ĐẠI NAM
UNIVERSITY

THIẾT KẾ WEB VÀ TRIỂN KHAI HỆ THỐNG PHẦN MỀM

(FIT4014 - WEB)

QUY ĐỊNH TRONG LỚP HỌC

- Tuân thủ Tuyệt đối các quy định sau:
  - Tham dự đầy đủ buổi học, tự học, chuẩn bị slides, demo.
  - Tôn trọng lớp học, đúng giờ, không gây ảnh hưởng, tuân thủ văn hóa học đường.
  - Máy tính chỉ sử dụng cho mục đích học

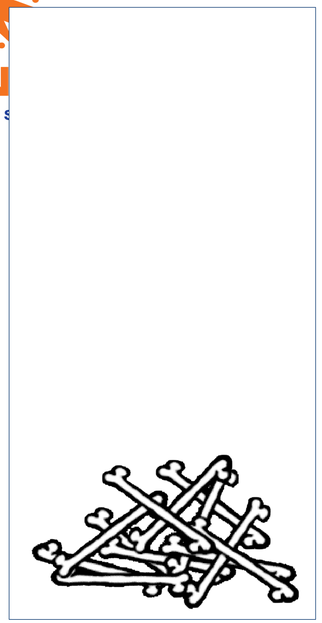
KHÔNG NÓI CHUYỆN RIÊNG

KHÔNG SỬ DỤNG ĐIỆN THOẠI

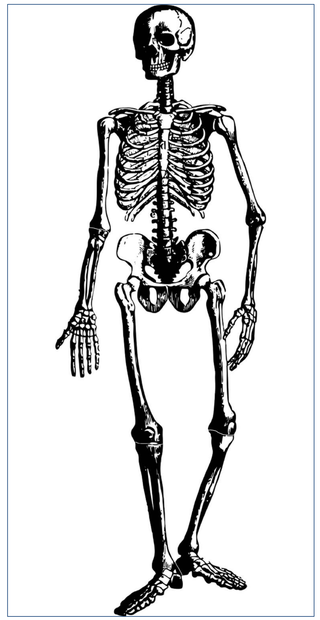
KHÔNG NGỦ

CODE ĐẦY ĐỦ

NỘI DUNG BUỔI 4



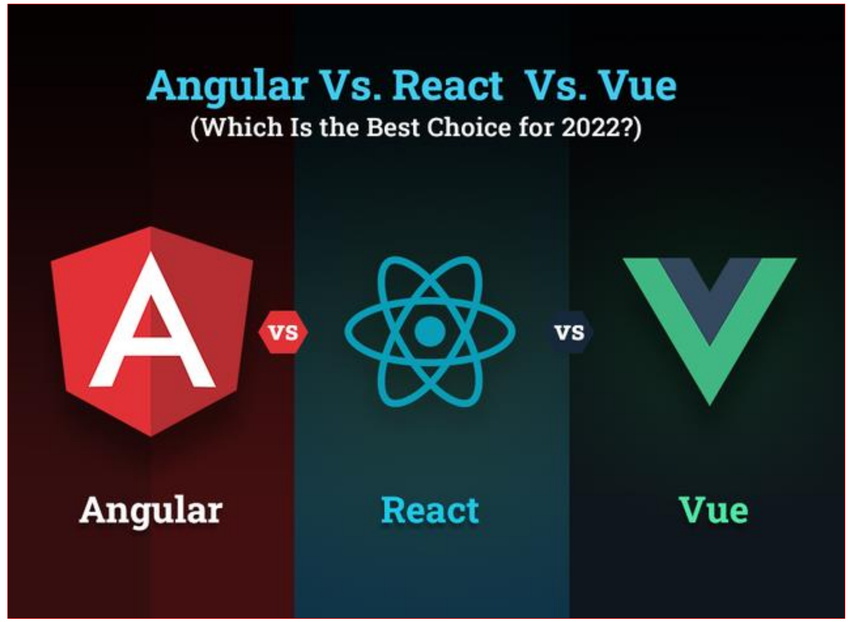
WORDS + IMAGES



HTML

CSS

JAVASCRIPT



FRONTEND FRAMEWORK

CHƯƠNG III. CSS

Nội dung

01 Float và Clear

02 Attribute selectors (Bộ chọn thuộc tính)

03 Units (Đơn vị)

04 Specificity và !important

05 Math Functions (Hàm toán học)

06 CSS function attr()

07 Variables (Biến)

08 Box Sizing (Kích thước hộp)

09 Khái niệm Responsive

10 Media Queries

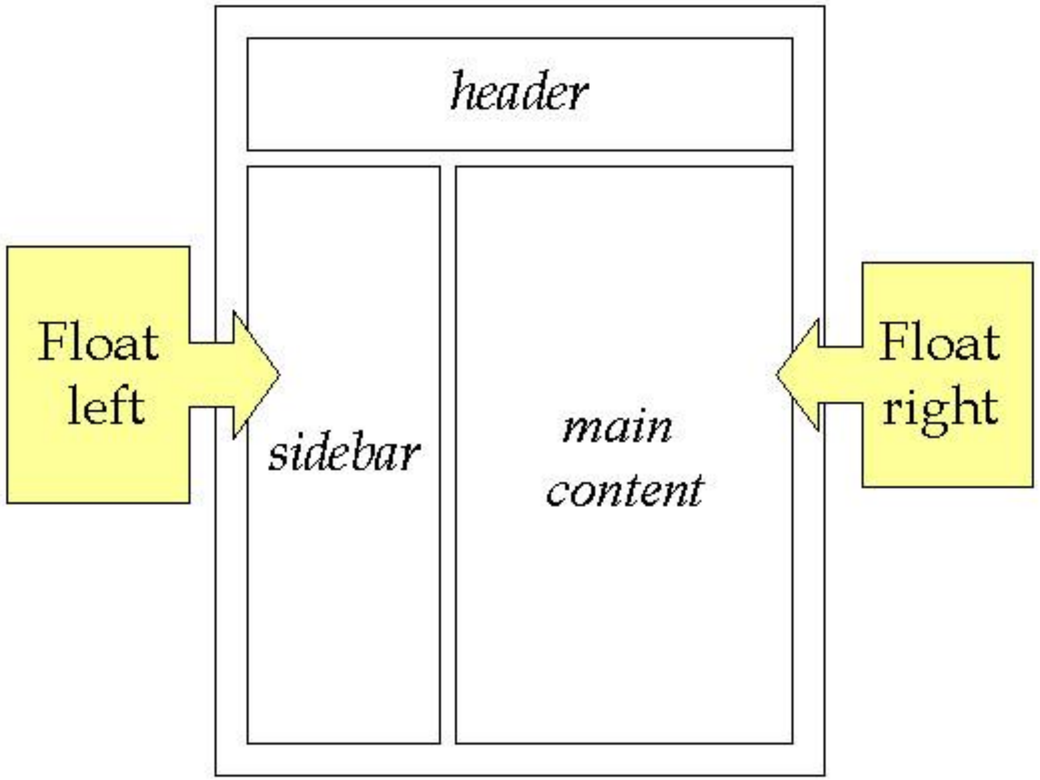
01. Float và Clear

1.1. Float

Thuộc tính float là một thuộc tính được sử dụng để chuyển một phần tử sang góc trái hoặc góc bên phải của không gian bao quanh nó.

Float có các giá trị sau:

- float: left; // Phần tử nằm phía bên trái.
- float: right; // Phần tử nằm phía bên phải.
- float: none; // Phần tử nằm tại chính vị trí của nó (Mặc định).



01. Float và Clear

1.2. Clear

Thuộc tính Clear là một thuộc tính ngược lại hoàn toàn với Float.

Ngăn chặn phần tử B chiếm vùng không gian của phần tử A (với A là phần tử sử dụng float).

Thuộc tính clear có các giá trị sau:

- clear: left;
  - Ngăn chặn phần tử B chiếm chỗ phần tử A, khi phần tử A có sử dụng float: left;
  - Thuộc tính clear: left; không có tác dụng khi phần tử A sử dụng float: right;
- clear: right;
  - Ngăn chặn phần tử B chiếm chỗ phần tử A, khi phần tử A có sử dụng float: right;
  - Thuộc tính clear: right; không có tác dụng khi phần tử A sử dụng float: left;

