văn bản**
- Được sử dụng để **định vị** và **định dạng** các **phần tử** trên **trang HTML**
- Giúp cải thiện giao diện, **tăng sức hấp dẫn trực quan** của **trang HTML**



Cách chèn CSS vào HTML

❑ Inline Style Sheet:

- Chèn **CSS** trực tiếp vào **một phần tử HTML** thông qua thuộc tính **style**
- **Cú pháp:** `<tagname style="property: value; ..."> Nội dung...</tagname>`
- Chỉ có ảnh hưởng trên một phần tử HTML đang thêm thuộc tính **style**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Inline Style Sheet</title>
</head>
<body>
  <h1 style="color: red;">This is a Heading</h1>
</body>
</html>
```

Cách chèn CSS vào HTML

❑ Internal Style Sheet (còn gọi là Embedding Style Sheet):

○ Chèn **CSS** vào một trang HTML thông qua cặp thẻ `<style></style>`

○ Cú pháp:

`<style>`

`selector { property: value; }`

`</style>`

○ Chỉ ảnh hưởng trong phạm vi toàn trang HTML chứa cặp thẻ `style`

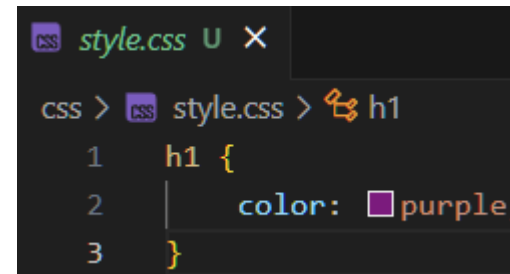
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Internal Style Sheet</title>
  <style>
    h1 {
      color: green;
    }
  </style>
</head>
<body>
  <h1>This is a Heading</h1>
</body>
</html>
```

Cách chèn CSS vào HTML

❑ External Style Sheet (còn gọi là Linking Style Sheet):

- Chèn tập tin **CSS** (*.css) đã tạo vào **nhiều trang HTML** thông qua thẻ `<link>`
- **Cú pháp:** `<link rel="stylesheet" type="text/css" href="./style.css">`
- Có khả năng ảnh hưởng trong phạm vi nhiều trang HTML khác nhau

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>External Style Sheet</title>
  <link rel="stylesheet" href="./css/style.css" type="text/css">
</head>
<body>
  <h1>This is a Heading</h1>
</body>
</html>
```



```
css style.css U X
css > css style.css > h1
1 h1 {
2   color: purple;
3 }
```

Cách chèn CSS vào HTML

❑ Thứ tự ưu tiên các cách chèn CSS:

○ Là thứ tự ưu tiên áp dụng các cách chèn CSS khác nhau trên cùng file HTML (sắp xếp theo thứ tự tăng dần):

**External CSS < Internal CSS < Inline CSS
< Important CSS (!important)**

○ **Lưu ý:** tránh lạm dụng **!important**, thứ tự ưu tiên CSS còn phụ thuộc vào độ cụ thể (**specificity**) của selector.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Import CSS Priority Specificity</title>
  <!-- External CSS -->
  <link rel="stylesheet" href="./css/style.css" />

  <!-- Internal CSS -->
  <style>
    h1 { color: blue; }
  </style>
</head>
<body>
  <h1 style="color: red;">This is a Heading</h1>
</body>
</html>
```


Selector trong CSS

❑ Selector là gì? Công dụng ra sao? Có mấy loại?

- Selector (bộ chọn) là **cách chọn phần tử HTML cần thêm CSS**
- Có **5 loại selector** (bộ chọn) **chính** trong CSS:

Loại bộ chọn	Mô tả	Ví dụ
Selector cơ bản	Chọn theo tên thẻ, class, id	p, .title, #header
Selector tổ hợp	Kết hợp nhiều selector	div p, ul > li, h1 + p
Selector thuộc tính	Chọn phần tử có thuộc tính	input[type="text"]
Selector giả lớp	Chọn trạng thái đặc biệt của phần tử	a: hover, input: focus
Selector giả phần tử	Chọn một phần tử cụ thể bên trong phần tử hiện tại	p::first-line, div::before

Selector trong CSS

❑ Ví dụ các Selector thường dùng:

Bộ chọn	Mô tả	Ví dụ
*	*	Chọn tất cả các phần tử
.class	.header	Chọn tất cả các phần tử có <i>class</i> = "header"
#id	#content	Chọn phần tử duy nhất có <i>id</i> = "content"
element	p	Chọn tất cả các <i>tag</i> <p>
element1, element2	div, p	Chọn tất cả các <i>tag</i> <div> & <tag p>

Selector trong CSS

❑ Ví dụ các Selector thường dùng:

Bộ chọn	Mô tả	Ví dụ
<i>element1 > element2</i>	div > p	Chọn tag <p> là con trực tiếp của tag <div>
<i>element1 + element2</i>	div + p	Chọn 1 tag <p> nằm ngay sau tag <div> (cùng cấp)
<i>element1 ~ element2</i>	div ~ p	Chọn tất cả các tag <p> được đặt sau tag <div> (cùng cấp)
<i>element[attribute="value"]</i>	div[class="demo"]	Chọn tất cả các thẻ div có thuộc tính class="demo"

Các CSS thường dùng

❑ CSS Fonts, Colors:

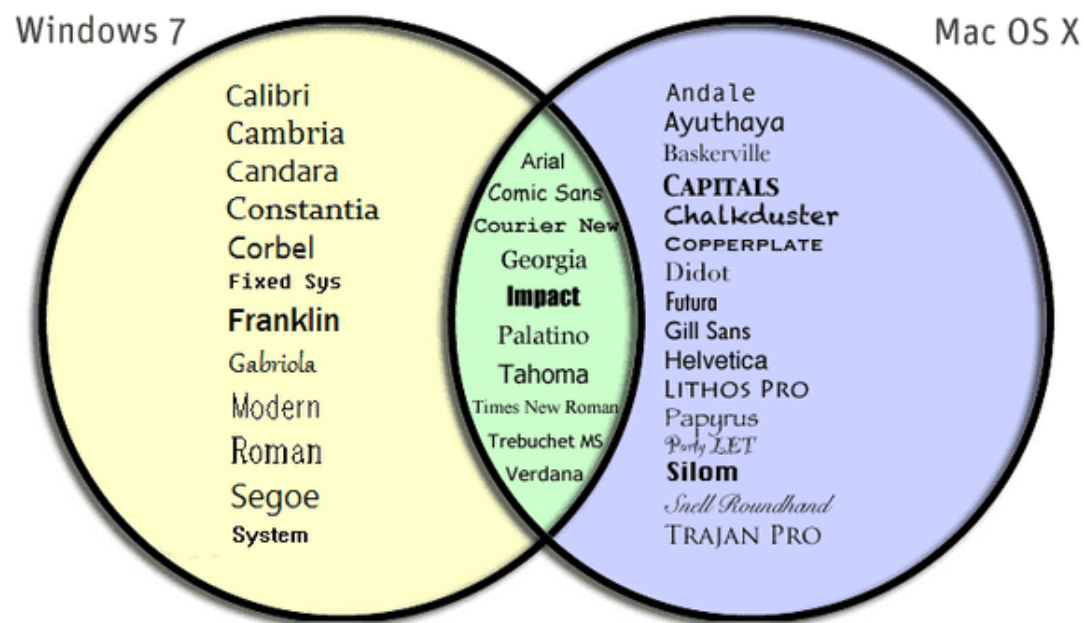
○ CSS Fonts dùng để định dạng CSS cho văn bản trên trang HTML

Thuộc tính	Mô tả	Các giá trị phổ biến
font-family	Tên phong chữ cụ thể và chung chung	font-family: Cambria, "Times New Roman", serif; font-family: Verdana, Arial, sans-serif;
font-size	Kích thước tuyệt đối hoặc tương đối của phong chữ	font-size: 1.5em; font-size: 50%; font-size: x-large; font-size: 14pt;
font-weight	Trọng lượng của một phong chữ	font-weight: bold; font-weight: bolder;
font-style	Phong cách của một phong chữ	font-style: normal; font-style: italic;
line-height	Sự canh chỉnh chiều cao của chữ	line-height: 25px;
color	Màu của chữ	color: red; color: green; color: blue;

Các CSS thường dùng

❑ CSS Fonts, Colors:

- Các fonts chung: https://www.w3schools.com/cssref/css_websafe_fonts.asp
- Trình duyệt sẽ không đọc được fonts bạn tự cài không có trong fonts chung

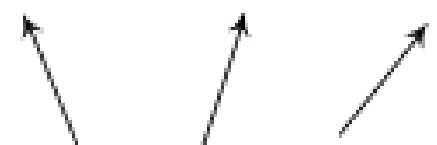


Các CSS thường dùng

❑ CSS Fonts, Colors:

- Thuộc tính **font-family** có thể bao gồm nhiều giá trị khác nhau
- **Nên cung cấp nhiều hơn một giá trị cho font-family** để phòng trường hợp trình duyệt không hỗ trợ phông chữ chính, các **fallback values** sẽ được gọi.
- Ví dụ: **font-family: Cambria, "Times New Roman", serif;**

```
h1 {  
  font-family: "Times New Roman", Palatino, Georgia, Serif;  
}
```



Fallback fonts

Các CSS thường dùng

❑ CSS Fonts, Colors:

- Thuộc tính **color** để thiết lập màu chữ, có 3 giá trị màu chính thường dùng:
 - **color name**: màu sắc được xác định bằng **tên màu**. Ví dụ: red, green, blue...
 - **HEX value**: màu sắc được xác định bằng **trị số hexa** theo **cấu trúc #rrggbb**
Cú pháp: **rr (red) gg (green) bb (blue)** ứng với giá trị hexa từ **00 -> ff**
 - **RGB value**: màu sắc được xác định theo **cấu trúc: rgb(red, green, blue)**
Cú pháp: **red, green, blue** ứng với giá trị từ **0 -> 255**

Các CSS thường dùng

❑ Box Model:

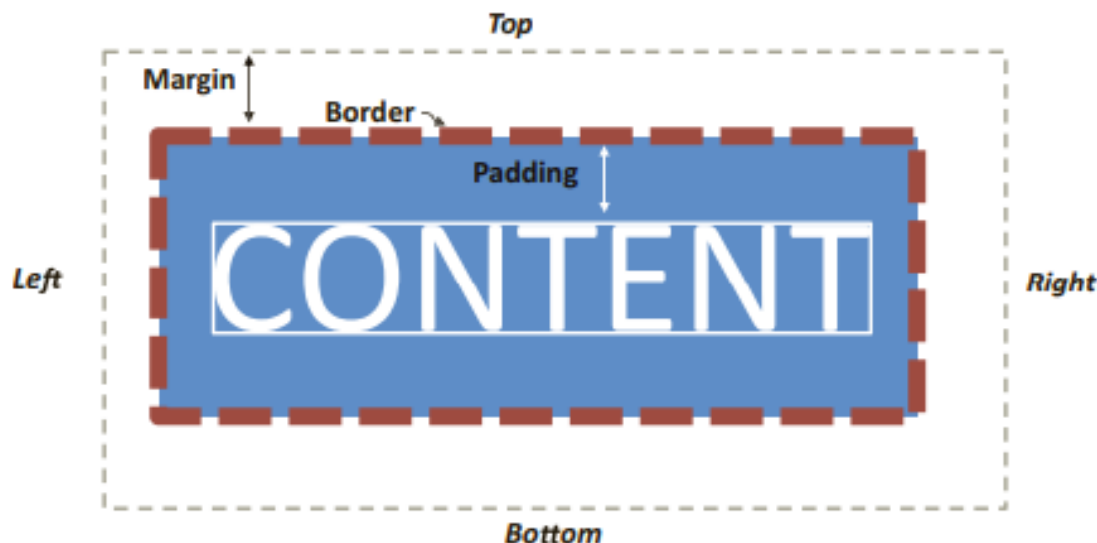
- **Box Model** là một **kỹ thuật cơ bản nhất trong CSS Layout**
- Được **sử dụng để mô tả về khoảng cách** mà mỗi phần tử **HTML** sở hữu.
- Bao gồm **4 phần** quan trọng đó là: **Margin, Border, Padding, Content**



Các CSS thường dùng

❑ CSS Padding:

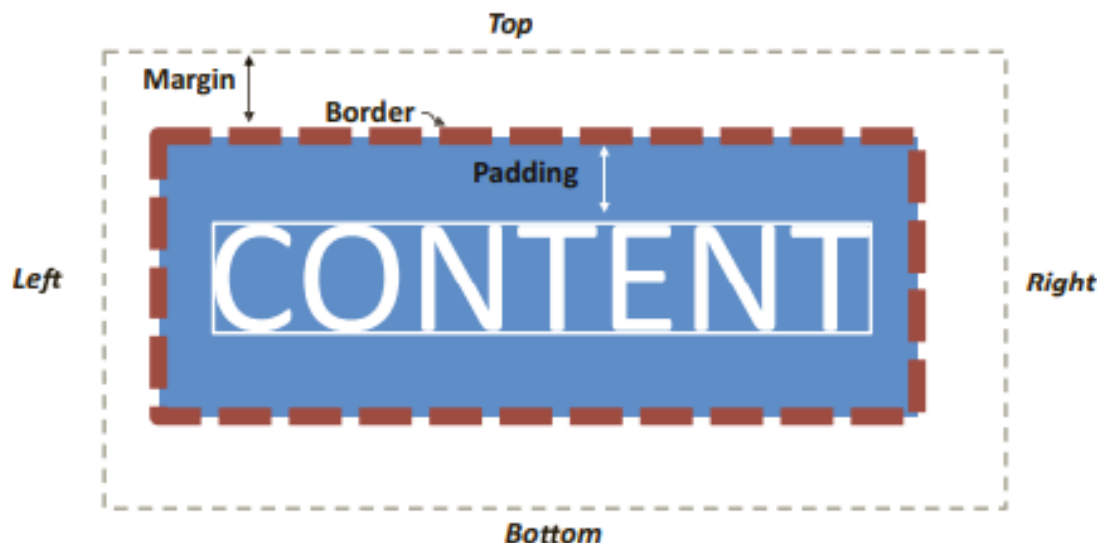
- **padding**: là khoảng cách từ border của 1 phần tử tới nội dung nằm bên trong phần tử đó. Ví dụ: **padding: 35px;**
- Các giá trị chính: **padding-top, padding-right, padding-bottom, padding-left**



Các CSS thường dùng

❑ CSS Padding:

- **margin**: là khoảng cách giữa các phần tử HTML với nhau. Ví dụ: **margin: 15px;**
- Các giá trị chính: **margin-top**, **margin-right**, **margin-bottom**, **margin-left**
- **Lưu ý**: giá trị **auto** sẽ tự động canh đều 2 bên của phần tử. Ví dụ: **margin: auto;**



