(Ngày 15/10/2025)

# CSS (10) - Định dạng màu

## Bài tập và thực hành

Bài tập 3.1a. Viết lại các đoạn mã trong phần lý thuyết.

### **Xác định màu**

### Có hai cách phổ biến để xác định một màu, một là dùng các tên màu định sẵn và hai là dùng giá trị số (mỗi màu có một giá trị số tương ứng).

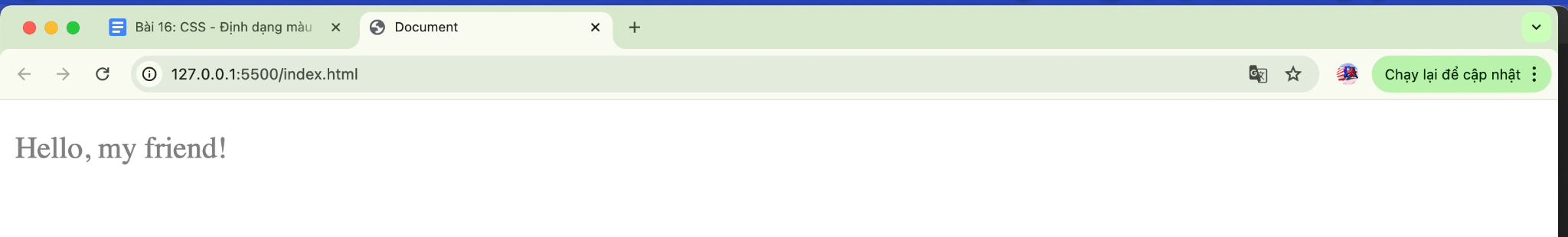
* **Tên màu**

Cách trực quan nhất để xác định một màu là dùng tên của nó, sử dụng các tên màu đã được đặc tả sẵn trong CSS. CSS3 hỗ trợ 140 tên màu.

Ví dụ:

p { color: gray; } /\* văn bản trong p sẽ có màu xám \*/

[Kết quả]



**Giá trị số**

Xác định màu bằng tên tuy đơn giản nhưng lại bị giới hạn về số lượng. Vì vậy, mọi người hay sử dụng giá trị số để xác định một màu, đó là giá trị RGB. Với RGB, chúng ta có thể xác định hàng triệu màu khác nhau.

* ***Hệ màu RGB***

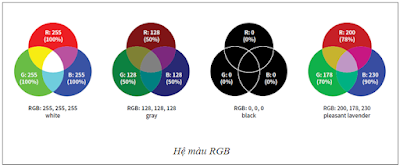
Máy tính tạo ra các màu khác nhau bằng cách kết hợp ba màu cơ bản (với tỉ lệ nhất định) là: đỏ (red), xanh lá cây (green) và xanh da trời (blue). Tên gọi RGB là viết tắt của ba chữ Red, Green và Blue.

Mỗi chỉ số trong tổ hợp RGB có giá trị nằm trong khoảng từ 0 tới 255. Nếu ba chỉ số có giá trị là 0, chúng ta sẽ có màu đen (black). Nếu ba chỉ số đều có giá trị là 255, chúng ta sẽ có màu trắng (white).

Như vậy, mọi điểm màu trên màn hình đều được mô tả bằng bộ ba giá trị RGB

Ví dụ: màu lavender sẽ có bộ ba giá trị là (200, 178, 230).

Xem hình minh họa về hệ màu RGB,



* ***Lấy màu***

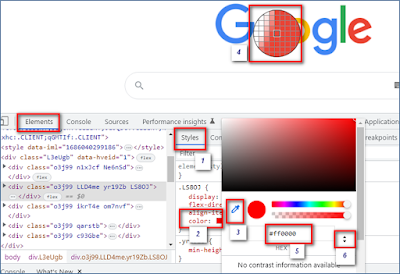
[Cách 1] Cách dễ nhất để lấy được giá trị RGB của một màu là dùng công cụ có tên là Color Picker. Bạn hãy vào trang Google, tìm kiếm và học cách sử dụng công cụ này.

[Cách 2] Cách khác để lấy giá trị màu là dùng Developer tools.

Nếu chúng ta muốn lấy một màu có sẵn trên một trang web nào đó (sao chép màu), ví dụ lấy màu của các chữ cái google, thì làm như sau:

– Mở trang google.com

– Mở công cụ Developer tools của trình duyệt



– Trong tab Elements/Styles [1], nhập một chỉ dẫn CSS có thuộc tính color, giá trị là một màu bất kì, ví dụ *color: red [2],* bấm chuột trái vào ô vuông màu đỏ sau chữ *color*

– Bấm chuột vào công cụ soi màu (eye droper) [3]

– Di chuyển công cụ soi màu vào vùng màu cần lấy [4], bấm chuột trái để lấy màu

– Lấy giá trị màu dạng HEX [5], dạng RGB [6]

Ngoài ra, chúng ta có thể sử dụng các công cụ xử lý ảnh để lấy màu, như Photoshop, Illustrator.

