

Buổi 3

Tổng quan về CSS

Định dạng văn bản và đối tượng

I. Các khái niệm về CSS

1. Nhược điểm của việc định dạng bằng HTML

Việc định dạng cho trang web bằng các thẻ HTML gặp phải một số bất lợi sau:

- Không thể tái sử dụng, dù là trong cùng 1 trang web hay giữa nhiều trang web với nhau.

Ví dụ: Có đoạn văn bản sau:

Hôm nay trời đẹp quá

Kìa có một con cá

Miệng đang ngậm chiếc lá

Ôi bài thơ hay quá

Để định dạng đoạn văn bản trên, cần phải viết mã nguồn HTML như sau:

```
<font color="red">Hôm nay trời đẹp quá</font>
```

```
<font color="blue">Kìa có một con cá</font>
```

```
<font color="red">Miệng đang ngậm chiếc lá</font>
```

```
<font color="blue">Ôi bài thơ hay quá</font>
```

➔ Việc định dạng phải được viết đi viết lại nhiều lần mặc dù định dạng không khác nhau.

- Số lượng các tính chất có thể được định dạng còn khá hạn chế.
- Một khi đã định dạng, việc thay đổi định dạng khác rất khó khăn, hoặc bất khả thi.

Ví dụ: Xem ví dụ 3.1. Việc thay đổi định dạng chỉ bằng HTML là không thể.

Ngoài ra, về mặt bản chất, HTML chỉ nên dùng để mô tả cấu trúc các thành phần của trang web chứ không phải dùng để định dạng.

Do đó, tổ chức W3C¹ đã tạo ra ngôn ngữ CSS, đảm nhiệm phần định dạng cho trang web.

2. Tổng quan về CSS

CSS (*Cascading Style Sheet*) là 1 ngôn ngữ dùng để định dạng cho các thành phần HTML trên trang web.

Cú pháp:

```
[selector] {  
    [att1]: [value1];  
    [att2]: [value2];  
    ...  
}
```

Trong đó:

- [selector] là thẻ HTML muốn định dạng (có thể kèm theo cả id và class).
- att1, att2... là tên các thuộc tính CSS.
- value1, value2... là tên các giá trị tương ứng.

¹ World Wide Web Consortium

Ví dụ: Đoạn mã CSS định dạng chữ màu đỏ và canh giữa cho tất cả thẻ <p>:

```
p {  
    color: red;  
    text-align: center;  
}
```

Có 3 cách để cài đặt CSS cho 1 file HTML:

- External CSS: Mã nguồn CSS được lưu trong 1 file có đuôi .css và được chèn vào trang web thông qua thẻ <link> nằm trong thẻ <head>:

```
<link rel="stylesheet" href="css/style.css">
```

- Internal CSS: Mã nguồn CSS được đặt trong thẻ <style> nằm trong thẻ <head>:

```
<style>  
    p {  
        color: red;  
        text-align: center;  
    }  
</style>
```

- Inline CSS: Mã nguồn CSS được đặt trong thuộc tính style của thẻ HTML:

```
<p style="color: red; text-align: center;">Hello World</p>
```

Lưu ý: Xét về độ ưu tiên thì External CSS < Internal CSS < Inline CSS.

3. HTML id và class

Tất cả thẻ HTML đều cung cấp 2 thuộc tính id và class để định danh thẻ đó, hỗ trợ cho việc cài đặt CSS cũng như JavaScript sau này.

- ID dùng để định danh cho 1 thẻ, do đó ID không được trùng nhau, và mỗi thẻ chỉ có tối đa 1 ID.
- Class dùng để phân loại cho 1 thẻ, do đó class có thể trùng nhau, và mỗi thẻ có thể thuộc nhiều class.

ID và class do lập trình viên tự đặt theo quy tắc: có phân biệt chữ hoa và chữ thường, có ít nhất 1 chữ cái, và không bao gồm khoảng trắng. Một thẻ có thể có cả ID và class.