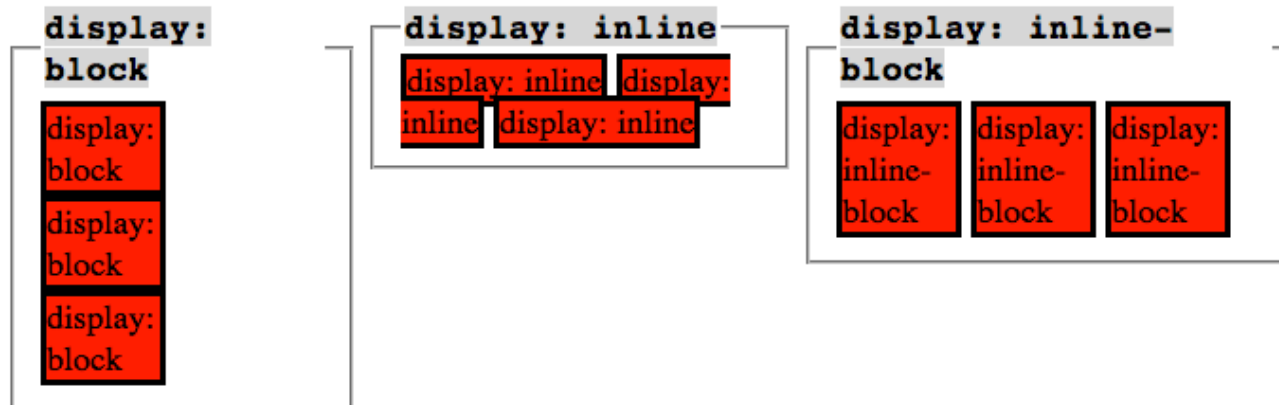
Các CSS thường dùng

❑ CSS Display:

- Mỗi một phần tử HTML đều có một giá trị hiển thị mặc định.
- CSS display có 4 giá trị hiển thị chính: **block**, **inline**, **inline-block**, **none**

block vs inline vs inline-block

Below are a bunch of `<div style="width: 50px"...>` with different `display:` settings.



Các CSS thường dùng

□ CSS Display:

- Một phần tử ở cấp độ **inline** không thể chứa một phần tử ở cấp độ **block**.
- `<div>`, `` thường sử dụng các thuộc tính **style**, **class**, **id** đi kèm.
- Ví dụ thẻ `<div>`:

```
<div style="background-color:black;color:white;padding:20px;">  
  <h2>London</h2>  
  <p>London is the capital city of England. It is the most populous city in the  
United Kingdom, with a metropolitan area of over 13 million inhabitants.</p>  
  <p>Standing on the River Thames, London has been a major settlement for two  
millennia, its history going back to its founding by the Romans, who named it  
Londinium.</p>  
</div>
```

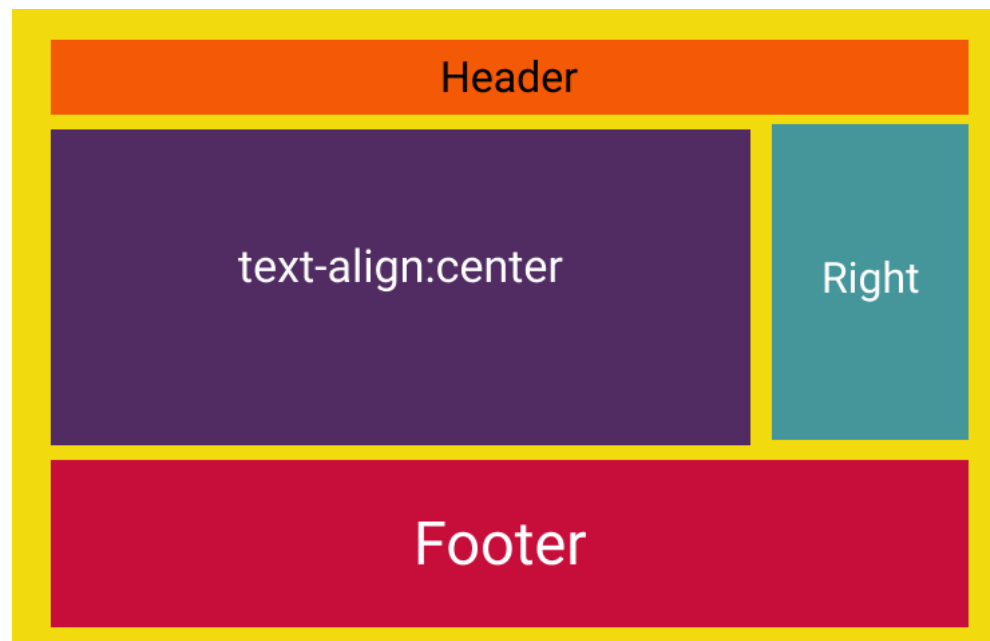
- Ví dụ thẻ `<div>`:

```
<p>My mother has <span style="color:blue;font-weight:bold">blue</span> eyes and  
my father has <span style="color:darkolivegreen;font-weight:bold">dark  
green</span> eyes.</p>
```

Các CSS thường dùng

❑ CSS Alignment:

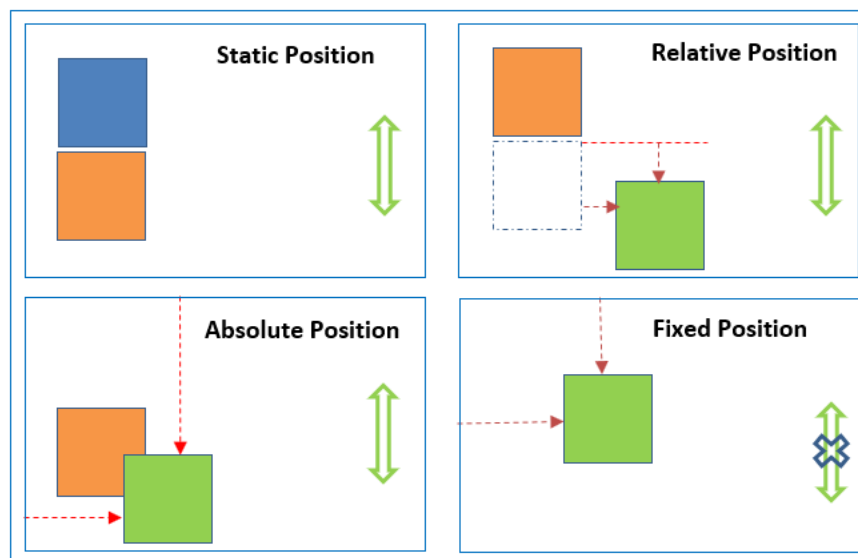
- **text-align**: điều chỉnh vị trí đoạn văn bản theo chiều ngang
- Các giá trị chính: **left**, **right**, **center**, **justify** (canh đều)



