

BÀI 3:

CÁC ĐỊNH DẠNG CSS CƠ BẢN

Thiết kế web

Bộ môn Kỹ thuật và mạng máy

tính
Ngô Văn Bình, Trần Phương Nhung, Nguyễn Bá Nghiễn, Nguyễn Trung Phú

Nội dung bài học

Tổng quan về CSS

01

CSS background

02

CSS text

03

CSS font

04

PHẦN 1

TỔNG QUAN VỀ CSS

CSS là gì (What is CSS?)

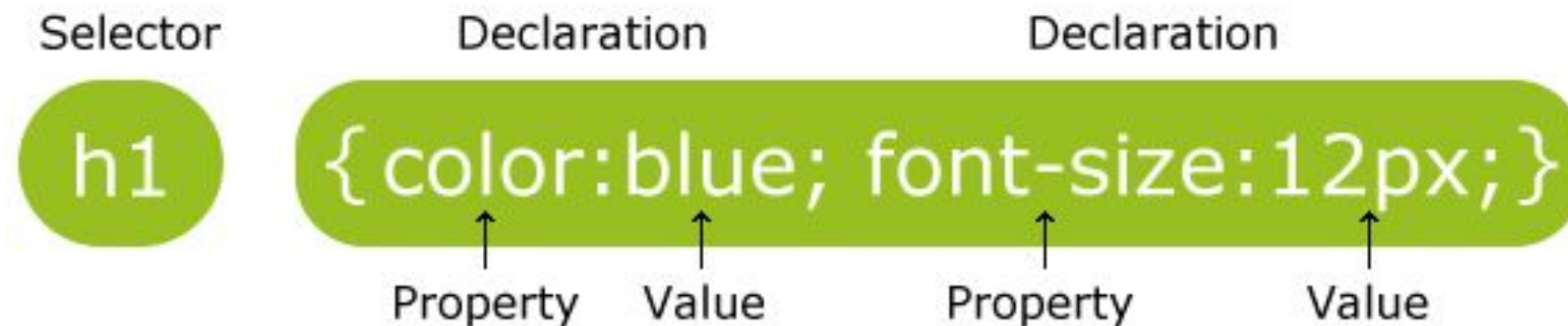
- CSS viết tắt cho cụm từ Cascading Style Sheets
- CSS định nghĩa cách thức hiển thị các phần tử HTML
- CSS được thêm vào HTML 4.0 để giải quyết vấn đề của phiên bản HTML 3.2.
- Sử dụng CSS tiết kiệm thời gian cho thiết kế web
- CSS có thể chứa trong một file văn bản không định dạng có phần mở rộng .css

Cú pháp của CSS

- Cú pháp CSS gồm 2 phần: selector và declaration (property và value) hay còn gọi là khối định dạng

Selector { **property**:**value**; }

- Ví dụ:



Cú pháp của CSS (tiếp)

- Khối định dạng: Chứa 1 hoặc nhiều khai báo, được ngăn cách với nhau bởi dấu chấm phẩy.
- Cú pháp của một khai báo: **property:value;**
- Nếu có nhiều khai báo thì mỗi khai báo nên viết trên một dòng.

Cú pháp của CSS (tiếp)

- Khối định dạng: Chứa 1 hoặc nhiều khai báo, được ngăn cách với nhau bởi dấu chấm phẩy.
- Cú pháp của một khai báo: **property:value;**
- Nếu có nhiều khai báo thì mỗi khai báo nên viết trên một dòng.

Cú pháp của CSS (tiếp)

- **Selector**: được sử dụng để xác định phần tử HTML cần định dạng. Ta có thể xác định một phần tử HTML thông qua tên (tag name), id, lớp (class), ... của phần tử đó.
- **Property**: là thuộc tính quy định cách trình bày. Ví dụ: background-color, font-family...
- **Value**: Giá trị của thuộc tính.
- **Chú thích cho CSS**: Dòng chú thích được đặt trong cặp dấu: /* chú thích */.