### **Cách viết giá trị màu RGB**

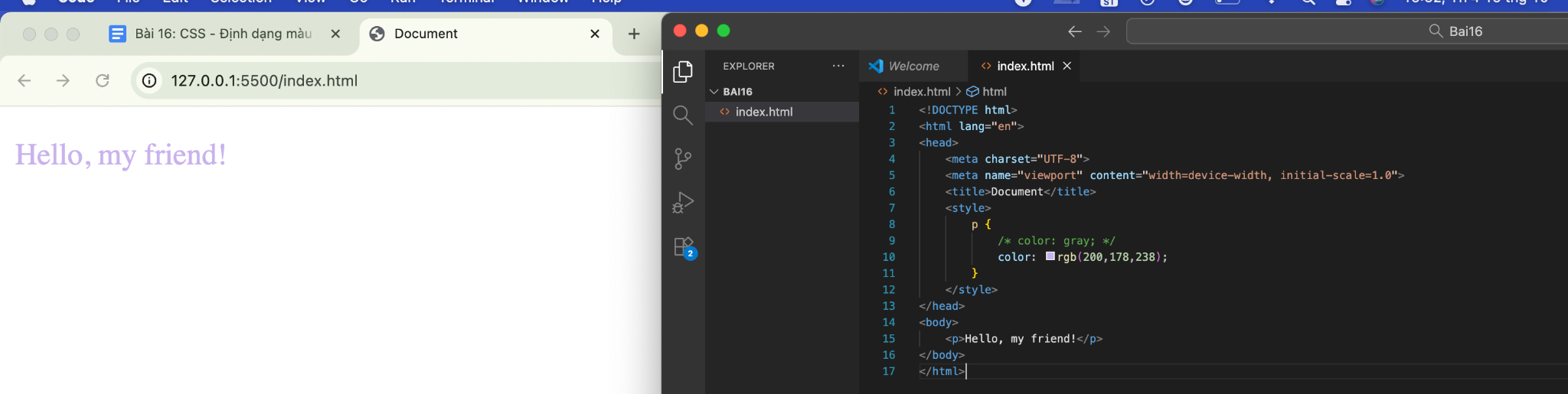
Có ba cách để viết giá trị màu RGB trong CSS, một là dùng giá trị hệ 10, hai là dùng %, ba là dùng giá trị hệ 16.

* ***Dùng giá trị hệ 10***

Bắt đầu là kí hiệu *rgb*, sau đó là ba giá trị hệ 10 của ba màu cơ bản (RGB), mỗi giá trị nằm trong khoảng từ 0 tới 255, ví dụ,

color: rgb(200, 178, 230);

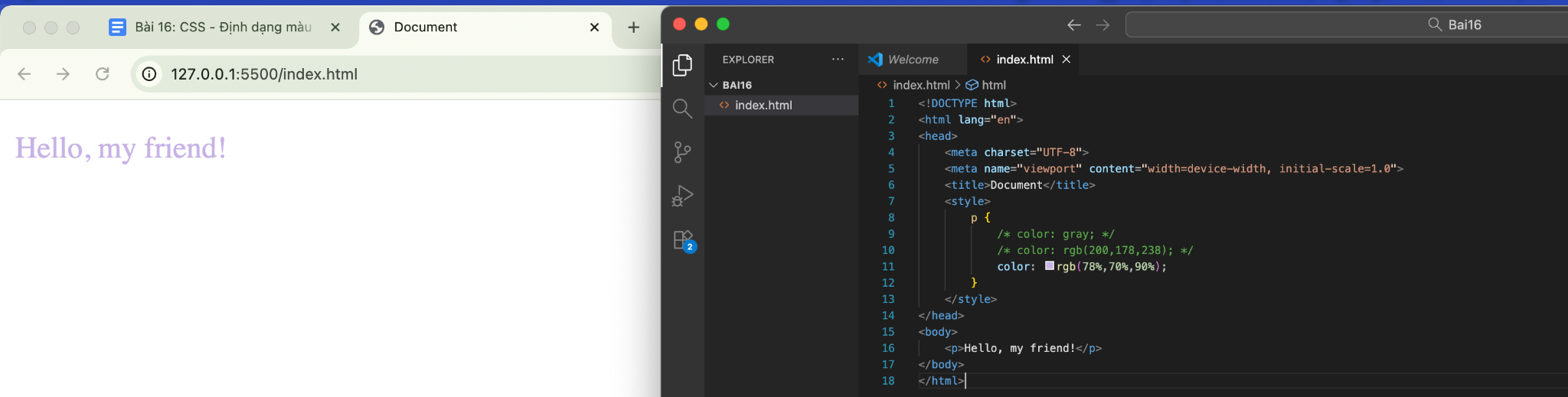
[Kết quả]



* ***Dùng giá trị %,***

Bắt đầu là kí hiệu *rgb*, sau đó là ba giá trị % của ba màu cơ bản, giá trị % được tính bằng cách lấy “giá trị hệ 10”/255\*100%, ví dụ,

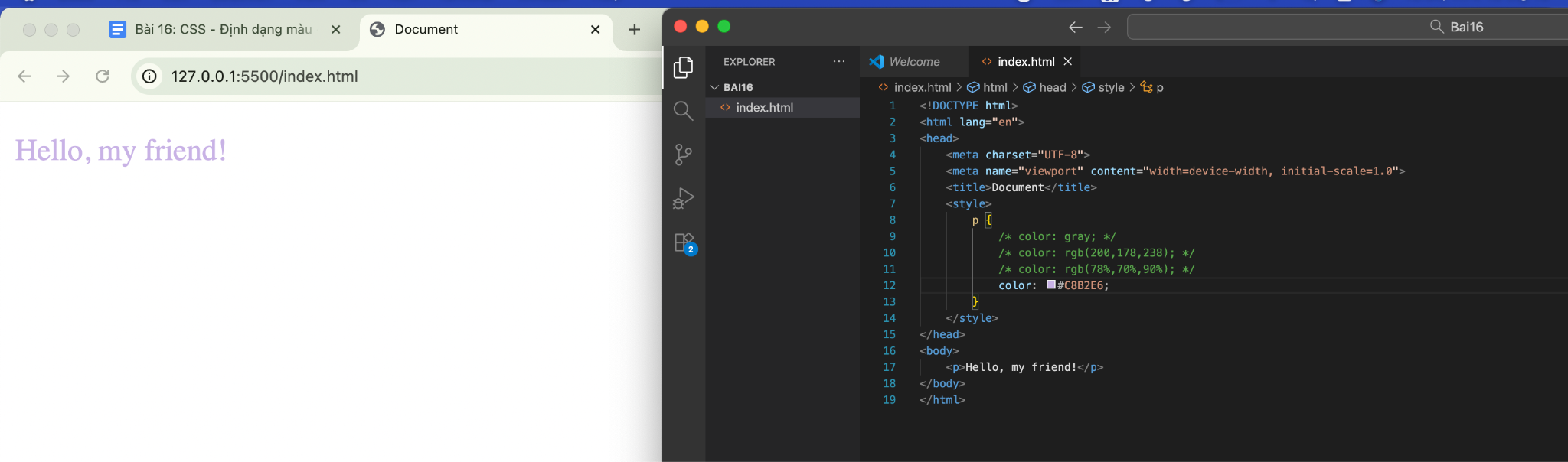
color: rgb(78%, 70%, 90%);