Các CSS thường dùng

❑ CSS Alignment:

- **position**: là thuộc tính CSS được dùng để điều chỉnh vị trí phần tử HTML
- Các **giá trị chính** của position: **static**, **relative**, **absolute**, **fixed**
- Các **thuộc tính** của **position** để căn chỉnh vị trí phần tử: **top**, **left**, **right**, **bottom**



Các CSS thường dùng

□ CSS Alignment:

- **position**: là thuộc tính CSS được dùng để điều chỉnh vị trí phần tử HTML
- **Cách hiển thị các giá trị** của thuộc tính **position**:
 - **static**: mặc định, không dùng được top, left, right, bottom
 - **relative**: so với vị trí ban đầu của chính nó
 - **absolute**: so với phần tử cha gần nhất có position khác static
 - **fixed**: so với viewport (cửa sổ trình duyệt), dùng được top, left, right, bottom
- **Cách sử dụng thường dùng**:
 - **Ở thẻ cha**: thường dùng **position: relative** để thẻ cha không di chuyển
 - **Ở thẻ con**: thường dùng **position: absolute** để thẻ con di chuyển bên trong thẻ cha.

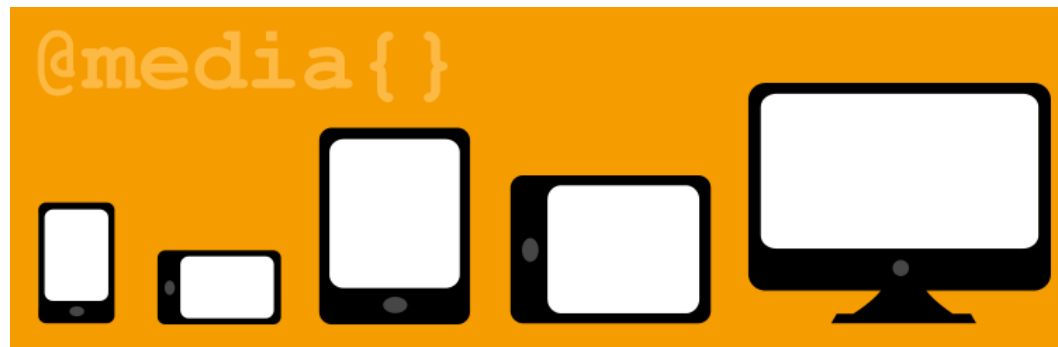
Các CSS thường dùng

❑ CSS Media Queries:

○ **@media** giúp thiết kế giao diện Website **Responsive** (hiển thị linh hoạt, thích ứng tốt trên nhiều loại thiết bị: mobile, tablet, desktop,...)

○ Các tính năng thường dùng của media queries:

- Giao diện cần thay đổi theo **màn hình nhỏ (mobile-first)**
- Ẩn/hiện một phần tử tùy theo thiết bị
- Tùy chỉnh bố cục, font, padding/margin cho các loại màn hình khác nhau

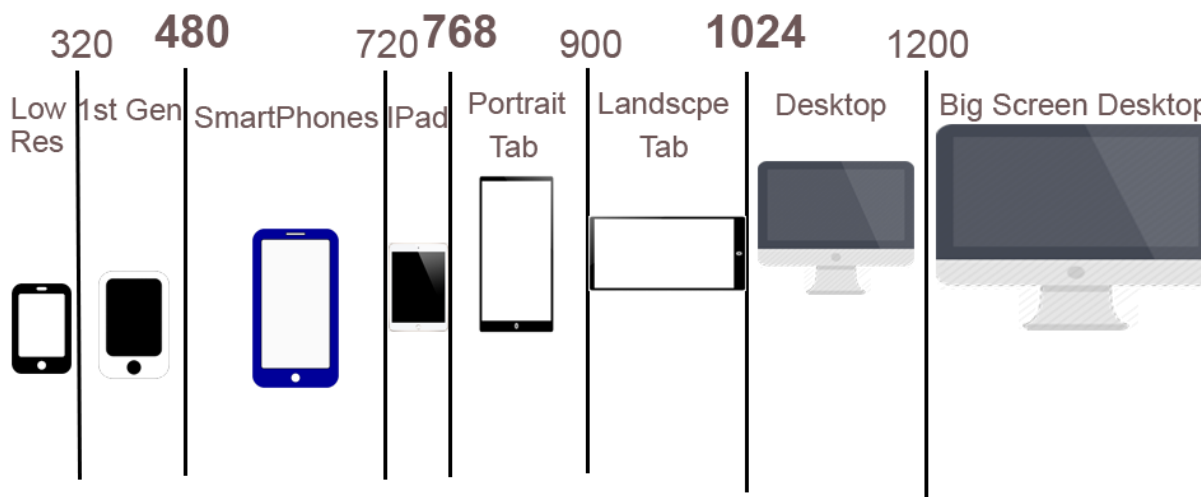


Các CSS thường dùng

❑ CSS Media Queries:

○ Cú pháp:

`@media not|only mediatype and (expressions) {
 CSS-Codes;
}`



Các CSS thường dùng

❑ Điểm ngắt (break points):

- **Media queries** có thể xác định loại thiết bị hiện tại thông qua viewport size bằng cách thiết lập một điểm ngắt (breakpoints) cho trang HTML
- **Breakpoints** là điểm chuyển giao giữa 2 định kiểu màn hình khác nhau giúp trang Web dễ đọc hơn trên từng loại màn hình khác nhau:

Thiết bị	Độ rộng viewport nhỏ nhất	Độ rộng viewport lớn nhất
Smartphones cỡ nhỏ	320px	480px
Tablets, smartphones cỡ lớn	481px	768px
Tablets (hướng ngang), laptops, màn hình desktop cỡ nhỏ	769px	1279px
Màn hình desktop cỡ lớn	1280px	NA

Bố cục Flexbox

□ Giới thiệu:

○ Là một **phương pháp** hiệu quả để **bố cục**, **canh chỉnh** và **phân phối không gian** giữa các phần tử trong trang web ngay cả khi **viewport** và **size** của các phần tử của bạn không xác định.

Complex flexbox example

First article

Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth gastropub hammock.

Second article

Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth gastropub hammock.

Smile Laugh Wink Shrug

Blush

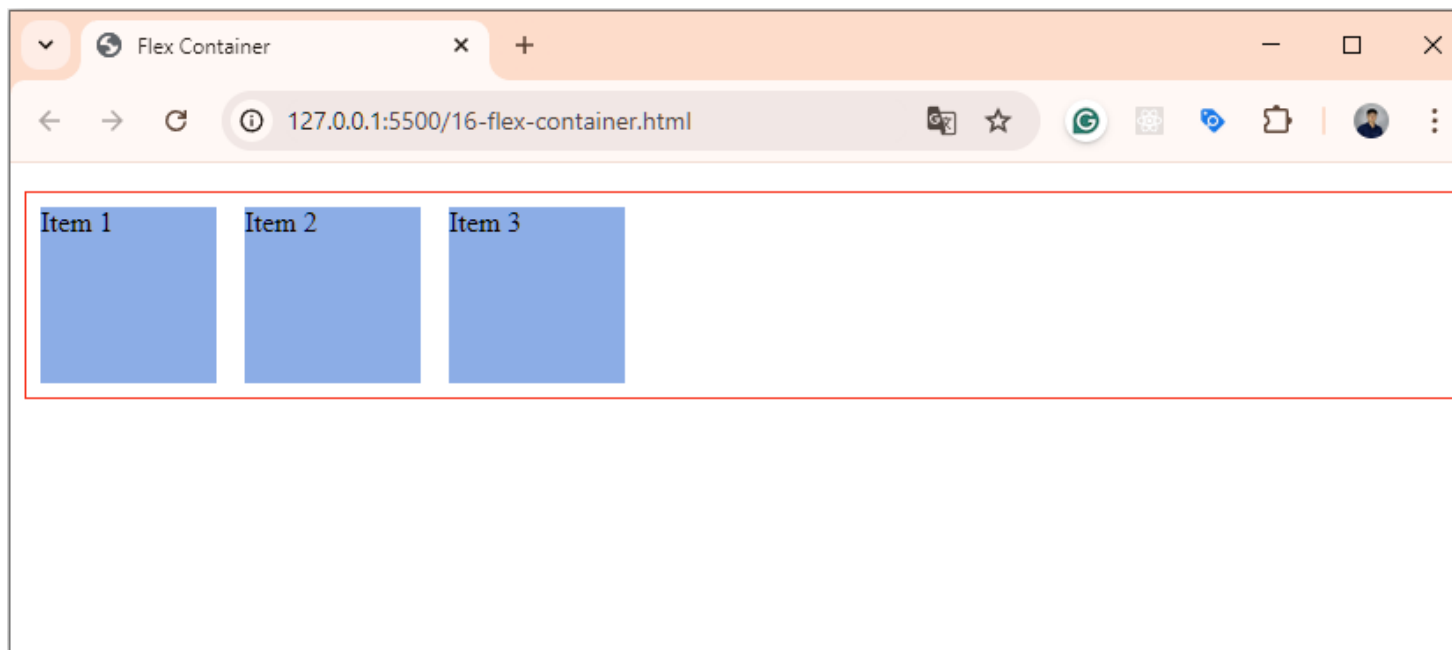
Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth gastropub hammock.

Cray food truck brunch, XOXO +1 keffiyeh pickled chambray waistcoat ennui. Organic small batch paleo 8-bit. Intelligentsia umami wayfarers pickled, asymmetrical kombucha letterpress kitsch leggings cold-pressed squid chartreuse put a bird on it. Listicie pickled man bun cornhole heirloom art party.