01. Float và Clear

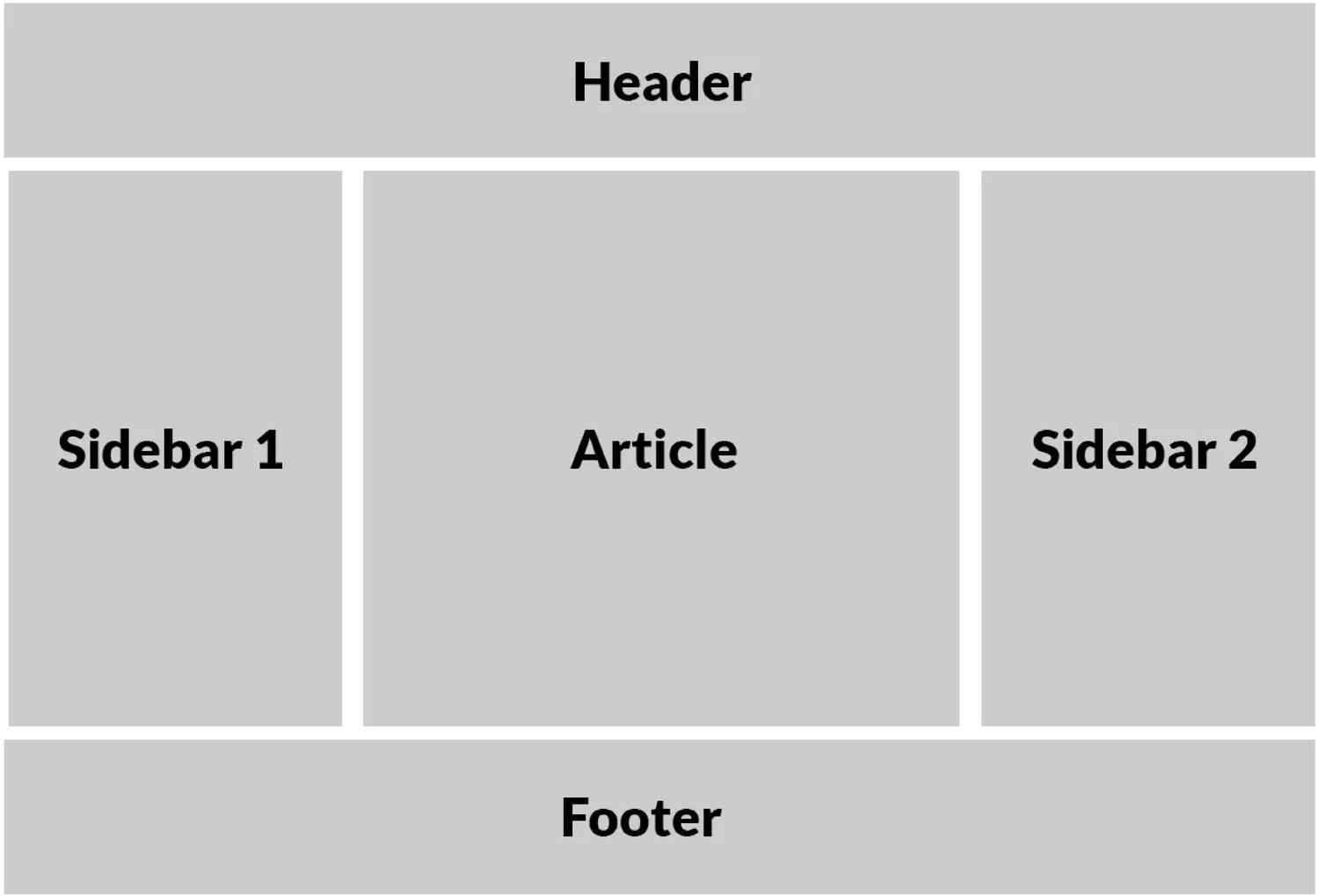
1.2. Clear

Thuộc tính clear có các giá trị sau:

- clear: bh;
  - Khi cả 2 phần tử A và phần tử B sử dụng float, chúng ta không thể sử dụng clear: left; hay clear: right; để ngăn chặn thành phần C chiếm vùng không gian còn trống, trong trường hợp này ta sử dụng thuộc tính clear: both; để ngăn chặn sự chiếm vùng của thành phần C.
  - Thuộc tính clear: both; có thể ngăn chặn sự chiếm vùng không gian cả khi chỉ có một phần tử sử dụng float (left hoặc right).
- clear: none;
  - Thuộc tính clear: none; là dạng phục hồi khi Thành phần sử dụng thuộc tính clear:

01. Float và Clear

1.3. Ví dụ



02. Attribute selectors (Bộ chọn thuộc tính)

Bộ chọn attribute được sử dụng để chọn các phần tử có thuộc tính được chỉ định.

Một số kiểu chọn:

[attribute]

- Chọn tất cả các phần tử có thuộc tính [attribute].

[attribute="value"]

- Chọn tất cả các phần tử có thuộc tính [attribute] phải chứa giá trị value.
- Chỉ được có giá trị duy nhất là value, nếu có thêm giá trị khác thì sẽ không được chọn.
- Giá trị có thể bao gồm nhiều từ.

02. Attribute selectors (Bộ chọn thuộc tính)

Bộ chọn attribute được sử dụng để chọn các phần tử có thuộc tính được chỉ định.

Một số kiểu chọn:

[attribute~="value"]

- Chọn tất cả các phần tử có thuộc tính [attribute] phải chứa giá trị value.
- Nhưng value đó phải đứng độc lập, không được viết liền với từ khác.
- value phải là 1 từ khóa duy nhất.

[attribute|="value"]

- Chọn các phần tử có thuộc tính [attribute], giá trị có thể chính xác là giá trị được chỉ định hoặc giá trị được chỉ định theo sau dấu gạch nối (-).
- Lưu ý: Giá trị phải là một từ nguyên vẹn, đơn lẻ như class="top" hoặc theo sau là dấu gạch ngang (-), như class="top-text".

02. Attribute selectors (Bộ chọn thuộc tính)

Bộ chọn attribute được sử dụng để chọn các phần tử có thuộc tính được chỉ định.

Một số kiểu chọn:

[attribute^="value"]

- Chọn tất cả các phần tử có thuộc tính [attribute] bắt đầu bằng giá trị value
- Khác với [attribute|="value"] ở chỗ nó không có ngoại lệ và chọn tất cả các phần tử bắt đầu bằng value, được phép viết liền với các từ khác.

[attribute$="value"]

- Chọn tất cả các phần tử có thuộc tính [attribute] kết thúc bằng giá trị value.
- Lưu ý: Giá trị có thể được viết liền với từ khác.

02. Attribute selectors (Bộ chọn thuộc tính)

Bộ chọn attribute được sử dụng để chọn các phần tử có thuộc tính được chỉ định.

Một số kiểu chọn:

[attribute*="value"]

- Chọn tất cả các phần tử có thuộc tính [attribute] chứa giá trị value.
- Lưu ý: Giá trị không nhất thiết phải là một từ nguyên vẹn!