[Kết quả]

* ***Dùng giá trị hệ 16***

Bắt đầu là kí hiệu #, sau đó là ba giá trị hệ 16 của ba màu cơ bản, mỗi giá trị màu được biểu diễn bằng hai kí số hệ 16, ví dụ,

color: #C8B2E6;

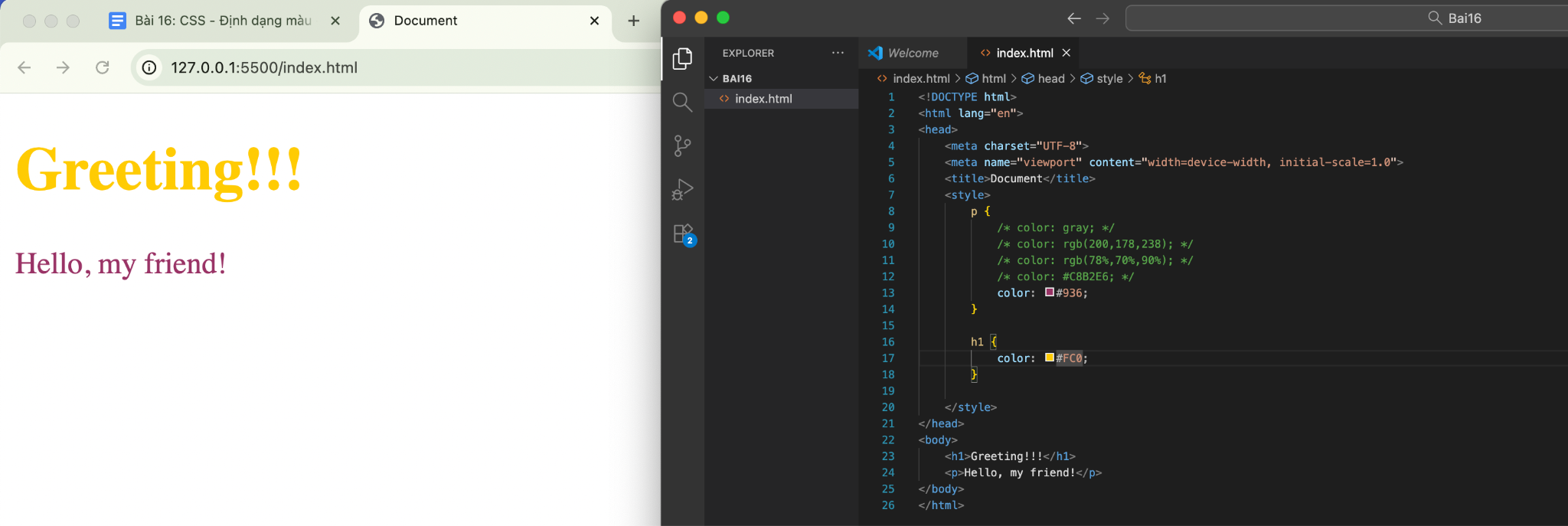
[Kết quả]

Trong trường hợp giá trị của mỗi màu cơ bản là hai kí số giống nhau thì có thể viết rút gọn, ví dụ,

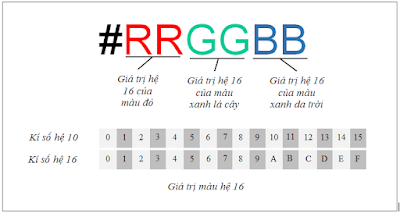
color: #FFCC00 hoặc color: #993366

thì có thể viết rút gọn thành,

color: #FC0 hoặc color: #936

[Kết quả]

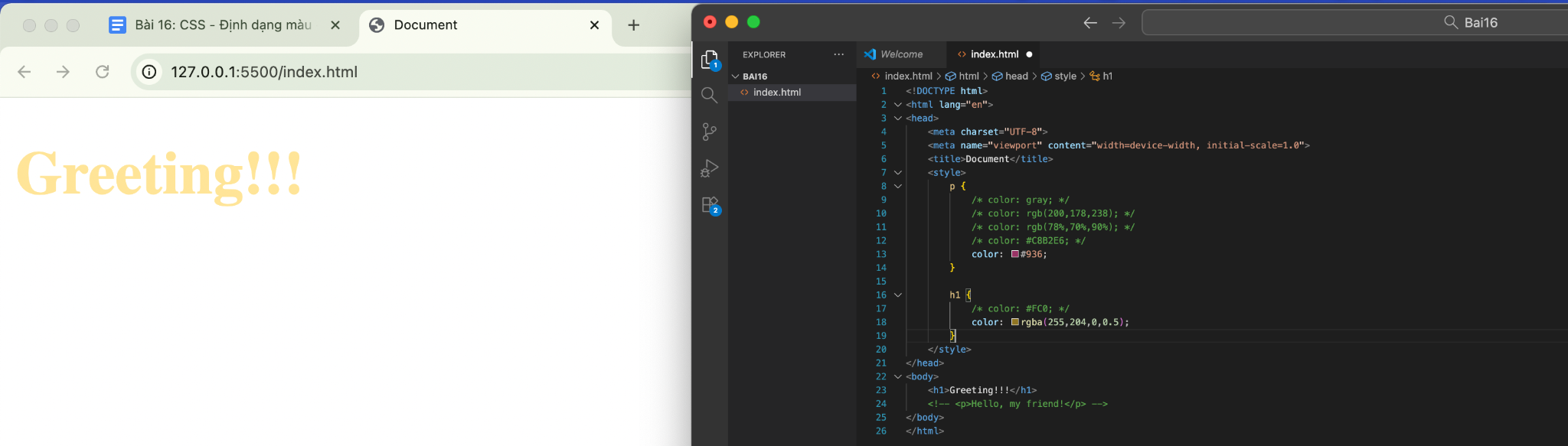
Xem hình minh họa về giá trị màu hệ 16,



* **Màu RGBa**

Màu *RGBa* cho phép xác lập một màu bất kì cùng với độ mờ của màu. Chữ *a* là viết tắt của *alpha*, được sử dụng để thiết lập độ mờ của màu, nếu *a* có giá trị 0: tương ứng với màu trong suốt hay mờ nhiều nhất (transparent), nếu *a* có giá trị 1: tương ứng với màu ít mờ nhất hay đậm đặc nhất (opaque).