Ví dụ:

```
<p id="loiChao"></p>  
<h1 class="ThongBao"></h1>  
<a id="lnkIndex" class="link homepage"></a>
```

II. CSS Selector

Selector của CSS dùng để chọn ra phần tử cần định dạng. Có thể chia selector thành 5 loại:

- Simple selector: Dựa vào tên thẻ, ID, class.
- Combinator selector: Dựa vào mối quan hệ, cấu trúc phân cấp của thẻ.
- Pseudo-class selector: Dựa vào các trạng thái đặc biệt của thẻ.
- Pseudo-element selector: Dựa vào một thành phần của thẻ.
- Attribute selector: Dựa vào thuộc tính hoặc giá trị thuộc tính của thẻ.

Trong khuôn khổ tài liệu buổi 3, chúng ta chỉ tìm hiểu về 2 loại đầu tiên.

1. Simple selector

Đây là loại selector đơn giản nhất, chỉ dựa vào tên thẻ, ID, class để chọn ra thẻ cần định dạng. Để chọn theo ID, dùng selector #id. Để chọn theo class, dùng selector .class.

Ví dụ: Cho đoạn code HTML sau:

```
<p class="TieuDe">Dòng 1</p>
<p id="p1">Dòng 2</p>
<p class="TieuDe"><span>Dòng 3</span></p>
<p><span class="NoiDung">Dòng 4</span></p>
```

Sau đây là kết quả của một số đoạn CSS:

Code CSS	Kết quả
<pre>p { color: red; }</pre>	Dòng 1 Dòng 2 Dòng 3 Dòng 4
<pre>p.TieuDe { color: red; }</pre>	Dòng 1 Dòng 2 Dòng 3 Dòng 4
<pre>span.NoiDung { color: red; }</pre>	Dòng 1 Dòng 2 Dòng 3 Dòng 4
<pre>#p1, .NoiDung { color: red; }</pre>	Dòng 1 Dòng 2 Dòng 3 Dòng 4

Để chọn tất cả các thẻ, chúng ta dùng selector *.

2. Combinator selector

Đôi khi, thẻ HTML cần định dạng không được cài đặt ID và class khiến cho việc viết selector cho thẻ đó gặp khó khăn. Khi đó, chúng ta có thể viết selector theo cấu trúc phân cấp.

Ví dụ: Cho đoạn code HTML sau:

```
<body>
  <div>
    <span>Xin chào</span>
  </div>
  <span>Đây là tiết học vào CSS</span>
</body>
```

Trong ví dụ trên, nếu muốn cài đặt CSS cho dòng chữ *Xin chào*, chúng ta cần viết selector đến thẻ chứa dòng chữ đó. Nhưng nếu viết như sau thì cả 2 thẻ đều sẽ bị ảnh hưởng:

```
span {
    color: red;
}
```

Do đó, chúng ta viết CSS theo cấu trúc phân cấp như sau:

```
div span {
    color: red;
}
```

Selector `div span` sẽ chọn tất cả thẻ nằm trong thẻ <div> (không nhất thiết phải là thẻ con trực tiếp). Để chọn thẻ con trực tiếp, chúng ta dùng selector `div > span`.

3. Kết hợp nhiều loại selector

Các loại selector có thể kết hợp với nhau để chọn ra phần tử cần định dạng chính xác hơn.

Ví dụ: Cho đoạn code HTML như ví dụ ở mục II.1. Sau đây là kết quả của một số đoạn CSS:

Code CSS	Kết quả
<pre>p.TieuDe span { color: red; }</pre>	Dòng 1 Dòng 2 Dòng 3 Dòng 4
<pre>p span.NoiDung { color: red; }</pre>	Dòng 1 Dòng 2 Dòng 3 Dòng 4

III. Định dạng văn bản

1. Các thuộc tính font

a) `font-family`: Quy định tên font hoặc nhóm font cần dùng. Nên liệt kê nhiều font, cách nhau bằng dấu phẩy và kết thúc bằng 1 nhóm font. Hiện tại có 3 nhóm font chính:

- serif: Gồm 1 số font như Times New Roman, Georgia...
- san-serif: Gồm 1 số font như Calibri, Arial, Verdana...
- monospace: Gồm 1 số font như Courier New, Consolas, Lucida Console...