Một số selector hay sử dụng

- **Element (tag name) selector:** Sử dụng tên của thẻ HTML để xác định thẻ được định dạng thuộc tính.
 - Ví dụ:

```
h1 {  
    text-align:center;  
    color:red;  
}
```

Một số selector hay sử dụng (tiếp)

- **id selector:**

- Sử dụng thuộc tính id của thẻ HTML để lựa chọn thẻ được định dạng thuộc tính.
- id nên là duy nhất trong một trang. Do đó, ta sử dụng id selector nếu ta muốn lựa chọn một phần tử duy nhất để định dạng.
- Cú pháp của id selector là: **#id** của phần tử
- Ví dụ:

```
#css { color:blue;text-align:center; }
```

□ Chú ý: Không đặt id bắt đầu bằng số

Một số selector hay sử dụng (tiếp)

- **class selector:**

- Sử dụng thuộc tính class của các thành phần trong tài liệu HTML để lựa chọn cho các thành phần đó.
- Sử dụng class selector cho một nhóm các thành phần có cùng định dạng giống nhau.
- Cú pháp của class selector là: **.tên của class**
- Ví dụ:

```
.web { color:green;text-align:center; }
```

Một số selector hay sử dụng (tiếp)

- **universal selector ***:

- Khi muốn áp dụng thuộc tính định dạng với tất cả các thành phần trong tài liệu HTML.
- Cú pháp của universal selector là: `*{thuộc tính1:giá trị1; ,...}`
- Ví dụ:

```
* { color:yellow;text-align:center; }
```

Một số selector hay sử dụng (tiếp)

- **descendant selector:**

- Sử dụng khi muốn áp dụng các thuộc tính định dạng với tất cả các thành phần là con (hậu duệ) của một thành phần khác. Hậu duệ có thể được lồng rất sâu.
- Cú pháp của descendant selector là: **cha con**{**thuộc tính1:giá trị1; ,...**}
- Ví dụ:

```
.container .hello-class{ color:red;}
```

Một số selector hay sử dụng (tiếp)

- **child selector :**

- Không giống như Descendant Selector, Child Selector được sử dụng khi muốn áp dụng thuộc tính định dạng với tất cả các thành phần là con (hậu duệ) trực tiếp của một thành phần khác.
- Cú pháp của child selector là: **cha>con**{thuộc tính1:giá trị1; ,...}
- Ví dụ:
`.paragraph-container > .hello-class{ color:blue; }`

Grouping selectors

```
h1 {
  text-align: center;
  color: red;
}
```

```
h2 {
  text-align: center;
  color: red;
}
```

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

```
h1, h2, p {
  text-align: center;
  color: red;
}
```

CSS làm việc như thế nào?

- Khi trình duyệt đọc CSS thì nó sẽ định dạng tài liệu dựa vào thông tin trong CSS.
- Có 3 cách để chèn CSS vào tài liệu HTML
 - CSS nội tuyến (Inline CSS)
 - CSS bên trong (Internal CSS)
 - CSS bên ngoài (External CSS)

Inline Style

- Style được định nghĩa trong thuộc tính style của một thẻ. Sử dụng khi muốn định dạng xác định cho một thẻ. Nên tránh sử dụng, thay nó bằng **id selector**.
- Ví dụ:

```
<p style="background-color:red;blue;">
```

This is a heading.

```
</h1
```

Internal Style Sheets

- Style sheet được định nghĩa trong thẻ **<style>** đặt trong phần head của tài liệu HTML. Sử dụng khi một trang có định dạng khác với các trang khác.
- Ví dụ:

```
<head>
  <style>
    p.Two{
      background-color:blue;
    }
  </style>
</head>
```

External Style Sheet

- Style sheet được định nghĩa trong 1 file được soạn thảo bằng hệ soạn thảo không định dạng (notepad) và lưu với phần mở rộng **.css**.
- Các trang web cần sử dụng định dạng cần tham chiếu đến file này bằng cách sử dụng thẻ **<link>** đặt trong phần head có cú pháp như sau:

```
<link rel="stylesheet" type="text/css" href="đường  
dẫn và tên file .css">
```

- Sử dụng khi có nhiều trang web có định dạng giống nhau.