Ví dụ:



Ví dụ, xem hình minh họa,



### **Màu chữ & màu nền**

**Màu chữ**

Chúng ta có thể thiết lập màu chữ (foreground color) cho mọi phần tử HTML. Khi thiết lập màu chữ, màu đường viền (border) cũng được thiết lập. Để thiết lập màu chữ, sử dụng thuộc tính *color*.

Thuộc tính co*lor*:

– Giá trị: *giá trị màu (tên hoặc số)*

– Mặc định: *tùy trình duyệt và thiết lập của người dùng*

– Áp dụng: *cho mọi phần tử*

– Kế thừa: *có*

Ví dụ, chỉ dẫn CSS sau sẽ thiết lập màu chữ xanh nhạt cho phần tử *blockquote*, với các giá trị R:80, G:140 và B:25, đổi giá trị này sang hệ 16 sẽ là #508C19. Hai phần tử *p* và *em* là hậu duệ của phần tử *blockquote* nên cũng được kế thừa màu vừa thiết lập.

[HTML]

<blockquote>

<p>Đây là ví dụ về màu chữ <em>(foreground color)</em></p>

</blockquote>

[CSS]

blockquote {

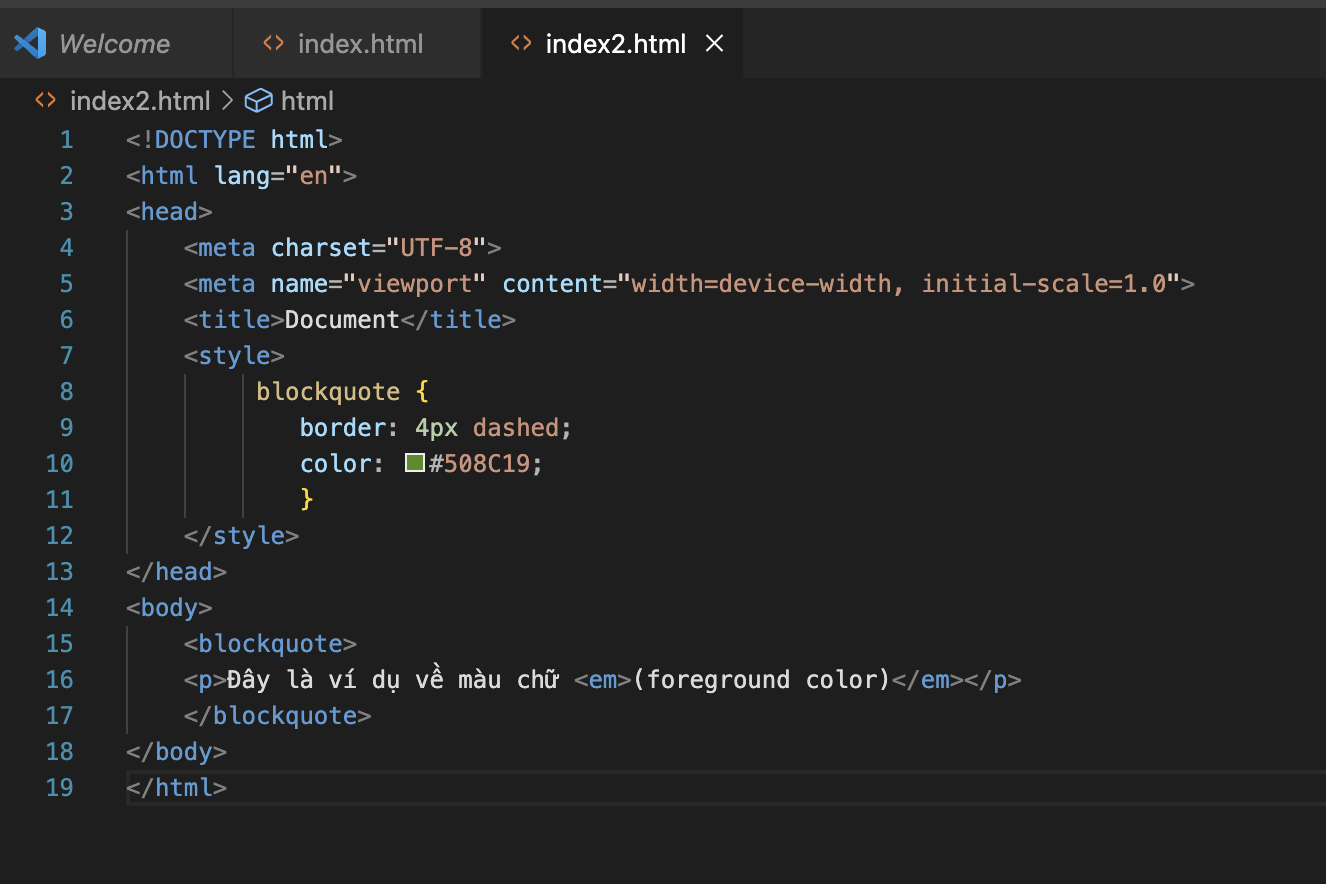
border: 4px dashed;