Ví dụ:

```
p {  
    font-family: "Times New Roman", Georgia, serif;  
}
```

b) `font-style`: Quy định kiểu chữ thường hay in nghiêng, gồm các giá trị: `normal` (mặc định) hoặc `italic`.

Ví dụ:

```
p {  
    font-style: italic;  
}
```

c) `font-variant`: Quy định kiểu chữ thường hay in hoa, gồm các giá trị: `normal` (mặc định) hoặc `small-caps`.

Ví dụ:

```
p {  
    font-variant: small-caps;  
}
```

Sự khác nhau giữa văn bản in hoa bình thường và `small-caps`:

ĐÂY LÀ ĐOẠN VĂN BẢN SMALL-CAPS

ĐÂY LÀ VĂN BẢN IN HOA BẰNG CAPS LOCK

d) `font-weight`: Quy định kiểu chữ thường hay in đậm, gồm các giá trị: `normal` (mặc định), `bold`, `bolder`, `lighter`, `100`, `200`, `300`,... , `900`. Tuy nhiên, trên thực tế đa số các font chỉ hỗ trợ mức `normal` (400) và `bold` (700).

Ví dụ:

```
p {  
    font-weight: bold;  
}
```

e) font-size: Quy định kích thước chữ, giá trị là 1 con số theo kiểu tuyệt đối hoặc tương đối.
Các giá trị kích thước tuyệt đối:

- XX-SMALL
- X-SMALL: tương đương <h6>.
- SMALL: tương đương <h5>.
- MEDIUM: tương đương <h4>.
- LARGE: tương đương <h3>.
- X-LARGE: tương đương <h2>.
- XX-LARGE: tương đương <h1>.
- Dùng theo các đơn vị như pt, pc...

Các giá trị kích thước tương đối: Dùng theo đơn vị px (tùy thuộc màn hình) hoặc % (tùy thuộc phần tử cha).

Ví dụ:

```
p {  
    font-size: 16px;  
}
```

f) font: Là thuộc tính tổng hợp của 5 thuộc tính trên.

Cú pháp:

```
font: [font-style] [font-variant] [font-weight] [font-size] [font-family];
```

Ví dụ:

```
p {  
    font: italic small-caps bold 32px "Times New Roman", Georgia, serif;  
}
```

2. Các thuộc tính khác

CSS còn cung cấp một số thuộc tính khác để định dạng văn bản:

- letter-spacing và word-spacing: Quy định khoảng cách giữa các kí tự và các từ, giá trị là normal (mặc định) hoặc 1 con số. Nếu mang giá trị âm, các kí tự/từ sẽ đè lên nhau.
- line-height: Quy định chiều cao dòng chữ, giá trị là normal (mặc định) hoặc 1 con số.
- text-indent: Quy định thụt đầu dòng, giá trị là 0 (mặc định) hoặc 1 con số.
- text-align: Quy định canh lề, giá trị là left (mặc định), right, center hoặc justify.
- text-decoration: Quy định kiểu trang trí chữ, giá trị là none (mặc định), underline, overline hoặc line-through.
- text-transform: Quy định kiểu chữ là in hoa hay in thường, giá trị là none (mặc định), capitalize, uppercase hoặc lowercase.

IV. Định dạng nền

CSS cung cấp các thuộc tính sau để định dạng nền cho đối tượng:

- a) background-color: Quy định màu nền cho đối tượng. Màu trong CSS được quy định tương tự như màu trong HTML²: tên màu, mã RGB, mã HEX.

