


TRƯỜNG ĐẠI HỌC MỞ TP. HCM  
Cơ hội học tập cho mọi người

# ỨNG DỤNG WEB

Nguyễn Thị Mai Trang

1




TRƯỜNG ĐẠI HỌC MỞ TP. HCM  
Cơ hội học tập cho mọi người

## Mục tiêu

- Trình bày được ý nghĩa và vai trò của CSS trong thiết kế web.
- Sử dụng được cú pháp CSS trong thiết kế giao diện web.
- Thiết kế được bố cục trang web.
- Thiết lập được các hiệu ứng cho các phần tử trên trang web.
- Trình bày được các đặc trưng cơ bản của CSS3.
- Sử dụng Bootstrap cơ bản trong thiết kế giao diện web responsive.

3




TRƯỜNG ĐẠI HỌC MỞ TP. HCM  
Cơ hội học tập cho mọi người

## Chương 3

### CSS (Cascading Style Sheets)

2



TRƯỜNG ĐẠI HỌC MỞ TP. HCM  
Cơ hội học tập cho mọi người

## Nội dung

1. Giới thiệu CSS
2. Các bộ chọn trong CSS
3. Định dạng trang web với CSS

4

### 3.1. Giới thiệu CSS

- HTML tạo nội dung trang web.
- CSS (**C**ascading **S**tyle **S**heets) làm đẹp trang web.
- CSS là cấu trúc kỹ thuật định kiểu, một bộ quy tắc để mô tả cách hiển thị của các đối tượng trên trang web.
  - Thay đổi màu sắc các thành phần trên trang web
  - Thêm hình nền, đường viền
  - Thay đổi giao diện trực quan của các phần tử (như danh sách, liên kết,...)
  - Tùy biến bố cục của trang web.

5

5

### Giới thiệu CSS (tt)

- CSS đã được hỗ trợ bởi tất cả các trình duyệt.
- Viết code CSS bằng các phần mềm Text Editor: Notepad++, Edit Plus, Dreamweaver, ...
- Các phiên bản CSS
  - CSS1: 1996, cung cấp khả năng định dạng chuẩn, chưa hỗ trợ một số các tính năng định dạng khác như khoảng trắng, khoảng cách chữ cái, ....
  - CSS2, CSS2.1: bổ sung các tính năng định dạng về vị trí, đa phương tiện, định vị, in ấn,...
  - CSS3: bổ sung khá nhiều tính năng và module đáp ứng với công nghệ web hiện đại như tự động tải font chữ, hiệu ứng màu linh hoạt, biến đổi văn bản, hoạt hình,...

7

7

### Giới thiệu CSS (tt)

- Ưu điểm:
  - Cung cấp các thuộc tính định dạng mà các phần tử HTML không hỗ trợ trực tiếp như: leading, margins, indents, point sizes, background colors, ...
  - Tiết kiệm thời gian, không gian lưu trữ
  - Giúp lệnh HTML đơn giản hơn vì không phải kết hợp các thuộc tính phức tạp trong mỗi phần tử như trước đây, mà tất cả các định dạng được lưu trữ trong tập tin CSS riêng biệt.
  - Cho phép sử dụng cùng một kiểu định dạng trên nhiều phần tử và trang → tính nhất quán trong việc trình bày của website
  - Thay đổi thiết kế trang web dễ dàng mà không cần hiệu chỉnh trực tiếp trên các thẻ HTML

6


6

### Giới thiệu CSS (tt)

- CSS có ba loại:
  - **Inline Style Sheet**: style định nghĩa bên trong thẻ HTML.
  - **Internal Style Sheet**: style định nghĩa trong thẻ `<style></style>` trên trang HTML.
  - **External Style Sheet**: style định nghĩa trong tập tin có phần mở rộng là css, được tham chiếu đến trong trang HTML.

8

8


**Giới thiệu CSS (tt)**


- **Inline Style Sheet:**
  - Sử dụng style ngay trong thẻ HTML cần định dạng
  - Cú pháp: `<TagName style="property1: value1; property2: value2;...">`  
Nội dung văn bản muốn định dạng
  - Ví dụ: `</TagName>`

```

<body>
<h1 style="color:blue;margin-left:30px;">Chữ màu xanh dương, cách lề trái 30 pixel</h1>
<p>Đoạn văn bản với định dạng mặc định</p>
</body>
  
```

**Chữ màu xanh dương, cách lề trái 30 pixel**  
 Đoạn văn bản với định dạng mặc định

9



**Giới thiệu CSS (tt)**

- **Internal Style Sheet:**
  - Thích hợp cho trang web riêng lẻ
  - Tạo một thẻ <style> đặt trong phần <head>
  - Cú pháp:
 

```

<head>
  <style> hoặc <style type="text/css">
    Selector_namer {
      Property1:value1;
      Property2:value2;
      ....
    }
  </style>
</head>
          
```

11


**Giới thiệu CSS (tt)**

- **Inline Style Sheet (tt):**
  - Hạn chế:
    - Chỉ áp dụng cho từng phần tử riêng lẻ → không hiệu quả đối với trang web có nhiều phần tử cần định dạng → thiếu tính nhất quán đối với website lớn, có nhiều trang.
    - Pha trộn nội dung với hình thức thể hiện → mã lệnh HTML trở nên rối rắm → cập nhật, thay đổi hình thức các phần tử trên trang web khó khăn.

10


**Giới thiệu CSS (tt)**

- **Internal Style Sheet:**  
ví dụ:
 

```

<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>CSS</title>
  <style>
    body {
      color: blue;
      background-color: lightgray;
    }
    h1 {
      color: red;
      background-color: yellow;
    }
  </style>
</head>
<body>
  <h1>Chữ màu đỏ trên nền màu vàng</h1>
  <p>Chữ màu xanh trên nền màu xám</p>
</body>
</html>
      
```

12

**TRƯỜNG ĐẠI HỌC MỞ TP. HCM**  
Cơ hội học tập cho mọi người

## Giới thiệu CSS (tt)

- External Style Sheet
  - Các style được viết trong một file CSS (\*.css) riêng bên ngoài và được liên kết đến từ trang HTML
  - Tất cả các trang web có tạo liên kết đến file .css đều bị ảnh hưởng
  - Trang web liên kết đến file .css bằng thuộc tính **href** trong thẻ **<link>**
  - Cú pháp:

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

13

**TRƯỜNG ĐẠI HỌC MỞ TP. HCM**  
Cơ hội học tập cho mọi người

## Giới thiệu CSS (tt)

- External Style Sheet (tt)

```
graph TD
    A[CSS External File] --> B[Web page]
    A --> C[Web page]
    A --> D[Web page]
    A --> E[Web page]
```

15

**TRƯỜNG ĐẠI HỌC MỞ TP. HCM**  
Cơ hội học tập cho mọi người

## Giới thiệu CSS (tt)

- External Style Sheet (tt)
  - Cách tạo file CSS: sử dụng một trình soạn thảo tạo tập tin lưu với dạng plain text, tên file có phần mở rộng là css
  - Nhập các selector muốn định dạng thuộc tính theo cú pháp:

```
Selector { "property1: value1; property2: value2;...">
```

14

**TRƯỜNG ĐẠI HỌC MỞ TP. HCM**  
Cơ hội học tập cho mọi người

## Giới thiệu CSS (tt)

- External Style Sheet (tt)
  - Ví dụ: cấu trúc thư mục web như sau:
- Nội dung trang exstyle.css

```
graph TD
    web[web] --> css[css_external-style.html]
    web --> style[style]
    style --> exstyle[exstyle.css]
```

```
body { color: blue; }
h1 { color: red; }
#link {
    color: red;
    font-weight: bold;
}
.yellow { background-color: yellow; }
```

16

**Giới thiệu CSS (tt)**

- External Style Sheet (tt)
  - Nội dung trang css external-style.html:

```
<head> <link rel="stylesheet" type="text/css" href=" style/exstyle.css "> </head>
```

**CSS External Style Sheet**

CSS External Style Sheet được viết trong tập tin có phần mở rộng là css  
Tập tin HTML liên kết đến tập tin CSS dùng thẻ <link>

```
<body>
<h1>CSS External Style Sheet</h1>
<p>CSS <em class="yellow">External Style Sheet</em> được viết trong tập tin có
phần mở rộng là css</p>
<p>Tập tin HTML <span id="link">liên kết</span> đến tập tin CSS dùng thẻ
<span id="link">&lt;link&gt;</span></p>
</body>
```

17

**Giới thiệu CSS (tt)**

- Độ ưu tiên của CSS
  - Ví dụ: với các khai báo style cho phần tử <h1> như sau, khi hiển thị trên trình duyệt, dòng chữ “Minh họa độ ưu tiên trong CSS” sẽ có màu đen

```
<head>
<style>
h1 {color: red;}
</style>
</head>
<body>
<h1 style="color: black">Minh họa độ ưu tiên trong CSS </h1>
</body>
```

Ưu tiên hơn

19

**Giới thiệu CSS (tt)**

- Độ ưu tiên của CSS từ cao xuống thấp

```

CSS inline  ⇒  CSS bên trong  ⇒  CSS bên ngoài
                ↓
            CSS mặc định của trình duyệt
  
```

- Lưu ý: khi áp dụng style CSS cho một phần tử, nó sẽ được áp dụng cho tất cả các phần tử con (các phần tử chứa bên trong phần tử đó)


18

**Giới thiệu CSS (tt)**

- Độ ưu tiên của CSS (tt)
  - Thay đổi độ ưu tiên: sử dụng thuộc tính **!important**
  - Đặt thuộc tính này sau thuộc tính muốn áp dụng theo cú pháp :  
**selector { property:value !important }**

```
<head> <style>
p { text-align:right; color:red !important; }
</style> </head>
<body>
<p style="text-align:center; color:blue" >
Đoạn văn bản này khi hiển thị trên trình duyệt có màu đỏ, canh lề giữa.
</p>
</body>
```

20



**TRƯỜNG ĐẠI HỌC MỞ TP. HCM**  
Cơ hội học tập cho mọi người

# Nội dung

---

1. Giới thiệu CSS
2. **Các bộ chọn trong CSS**
3. Định dạng trang web với CSS

21


**TRƯỜNG ĐẠI HỌC MỞ TP. HCM**  
Cơ hội học tập cho mọi người

## 3.2 Các bộ chọn trong CSS

- Cú pháp CSS
- Các bộ chọn trong CSS

22

22



**TRƯỜNG ĐẠI HỌC MỞ TP. HCM**  
Cơ hội học tập cho mọi người

## 3.2.1 Cú pháp CSS

---

- Một bộ chọn trong CSS:
 


Phần tử chọn

{thuộctính: giá trị; thuộctính: giá trị;... }

} Phần mô tả
- Phần tử chọn (Selector): đối tượng lựa chọn
  - Thẻ HTML: viết bình thường, ví dụ: **div**, **p**, **h1**, **h2**, **a**,...
  - class: sử dụng dấu chấm phía trước class-name, ví dụ: **.banner**
  - id: sử dụng dấu # phía trước định danh id, ví dụ: **#banner**
  - Chọn theo các cú pháp đặc biệt khác
- Phần mô tả (Declaration): có dạng {**thuộctính: giá trị**}, nếu có nhiều thuộc tính thì các bộ cách nhau dấu chấm phẩy (;)
- Ví dụ:
 