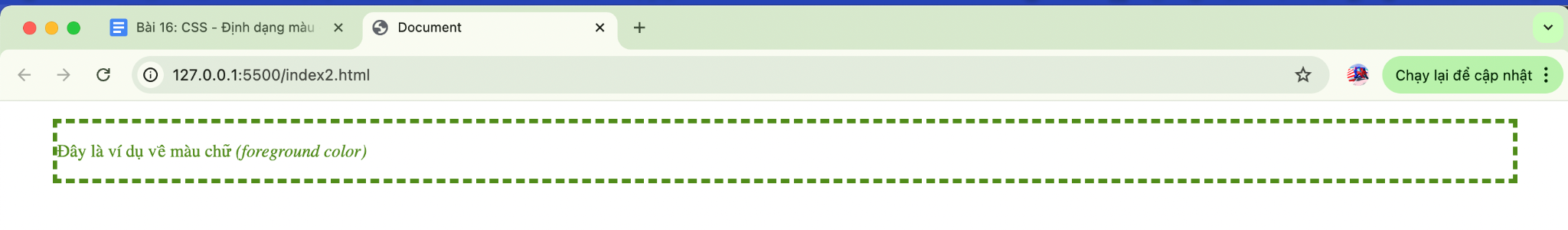
color: #508C19;

}

[Kết quả]







Nếu thuộc tính *border-color* cũng được thiết lập thì nó sẽ thay thế màu của đường viền đang được thiết lập bằng thuộc tính *color*. Nghĩa là thuộc tính *border-color* có độ ưu tiên cao hơn thuộc tính *color*.

**Màu nền**

Sử dụng thuộc tính *background-color* để thiết lập màu nền cho các phần tử HTML.

Thuộc tính *background-color*:

– Giá trị: *giá trị màu (tên hoặc số) | transparent*

– Mặc định: *transparent (trong suốt)*

– Áp dụng: *cho mọi phần tử*

– Kế thừa: *không*

Khung nền (background, hoặc canvas) bao gồm: vùng chứa nội dung, khoảng đệm phía trong đường viền (padding), nền chứa đường viền, không bao gồm vùng đệm phía ngoài đường viền (margin).

Mặc định, thuộc tính *background-color* sẽ đổ màu cho toàn bộ phần khung nền, hay được gọi tắt là nền.

Ví dụ, định dạng màu nền cho phần tử *blockquote*,

blockquote {

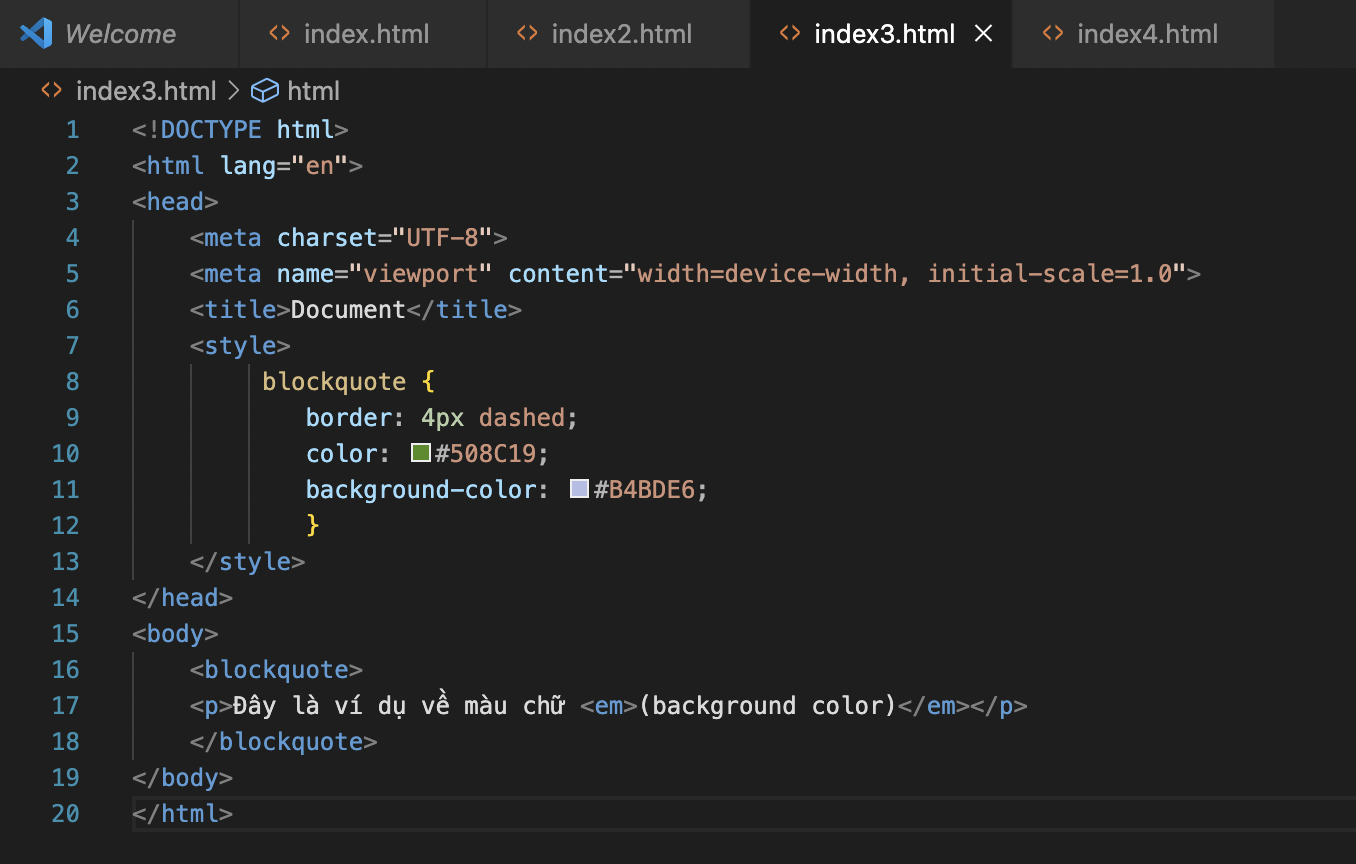
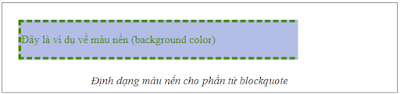
border: 4px dashed;