Bố cục Flexbox

❑ Thuộc tính display:

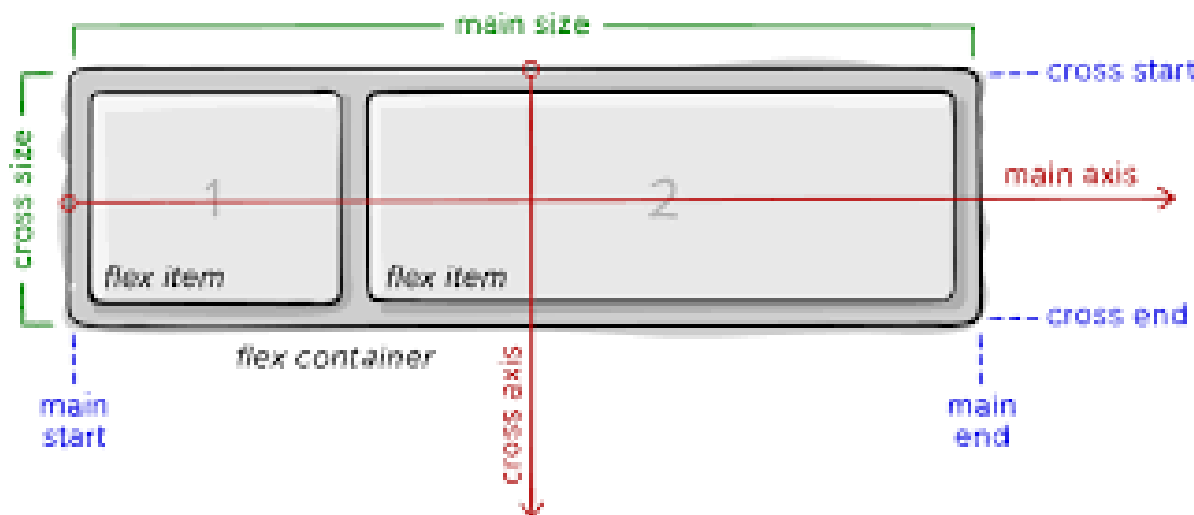
- Là thuộc tính để khởi tạo không gian chứa các phần tử flex
- Cú pháp: **display:flex** hoặc **display:inline-flex**



Bố cục Flexbox

❑ Thuộc tính flex-direction:

- Là **thuộc tính** giúp kiểm soát hướng mà các **flex item** đặt dọc theo trục chính
- 4 giá trị chính: **row**, **column**, **row-reverse**, **column-reverse**
- Có 2 hướng chính: **main-axis** và **cross-axis**



Bố cục Flexbox

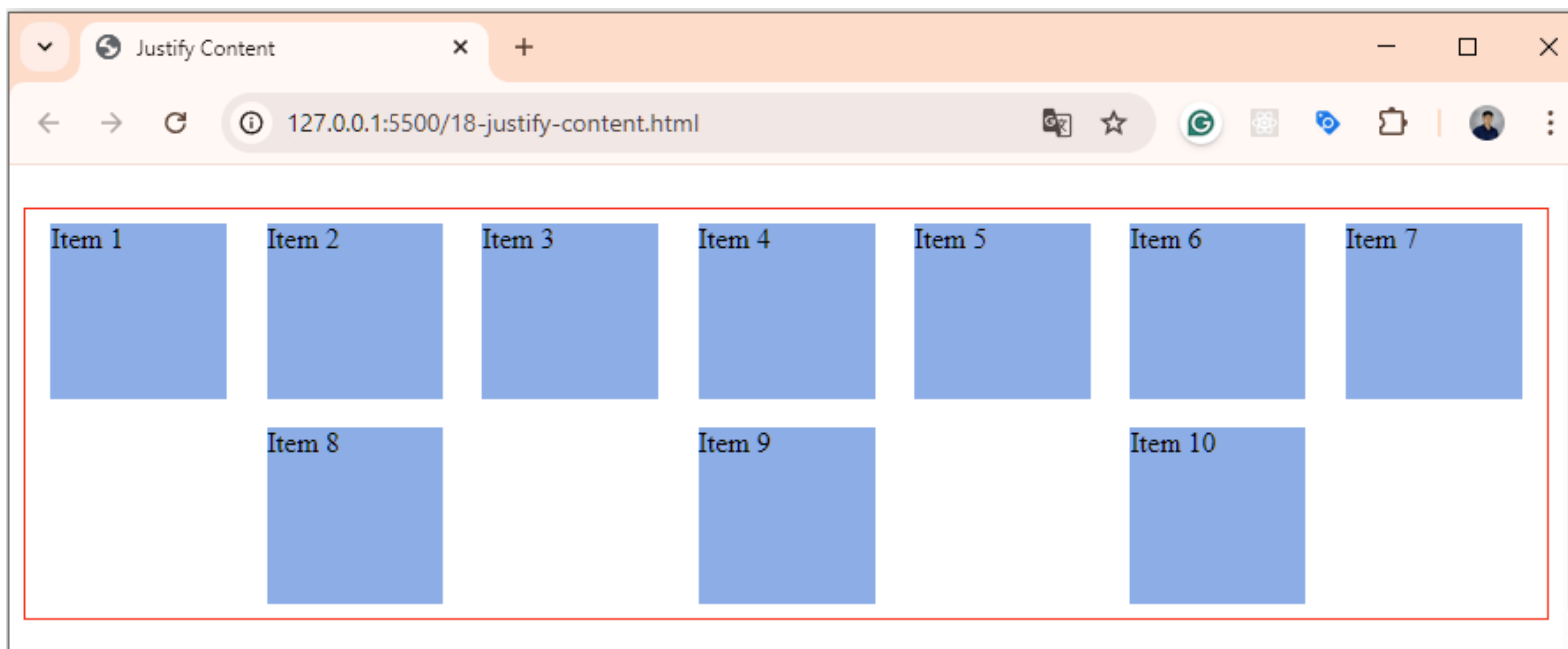
❑ Thuộc tính flex-flow:

- Là sự kết hợp 2 thuộc tính **flex-direction** và **flex-wrap** trên các **flex item**
 - **flex-direction**: kiểm soát hướng mà các **flex item** đặt dọc theo trục chính
 - 4 giá trị chính: **row**, **column**, **row-reverse**, **column-reverse**
 - Có 2 hướng chính: **main-axis** và **cross-axis**
 - **flex-wrap**: bao bọc các **flex item** và quyết định có xuống dòng hay không
 - 3 giá trị chính:
 - **nowrap**: không xuống dòng, không giữ nguyên kích thước các flex item
 - **wrap**: xuống dòng, giữ nguyên kích thước các flex item
 - **wrap-reverse** : giống wrap nhưng thứ tự sắp xếp ngược lại
- Các giá trị của **flex-flow**: **row wrap**, **row nowrap**, **column wrap**, **column nowrap**,...

Bố cục Flexbox

❑ Thuộc tính justify-content:

- Là thuộc tính **giúp canh chỉnh nội dung** của các flex item trên trục chính
- Có 5 giá trị chính: **flex-start**, **flex-end**, **center**, **space-between**, **space-around**



Bố cục Flexbox

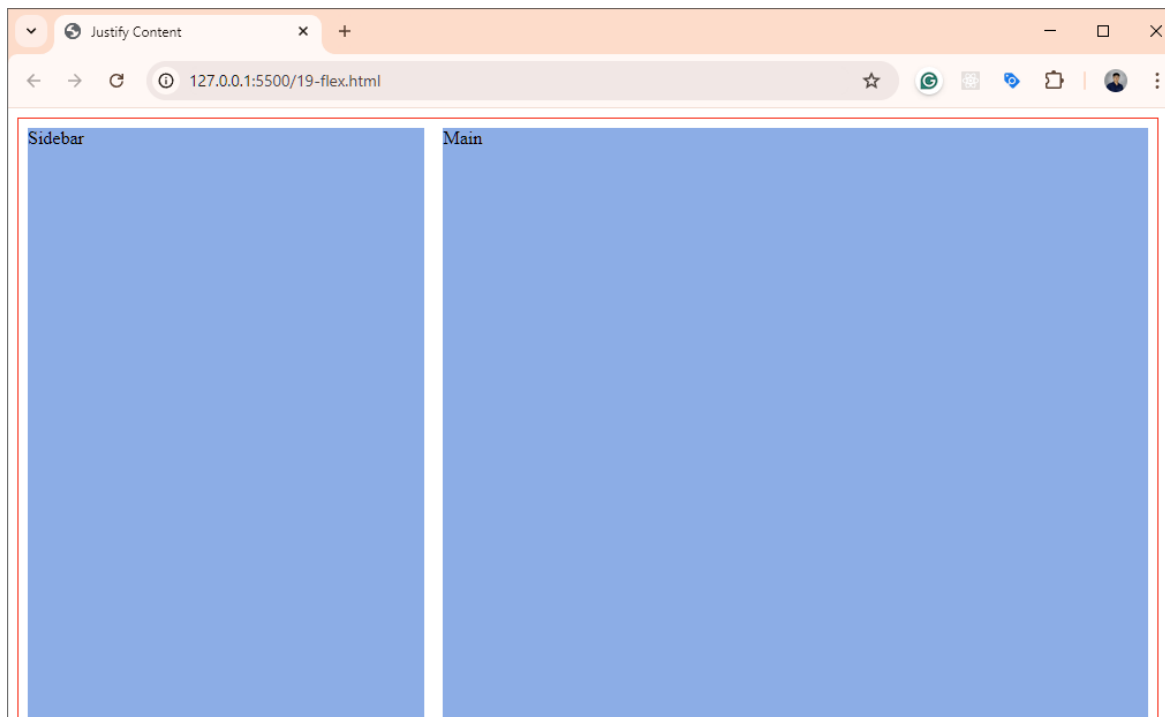
❑ Thuộc tính flex:

- Là sự kết hợp 3 thuộc tính flex-grow, flex-shrink, flex-basis trên các flex item
- Ý nghĩa từng thuộc tính:
 - **flex-grow**: quy định mức độ mở rộng của phần tử khi còn chỗ trống
 - **flex-shrink**: quy định mức độ thu hẹp của phần tử khi hết chỗ trống
 - **flex-basis**: quy định kích thước ban đầu của phần tử (trước khi grow/shrink)
- Ví dụ: **flex: 2 1 auto**; tương đương: flex-grow: 2; flex-shrink: 1; flex-basis: auto;

Bố cục Flexbox

❑ Thuộc tính flex:

○ **Minh họa:** thiết kế bố cục trang web gồm 2 phần chính sidebar, main như sau:



Tóm tắt bài học

- ☐ Tổng quan về CSS
- ☐ Cấu tạo một khai báo CSS
- ☐ Cách chèn CSS vào HTML
- ☐ Selector trong CSS
- ☐ Các CSS thường dùng