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

23



**TRƯỜNG ĐẠI HỌC MỞ TP. HCM**  
Cơ hội học tập cho mọi người

## Cú pháp CSS (tt)

---

- Nếu value là chuỗi có khoảng trắng thì phải đặt trong dấu ngoặc kép, ví dụ: `p{ font-family : "Times New Roman" }`
- Nhiều thuộc tính có giá trị giống nhau, có thể được gom nhóm bằng dấu phẩy

```
h1 {
    color:#0000FF;
    text-transform:uppercase;
}
h2 {
    color:#0000FF;
    text-transform:uppercase;
}
```

➔

```
h1, h2 {
    color:#0000FF;
    text-transform:uppercase;
}
```

24

24

## Cú pháp CSS (tt)

- Bộ chọn **class** (lớp): định nghĩa nhiều style khác nhau cho cùng một phần tử hoặc cho nhiều phần tử.
  - Cùng một phần tử: **<tagname>.<classname> {properties}**
  - Nhiều phần tử: **.<classname>{properties}**
  - Khai báo: **<tagname class = "classname" ...>**
  - Ví dụ:

```
p { color: red; text-align: center; }
p.lineone { color:blue; font-size: 18px }
p.linetwo { font-size: 16px }
```

25

25

## Cú pháp CSS (tt)

- Ví dụ sử dụng style cho nhiều tag

```
<head>
<meta charset="utf-8">
<style>
  .dong1 { color:blue; font-size: 18px }
  .dong2 { font-size: 20px; font-weight: bold}
</style>
</head>
<body>
  <p class="dong1"> Đoạn này áp dụng style của class dong1 </p>
  <div class="dong2"> Khối này áp dụng style của class dong2</div>
  <div class="dong1"> Khối này áp dụng style của class dong1</div>
</body>
```

Đoạn này áp dụng style của class dong1

Khối này áp dụng style của class dong2

Khối này áp dụng style của class dong1

27

27

## Cú pháp CSS (tt)

- Ví dụ sử dụng style cho tag <p>

```
<head>
<meta charset="utf-8">
<style>
  p { color: red; text-align: center; }
  p.dong1 { color:blue; font-size: 18px }
  p.dong2 { font-size: 20px; font-weight: bold}
</style>
</head>
<body>
  <p> Đoạn này chỉ áp dụng style của phần tử <p> </p>
  <p class="dong1"> Đoạn này áp dụng style của phần tử <p> và class dong1</p>
  <p class="dong2"> Đoạn này áp dụng style của phần tử <p> và class dong2</p>
</body>
```

Đoạn này chỉ áp dụng style của phần tử &lt;p&gt;

Đoạn này áp dụng style của phần tử &lt;p&gt; và class dong1

Đoạn này áp dụng style của phần tử &lt;p&gt; và class dong2

26

26


## Cú pháp CSS (tt)

- Bộ chọn là id:

- id là thuộc tính của phần tử trong trang HTML.
- Giá trị của thuộc tính id nên là duy nhất.
- Do đó, mỗi bộ chọn là id chỉ áp dụng style cho một phần tử.
- Sử dụng: **#tagid { properties }**
- Khai báo trong html: **<tagname id = "tagid" ...>**

28

28



## Cú pháp CSS (tt)

- Bộ chọn là id: Ví dụ

```

<head>
<meta charset="utf-8">
<style>
  p { color: red; text-align: center; }
  #dong1 { color: blue; font-size: 18px; }
  #dong2 { font-size: 20px; font-weight: bold; }
</style>
</head>
<body>
<p> Đoạn này chỉ áp dụng style của phần tử <p> &lt; p &gt; </p>
<p id="dong1"> Đoạn này áp dụng style của phần tử &lt; p &gt; và #dong1 </p>
<p id="dong2"> Đoạn này áp dụng style của phần tử &lt; p &gt; và #dong2 </p>
</body>


```

Đoạn này chỉ áp dụng style của phần tử <p>

Đoạn này áp dụng style của phần tử <p> và #dong1

Đoạn này áp dụng style của phần tử <p> và #dong2


29



## 3.2.2 Các bộ chọn CSS

Bộ chọn (Selector)	Mô tả	Ví dụ
E	Tất cả các phần tử E	<code>h1 {color: red;}</code> /* thiết lập màu chữ đỏ cho các thẻ h1 */
*	Tất cả các phần tử	<code>* {color: red;}</code> /* thiết lập màu chữ đỏ cho tất cả các phần tử */
E, F, G	Chọn một nhóm các phần tử E, F, G	<code>h1, h2, h3 {background-color: red;}</code> /* thiết lập màu nền đỏ cho các thẻ h1, h2, h3 */
#id	Phần tử có thuộc tính id="id"	<code>#test {color: green;}</code> /* thiết lập màu chữ xanh lá cho phần tử có thuộc tính id='test' */
E#id	Chọn các phần tử E có thuộc tính id="id"	<code>h3#contact {color: red;}</code> /* thiết lập màu chữ đỏ cho thẻ h3 có thuộc tính id='contact' */

31



## Cú pháp CSS (tt)

- Ghi chú trong CSS
  - Cú pháp : `/* Ghi chú */`
  - Ví dụ:


```

p {
  color: red;
  /* This is a single-line comment */
  text-align: center;
}

/* This is
a multi-line
comment */

```

30




## Các bộ chọn CSS (tt)


Bộ chọn (Selector)	Mô tả	Ví dụ
.classname	Chọn tất cả các phần tử có thuộc tính class="classname"	<code>.note {color: yellow;}</code> /* thiết lập màu chữ vàng cho tất cả các phần tử có thuộc tính class='note' */
E.classname	Chọn các phần tử E có thuộc tính class="classname"	<code>h1.note {text-decoration: underline;}</code> /* thiết lập font chữ có gạch dưới cho các thẻ h1 có thuộc tính class='note' */
E F	Chọn các phần tử con F nằm trong phần tử E	<code>p strong {background-color: yellow;}</code> /* thiết lập màu nền vàng cho tất cả các thẻ strong nằm trong thẻ p */
E > F	Chọn phần tử F có phần tử cha là E	<code>body &gt; p {background-color: green;}</code> /* thiết lập màu nền xanh lục cho tất cả thẻ p có phần tử cha là body */

32




 <b>Các bộ chọn CSS (tt)</b>		
Bộ chọn (Selector)	Mô tả	Ví dụ
E + F	Chọn phần tử F liền kề ngay sau phần tử E	<b>h1 + p {color: red;}</b> /* thiết lập màu chữ đỏ cho thẻ p liền kề sau thẻ h1 */
E ~ F	Chọn tất cả các phần tử F sau phần tử E	<b>p ~ strong {font-style: italic;}</b> /* thiết lập chữ in nghiêng trong tất cả các thẻ strong sau thẻ p */
E[attr]	Chọn tất cả các phần tử E có thuộc tính attr	<b>a[href] {background-color: yellow;}</b> /* thiết lập màu nền vàng cho các thẻ a có thuộc tính href */
E[attr="value"]	Chọn các phần tử E có thuộc tính attr=value	<b>a[href="http://www.abc.com"]{font-weight: bold;}</b> /* thiết lập chữ in đậm cho các thẻ a có thuộc tính href="http://www.abc.com" */


33

 <b>Các bộ chọn CSS (tt)</b>		
Bộ chọn (Selector)	Mô tả	Ví dụ
E[attr*=value]	Chọn các phần tử E có thuộc tính attr mà giá trị của nó chứa chuỗi value.	<b>div[id*="section"] {color: red;}</b> /* thiết lập màu chữ đỏ cho các thẻ div mà thuộc tính id chứa chuỗi "section" */
:after Hoặc ::after	Thiết lập style áp dụng ngay sau phần tử chỉ định.	<b>div:after {content:url(banana.gif);}</b> Hoặc <b>div::after {content:url (banana.gif);}</b> /* chèn ảnh banana.gif ngay sau tất cả các thẻ div */
:before Hoặc ::before	Thiết lập style áp dụng ngay trước phần tử chỉ định.	<b>div:before {content:url(banana.gif.gif);}</b> Hoặc <b>div::before {content:url(banana.gif.gif);}</b> /* chèn ảnh banana.gif ngay trước tất cả các thẻ div */


35

 <b>Các bộ chọn CSS (tt)</b>		
Bộ chọn (Selector)	Mô tả	Ví dụ
E[attr~=value]	Chọn các phần tử E có thuộc tính attr mà giá trị của nó có chứa từ value	<b>p[title~="test"] { font-style:italic; }</b> /* thiết lập chữ in nghiêng trong các thẻ p mà giá trị của thuộc tính title có chứa từ "test" */
E[attr^=value]	Chọn các phần tử E có thuộc tính attr mà giá trị của nó bắt đầu là value	<b>h1[id^="abc"] { color: red; }</b> /* thiết lập màu chữ đỏ cho các thẻ h1 mà thuộc tính id có ba ký tự đầu là "abc" */
E[attr\$=value]	Chọn các phần tử E có thuộc tính attr mà giá trị của nó kết thúc là value	<b>h1[id\$="xyz"] { color: blue; }</b> /* thiết lập màu chữ xanh dương cho các thẻ h1 mà thuộc tính id có ba ký tự cuối là "xyz" */


34

 <b>Các bộ chọn CSS (tt)</b>		
Bộ chọn (Selector)	Mô tả	Ví dụ
:checked	Chọn các phần tử có trạng thái được checked	<b>:checked {color: red;}</b> /* thiết lập màu chữ đỏ cho các phần tử có trạng thái được checked */
:default	Chọn các phần tử là mặc định (default) trong tập hợp các phần tử tương tự nhau	<b>:default {background-color: red;}</b> /* thiết lập màu nền đỏ cho phần tử mặc định (ví dụ submit button) */
:disabled	Chọn các phần tử đang ở trạng thái vô hiệu hóa (disabled)	<b>input:disabled {background-color: gray;}</b> /* thiết lập màu nền xám cho các thẻ input bị vô hiệu hóa */


36

 <b>Các bộ chọn CSS (tt)</b>		
Bộ chọn (Selector)	Mô tả	Ví dụ
:empty	Chọn phần tử rỗng	<code>div:empty {display: none;}</code> /* ẩn các thẻ div không chứa nội dung bên trong */
:enabled	Chọn các phần tử không ở trạng thái vô hiệu hóa	<code>input:enabled {background-color:white;}</code> /* thiết lập màu nền trắng cho các thẻ input không bị vô hiệu hóa */
:first-child	Chỉ chọn các phần tử là con đầu tiên của phần tử khác.	<code>p:first-child { color: red;}</code> /* thiết lập màu chữ đỏ cho các thẻ p đầu tiên nằm trong phần tử khác */
:first-letter	Chọn ký tự đầu trong nội dung của phần tử	<code>p:first-letter {font-size: larger;}</code> /* thiết lập font chữ lớn (larger) cho các ký tự đầu trong các thẻ p */