03. Units (Đơn vị)

Độ dài tuyệt đối

Là loại đơn vị có giá trị cố định và thể hiện chính xác chiều dài kích thước nó hiển thị, không phụ thuộc cũng như không thay đổi bởi bất kỳ tác động bên ngoài nào.

Một số đơn vị:

- cm
- mm
- in: inches (1in = 96px = 2.54cm)
- px: pixels (1px = 1/96 in) tương ứng với một điểm ảnh trên máy tính.
- pt: points (1pt = 1/72 in)
- pc: picas (1pc = 12 pt)

03. Units (Đơn vị)

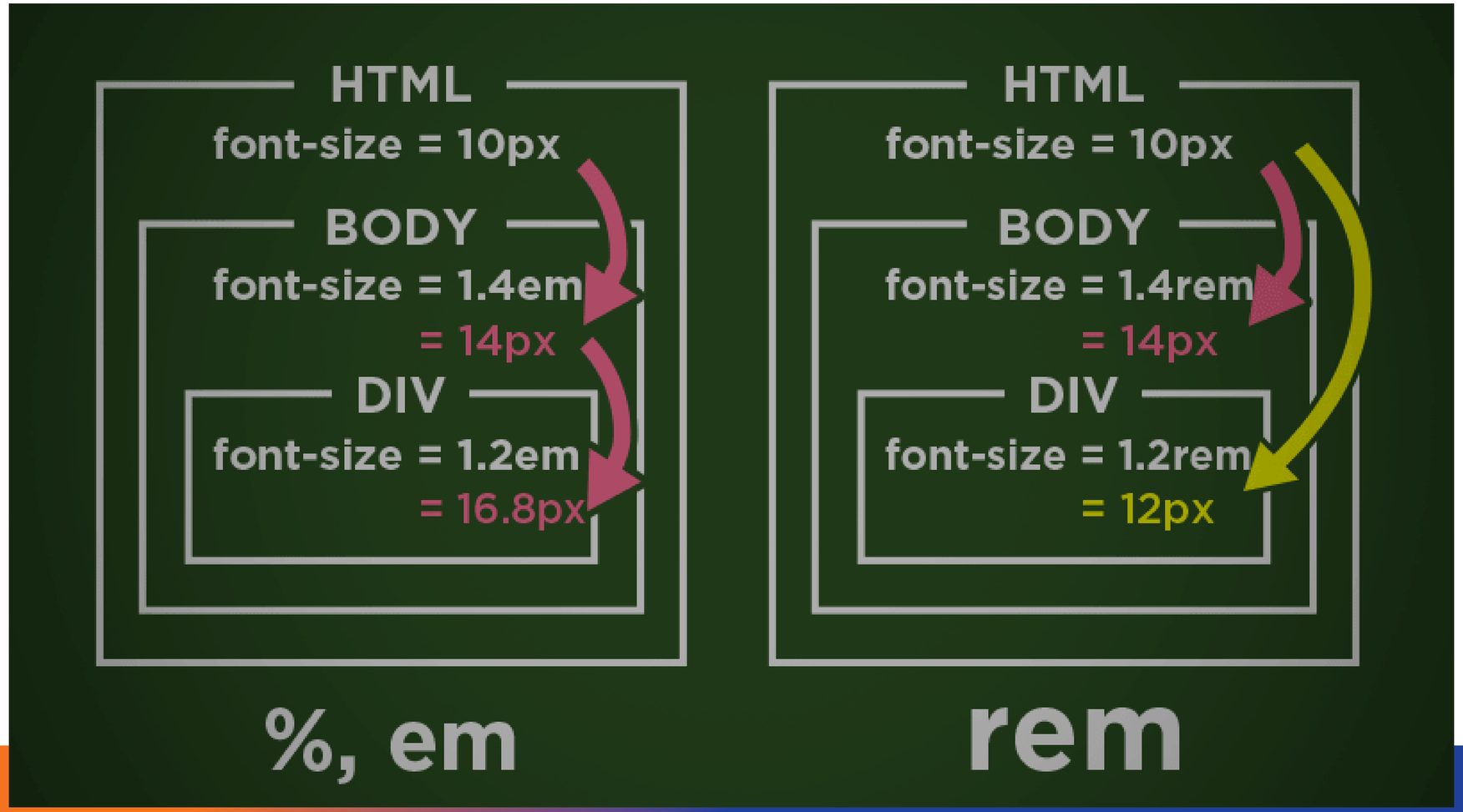
Độ dài tương đối

Là đơn vị đo lường được sử dụng trong CSS ở mức tương đối, thường phụ thuộc vào 1 thành phần nào đó thì mới xác định được giá trị của nó.

Một số đơn vị:

- %: giá trị tương đối so với phần tử cha.
- rem: giá trị tương đối với font-size của phần tử gốc. Phần tử gốc ở đây là thẻ html.
- em: 1em tương đương với kích cỡ của font-size của phần tử cha có định nghĩa font-size.
- vw (viewport width): 1vw tương đương với 1% của chiều rộng cửa sổ trình duyệt.
- vh (viewport height): 1vh tương đương với 1% của chiều cao cửa sổ trình duyệt.
- ex: 1ex tương đương với chiều cao (height) của một chữ x (in thường) của font hiện tại.
- ch: 1ch tương đương với chiều rộng (width) của một số 0 theo size hiện tại.