Multiple Style sheets

- Nếu một số thuộc tính cùng được định nghĩa cho một **selector** trong các **style sheets** khác nhau thì giá trị của nó sẽ được kế thừa từ những **style sheets** có độ ưu tiên cao hơn.

- Ví dụ:

External style sheet

```
h1 {
  color: yellow;
  margin-left: 20px;
}
```

Internal style sheet

```
h1 {
  color: red;
}
```

h1 { color: red; margin-left: 20px; }

Thứ tự ưu tiên của các style sheets

- Style nào sẽ được sử dụng khi có nhiều hơn 1 style xác định cho một thành phần HTML?
- Các Styles sẽ được sắp xếp (cascade) thành một style ảo duy nhất. Thứ tự ưu tiên của các style như sau:
 - Inline style
 - Internal và external style
 - Browse default
- Giá trị của các thuộc tính trong style có độ ưu tiên cao sẽ ghi đè giá trị thuộc tính trong style có độ ưu tiên thấp hơn.

PHẦN 2

CSS BACKGROUND

Giới thiệu

- Sử dụng để tạo các hiệu ứng nền cho các thành phần HTML.
- Các thuộc tính của CSS background
 - background-color
 - background-image
 - background-repeat
 - background-attachment
 - background-size
 - background-position

Ý nghĩa các thuộc tính

- **background-color**: Xác định màu nền. Màu nền có thể xác định bằng tên màu, rgb hoặc hexa (#ffffff).

- Ví dụ:

```
P {background-color: blue;}
```

```
h1 {background-color: #0066FF;}
```


Ý nghĩa các thuộc tính

- **background-image**: Xác định ảnh nền. Giá trị của thuộc tính là: `url(ảnh nền)`.

- Ví dụ

```
P {background-image: url(c:/anhnen.jpg) ; }
```

Ý nghĩa các thuộc tính

- **background-repeat**: Xác định ảnh nền có được lặp lại hay không. Giá trị của thuộc tính này có thể là:
 - repeat-x: lặp ảnh nền theo chiều ngang
 - repeat-y: lặp ảnh nền theo chiều đứng
 - repeat: lặp ảnh nền theo cả hai chiều (mặc định)
 - no-repeat: không lặp ảnh nền.

Ý nghĩa các thuộc tính (tiếp)

- **background-attachment**: Xác định ảnh có trôi hay cố định khi ta cuộn nội dung trang web hay không. Giá trị của thuộc tính này là:
 - **scroll** (default) : Ảnh nền sẽ cuộn cùng nội dung trang web, đây là giá trị mặc định.
 - **fixed**: Cố định ảnh nền so với nội dung trang web. Khi áp dụng giá trị này, ảnh nền sẽ đứng yên khi bạn đang cuộn trang web.