37

 <b>Các bộ chọn CSS (tt)</b>		
Bộ chọn (Selector)	Mô tả	Ví dụ
:hover	Chọn phần tử đang nhận con trỏ chuột	<code>p:hover {background-color: yellow;}</code> /* thiết lập màu nền vàng cho thẻ p khi con trỏ chuột đi vào vùng nội dung trên thẻ p */
:lang(value)	Chọn phần tử có thuộc tính lang="value"	<code>*:lang(fr) {color: blue;}</code> /* thiết lập màu chữ xanh dương cho tất cả các phần tử có thuộc tính lang="fr" */
:last-child	Chỉ chọn các phần tử là con cuối cùng của phần tử khác.	<code>p:last-child {font-size: small;}</code> /* thiết lập font chữ nhỏ (small) cho các thẻ p cuối cùng bên trong mỗi phần tử khác */
:last-of-type	Chọn phần tử con cuối cùng trong nhóm các phần tử cùng loại nằm trong một phần tử cha.	<code>p:last-of-type {background-color: red;}</code> /* thiết lập màu nền đỏ cho các thẻ p cuối cùng trong mỗi phần tử khác */


39

 <b>Các bộ chọn CSS (tt)</b>		
Bộ chọn (Selector)	Mô tả	Ví dụ
:first-line Hoặc ::first-line	Chọn dòng đầu tiên của phần tử	<code>p:first-line {color: blue;}</code> Hoặc <code>p::first-line {color: red;}</code> /* thiết lập màu chữ xanh dương cho dòng đầu tiên của các thẻ p */
:first-of-type	Chọn phần tử con đầu tiên trong nhóm các phần tử cùng loại nằm trong một phần tử cha.	<code>p:first-of-type { background-color: red; }</code> /* thiết lập màu nền đỏ cho các thẻ p đầu tiên trong mỗi phần tử cha */
:focus	Chọn phần tử đang nhận focus	<code>input:focus {background-color:yellow;}</code> /* thiết lập màu nền vàng cho phần tử input đang nhận focus */


38

 <b>Các bộ chọn CSS (tt)</b>		
Bộ chọn (Selector)	Mô tả	Ví dụ
:not(selector)	Chọn các phần tử không thuộc bộ chọn selector	<code>:not(p) { color: red; }</code> /* thiết lập màu chữ đỏ cho các nội dung không nằm trong thẻ p */
:nth-child(n) :nth-child(odd) :nth-child(even)	Chọn phần tử con thứ n theo thứ tự từ đầu odd: chọn phần tử lẻ even: chọn phần tử chẵn	<code>div:nth-child(2) {background-color: red;}</code> /* thiết lập màu nền đỏ cho phần tử div là con thứ 2 trong phần tử khác */
:nth-last-child(n)	Chọn phần tử con thứ n theo thứ tự từ cuối.	<code>p:nth-last-child(3) {color: yellow;}</code> /* thiết lập màu chữ vàng cho phần tử p là con thứ ba theo thứ tự từ cuối */
:nth-of-type(n)	Chọn phần tử con thứ n trong nhóm các phần tử cùng loại nằm trong một phần tử cha.	<code>p:nth-of-type(2) {background-color: red;}</code> /* thiết lập màu nền đỏ cho phần tử con p thứ hai trong nhóm các phần tử p nằm trong một phần tử cha */


40

 <b>Các bộ chọn CSS (tt)</b>		
Bộ chọn (Selector)	Mô tả	Ví dụ
:nth-last-of-type (n)	Chọn phần tử con thứ n trong nhóm các phần tử cùng loại nằm trong một phần tử cha theo thứ tự từ cuối.	<code>p:nth-last-of-type(2) {color:red;}</code> /* thiết lập màu chữ đỏ cho phần tử con p thứ hai trong nhóm các phần tử p nằm trong một phần tử cha theo thứ tự từ cuối */
:only-child	Chọn phần tử con duy nhất của phần tử khác	<code>h1:only-child {color: blue;}</code> /* thiết lập màu chữ xanh dương cho thẻ h1, nếu nó là con duy nhất trong phần tử khác */
:only-of-type	Chọn một phần tử con duy nhất nếu trong phần tử cha đó không chứa phần tử nào khác cùng loại với phần tử con này	<code>p:only-of-type {font-weight: bold;}</code> /*thiết lập font chữ in đậm cho thẻ p duy nhất nằm trong phần tử khác */


41

 <b>Nội dung</b>	
<ol style="list-style-type: none"> <li>1. Giới thiệu CSS</li> <li>2. Các bộ chọn CSS</li> <li><b>3. Định dạng trang web với CSS</b></li> </ol>	


43

 <b>Các bộ chọn CSS (tt)</b>		
Bộ chọn (Selector)	Mô tả	Ví dụ
:root	Chọn phần tử gốc của tài liệu	<code>:root {background-color: blue;}</code> /* thiết lập màu nền xanh dương cho trang web */
::selection Hoặc ::-moz-selection (Firefox)	Phần nội dung được chọn (selected)	<code>::selection {color: red;background: yellow;}</code> /* Code cho Firefox */ <code>::-moz-selection { color: red; background: yellow;}</code> /*Thiết lập màu chữ đỏ, nền vàng cho phần văn bản được chọn*/

42

 <b>3.3. Định dạng trang web với CSS</b>	
<ul style="list-style-type: none"> <li>• Màu và nền</li> <li>• Text và Font</li> <li>• Liên kết</li> <li>• Danh sách</li> <li>• Box Model</li> <li>• Bảng</li> </ul>	
<ul style="list-style-type: none"> <li>• CSS Display</li> <li>• CSS Position</li> <li>• CSS Float &amp; Clear</li> <li>• Layers</li> <li>• Giới thiệu CSS3</li> <li>• <b>Giới thiệu Bootstrap</b></li> </ul>	


44



## Màu và nền

- Các dạng màu trong CSS
  - 6 - Hex color:
    - 6 chữ số thập lục phân biểu diễn tổ hợp ba màu đỏ (red), xanh lá (green), xanh dương (blue) đặt sau tiền tố #
    - Ví dụ: #00ff00;, #55aa88;...
  - RGB:
    - Dạng rgb(red,green,blue), mỗi giá trị từ 0 - 255 hoặc theo tỷ lệ %
    - Ví dụ: color:rgb(204,0,51);, color:rgb(0%,10%,50%);
  - RGBA:
    - rgba(red,green,blue,alpha), trong đó alpha là độ rõ của màu, có giá trị từ 0-1, 1 là rõ hoàn toàn.
    - ví dụ color:rgba(255,0,0,0.4);
  - Màu theo tên:
    - "red", "green", "blue", "yellow",...

45



## Màu và nền (tt)

- background-color: màu nền

**Ví dụ sử dụng CSS background-color!**

Văn bản trong thẻ < body >

Văn bản trong thẻ < div >

Văn bản trong thẻ < p >

Văn bản trong thẻ < p >

Văn bản trong thẻ < p >


Văn bản trong thẻ < p >

Văn bản trong thẻ < div >

Văn bản trong thẻ < body >

```
<head> <style>
  body{background-color: yellow;}
  h1 {background-color: rgb(80,231,22);}
  p {background-color:rgba(50,203,179,0.5);}
  div {background-color:#EDC0E9;}
</style></head>
<body>
<h1>Ví dụ sử dụng CSS background-color!</h1>
Văn bản trong thẻ &lt; body &gt;
<div>
  Văn bản trong thẻ &lt; div &gt;
  <p>Văn bản trong thẻ &lt; p &gt;</p>
  <p>Văn bản trong thẻ &lt; p &gt;</p>
  <p>Văn bản trong thẻ &lt; p &gt;</p>
  <p>Văn bản trong thẻ &lt; p &gt;</p>
  Văn bản trong thẻ &lt; div &gt;
</div>
Văn bản trong thẻ &lt; body &gt;
</body>
```


47



## Màu và nền

- CSS background: dùng để định nghĩa hiệu ứng nền của phần tử, bao gồm các thuộc tính:
  - background-color
  - background-image
  - background-repeat
  - background-attachment
  - background-position

46



## Màu và nền (tt)

- background-image: chỉ định một tập tin ảnh làm nền

```
body {
  background-image: url("images/oulogo.png");
}
```

- oulogo.png là tập tin ảnh nằm trong thư mục images cùng thư mục với tập tin html
- Có thể kết hợp với thuộc tính **background-repeat** lặp ảnh theo hướng ngang và dọc

48

**Màu và nền (tt)**

- **background-repeat:**
  - Mặc định: hình sẽ lặp lại ảnh theo cả hai hướng ngang và dọc

```
body{
  background-image: url(Images/oulogo.png);
}
```



49

**Màu và nền (tt)**

- **background-repeat:**
  - repeat-y: lặp lại ảnh theo hướng dọc

```
body {
  background-image: url(Images/oulogo.png);
  background-repeat: repeat-y;
}
```



51

**Màu và nền (tt)**

- **background-repeat:**
  - repeat-x: lặp lại ảnh theo hướng ngang

```
body {
  background-image: url(Images/oulogo.png);
  background-repeat: repeat-x;
}
```



50

**Màu và nền (tt)**

- **background-repeat:**
  - space: lặp lại ảnh theo hướng ngang, giữa mỗi ảnh có khoảng trống

```
body {
  background-image: url(Images/oulogo.png);
  background-repeat: space;
}
```

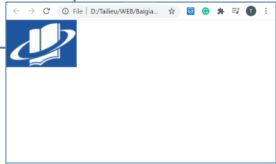


52

**Màu và nền (tt)**

- **background-repeat:**
  - no-repeat: không lặp

```
body {
  background-image: url(Images/oulogo.png);
  background-repeat: no-repeat;
}
```



53

**Màu và nền (tt)**

- **background-position:** định vị trí cho ảnh nền trong phần tử chứa nó, với các giá trị:
  - x% y%: tỷ lệ phần trăm theo trục x (ngang) và y (trục dọc)
  - x, y: vị trí theo chiều ngang, chiều dọc với đơn vị pixel
  - left top, left center, left bottom
  - right top, right center, right bottom
  - center top, center center, center bottom
  - Nếu không chỉ định thuộc tính background-position, giá trị mặc định là left top
  - Tham khảo: [https://www.w3schools.com/cssref/playit.asp?filename=playcss\\_background-position](https://www.w3schools.com/cssref/playit.asp?filename=playcss_background-position)

55

**Màu và nền (tt)**

- **background-attachment:** thiết lập tính cố định của ảnh nền, có 2 giá trị:
  - scroll: ảnh nền sẽ cuộn cùng nội dung trang web (mặc định).
  - fixed: cố định ảnh nền so với nội dung trang web, ảnh nền sẽ đứng yên khi người dùng cuộn trang web.

```
body {
  background-image: url(Images/flower.png);
  background-repeat: no-repeat;
  background-attachment: fixed;
}
```

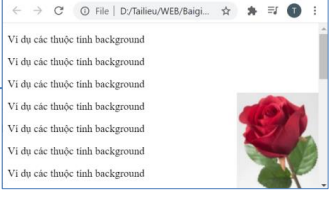


54


**Màu và nền (tt)**

- **background-position:** ví dụ

```
body{
  background-image: url(Images/flower.png);
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-position: bottom right;
}
```