03. Units (Đơn vị)

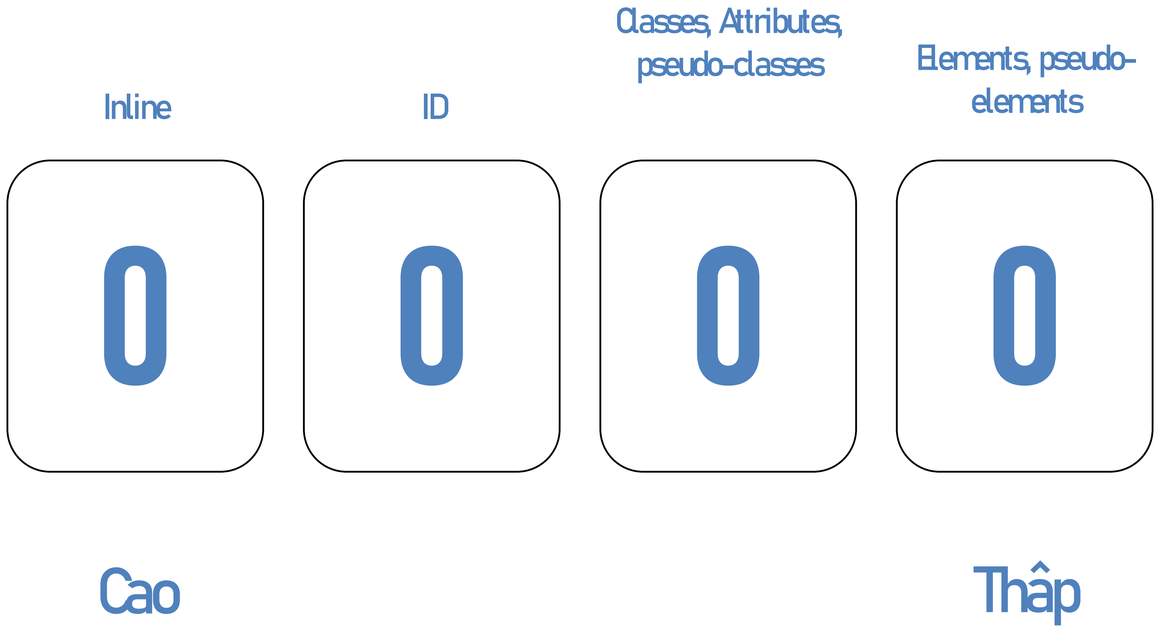


04. Specificity (Tính đặc hiệu) và !important (Quan trọng)

4.1. Specificity (Tính đặc hiệu)

Tính đặc hiệu (hay độ ưu tiên) là cách mà trình duyệt quyết định sẽ áp dụng thuộc tính css nào với một phần tử khi có nhiều quy tắc css cùng trỏ đến phần tử đó.

Sự phân cấp tính đặc hiệu: Mỗi selector đều có vị trí của nó trong hệ thống phân cấp. Độ đặc hiệu của selector có 4 mức (như hình):



04. Specificity (Tính đặc hiệu) và !important (Quan trọng)

4.1. Specificity (Tính đặc hiệu)

Làm sao để tính toán tính đặc hiệu? Ta biểu diễn tương đối tính đặc hiệu của một selector như dưới đây, sau đó xem giá trị nào càng cao thì càng được ưu tiên:

- 1-0-0-0: Inline styles
- 0-X-0-0: Số lượng ID selector
- 0-0-Y-0: Số lượng Classes, attributes và pseudo-classes
- 0-0-0-Z: Số lượng Elements và pseudo-elements
- Viết CSS theo kiểu Internal và External không có độ ưu tiên.
- Universal selector (*) và combinators selector (+, >, ~) không làm tăng tính đặc hiệu.

04. Specificity (Tính đặc hiệu) và !important (Quan trọng)

4.1. Specificity (Tính đặc hiệu)

Ví dụ: Tính toán tính đặc hiệu của các bộ chọn sau:

- h2 { color: red; }
- h2.title { color: blue; }
- div h2.title { color: orange; }
- .head h2.title {color: green; }
- div.head h2.title {color: gray; }
- div.head h2.title:last-child {color: yellow; }
- #main div.head h2.title {color: pink; }
- #main div#head h2.title {color: purple; }
- #main div#head h2.title::first-letter {color: red; }
- #main div#head h2::first-letter {color: yellow; }

Code HTML mẫu:

```
<main id="main">
    <div class="head" id="head">
      <h2 class="title">
        Tiêu đề 1
      </h2>
      <h2 class="title">
        Tiêu đề 2
      </h2>
    </div>
</main>
```

04. Specificity (Tính đặc hiệu) và !important (Quan trọng)

4.2. !important (Quan trọng)

Important được sử dụng để thay đổi thứ tự ưu tiên của CSS, khi một thuộc tính CSS nào đó được gán lệnh Important thì đồng nghĩa với việc nó sẽ có mức ưu tiên cao nhất. Nếu có nhiều thuộc tính cùng có !important thì lại quay về bài toán tính độ ưu tiên.