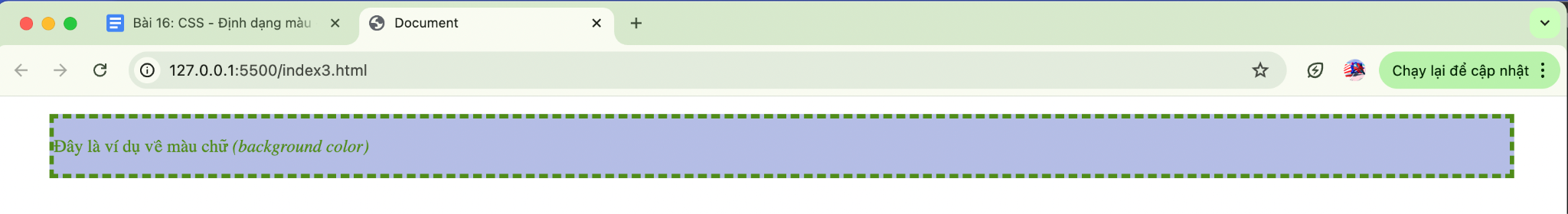
color: #508C19;

background-color: #B4BDE6;

}

[Kết quả]





Mặc dù màu nền không được kế thừa xuống các hậu duệ, tuy nhiên, do màu nền mặc định của các phần tử là trong suốt (transparent), nên màu nền của phần-tử-cha cũng được xem như là màu nền của các phần tử hậu duệ. Ví dụ, chúng ta có thể thiết lập màu nền cho toàn bộ trang web bằng cách thiết lập màu nền cho phần tử *body*. Sau đó, chúng ta cũng có thể thay đổi màu nền cho từng phần tử trong một trang web, dù là phần tử kiểu block hay kiểu inline.

**Giới hạn vùng nền**

Như đã biết, vùng nền (background) mặc định của một phần tử sẽ được tính từ viền ngoài của đường biên (border). Tuy nhiên, chúng ta có thể sử dụng thuộc tính *background-clip* để giới hạn vùng nền theo các tiêu chí khác nhau.

Thuộc tính *background-clip*:

– Giá trị: *border-box | padding-box | content-box*

– Mặc định: *border-box*

– Áp dụng: *cho mọi phần tử*

– Kế thừa: *không*

Ngoài giá trị mặc định, nếu giá trị là *padding-box*, vùng nền được tính từ viền ngoài của vùng padding. Nếu giá trị là *content-box*, vùng nền được tính từ viền ngoài của vùng nội dung.

Ví dụ:

[HTML]

<blockquote>

<p>Đây là ví dụ về background-clip</p>

</blockquote>

[CSS]

blockquote {

border: 4px dashed;

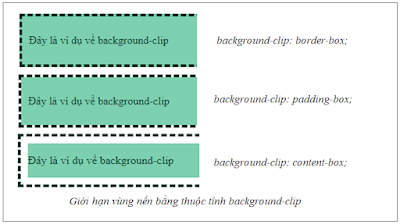
padding: 10px;

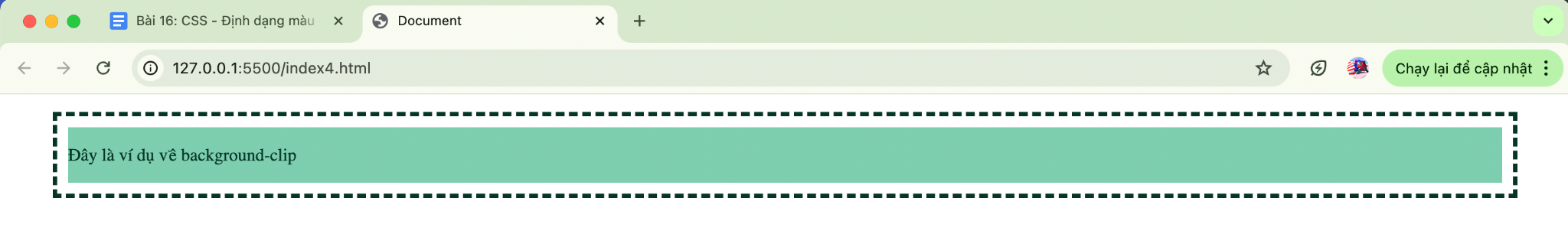
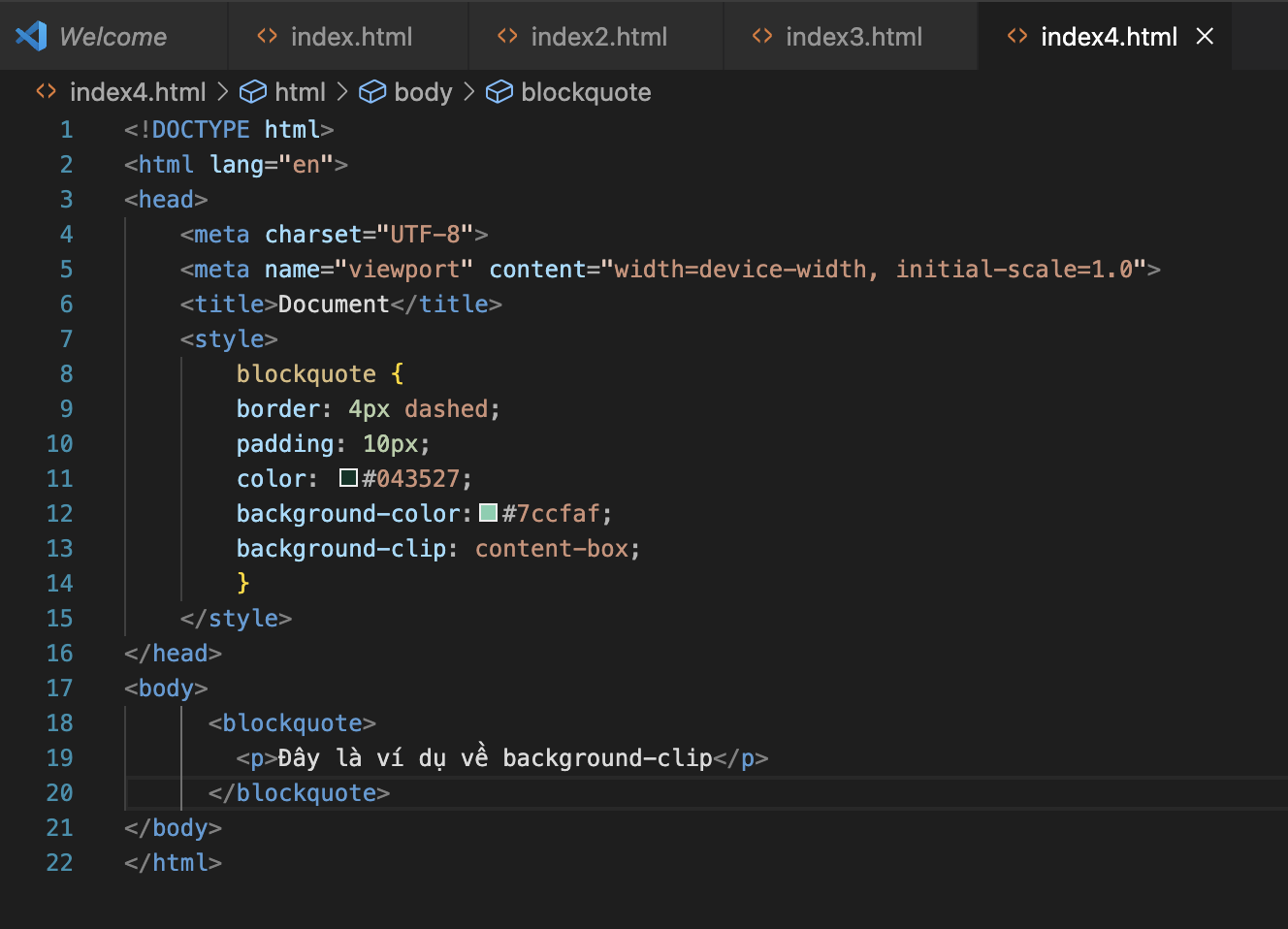
color: #043527;