56




**Màu và nền (tt)**

- **Sử dụng background rút gọn:**
  - Có thể viết theo cú pháp rút gọn, ví dụ:

```
body {
  background: #ffffff url(images/flower.png) no-repeat right top;
}
```
  - Thứ tự các thuộc tính:

```
background: [<background-image>] [<background-color>]
[<background-position>] [<background-size>]
[<background-repeat>] [<background-origin>] [<background-clip>]
[<background-attachment>] [<initial|inherit>];
```


57



**Text và Font**

- Các thuộc tính định dạng văn bản:
  - **color**: màu chữ
  - **text – align**: canh lề văn bản theo phương ngang so với phần tử chứa nó.
  - **vertical – align**: canh lề văn bản theo phương đứng so với phần tử chứa nó.
  - **text – decoration**: chỉ định văn bản được gạch dưới, gạch ngang hoặc nhấp nháy.
  - **text – indent**: chỉ định thụt lề trái cho văn bản
  - **text – transform**: chỉ định văn bản hiển thị dạng chữ hoa, chữ thường.
  - **text – shadow**: chỉ định văn bản có bóng
  - **letter – spacing**: khoảng cách giữa các ký tự trong văn bản
  - **word – spacing**: khoảng cách giữa các từ
  - **white – space**: độ co giãn của các khoảng trắng trong văn bản
  - **direction**: hướng của văn bản

59




**3.3. Định dạng trang web với CSS**

- Màu và nền
- **Text và Font**
- Liên kết
- Danh sách
- Box Model
- Bảng

- CSS Display
- CSS Position
- CSS Float & Clear
- Layers
- Giới thiệu CSS3

58



**Text và Font (tt)**

- **Thiết lập màu chữ: color**

```
<style>
body {
  color: blue;
  background-color: yellow;
}
span.color {color: red;}
</style>
```

```
<body>
<h3>color</h3>
<p>Nền trang web có màu vàng</p>
<p>
  Đoạn văn bản này hiển thị bình thường
  <span class="color"> Đoạn này có màu đỏ
</span>
  <br> Đoạn văn bản này hiển thị bình thường
</p>
```


**color**

Nền trang web có màu vàng

Đoạn văn bản này hiển thị bình thường **Đoạn này có màu đỏ**

Đoạn văn bản này hiển thị bình thường

60



**Text và Font (tt)**

- Canh lề văn bản theo chiều ngang: **text-align**
  - Gồm các giá trị: left (mặc định), right, center, justify

```
<style>
h1 {text-align: center;}
p.right {text-align: right;}
p.justify {text-align: justify;}
</style>

<body>
<h1> ... </h1>
<p class = "right"> ... </p>
<p class = "justify"> .... </p>
</body>
```


**Tiêu đề được canh giữa**

Đoạn này được canh lề phải

Đoạn này được canh đều hai bên. Đoạn này được canh đều hai bên. Đoạn này được canh đều hai bên. Đoạn này được canh đều hai bên. Đoạn này được canh đều hai bên. Đoạn này được canh đều hai bên.

Đoạn văn bản này hiển thị mặc định (canh lề trái)

61




**Text và Font (tt)**

- Các giá trị của **vertical-align**:
  - middle: canh giữa theo chiều dọc
  - bottom: lề dưới của văn bản/hình ảnh bằng với lề dưới của phần tử thấp nhất trên dòng.
  - text-bottom: lề dưới của văn bản/hình ảnh bằng với lề dưới của phần văn bản thấp nhất trên dòng.
  - Giá trị theo px hoặc %.

```
<style>
img{width: 50px; height: 50px;}
.plain {color:#FF0000;}
.baseline {vertical-align:baseline;}
.sub {vertical-align:sub;}
.super {vertical-align:super;}
.top {vertical-align:top;}
.texttop {vertical-align:text-top;}
.middle {vertical-align:middle;}
.bottom {vertical-align:bottom;}
.textbottom {vertical-align:text-bottom;}
.percent100 {vertical-align:100%;}
.pixel0 {vertical-align:10px;}
.pixel20 {vertical-align:20px;}
</style>
```


63



**Text và Font (tt)**

- Canh lề văn bản theo chiều dọc: **vertical-align**
  - Các giá trị của **vertical-align**:
    - baseline: canh lề như phần tử chứa nó (mặc định).
    - sub: tạo chỉ số dưới.
    - super: tạo chỉ số trên.
    - top: lề trên của văn bản/hình ảnh bằng với lề trên của phần tử cao nhất trên dòng.
    - text-top: lề trên của văn bản/hình ảnh bằng với lề trên của phần văn bản cao nhất trên dòng.

62



**Text và Font (tt)**

- **vertical-align (tt)**:

Plain text no vertical align	Plain text baseline	Plain text sub
Plain text super	Plain text top	Plain text texttop
Plain text middle	Plain text bottom	Plain text textbottom
100% Plain text	10px Plain text	20px Plain text

64



**Text và Font (tt)**

- Hiệu ứng gạch trên, dưới, ngang chữ: **text-decoration**
  - Gồm các giá trị: none, overline, line-through, underline

```
h1 {
  text-decoration: overline;
}
h2 {
  text-decoration: line-through;
}
h3 {
  text-decoration: underline;
}
```

**This is heading 1**  
~~This is heading 2~~  
This is heading 3

65

**Text và Font (tt)**

- Định chiều cao của dòng văn bản: **line-height**

```
<style>
p.small { line-height: 70%; }
p.big { line-height: 200%; }
</style>
```
- Thụt đầu dòng văn bản: **text-indent**

```
p {
  text-indent: 50px;
}
```

67

**Text và Font (tt)**

- Định khoảng cách các ký tự: **letter-spacing**

```
<style>
h1 { letter-spacing: 3px; }
h2 { letter-spacing: -3px; }
</style>
```

**This is heading 1**  
**This is heading 2**
- Định khoảng cách giữa các từ: **word-spacing**.
 

```
<style>
.two { word-spacing: 20px; }
</style>
```

Văn bản bình thường  
 Khoảng cách giữa các từ là 20 pixel

66


**Text và Font (tt)**

- Chuyển đổi kiểu chữ: **text-transform**
  - Gồm các giá trị: none, uppercase, lowercase, capitalize
  - Ví dụ:
 

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

THIS IS SOME TEXT.  
 this is some text.  
 This Is Some Text.

68


**Text và Font (tt)**

- Tạo bóng cho chữ: **text-shadow**
  - Cú pháp: **text-shadow: h-shadow v-shadow blur-radius color|none|initial|inherit;**

```

<style>
.shadow {text-shadow: 5px 5px #aaa;}
.shadow-blur-radius {text-shadow: 5px 5px 5px #aaa;}
</style>


<body>
<h1 class="shadow">text shadow no blur</h1>
<h1 class="shadow-blur-radius">text shadow blur-radius</h1>
</body>

```

text shadow no blur

text shadow blur-radius

69


**Text và Font (tt)**

- Thiết lập kích thước font chữ cho văn bản: **font-size**

Dạng chuỗi	Đơn vị là em	Đơn vị là px	Đơn vị là pt	Theo tỷ lệ %
xx-small	.50 em	8 px	6 pt	50 %
x-small	.60 em	11px	8 pt	60 %
small	.75 em	13 px	10 pt	75 %
medium	1 em	16 px	12 pt	100 %
large	1.15 em	18 px	13.5 pt	110 %
x-large	1.5 em	24 px	18 pt	150 %
xx-large	2 em	30 px	24 pt	200 %

71


**Text và Font (tt)**

- Thiết lập font chữ cho văn bản: **font-family**
  - Cú pháp:

```

selector {
font-family: "font name 1", fontname2, fontname3;
}


```
  - Nếu tên font dài, có khoảng trắng phải đặt trong dấu nháy kép ("")
  - Thứ tự ưu tiên font chữ được áp dụng được xếp theo thứ tự liệt kê (các font cách nhau dấu phẩy), nếu "font name 1" không có trong hệ thống → áp dụng fontname2,...
  - Ví dụ:

```

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

```

70


**Text và Font (tt)**

- font-size: theo đơn vị là pixel**

```

h1 { font-size: 40px; }
h2 { font-size: 30px; }
p { font-size: 14px; }

```

This is heading 1

This is heading 2

This is a paragraph.

This is another paragraph.
- font-size: theo đơn vị là em**

```

h1 { font-size: 2.5em; /* 40px/16=2.5em */ }
h2 { font-size: 1.875em; /* 30px/16=1.875em */ }
p { font-size: 0.875em; /* 14px/16=0.875em */ }

```


This is heading 1

This is heading 2

This is a paragraph.

Specifying the font-size in em allows all major browsers with older versions of IE. When resizing the text, it beco

72



**Text và Font (tt)**

- Thiết lập chế độ in đậm/không in đậm: **font-weight**
  - Gồm các giá trị: normal, bold,... hoặc giá trị nguyên (100-900, trong đó 700 = bold)


```
p.normal { font-weight: normal; }
p.thick { font-weight: bold; }
p.thicker { font-weight: 900; }
```

This is a paragraph.

**This is a paragraph.**

**This is a paragraph.**


73



**3.3. Định dạng trang web với CSS**

- Màu và nền
- Text và Font
- Liên kết**
- Danh sách
- Box Model
- Bảng
- CSS Display
- CSS Position
- CSS Float & Clear
- Layers
- Giới thiệu CSS3

75



**Text và Font (tt)**

- Thiết lập kiểu font chữ cho văn bản: **font-style**
  - Gồm các giá trị: normal, italic, oblique (in xiên- ít hỗ trợ)
- Thiết lập kiểu chữ in hoa hay thường: **font-variant**
  - Gồm các giá trị: normal, small-caps, initial, inherit

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

This is a paragraph in normal style.