Ví dụ:

- h2 { color: red!important; }
- h2.title { color: blue; }
- div h2.title { color: orange; }
- .head h2.title {color: green; }
- div.head h2.title {color: gray; }
- #main div.head h2.title {color: pink; }

05. Math Functions (Hàm toán học)

Các hàm toán học CSS cho phép các biểu thức toán học được sử dụng làm giá trị thuộc tính.

Các hàm toán học:

Hàm calc()

- Thực hiện một phép tính, kết quả sẽ lấy làm giá trị thuộc tính. Các toán tử sau có thể được sử dụng: + - * /
- Ví dụ: width: calc(100% - 100px);

Hàm max()

- Sử dụng giá trị lớn nhất, từ danh sách giá trị được phân tách bằng dấu phẩy, làm giá trị thuộc tính.
- Cú pháp: max(value1, value2, ...)
- Ví dụ: width: max(50%, 300px);

05. Math Functions (Hàm toán học)

Các hàm toán học CSS cho phép các biểu thức toán học được sử dụng làm giá trị thuộc tính.

Các hàm toán học:

Hàm min()

- Sử dụng giá trị nhỏ nhất, từ danh sách giá trị được phân tách bằng dấu phẩy, làm giá trị thuộc tính.
- Cú pháp: min(value1, value2, ...)
- Ví dụ: width: min(50%, 300px);

06. CSS function attr()

attr () là một hàm CSS trả về giá trị của một thuộc tính

Ví dụ:

```
<style>
[data-text]::before {
content: attr(data-text);
}
</style>
```

```
<span data-text="Xin chào ">Đặng Phương Nam</span>
```

ĐẠI NAM UNIVERSITY

07. Variables (Biến)

Mục đích của khai báo biến là để có thể sử dụng được ở nhiều nơi.

Kiểu global (toàn cục): có thể sử dụng được nhiều nơi trong file css.

Kiểu local (cục bộ): chỉ sử dụng được ở trong phạm vi của nó, những đoạn css khác không sử dụng được.

07. Variables (Biến)

Cách tạo biến global:

```
:root {
--ten-bien: giá trị;
}
```

Cách sử dụng biến:

```
var(--ten-bien);
```

Ví dụ:

```
:root {
--gray: #333333;
--white: #ffffff;
}
body {
color: var(--gray);
background-color: var(--white);
}
```

07. Variables (Biến)

Cách tạo biến local:

```
selector {
    --ten-bien: giá trị;
}
```

Cách sử dụng biến:

```
selector {
    --ten-bien: giá trị;
    var(--ten-bien);
}
```

Ví dụ:

```
h1 {
    --local-color: blue;
    color: var(--local-color);
}
p {
    color: var(--local-color);
}
```

08. Box Sizing (Kích thước hộp)

Mặc định khi ta sử dụng thuộc tính width, height thì chỉ là áp dụng cho phần content của 1 element.

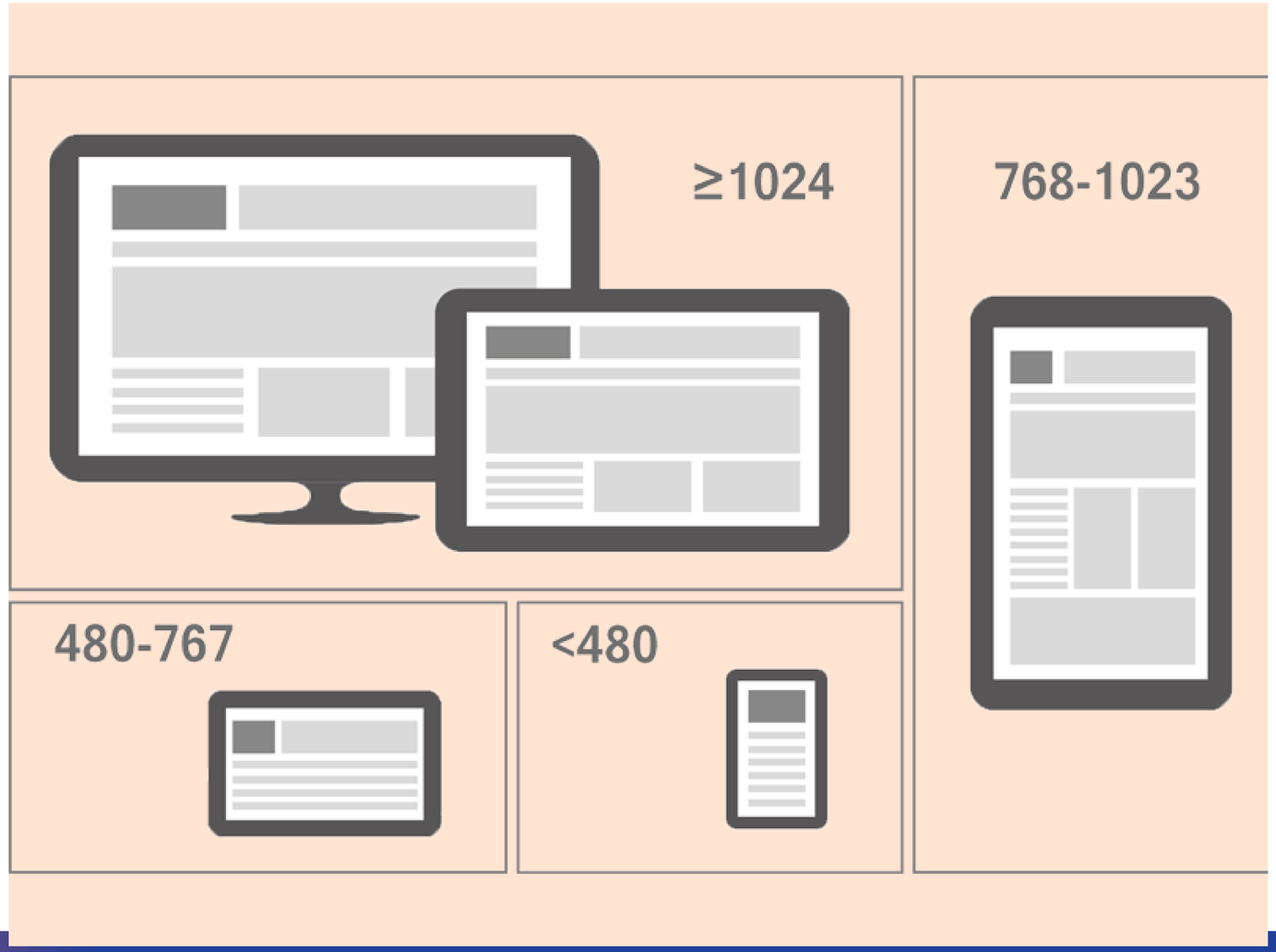
2 loại box-sizing:

- content-box:
  - Mặc định. Width/height chỉ bao gồm "nội dung" của phần tử.
  - width, height = content
- border-box:
  - Width/height bao gồm: content, padding, border của phần tử.
  - width, height = content + padding + border.

09. Khái niệm Responsive

Responsive là để chỉ một website có thể hiển thị tương thích trên mọi thiết bị, như máy tính, máy tính bảng, điện thoại.

Sử dụng media query để responsive giao diện.



09. Khái niệm Responsive

Khai báo meta viewport:

- <meta name="viewport" content="width=device-width, initial-scale=1.0">
- Trong đó:
  - viewport: Giúp trình duyệt hiểu thẻ meta này dùng để thiết lập khung nhìn.
  - width=device-width: Đặt chiều rộng bằng chiều rộng thiết bị (device-width).
  - initial-scale=1.0: Mức thu phóng của website. initial-scale thường được đặt bằng 1 (có thể tăng giá trị lên nhưng không được khuyến nghị).

10. Media Queries

@media CSS là một tính năng mới của CSS3.

Tính năng này cho phép ta tùy chỉnh CSS cho nhiều thiết bị khác nhau từ máy tính cho đến ipad, điện thoại và các thiết bị in ấn.

Cú pháp:

```
@media mediaType and (mediaFeature) {
    // Code
}
```

Trong đó:

- mediaType gồm các thuộc tính hay sử dụng sau:
  - all: Dùng cho mọi thiết bị
  - print: Dùng cho máy in
  - screen: Dùng cho máy tính và các thiết bị di động
- mediaFeature gồm các thuộc tính hay sử dụng sau:
  - max-width: Chiều rộng tối đa của viewport
  - min-width: Chiều rộng tối thiểu của viewport

10. Media Queries

PC first (Responsive):

- PC first là khái niệm để chỉ việc lập trình giao diện từ màn hình to đến màn hình nhỏ.
- Để làm việc với mô hình này chúng ta sử dụng max-width trong media query.
- Dưới đây là các media query điển hình:

Responsive Design

```
<style>
    /*Ipad ngang(1024 x 768)*/
@media screen and (max-width: 1024px) {
    /* Code */
}
    /*Ipad dọc(768 x 1024)*/
@media screen and (max-width: 768px) {
    /* Code */
}
    /* Smart phone (480 x 640)*/
@media screen and (max-width: 480px) {
    /* Code */
}
    /* Smart phone nhỏ */
@media screen and (max-width: 320px) {
    /* Code */
}
</style>
```