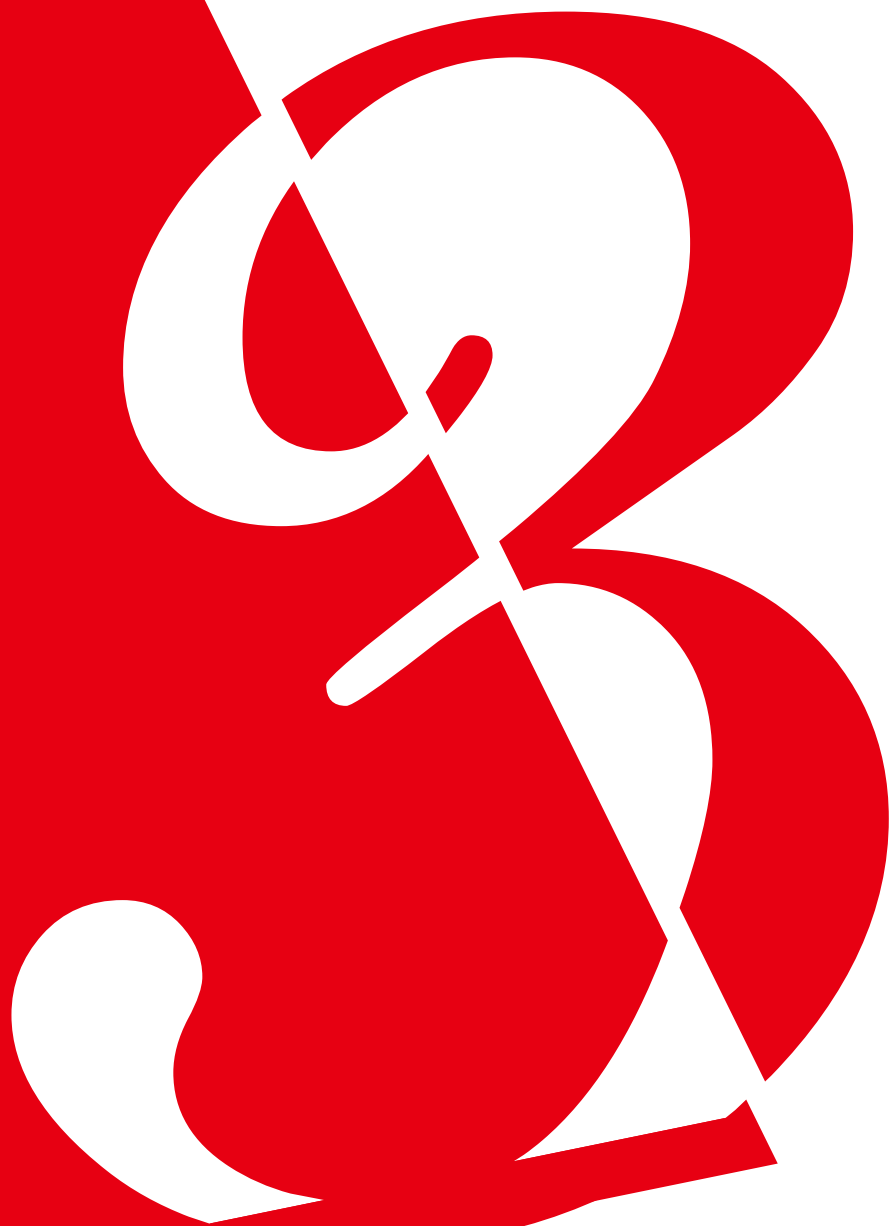
Ý nghĩa các thuộc tính (tiếp)

- **background-size**: thiết lập lại kích thước cho hình nền
Giá trị của thuộc tính là: **width height**.
- Ví dụ

```
body {background-image: url(c:/anhnen.jpg) ;  
      background-size: 101px 62px;}
```

Ý nghĩa các thuộc tính (tiếp)

- **background-position**: Xác định vị trí bắt đầu của ảnh nền. Các giá trị có thể của thuộc tính này là:
 - **x% y%**: góc trên bên trái là 0% 0%, góc dưới bên phải là 100% 100%. Nếu xác định một giá trị thì giá trị còn lại là 50%. Giá trị mặc định là 0% 0%.
 - **xpos ypos**: xác định tọa độ của điểm bắt đầu của ảnh. Tọa độ của điểm góc trên bên trái là 0 0.
 - Các từ định nghĩa sẵn: left top; left center; left bottom; right top; right center; right bottom; center top; center center; center bottom.



PHẦN 3

CSS TEXT

Giới thiệu

- Sử dụng CSS text để định dạng cho văn bản như: màu chữ, căn lề, thụt đầu các đoạn, khoảng cách giữa các ký tự trong từ, khoảng cách giữa các từ trong câu, các hiệu ứng như gạch chân, ...

Ý nghĩa các thuộc tính

- **Color:** Sử dụng thuộc tính này để đặt màu cho văn bản.

- Ví dụ:

```
p{color: blue;}
```

```
h1{color: #ff0000;}
```

```
h2 {color: rgb(0,255,255) ;}
```


Ý nghĩa các thuộc tính (tiếp)

- **text-align:** Sử dụng thuộc tính này để căn lề cho văn bản
 - Ví dụ:

```
p {  
  text-align: justify;  
}  
h1 {  
  text-align: right;  
}  
h2 {text-align: center;}
```

Ý nghĩa các thuộc tính (tiếp)

- **text-indent**: Sử dụng thuộc tính này để thụt đầu đoạn văn bản.