Ví dụ:

```
div {  
    background-color: red;  
}
```

- b) background-image: Quy định ảnh nền cho đối tượng. **Khuyến cáo** chỉ sử dụng ảnh nền không ảnh hưởng đến việc đọc văn bản, nội dung trên trang web.

Ví dụ:

```
div {  
    background-image: url("img/bg-image.jpg");  
}
```

Lưu ý: Khi có cả màu nền và ảnh nền, trình duyệt web sẽ ưu tiên sử dụng ảnh nền.

- c) background-repeat: Quy định cách lặp lại ảnh nền (với trường hợp ảnh nền nhỏ hơn kích thước trang web), gồm các giá trị: repeat (mặc định), repeat-x, repeat-y và no-repeat.

Ví dụ: Xem ví dụ 3.2.

- d) background-attachment: Quy định ảnh nền có cuộn theo trang web hay không, gồm 2 giá trị: scroll (mặc định) và fixed.

Ví dụ: Xem ví dụ 3.3 (scroll) và 3.4 (fixed).

- e) background-position: Quy định vị trí đặt ảnh nền.

Cú pháp:

```
background-position: [Vị trí ngang] [Vị trí dọc];
```

Trong đó:

- [Vị trí ngang] gồm có các giá trị: left, center, right hoặc theo %.
- [Vị trí dọc] gồm có các giá trị: top, center, bottom hoặc theo %.

Ví dụ: Xem ví dụ 3.5.

- f) background: Là thuộc tính tổng hợp của 5 thuộc tính trên.

Cú pháp:

```
background: [background-color] [background-image] [background-repeat]  
[background-attachment] [background-position];
```

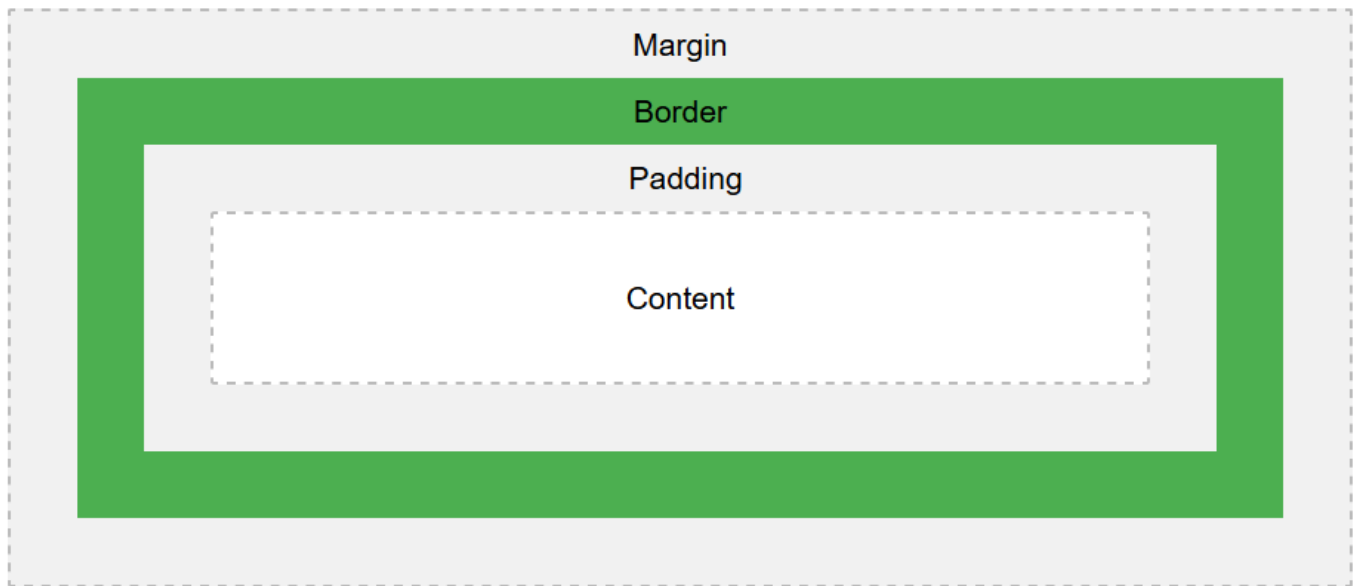
Ví dụ:

```
div {  
    background: url("img/bg-image.jpg") no-repeat fixed right top;  
}
```