10. Media Queries

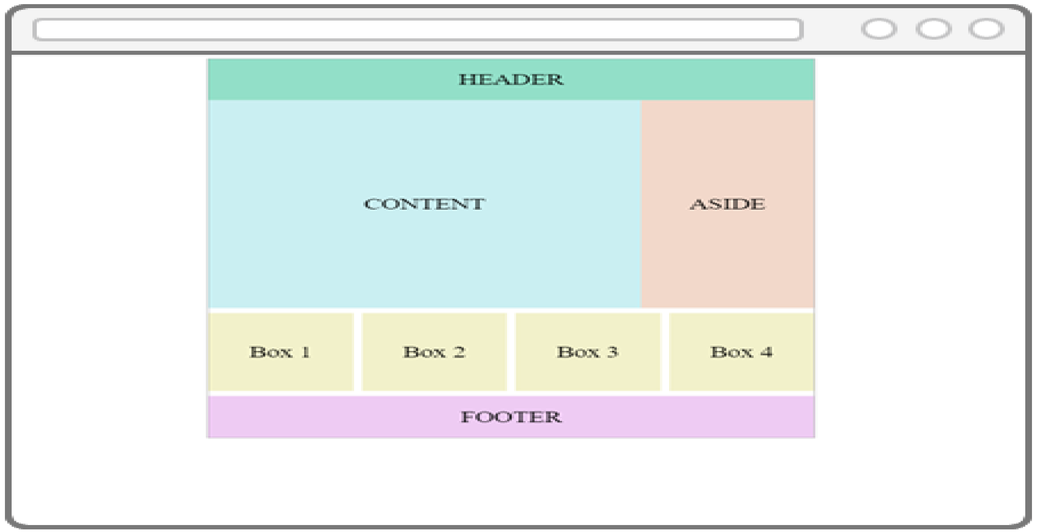
Mobile first:

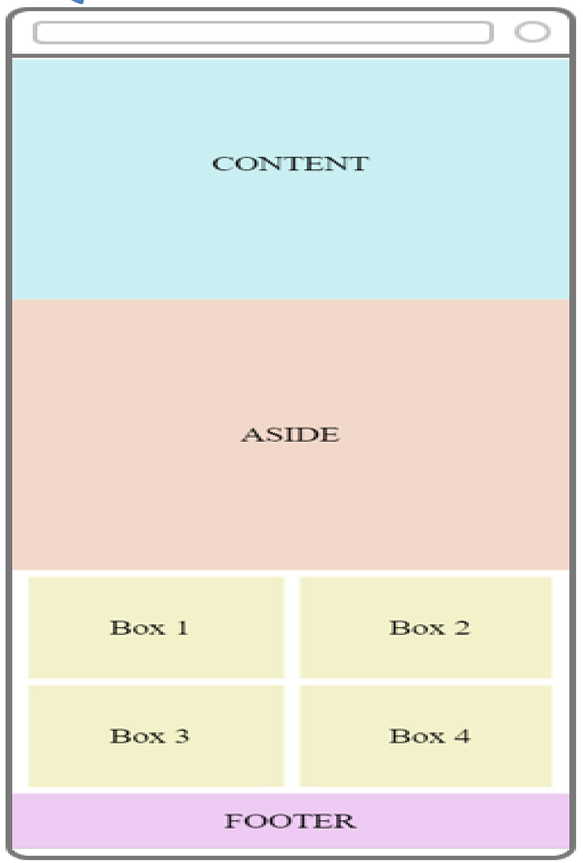
- Mobile first là khái niệm để chỉ việc lập trình giao diện từ màn hình nhỏ đến màn hình to.
- Để làm việc với mô hình này chúng ta sử dụng min-width trong media query.
- Dưới đây là các media query điển hình:

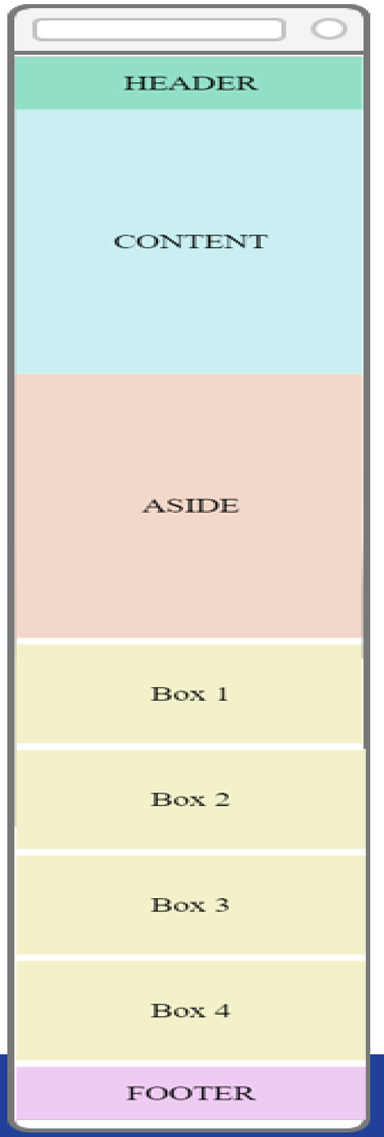
Mobile-First Design

```
<style>
    /* Smart phone nhỏ */
@media screen and (min-width: 320px) {
    /* Code */
}
    /* Smart phone nhỏ(480 x 640) */
@media screen and (min-width: 480px) {
    /* Code */
}
    /* Ipad dọc(768 x 1024) */
@media screen and (min-width: 768px) {
    /* Code */
}
    /* Ipad ngang(1024 x 768) */
@media screen and (min-width: 1024px) {
    /* Code */
}
</style>
```

10. Media Queries







HieuTC
WEB | FRONTEND | BACKEND
THÚY LOI UNIVERSITY