*This is a paragraph in italic style.*

*This is a paragraph in oblique style.*

**PLAY WITH THE TWO DIFFERENT FONT VARIANTS!**

CHANGE FONT VARIANT BY CLICKING THE RADIOBUTTONS

74



**Định dạng liên kết**

- CSS cho phép thiết lập style của liên kết với nhiều hiệu ứng khác nhau như màu chữ, màu nền, font chữ,...
- Có thể thiết lập style cho liên kết trong nhiều trạng thái khác nhau như:
  - a:link – liên kết chưa được thăm
  - a:visited – liên kết đã được thăm
  - a:hover – con trỏ chuột đang nằm trên liên kết
  - a:active – khi nhấn giữ chuột trên liên kết
- Lưu ý: cần đảm bảo theo thứ tự liệt kê như trên

76

**TRƯỜNG ĐẠI HỌC MỸ TP. HCM**  
Cơ hội học tập cho mọi người

## Định dạng liên kết (tt)

```

a { font-size:1.1em; font-weight:bold; }
a:link { color:#0000ff; text-decoration:none; }
a:visited { color:#333399;text-decoration:none; }
a:hover {
  background-color:#ffff00;
  text-decoration:underline;
  font-size: 1.5em;
}
a:active { color:#0033ff;text-decoration:underline; }

```

**Ví dụ sử dụng CSS Links**

- [Web site Cùng học lập trình](#) (liên kết khi đưa con trỏ chuột)
- [Web site học lập trình W3C](#) (liên kết ở trạng thái mặc định)
- [Tìm kiếm với Google](#) (liên kết đã thăm)

77

**TRƯỜNG ĐẠI HỌC MỸ TP. HCM**  
Cơ hội học tập cho mọi người

## Định dạng danh sách

- Thay đổi ký hiệu đầu dòng: **list-style-type**
  - Ví dụ:
 

```

ul.a {
  list-style-type: circle;
}
ul.b {
  list-style-type: square;
}
ol.c {
  list-style-type: upper-roman;
}
ol.d {
  list-style-type: lower-alpha;
}

```

**Example of unordered lists:**

- o Coffee
- o Tea
- o Coca Cola

**Example of ordered lists:**

- I Coffee
- II Tea
- III Coca Cola

79

**TRƯỜNG ĐẠI HỌC MỸ TP. HCM**  
Cơ hội học tập cho mọi người

## 3.3. Định dạng trang web với CSS

- Màu và nền
- Text và Font
- Liên kết
- **Danh sách**
- Box Model
- Bảng
- CSS Display
- CSS Position
- CSS Float & Clear
- Layers
- Giới thiệu CSS3

78

**TRƯỜNG ĐẠI HỌC MỸ TP. HCM**  
Cơ hội học tập cho mọi người

## Định dạng danh sách

- Các thuộc tính định dạng danh sách trong CSS

Thuộc tính	Ý nghĩa
list-style-type	Loại ký hiệu cho các phần tử trong danh sách
list-style-position	Thiết lập thuộc tính thụt lề của danh sách
list-style-image	Thiết lập ký hiệu đầu dòng là tập tin hình
list-style	Thiết lập một cách ngắn gọn danh sách các thuộc tính bên trên

80

**TRƯỜNG ĐẠI HỌC MỸ TP. HCM**  
Cơ hội học tập cho mọi người

## Định dạng danh sách (tt)

- Thiết lập loại ký hiệu đầu dòng: **list-style-type**
  - Đối với danh sách không thứ tự:

Giá trị	Loại ký hiệu
none	Không có
disc (mặc định)	Hình tròn có tô nền
circle	Hình tròn không tô nền
square	Hình vuông có tô nền

81

**TRƯỜNG ĐẠI HỌC MỸ TP. HCM**  
Cơ hội học tập cho mọi người

## Định dạng danh sách (tt)

- list-style-type**

```
<style>
li.a {list-style:none;}
li.b {list-style:disc;}
li.c {list-style:circle;}
li.d {list-style:square;}
li.e {list-style:decimal;}
li.f {list-style:lower-alpha;}
li.g {list-style:upper-alpha;}
li.h {list-style:lower-roman;}
li.i {list-style:upper-roman;}
</style>
```

Không có ký hiệu đầu dòng

  - Ký hiệu là disc
  - Ký hiệu là circle
  - Ký hiệu là square
  - Ký hiệu là số thứ tự thường
  - Ký hiệu là chữ cái thường
  - Ký hiệu là chữ cái in hoa
  - Ký hiệu là số la mã thường
  - Ký hiệu là số la mã in hoa

83

**TRƯỜNG ĐẠI HỌC MỸ TP. HCM**  
Cơ hội học tập cho mọi người

## Định dạng danh sách (tt)

- list-style-type**: danh sách có thứ tự:

Giá trị	Loại ký hiệu	Ví dụ
decimal	Số thứ tự	1,2,3,4,5,...
decimal-leading-zero	Số thứ tự trước có '0'	01,02,03,04,...
lower-alpha	Chữ cái thường	a,b,c,...
upper-alpha	Chữ cái in hoa	A,B,C,...
lower-roman	Chữ cái la mã thường	i,ii,iii,iv,...
upper-roman	Chữ cái la mã in hoa	I,II,III,IV,...

82

**TRƯỜNG ĐẠI HỌC MỸ TP. HCM**  
Cơ hội học tập cho mọi người

## Định dạng danh sách (tt)

- Thiết lập vị trí ký hiệu đầu dòng: **list-style-position**
  - Gồm hai giá trị là inside (ký hiệu nằm bên trong khối) và outside (ký hiệu nằm bên ngoài khối)

```
ul.a {
  list-style-position: outside;}
ul.b {
  list-style-position: inside;
}
```

Outside:

- Coffee
- Tea
- Coca-cola

Inside:

- Coffee
- Tea
- Coca-cola

84

**TRƯỜNG ĐẠI HỌC MỞ TP. HCM**  
Cơ hội học tập cho mọi người

## Định dạng danh sách (tt)

- Thiết lập ký hiệu đầu dòng là tập tin hình: **list-style-image**

```
ul {
  list-style-image: url('check.png');
}
```
- Thuộc tính **list-style** có thể viết ngắn gọn hơn, ví dụ:

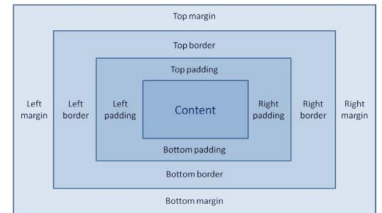
```
ul {
  list-style: square inside url("check.png");
}
```
- Thứ tự các thuộc tính phải được sắp xếp theo trình tự: list-style-type, list-style-position, list-style-image

85

**TRƯỜNG ĐẠI HỌC MỞ TP. HCM**  
Cơ hội học tập cho mọi người

## CSS Box Model

- Mô hình hộp gồm các thành phần sau:
  - Content: Vùng nội dung bên trong khung
  - Padding: vùng trống bao quanh nội dung (trong suốt)
  - Border: đường biên bao quanh phần padding và content
  - Margin: vùng trống bên ngoài bao quanh border



87

**TRƯỜNG ĐẠI HỌC MỞ TP. HCM**  
Cơ hội học tập cho mọi người

## 3.3. Định dạng trang web với CSS

- Màu và nền
- Text và Font
- Liên kết
- Danh sách
- Box Model**
- Bảng
- CSS Display
- CSS Position
- CSS Float & Clear
- Layers
- Giới thiệu CSS3

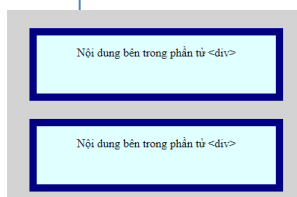
86

**TRƯỜNG ĐẠI HỌC MỞ TP. HCM**  
Cơ hội học tập cho mọi người

## CSS Box Model (tt)

- Để tạo Box Model, thường sử dụng phần tử `<div>`

```
<head> <style>
body { background-color: lightgrey; }
div {
  background-color: lightcyan;
  width: 300px; height: 100px; padding: 15px;
  border: 10px solid navy;
  margin: 25px; text-align: center;
}
</style></head>
<body>
<div>Nội dung bên trong phần tử <div></div>
<div>Nội dung bên trong phần tử <div></div>
</body>
```



88

## CSS Box Model (tt)

- **Thuộc tính border:** đường viền bao quanh đối tượng
  - **border-color:** màu đường viền.
  - **border-width:** độ dày đường viền
    - Các giá trị thin, medium, thick hoặc giá trị số với đơn vị là px.
    - Độ dày cho từng đường viền: border-bottom-width, border-right-width, border-top-width, border-left-width.
  - **border-style:** kiểu đường viền, với các giá trị thường dùng như sau:
    - none (không có đường viền)
    - solid (đường viền liền nét)
    - dotted (đường viền là dây các dấu chấm)
    - dashed (đường viền là các đường gạch đứt nét)
    - double (đường viền hai gạch)
- Có thể khai báo ngắn gọn dạng: `div {border:10px solid red;}`

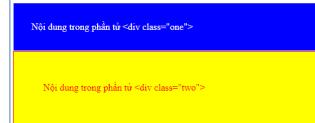
89

89

## CSS Box Model (tt)

- **Thuộc tính padding:** khoảng cách giữa phần tử HTML và nội dung của nó

```
<head>
<style>
  div { border: solid 1px; }
  div:first-child { color: white; background: blue; padding: 30px; }
  div:last-child { color: red; background: yellow; padding: 50px; }
</style>
</head>
<body>
  <div> ... </div>
  <div> ... </div>
</body>
```



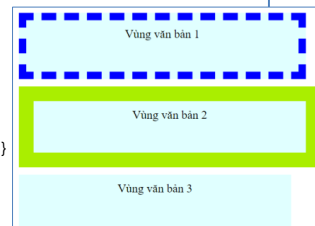
91

91

## CSS Box Model (tt)

- **Thuộc tính border:** ví dụ

```
<head>
<style>
  body { background-color: lightgrey; }
  div { background-color: lightcyan;
        width: 350px; height: 50px;
        padding: 10px; margin: 10px; text-align: center; }
  div:first-child { border-style: dashed;
                    border-width: 10px; border-color: blue; }
  div:nth-child(2) { border: 20px solid #aeee00; }
</style>
</head>
<body>
  <div>...</div> <div>...</div> <div>...</div>
</body>
```



90


90

## CSS Box Model (tt)

- **Thuộc tính margin:** khoảng cách giữa 2 phần tử HTML, có các giá trị:
  - margin: khoảng cách giữa phần tử và các phần tử xung quanh:
    - margin: 25px 50px 75px 100px; (top, right, bottom, left)
    - margin: 25px 50px 75px; (top, (right – left), bottom)
    - margin: 25px 50px; ((top – bottom), (right – left))
    - margin: 25px; (áp dụng cho tất cả các chiều)
    - margin: 0px auto; (phần tử nằm giữa phần tử chứa nó)
  - margin-left: khoảng cách giữa phần tử và phần tử bên trái nó
  - margin-top: khoảng cách giữa phần tử và phần tử trên nó
  - margin-right: khoảng cách giữa phần tử và phần tử bên phải nó
  - margin-bottom: khoảng cách giữa phần tử và phần tử dưới nó

92

92



**CSS Box Model (tt)**

---

- **Thuộc tính margin:** ví dụ

```

<head>
<style>
  div { border: solid 1px; }
  div:nth-child(2) { margin-top: 80px; margin-bottom: 40px; }
</style>
</head>
<body>
<div> ... </div>
<div> ... </div>
<div> ... </div>
</body>
```

Nội dung trong phần tử <div>


---

Nội dung trong phần tử <div class="two">

---

Nội dung trong phần tử <div>

93



**CSS Box Model (tt)**

---


- **Thuộc tính outline:** ví dụ

```

<head>
<style>
  p { outline: 5px double red; }
</style>
</head>
<body>
<p>Vùng văn bản có sử dụng thuộc tính outline</p>
</body>
```

Vùng văn bản có sử dụng thuộc tính outline

95




**CSS Box Model (tt)**

---

- **Thuộc tính outline:** thiết lập các dạng đường viền bao ngoài phần tử HTML, gồm các thuộc tính có giá trị tương tự như thuộc tính border.
  - outline
  - outline-color
  - outline-style
  - outline-width

94




**CSS Box Model (tt)**

---

- **Chiều rộng, chiều cao hộp:** các thuộc tính
  - height: chiều cao
  - width: chiều rộng
  - line-height: kích thước chiều cao cho mỗi dòng văn bản trong hộp
  - max-height: chiều cao tối đa của hộp
  - min-height: chiều cao tối thiểu
  - max-width: chiều rộng tối đa
  - min-width: chiều rộng tối thiểu