V. Box model

Mọi đối tượng HTML đều có thể được xem như là một hình chữ nhật. Trong CSS, thuật ngữ box model dùng để chỉ một vùng hình chữ nhật bao quanh mỗi đối tượng HTML. Box model bao gồm margin, border, padding và nội dung của đối tượng đó:

² Xem lại phần VII, buổi 1



Trong đó:

- Content là phần nội dung của đối tượng HTML.
- Border là viền bao quanh đối tượng.
- Padding là khoảng cách từ nội dung đến viền.
- Margin là vùng trong suốt ở ngoài viền, là khoảng cách giữa các đối tượng HTML.

Kích thước đầy đủ mà 1 đối tượng HTML chiếm trên trang web được tính bằng kích thước phần nội dung cộng với kích thước của cả margin, border và padding (tính theo px):

- Total width = Left margin + Left border + Left padding + Width + Right padding + Right border + Right margin.
- Total height = Top margin + Top border + Top padding + Height + Bottom padding + Bottom border + Bottom margin.

Ví dụ: Với phần định dạng cho thẻ <div> như sau, thì thẻ <div> này rộng 350px và cao 130px:

```
div {  
    width: 320px;  
    height: 100px;  
    padding: 10px;  
    border: 5px solid gray;  
    margin: 0;  
}
```

1. Margin

Margin là 1 vùng trong suốt nằm bên ngoài phần viền bao quanh đối tượng. Margin thường dùng để tạo khoảng trống bao quanh đối tượng, hoặc để thay đổi khoảng cách giữa các đối tượng.

Có 2 cách để định dạng margin cho đối tượng:

- Dùng thuộc tính margin-top, margin-bottom, margin-left, margin-right để định dạng cho từng cạnh.
- Dùng thuộc tính margin để định dạng chung.

Với cách định dạng chung, giá trị của thuộc tính margin có thể có từ 1 đến 4 con số:

- 4 số: theo thứ tự là top, right, bottom, left.

Ví dụ:

```
margin: 20px 15px 30px 10px;
```

- 3 số: theo thứ tự là top, (left = right), bottom.

Ví dụ:

```
margin: 20px 15px 10px;
```

- 2 số: theo thứ tự là (top = bottom), (left = right).

Ví dụ:

```
margin: 20px 15px;
```

- 1 số: cả 4 margin đều bằng nhau.

Ví dụ:

```
margin: 20px;
```

2. Padding

Padding là 1 vùng trong suốt nằm giữa phần nội dung và viền bao quanh đối tượng. Padding thường dùng để tạo khoảng cách giữa nội dung với viền, tránh trường hợp nội dung nằm ngay sát viền sẽ không đẹp.

Cách định dạng padding cho đối tượng cũng tương tự như margin.

3. Border

Border là viền bao quanh đối tượng. Viền bao gồm độ dày, kiểu dáng và màu sắc.

- a) `border-width`: Quy định độ dày của viền, gồm các giá trị: thin, medium, thick hoặc một con số (tính theo px). Thuộc tính `border-width` có thể có từ 1 đến 4 giá trị tương tự như margin.

Ví dụ:

```
border-width: medium;
```

```
border-width: 3px 2px 3px 4px;
```

```
border-width: 1px;
```

- b) `border-style`: Quy định kiểu dáng của viền, gồm các giá trị: dotted, dashed, solid, double, groove, ridge, inset, outset, none, hidden.