background-color:#7ccfaf;

background-clip: content-box;

}





### **Làm mờ một phần tử**

Ở phần trước, chúng ta đã tìm hiểu cách làm mờ một phần tử bằng định dạng màu *RGBa*. Phần này giới thiệu một cách khác để làm mờ phần tử bằng thuộc tính *opacity*. Opacity có nghĩa là độ mờ.

Thuộc tính *opacity:*

– Giá trị: *giá trị số (trong khoảng từ 0 đến 1)*

– Mặc định: *1*

– Áp dụng: *cho mọi phần tử*

– Kế thừa: *không*

Giá trị của thuộc tính *opacity* càng nhỏ thì phần tử càng mờ, giá trị 0 là mờ nhất, giá trị 1 là rõ nhất.

Ví dụ,

body {

background-color: #333333;

}

h1 {

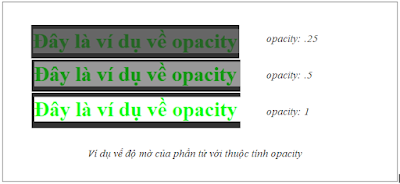
color: #00FF00;

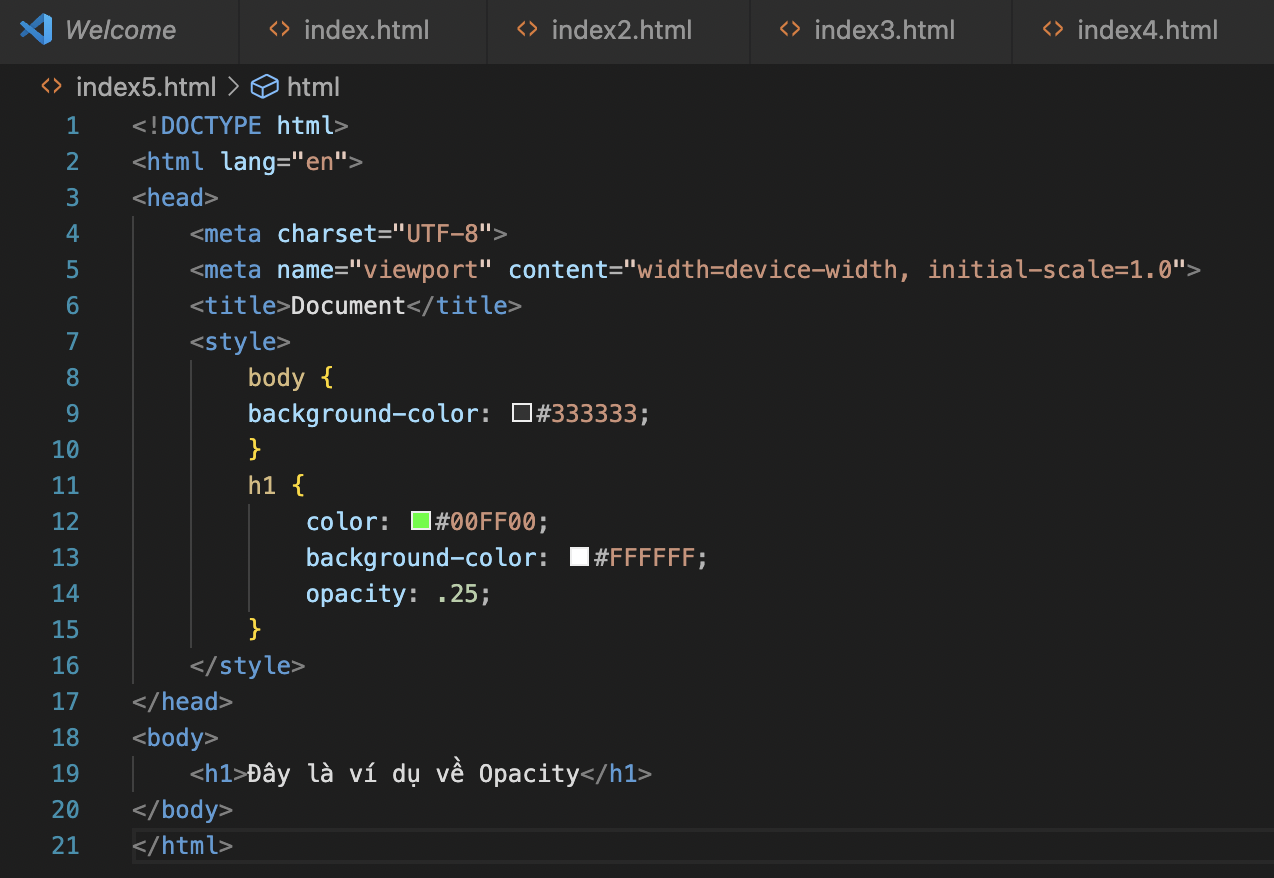
background-color: #FFFFFF;