- Ví dụ:

```
p {  
  text-indent: 20px;  
}
```

Ý nghĩa các thuộc tính (tiếp)

- **text-decoration**: Sử dụng thuộc tính này để thiết lập hoặc gỡ bỏ hiệu ứng (gạch chân, gạch đầu,...) cho văn bản.

- Ví dụ:

```
a {text-decoration: none;}
```

```
h1 {text-decoration: overline;}
```

```
h2 {text-decoration: underline;}
```

```
h3 {text-decoration: line-through;}
```

Ý nghĩa các thuộc tính (tiếp)

- **text-transform**: Sử dụng thuộc tính này để chuyển sang chữ thường hoặc chữ hoa.

- Ví dụ

```
p.uppercase { text-transform: uppercase; }
```

```
p.lowercase { text-transform: lowercase; }
```

```
p.capitalize { text-transform: capitalize; }
```

Ý nghĩa các thuộc tính (tiếp)

- **letter-spacing**: Sử dụng thuộc tính này để tăng hoặc giảm khoảng cách giữa các ký tự trong một từ.

- Ví dụ

```
p {letter-spacing: 2px;}
```

```
h1 {letter-spacing: -2px;}
```

Ý nghĩa các thuộc tính (tiếp)

- **word-spacing**: Sử dụng thuộc tính này để tăng hoặc giảm khoảng cách giữa các từ trong văn bản.
 - Ví dụ

```
p {word-spacing: 2px; }
```

```
h1 {word-spacing: -2px; }
```

Ý nghĩa các thuộc tính (tiếp)

- **line-height**: Sử dụng thuộc tính này để tăng hoặc giảm khoảng cách giữa các dòng trong văn bản.

- Ví dụ

```
p.small{line-height: 90%;}
```

```
p.big {line-height: 200%;}
```

Ý nghĩa các thuộc tính (tiếp)

- **direction**: Sử dụng thuộc tính này để xác định hướng viết văn bản từ trái sang phải hoặc từ phải sang trái.
 - Ví dụ

```
p.left {direction: ltr;}
```

```
p.right{ direction: rtl;}
```


Ý nghĩa các thuộc tính (tiếp)

- **text-shadow**: tạo bóng cho văn bản.

- Cú pháp:

`text-shadow: h-shadow v-shadow blur-radius color`

- Trong đó

- **h-shadow** : vị trí bóng ngang so với chữ, số âm sẽ đẩy lên trên và số dương sẽ đẩy xuống dưới
- **v-shadow** : vị trí bóng dọc so với chữ, số âm sẽ đẩy lui phía sau và số dương sẽ đẩy tới phía trước
- **blur-radius** : độ nhòe của chữ bóng, tính bằng pixel
- **color** : màu sắc của bóng

PHẦN 4

CSS FONT



Giới thiệu

- Dùng để xác định font family, font weight, font size và font style của văn bản

Ý nghĩa các thuộc tính

- Thuộc tính **font-family**: dùng để xác định font chữ cho văn bản.
 - Ví dụ

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

Ý nghĩa các thuộc tính (tiếp)

- Thuộc tính **font-size**: dùng để xác định kích thước của chữ cho văn bản. Ta có thể sử dụng đơn vị đo là px, % hoặc em.

- Ví dụ:

```
p.normal {font-size: 16px; /* cỡ chữ mặc định  
của trình duyệt web = 1em*/}
```

```
p.bigger {font-size: 1.5em;}
```

Ý nghĩa các thuộc tính (tiếp)

- Thuộc tính **font-style**: dùng để xác định kiểu chữ cho văn bản.
 - Ví dụ:

```
p.normal {font-style: normal;}
```

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

Ý nghĩa các thuộc tính (tiếp)

- Thuộc tính **font-weight**: dùng để xác định in đậm cho văn bản.
 - Ví dụ:

```
p.normal {font-wieght: normal;}
```

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

Ý nghĩa các thuộc tính (tiếp)

- Thuộc tính **font-variant**: dùng để xác định cho văn bản có in chữ hoa nhỏ (small cap) hay không.
 - Ví dụ:

```
p.normal {font-variant: normal;}
```

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


Xin cảm ơn