Có 2 cách để định dạng kiểu dáng của viền:

- Dùng thuộc tính `border-top-style`, `border-bottom-style`, `border-left-style`, `border-right-style` để định dạng cho từng cạnh.
- Dùng thuộc tính `border-style` để định dạng chung.

Thuộc tính `border-style` có thể có từ 1 đến 4 giá trị tương tự như margin.

Ví dụ: Xem ví dụ 3.6.

- c) `border-color`: Quy định màu sắc của viền. Màu trong CSS được quy định tương tự như màu trong HTML³: tên màu, mã RGB, mã HEX.

- d) `border`: Là thuộc tính tổng hợp của 3 thuộc tính trên.

Cú pháp:

```
border: [border-width] [border-style] [border-color];
```

Có thể thay thuộc tính `border` bằng thuộc tính `border-top`, `border-bottom`, `border-left`, `border-right` để định dạng cho từng cạnh.

Ví dụ:

```
border-top: 1px dashed black;
```

```
border-bottom: 2px dotted green;
```

```
border: 1px solid black;
```

³ Xem lại phần IV của buổi 2

e) `border-radius`: Quy định độ bo tròn của viền. Giá trị là 1 con số (tính theo px hoặc %).

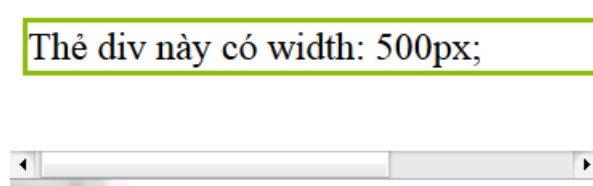
Ví dụ:

```
div {  
    border: 2px solid red;  
    border-radius: 5px;  
}
```

4. Width và height

Để quy định kích thước cho phần nội dung của đối tượng, chúng ta dùng thuộc tính `width` và `height`, gồm các giá trị: `auto`⁴ (mặc định), một con số (tính theo px) hoặc giá trị % tính theo đối tượng cha.

Khi sử dụng `width` và `height`, kích thước của đối tượng sẽ luôn cố định, kể cả khi vùng hiển thị đối tượng nhỏ hơn kích thước đã quy định, khi đó trình duyệt sẽ hiển thị một thanh cuộn ngang:



Để tránh trường hợp này, chúng ta sử dụng thuộc tính `max-width` và `max-height` thay cho 2 thuộc tính trên. Khi đó, nếu chiều rộng vùng hiển thị đối tượng lớn hơn kích thước đã quy định, đối tượng sẽ được hiển thị đầy đủ, ngược lại, đối tượng sẽ tự thu nhỏ kích thước lại.

Ví dụ: Xem ví dụ 3.7 và thu nhỏ kích thước cửa sổ trình duyệt để thấy sự khác biệt của thuộc tính `width` và `max-width`.

Tuy nhiên, khi thu nhỏ quá mức, vùng nội dung của đối tượng sẽ rất khó đọc:

Trường ĐH Khoa học Tự
nhiên - Tp.HCM tọa lạc ở
số 227 Nguyễn Văn Cừ,
Phường 4, Quận 5,
Tp.HCM trên 1 khuôn viên
rộng 1 hecta.
Trường hiện có 09 Khoa;
16 trung tâm đào tạo,
nghiên cứu khoa học -
chuyển giao công nghệ.
Hiện nay, Trường đang có
32 chuyên ngành ở bậc sau
đại học (SDH) và 13

Khi đó, chúng ta sử dụng thêm thuộc tính `min-width` và `min-height` để quy định kích thước tối thiểu của đối tượng.

Ví dụ: Xem ví dụ 3.8.

⁴ Kích thước mang giá trị `auto` tức là sẽ vừa đủ để chứa phần nội dung của đối tượng