96



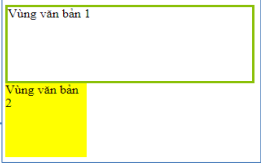


## CSS Box Model (tt)


- **Chiều rộng, chiều cao hộp**

```

<head>
<style>
  div: first-child { width: 300px; height: 100px; border: 3px solid #8AC007; }
  div: last-child { max-width: 100px; min-height: 100px; background-color: yellow; }
</style>
</head>
<body>
<div>Vùng văn bản 1</div>
<div>Vùng văn bản 2</div>
</body>
  
```




97



## CSS Tables

- Các thuộc tính thường được sử dụng:
  - Các thuộc tính về font chữ.
  - border: đường viền của bảng.
  - padding: khoảng cách giữa đường viền của ô và nội dung trong ô.
  - text-align: căn chỉnh văn bản trong ô theo chiều ngang.
  - vertical-align: căn chỉnh văn bản trong ô theo chiều dọc.
  - width: chiều rộng của bảng hoặc ô.
  - height: chiều cao của ô.
  - background-color: màu nền của bảng hoặc ô.
  - background-image: hình nền của bảng hoặc ô.
- Đối với phần tử <tr>: chỉ áp dụng thuộc tính màu và chiều cao


99



## 3.3. Định dạng trang web với CSS

- Màu và nền
- Text và Font
- Liên kết
- Danh sách
- Box Model
- **Bảng (Table)**
- CSS Display
- CSS Position
- CSS Float & Clear
- Layers
- Giới thiệu CSS3

98



## CSS Tables (tt)


- Thiết lập màu sắc cho đường biên, nền

```

<style>
table { border-collapse: collapse; }
table, th, td { border: 1px solid gray; padding: 5px; }
tr:nth-child(odd) { background-color: yellow; }
tr:nth-child(even) { background-color: cyan; }
th { background-color: blue; color: white; }
</style>
  
```

Môn học	Số tín chỉ	Học phí
Ứng dụng Web	3	1.500.000
Cơ sở lập trình	4	2.000.000
Kỹ thuật lập trình	4	2.000.000
Lập trình Web	3	1.500.000

100



## CSS Tables (tt)


- Thiết lập khoảng cách giữa các ô: **border-spacing**
  - Lưu ý:** giá trị của thuộc tính border-spacing
    - nếu chỉ có một giá trị: khoảng cách giữa một ô với ô bên cạnh và ô dưới nó đều bằng giá trị đó
    - nếu có hai giá trị: giá trị thứ nhất là khoảng cách giữa một ô và ô cạnh nó, còn giá trị thứ hai là khoảng cách giữa ô đó và ô dưới nó.

table: first-child { border-spacing: 15px; }
table: last-child { border-spacing: 2px 15px; }
td { padding: 5px; background-color: #aaa; border: 1px solid; }

**Thuộc tính border-spacing**

table one: cell a	table one: cell b
table one: cell c	table one: cell d
table two: cell a	table two: cell b
table two: cell c	table two: cell d

101




## CSS Tables (tt)

- Thuộc tính **empty-cells**:
  - Quy định ô rỗng có được hiển thị đường viền hay không
  - Có ba giá trị: show (hiển thị đường viền), hide (không hiển thị đường viền), inherit (kế thừa từ phần tử cha).

table { empty-cells: hide; }
td { padding: 5px; border: 1px solid; }

nội dung trong ô 1,1		nội dung trong ô 1,3
nội dung trong ô 1,4	nội dung trong ô 1,5	

103



## CSS Tables (tt)


- border-spacing: ví dụ**

table: first-child { border-spacing: 15px; }
table: last-child { border-spacing: 2px 15px; }
td { padding: 5px; background-color: #aaa; border: 1px solid; }

**Thuộc tính border-spacing**

table one: cell a	table one: cell b
table one: cell c	table one: cell d
table two: cell a	table two: cell b
table two: cell c	table two: cell d

102



## CSS Tables (tt)

- Thuộc tính **border-collapse**:
  - Cho/không cho phép gộp đường viền của bảng thành một đường đơn.
  - Có hai giá trị:
    - collapse
    - separate (mặc định).

```
<style>
table, td { margin-bottom: 20px; border: 2px solid; }
table.one { border-collapse: collapse; }
table.two { border-collapse: separate; }
</style>
```

104

**CSS Tables**

**table với border-collapse: collapse**

```
<h2>table với border-collapse: collapse</h2>
<table class="one">
<tr> <td >...</td> <td >...</td> <td >...</td> </tr>
<tr> <td >...</td> <td >...</td> <td >...</td> </tr>
</table>
<h2>table với border-collapse: separate</h2>
<table class="two">
<tr> <td >...</td> <td >...</td> <td >...</td> </tr>
<tr> <td >...</td> <td >...</td> <td >...</td> </tr>
</table>
<h2>table không thiết lập thuộc tính border-collapse</h2>
<table>
<tr> <td >...</td> <td >...</td> <td >...</td> </tr>
<tr> <td >...</td> <td >...</td> <td >...</td> </tr>
</table>
```

**table với border-collapse: collapse**

cell a collapse	cell b collapse
cell b collapse	cell a collapse

**table với border-collapse: separate**

cell a separate	cell b separate
cell b separate	cell a separate

**table không thiết lập thuộc tính border-collapse**

cell a	cell b
cell b	cell a

105

**CSS Tables (tt)**

- Canh lề văn bản theo chiều ngang: **text-align**

```
table, td, th {
  border: 1px solid black;
}
table {
  width: 100%;
}
th {
  text-align: left;
}
```

Firstname	Lastname	Savings
Peter	Griffin	\$100
Lois	Griffin	\$150
Joe	Swanson	\$300
Cleveland	Brown	\$250

107

**CSS Tables (tt)**

- Thiết lập chiều rộng, chiều cao: **width, height**

```
table, td, th {
  border: 1px solid black;
}
table {
  width: 100%;
}
th {
  height: 50px;
}
```

Firstname	Lastname	Savings
Peter	Griffin	\$100
Lois	Griffin	\$150
Joe	Swanson	\$300
Cleveland	Brown	\$250

106

**CSS Tables (tt)**

- Canh lề văn bản theo chiều dọc: **vertical-align**

```
table, td, th {
  border: 1px solid black;
}
td {
  height: 50px;
  vertical-align: bottom;
}
```

Firstname	Lastname	Savings
Peter	Griffin	\$100
Lois	Griffin	\$150
Joe	Swanson	\$300
Cleveland	Brown	\$250

108

**CSS Tables (tt)**

- Khoảng cách từ biên của ô đến nội dung: **padding**

```
table, td, th {
  border: 1px solid black;
}

td {
  padding: 15px;
}
```

Firstname	Lastname	Savings
Peter	Griffin	\$100
Lois	Griffin	\$150
Joe	Swanson	\$300
Cleveland	Brown	\$250

109

**3.3. Định dạng trang web với CSS**

- Màu và nền
- Text và Font
- Liên kết
- Danh sách
- Box Model
- Bảng
- CSS Display**
- CSS Position
- CSS Float & Clear
- Layers
- Giới thiệu CSS3

111

**CSS Tables (tt)**

- Thiết lập vị trí cho tiêu đề bảng: **caption-side** (HTML5):

```
table, td, th {
  border: 1px solid black;
}

caption {
  caption-side: bottom;
}
```

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Berglunds snabbköp	Christina Berglund	Sweden
Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK

Table 1.1 Customers

110

**CSS Display**

- Mỗi phần tử HTML có hai cách hiển thị mặc định
  - inline**: hiển thị trên cùng một hàng (không xuống dòng), bao gồm các phần tử như: `<span>`, `<a>`, `<img>`
  - block**: hiển thị theo khối (các phần tử không nằm trên cùng một hàng), bao gồm các phần tử như: `<div>`, `<h1>` - `<h6>`, `<p>`, `<form>`, `<header>`, `<footer>`, `<section>`, `<ul>`, `<ol>`, `<li>`
- Thay đổi cách hiển thị: sử dụng thuộc tính **display** với các giá trị:
  - `none`: không hiển thị
  - `inline`, `block`, `inline-block`
  - CSS3: `inline-table`, `table-row`, `list-item`, `flex`, `grid`

112

**CSS Display (tt)**

```
<head> <style>
div { display: inline-block; border: 1px solid; width: 100px; height: 50px; }
li {display: inline;}
span {display: block;}
p {display: none;}
</style></head>
<body>
<ul type="none"> <li>Trang chủ |</li> <li>Giới thiệu |</li> <li>Liên hệ</li> </ul>
<div>1</div> <div>2</div> <div>3</div>
<span>Đây là nội dung trong phần tử &lt;span>&gt;</span>
Đây là nội dung trang web
<p>Đây là nội dung trong phần tử &lt;p>&gt;</p>
</body>
```

113

113

**3.3. Định dạng trang web với CSS**

- Màu và nền
- Text và Font
- Liên kết
- Danh sách
- Box Model
- Bảng
- CSS Display
- CSS Position**
- CSS Float & Clear
- Layers
- Giới thiệu CSS3

115

115

**CSS Display (tt)**

**Ví dụ**

Trang chủ | Giới thiệu | Liên hệ

1 2 3

Đây là nội dung trong phần tử <span> Đây là nội dung trang web

Đây là nội dung trong phần tử <p>

thiết lập thuộc tính display như trong code

li {display: inline;} span {display: block;}

nếu không sử dụng thuộc tính display

114


114

**CSS Position**

- Mặc định, các phần tử HTML được sắp xếp theo thứ tự trong mã HTML.
- Tùy biến cách sắp xếp: sử dụng thuộc tính **position** với các giá trị:
  - static**: mặc định.
  - fixed**: vị trí cố định và không di chuyển khi cuộn trang web
  - absolute**: định vị tuyệt đối với các giá trị left, right, top, bottom theo phần tử chứa nó, vị trí phần tử không ảnh hưởng đến các phần tử khác.
  - relative**: định vị tương đối với các giá trị left, right, top, bottom. Vị trí phần tử được xác định so với các phần tử liên quan với nó, đồng thời nó chiếm giữ một khoảng không gian trên trang web theo thứ tự trong mã HTML → khi di chuyển phần tử này, nó sẽ ảnh hưởng đến các phần tử khác hoặc ngược lại.