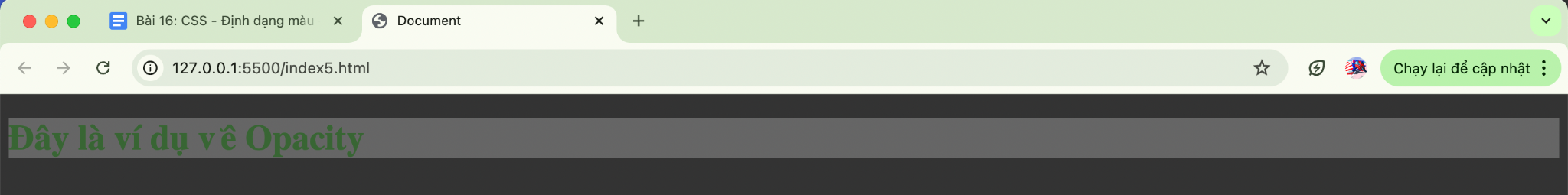
opacity: .25;

}

Hình dưới đây là một số ví dụ sử dụng thuộc tính *opacity*,







Bài tập 3.1b. Định dạng màu cho chữ Google, giống như trong trang google.com

[Gợi ý làm bài tập]

Bài tập 3.1b.

– Tạo thư mục dự án, tên Google, tạo tập tin index.html

– Kiểm tra trên trang google.com, thấy chữ Google là một ảnh, vì vậy chúng ta không thể xem được mã nguồn HTML và CSS để bắt chước theo. Vậy là phải tự sáng tạo ra cách làm.

– Sử dụng phần tử *p* để tạo chữ Google, vì các chữ có màu khác nhau nên sẽ sử dụng phần tử *span* để tách từng chữ, gán cho mỗi màu một class.

[index.html]

<p><span *class*="blue">G</span>

<span *class*="red">o</span>

<span *class*="yellow">o</span>

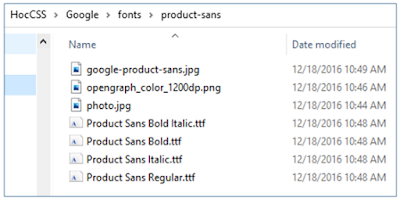
<span *class*="blue">g</span>

<span *class*="green">l</span>

<span *class*="red">e</span></p>

[CSS]

– Chọn phông chữ, tìm trên mạng biết được phông chữ mà google đang dùng là Product Sans, tải phông chữ này về máy. Ví dụ, tải từ đây: <https://www.dafontfree.io/google-logo-font/>. Giải nén, lưu vào trong thư mục fonts của dự án, xem hình:



– Khai báo để dùng được phông chữ Product Sans

@font-face {

font-family: "Product Sans";

src: **url**('fonts/product-sans/Product Sans Bold.ttf');

}

p {

font-family: "Product Sans", sans-serif;

}

– Thiết lập font-size cho chữ Google, có thể đặt chữ Google do chúng ta tạo ra bên cạnh chữ Google mẫu, khi nào kích thước chúng bằng nhau thì đó là font-size. Hoặc chuyên nghiệp hơn thì dùng các phần mềm xử lý ảnh để đo, ví dụ dùng photoshop online: <https://photoshoponlinemienphi.com/>.

font-size: 90px;

– Thu nhỏ khoảng cách giữa các kí tự, căn giữa dòng chữ

letter-spacing: -11px;

text-align: center;

– Dùng công cụ Developer tools để lấy màu của mỗi kí tự, viết mã CSS cho mỗi class.

Mã nguồn tham khảo:

[index.html]

<!DOCTYPE *html* >

<html>

<head>

<meta *charset*="utf-8">

<title>Google</title>

<style>

@font-face {

font-family: "Product Sans";

src: **url**('fonts/product-sans/Product Sans Bold.ttf');

}

p {

font-family: "Product Sans", sans-serif;

font-size: 90px;

letter-spacing: -11px;

text-align: center;

}

.blue { color: #4285f4; }

.red { color: #ea4335; }

.yellow { color: #fbbc05; }

.green { color: #34a853; }

</style>

</head>

<body>

<p><span *class*="blue">G</span>

<span *class*="red">o</span>

<span *class*="yellow">o</span>

<span *class*="blue">g</span>

<span *class*="green">l</span>

<span *class*="red">e</span></p>

</body>

</html>



[Kết quả]



## **Câu hỏi ôn tập**

Câu 1. Phần tử p sẽ có chữ màu gì, với chỉ dẫn CSS sau: p { color: rgb(255, 0, 0); }

A. Xanh lá cây (green)

B. Xanh nước biển (blue)

C. Trắng

**D. Đỏ**

Câu 2. Chỉ dẫn CSS nào sau đây hợp lệ?

A. p { color: #4455GG; }

**B. p { color: rgba(255, 0, 255, .5); }**

C. p { color: rgb(255, 0, 256, .5); }

D. p { color: rgb(255, 0, 256); }

Câu 3. Chỉ dẫn CSS nào sau đây không hợp lệ?

A. p { color:#4455FF; }

**B. p { color: rgba(255, 0, 255, 2); }**

C. p { color: rgba(255, 0, 254, .5); }

D. p { color: rgb(0, 0, 0); }

Câu 4. Muốn giới hạn vùng nền chỉ gồm phần nội dung của phần tử, cần thiết lập giá trị cho thuộc tính background-clip là gì?

A. border-box

B. none

**C. content-box**

D. padding-box