116

116



## CSS Position (tt)

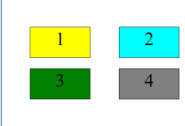
---

• **position: absolute**

```


<head> <style>
  div{ position: absolute; width: 100px; height: 80px;
    border: 1px solid; font-size: 2em; text-align: center;
  }
  div:first-child { top: 50px; left: 50px; background-color: yellow;}
  div:nth-child(2) { top: 50px; left: 200px; background-color: cyan;}
  div:nth-child(3) { top: 120px; left: 50px; background-color: green;}
  div:last-child{ top: 120px; left: 200px; background-color: gray;}
</style></head>
<body>
  <div>1</div> <div> 2 </div><div>3</div> <div> 4 </div>
</body>

```



117

117



## 3.3. Định dạng trang web với CSS


---

- Màu và nền
- Text và Font
- Liên kết
- Danh sách
- Box Model
- Bảng

- CSS Display
- CSS Position
- CSS Float & Clear**
- Layers
- Giới thiệu CSS3

119

119



## CSS Position (tt)

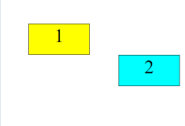
---

• **position: relative**

```


<head> <style>
  div{ position: relative; width: 100px; height: 80px;
    border: 1px solid; font-size: 2em; text-align: center;
  }
  div:first-child { top: 50px; left: 50px; background-color: yellow;}
  div:nth-child(2) { top: 50px; left: 200px; background-color: cyan;}
  div:nth-child(3) { top: 120px; left: 50px; background-color: green;}
  div:last-child{ top: 120px; left: 200px; background-color: gray;}
</style></head>
<body>
  <div>1</div> <div> 2 </div><div>3</div> <div> 4 </div>
</body>

```



118

118



## CSS Float và Clear

---

• **Thuộc tính float:**

- Tách một thành phần bất kỳ khỏi cách hiển thị thông thường để tạo chỗ cho những thành phần phía sau di chuyển lên và lấp đầy chỗ trống nếu có.
- Ví dụ: hình ảnh và văn bản bao quanh phần tử
- Thuộc tính này có ba giá trị:
  - left:** cố định phần tử về bên trái, các phần tử bên dưới sẽ được đẩy lên lấp đầy phần trống bên phải nó.
  - right:** cố định phần tử về bên phải, các phần tử bên dưới sẽ được đẩy lên lấp đầy phần trống bên trái nó.
  - none:** bình thường, đây là giá trị mặc định.

120

120

**CSS Float và Clear (tt)**

- Thuộc tính clear:** được sử dụng để xóa ảnh hưởng mà thuộc tính float đã áp đặt, gồm các giá trị: left, right, both.

```
<head>
<style>
div { float:left; width:100px; height: 80px;
border:2px solid black; padding:5px; margin:5px;
}
div:last-child { clear: left; background-color: yellow; }
</style>
</head>
<body>
<h1>Ví dụ về thuộc tính float - clear</h1>
<div>1</div> <div>2</div> <div>3</div> <div>4</div>
</body>
```

Ví dụ về thuộc tính float - clear

121

**3.3. Định dạng trang web với CSS**

- Màu và nền
- Text và Font
- Liên kết
- Danh sách
- Box Model
- Bảng
- CSS Display
- CSS Position
- CSS Float & Clear
- Layers**
- Giới thiệu CSS3

123

**CSS Float và Clear (tt)**

- Thuộc tính overflow:** Khi sử dụng float, một phần tử con cao hơn phần tử chứa nó sẽ tràn ra ngoài → sử dụng **overflow:auto** cho phần tử chứa

```
<head>
<style>
div { border: 2px solid #8AC007; margin-top: 20px; }
div:first-child { overflow: auto; }
img { float: right; width: 80px; height: 100px; }
</style>
</head>
<body>
<div>...</div>
<div>...</div>
</body>
```

Có sử dụng thuộc tính overflow: auto

Không sử dụng thuộc tính overflow: auto

122

**CSS Layers**


- Trong trang HTML, để các phần tử nằm chồng lên nhau:
  - Gán cho mỗi phần tử một giá trị nguyên trong thuộc tính **z-index**
  - Phần tử có z-index lớn sẽ nằm trên các phần tử có z-index nhỏ hơn.
  - Thuộc tính này **không** áp dụng cho các phần tử có thuộc tính **position:static**

```
img { position: absolute;
left: 0px; top: 0px; z-index: -1;
}
```

This is a heading

CSS: Because the image has a z-index of -1, it will be placed behind the text.

124



### 3.3. Định dạng trang web với CSS


---

- Màu và nền
- Text và Font
- Liên kết
- Danh sách
- Box Model
- Bảng

- CSS Display
- CSS Position
- CSS Float & Clear
- Layers
- Giới thiệu CSS3**

125

125




### CSS3 (tt)

---

- CSS3 Text Effects: tạo bóng cho văn bản.
- CSS3 Display: với các giá trị mới như [inline-table](#), [list-item](#), [flex](#), [grid](#)
- CSS3 Fonts: cho phép sử dụng bất kỳ font chữ nào, các font này được tải về tự động.
- CSS3 Transforms: biến đổi hình dạng, kích thước, vị trí phần tử.
- CSS3 Animation: tạo hình ảnh động, thay thế cho gif, flash, javascript.
- CSS3 Transition: thêm hiệu ứng cho phần tử, thay thế cho flash, javascript.
- CSS3 Multiple Columns: cho phép khai báo số lượng cột, độ rộng, khoảng cách từng cột
- → Tham khảo thêm trên <http://www.w3schools.com/> hoặc <https://developer.mozilla.org>

127

127




### Giới thiệu CSS3

---

- CSS3 được hỗ trợ tốt trong các trình duyệt mới như IE9, Firefox 4.0, Chrome 5.0, Opera 10.5,... các phiên bản sau
- Các thuộc tính CSS3 trên một số trình duyệt cần có thêm các tiền tố **-moz-** (Mozilla), **-webkit-** (WebKit: Chrome, Safari), **-o-** (Opera).
- Các đặc tính mới của CSS3:
  - CSS3 Borders - Rounded Corners: tạo đường viền tròn góc cho đối tượng.
  - CSS3 Border Image: sử dụng hình ảnh làm đường viền cho khung.
  - CSS3 Box Shadow: tạo bóng cho các phần tử dạng hình hộp.
  - CSS3 Background: thiết lập hình nền và kích thước hình nền.

126

126



### CSS3 (tt)


---

• <a href="#">animation</a>	• <a href="#">box-align</a>	• <a href="#">font-size-adjust</a>	• <a href="#">transform</a>
• <a href="#">appearance</a>	• <a href="#">box-direction</a>	• <a href="#">@keyframes</a>	• <a href="#">transform-origin</a>
• <a href="#">backface-visibility</a>	• <a href="#">box-flex</a>	• <a href="#">nav</a>	• <a href="#">transform-style</a>
• <a href="#">background-clip</a>	• <a href="#">box-ordinal-group</a>	• <a href="#">opacity</a>	• <a href="#">transition</a>
• <a href="#">background-origin</a>	• <a href="#">box-orient</a>	• <a href="#">perspective</a>	• <a href="#">word-break</a>
• <a href="#">background-size</a>	• <a href="#">box-pack</a>	• <a href="#">perspective-origin</a>	• <a href="#">word-wrap</a>
• <a href="#">background</a>	• <a href="#">box-sizing</a>	• <a href="#">resize</a>	
• <a href="#">border-image</a>	• <a href="#">box-shadow</a>	• <a href="#">text-justify</a>	
• <a href="#">border-radius</a>	• <a href="#">column</a>	• <a href="#">text-overflow</a>	
	• <a href="#">@font-face</a>	• <a href="#">text-shadow</a>	

128

128




**CSS3 Rounded Corners**


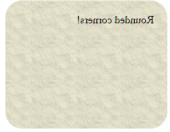
- Thuộc tính **border-radius**: tạo đường biên bo tròn các góc, áp dụng với các phần tử block

```


#rcorners1 {
  border-radius: 25px;
  background: #8AC007;
  padding: 20px;
  width: 200px;
  height: 150px;
}

#rcorners3 {
  border-radius: 25px;
  background: url(paper.gif);
  background-position: left top;
  background-repeat: repeat;
  padding: 20px;
  width: 200px;
  height: 150px;
}

```

129


**CSS3 Background**

- background-size**: ví dụ

```

.div1 {
  border: 1px solid black;
  height: 120px;
  width: 150px;
  background: url(img_flwr.gif);
  background-repeat: no-repeat;
  background-size: contain;
}

.div2 {
  border: 1px solid black;
  height: 120px;
  width: 150px;
  background: url(img_flwr.gif);
  background-repeat: no-repeat;
  background-size: cover;
}




.div3 {
  border: 1px solid black;
  height: 120px;
  width: 150px;
  background: url(img_flwr.gif);
  background-repeat: no-repeat;
}

```


background-size: contain:

background-size: cover:

No background-size defined:






131


**CSS3 Background**


- background-size**: thiết lập kích thước ảnh nền, gồm các giá trị:
  - chiều rộng, chiều cao
  - contain: hình tự resize để lấp đầy trong khung chứa và giữ nguyên tỷ lệ giữ chiều rộng, chiều cao
  - cover: hình lấp đầy để toàn bộ nội dung được phủ bởi hình nền

130


**CSS3 Background**

- background-origin**: chỉ định vị trí của hình nền, gồm các giá trị:
  - border-box: vị trí hình ở góc trên trái của phần border
  - padding-box: mặc định, vị trí hình ở góc trên trái của phần padding
  - content-box: vị trí hình ở góc trên trái của phần nội dung

132



**CSS3 Background**

- background-clip: ví dụ**

```
#example1 {
border: 10px dashed black;
padding: 25px;
background: url(paper.gif);
background-repeat: no-repeat;
background-origin: padding-box;
}
```

```
#example2 {
border: 10px dashed black;
padding: 25px;
background: url(paper.gif);
background-repeat: no-repeat;
background-origin: border-box;
}
```

```
#example3 {
border: 10px dashed black;
padding: 25px;
background: url(paper.gif);
background-repeat: no-repeat;
background-origin: content-box;
}
```

Hello World

The background image starts from the upper left corner of the padding edge.

Hello World


The background image starts from the upper left corner of the border.

Hello World

The background image starts from the upper left corner of the content.

133

133



**CSS3 Background**

- background-clip: ví dụ**

```
#example1 {
border: 10px dotted black;
padding: 35px;
background: yellow;
background-clip: padding-box;
}
```

```
#example2 {
border: 10px dotted black;
padding: 35px;
background: yellow;
background-clip: padding-box;
}
```

```
.div3 {
border: 1px solid black;
height: 120px;
width: 150px;
background: url(img_flwr.gif);
background-repeat: no-repeat;
}
```

Lorem Ipsum Dolor

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Lorem Ipsum Dolor


Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Lorem Ipsum Dolor

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

135

135




**CSS3 Background**

- background-clip: chỉ định vùng phủ của nền, gồm các giá trị:**
  - border-box: nền phủ đến hết vùng border
  - padding-box: nền phủ đến hết vùng padding
  - content-box: nền chỉ phủ vùng nội dung

134

134



**CSS3 Background**

- Background Gradients (CSS3):**
  - Hiện thị nền dưới dạng pha trộn màu
  - Có hai loại hàm: linear và radial
    - linear
      - linear-gradient()
      - repeating-linear-gradient()
    - radial
      - radial-gradient()
      - repeating-radial-gradient()


136

136

**CSS3 Background**

- **Background Gradients: linear**

```
.linear-gradient { background: linear-gradient(to left, red, blue); }
.diagonal-linear-gradient { background: linear-gradient(to left top, red, yellow 10%); }
.linear-gradient-rainbow {
  background: linear-gradient(to left, red, orange, yellow, green, blue, indigo, violet)
}
```



137

**CSS3 Border Images**

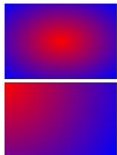
- **Thuộc tính border:** đường viền bao quanh đối tượng
  - **border-image:** dùng ảnh làm đường viền, bao gồm:
    - **border-image-source:** đường dẫn đến hình
    - **border-image-slice:** độ lệch được sử dụng để chia hình ảnh thành chín vùng (bốn góc, bốn cạnh và ở giữa)
    - **border-image-repeat:** gồm các giá trị:
      - repeat
      - round
      - space
      - stretch
  - Ví dụ: [https://www.w3schools.com/csSref/playdemo.asp?filename=playcss\\_border-image](https://www.w3schools.com/csSref/playdemo.asp?filename=playcss_border-image)

139

**CSS3 Background**

- **Background Gradients: radial**

```
.radial-gradient-simple { background: radial-gradient(red, blue); }
.radial-gradient { background: radial-gradient(circle farthest-corner at top left, red, blue); }
```




Giá trị	Ý nghĩa
circle	Hình dạng của gradient, giá trị là circle hoặc ellipse, mặc định là ellipse.
farthest-corner	Chỉ định hình dạng vùng kết thúc, với các giá trị: closest-side, farthest-side, closest-corner, farthest-corner
top left	Vị trí của tâm gradient

138

**CSS text-overflow**

- Chỉ định cách hiển thị nội dung văn bản: tràn, bị cắt, dấu chấm lửng (...) hoặc một chuỗi tùy chỉnh.
- Hai thuộc tính bắt buộc:
  - white-space: nowrap;
  - overflow: hidden;
- Các giá trị của thuộc tính text-overflow:
  - clip
  - ellipsis
  - một chuỗi bất kỳ (chỉ dùng trong firefox)

140



## CSS text-overflow

• Ví dụ:

```
div.a {
  white-space: nowrap;
  width: 50px;
  overflow: hidden;
  text-overflow: clip;
  border: 1px solid #000000;
}
```

```
div.b {
  white-space: nowrap;
  width: 50px;
  overflow: hidden;
  text-overflow: ellipsis;
  border: 1px solid #000000;
}
```


```
div.c {
  white-space: nowrap;
  width: 50px;
  overflow: hidden;
  text-overflow: "----";
  border: 1px solid #000000;
}
```

```
<body>
<div class="a">Hello world!</div>
<div class="b">Hello world!</div>
<div class="c">Hello world!</div>
</body>
```

Hello w  
 Hell...  
 Hell----

141

141




## Flexbox

- main start, main end: điểm bắt đầu và điểm kết thúc của container
- cross start, cross end: trục vuông góc với main start, main end.
- main axis: trục chính để điều khiển hướng mà các item sẽ hiển thị.
  - Mặc định là chiều ngang
  - Có thể sử dụng thuộc tính **flex-direction** để thay đổi trục của main axis với các giá trị: row, row-reverse, column, column-reverse
- cross axis: trục vuông góc của main axis.
- main size: kích thước của mỗi item theo chiều ngang (main axis)
- cross size: kích thước của mỗi item theo chiều dọc (cross axis).

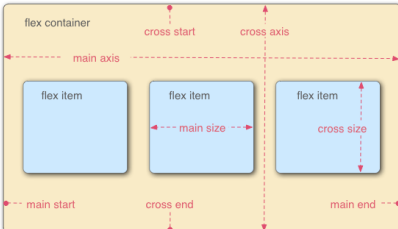
143

143



## Flexbox


- CSS3 cung cấp một kiểu dàn trang khá linh hoạt
- Sơ đồ cấu trúc Flexbox:
- Flexbox có hai thành phần chính:
  - Container
  - Item



Nguồn: [https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS\\_layout/Flexbox](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox)

142

142




## Flexbox

- container:
  - Phần tử chứa các phần tử khác
  - Có thể thiết lập cách hiển thị các phần tử bên trong theo chiều ngang hoặc chiều dọc
- item:
  - Các phần tử bên trong container
  - Có thể thiết lập cho phép item có độ rộng bằng bao nhiêu cột trong một container, hoặc thiết lập thứ tự hiển thị của nó.

144

144




## Flexbox

---

- Các thuộc tính thường dùng với thành phần container:
  - [flex-direction](#): row, column
  - [flex-wrap](#): wrap, nowrap
  - [flex-flow](#): cách viết ngắn gọn của flex-direction và flex-wrap
  - [justify-content](#)
  - [align-items](#)
  - [align-content](#)

145

145



## Flexbox

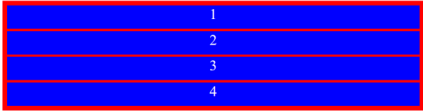
---

• Ví dụ:

```
<div class="container">
  <div class="item item1">1</div>
  <div class="item item2">2</div>
  <div class="item item3">3</div>
  <div class="item item4">4</div>
</div>
```


```
.container{
  background-color:red;
  max-width: 900px;
  max-height: 900px;
  margin: 0 auto;
  padding: 5px;
}

.item {
  background-color: blue;
  margin: 5px;
  color: white;
  height: 50px;
  text-align: center;
  font-size: 2em;
}
```



147

147




## Flexbox

---

- Các thuộc tính thường dùng với thành phần item:
  - [order](#)
  - [flex-grow](#)
  - [flex-shrink](#)
  - [flex-basis](#)
  - [flex](#)
  - [align-self](#)

146


146




## Flexbox

---

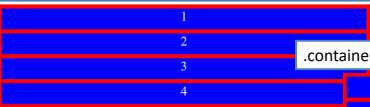
```
.container{ display:flex;....}
```



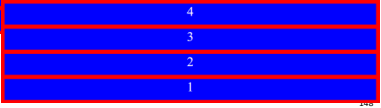
```
.container{ display:flex; flex-direction: row-reverse;....}
```



```
.container{ display:flex; flex-direction: column;...}
```



```
.container{ display:flex; flex-direction: column-reverse;...}
```



148

148

## CSS3 2D Transforms

- Thực hiện các phép biến đổi hình học (hình dạng, kích thước, vị trí)
- Sử dụng các phương thức:
  - translate(x,y): di chuyển đối tượng theo trục X và Y.
  - translateX(n): di chuyển đối tượng theo trục X.
  - translateY(n): di chuyển đối tượng theo trục Y.
  - scale(x,y): thay đổi kích thước của đối tượng theo chiều rộng và chiều cao.
  - scaleX(n): thay đổi kích thước của đối tượng theo chiều rộng.
  - scaleY(n): thay đổi kích thước của đối tượng theo chiều cao.
  - rotate(angle): xoay đối tượng xung quanh trọng tâm của nó theo một góc angle.
  - skew(x-angle,y-angle): xô lệch đối tượng theo hai trục X và Y.
  - skewX(angle): xô lệch đối tượng theo trục X một góc angle.
  - skewY(angle): xô lệch đối tượng theo trục Y một góc angle.
  - matrix(n,n,n,n,n,n): biến đổi đối tượng theo ma trận gồm sáu giá trị.

149

149

## Hoạt hình

- Thuộc tính transition:** thay đổi động hiệu ứng của phần tử với các tham số:
  - Loại hoạt hình (animation): xác định loại hoạt hình được áp dụng, gồm các giá trị: all (mặc định), color, length, width, percentage, opacity, number.
  - Thời gian hoạt hình (duration) tính bằng giây.
  - Tốc độ của hoạt hình: gồm các giá trị linear (tốc độ không thay đổi), ease (chậm→nhẹ→chậm), ease-in (bắt đầu chậm), ease-out (kết thúc chậm), ease-in-out (bắt đầu và kết thúc chậm).
  - Độ trì hoãn: thời gian bắt đầu áp dụng hoạt hình.

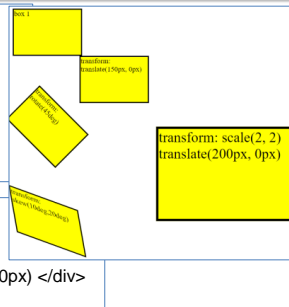
151

151

## CSS3 2D Transforms (tt)

```
#box1, #box2, #box3, #box4, #box5 {
  width: 150px; height: 100px;
  border: 3px solid black; background-color: yellow;
}
#box2 { transform: translate(150px, 0px); }
#box3 { transform: rotate(45deg); }
#box4 { transform: scale(2, 2) translate(200px, 0px); }
#box5 { transform: skew(10deg, 20deg); }
```

```
<div id = "box1">box 1</div>
<div id = "box2">transform: translate(150px, 0px)</div>
<div id = "box3">transform: rotate(45deg)</div>
<div id = "box4">transform: scale(2, 2) translate(200px, 0px) </div>
<div id = "box5">transform: skew(10deg,20deg)</div>
```



150

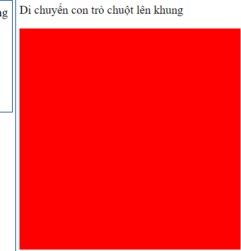
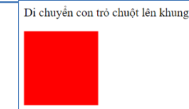
150

## Hoạt hình (tt)

- Thuộc tính transition:** ví dụ


```
div {
  width: 100px; height: 100px;
  background: red;
  transition: width 2s, height 4s;
}
div:hover { width: 300px; height: 300px; }
```

```
<body>
<p>Di chuyển con trỏ chuột lên khung</p>
<div></div>
</body>
```



152

152




**Hoạt hình (tt)**

- Thuộc tính animation:**

**animation: name duration timing-function delay iteration-count direction fill-mode play-state**
  - name: tên hoạt hình
  - duration: thời gian hoàn tất hoạt hình
  - timing-function: tốc độ
  - delay: độ trễ (chờ) trước khi bắt đầu
  - iteration-count: số lần hoạt hình
  - direction: hướng tiếp theo của hoạt hình (normal, reverse, alternate, alternate-reverse)
  - fill-mode: style cho phần tử khi hoạt hình không phát (none, forwards, backwards, both)
  - play-state: chỉ định chạy/tạm dừng hoạt hình (running, paused)

153



**Hoạt hình (tt)**


- Ví dụ: tạo năm khung chuyển động xoay vòng dựa vào vị trí từng khung ở mỗi thời điểm (0% - 100%):**

```

<head> <style>
  @keyframes anim {
    0% {left: 0px; top: 0px;}
    25% {left: 100px; top: 0px;}
    50% {left: 100px; top: 100px;}
    75% {left: 0px; top: 100px;}
    100% {left: 0px; top: 0px;}
  }
  #box { position: absolute;
    background-color: yellow;
    border: 1px solid black;
    animation: anim 5s linear infinite; }
</style></head>
<body> <div id = "box">Moving box</div></body>

```


155



**Hoạt hình (tt)**

- Các bước tạo hoạt hình cho phần tử:**
  - Tạo một animationname là tập hợp các khung hình
    - @keyframe animationname{ }
    - mỗi khung hình chỉ định một trạng thái của đối tượng (một cảnh trong hoạt hình).
  - Thiết lập giá trị phần trăm cho mỗi khung hình
    - khung hình đầu tiên là 0% (phần đầu của hoạt ảnh)
    - khung hình cuối cùng phải là 100% (phần cuối của hoạt ảnh)
    - các khung hình trung gian với các tỷ lệ % tăng dần.
    - Hoặc có thể dùng from {} (0%), to {} (100%)
  - Thiết lập style cho mỗi khung hình.
  - Áp dụng chuyển động cho phần tử: **animation**
  - Thiết lập các tham số chuyển động như thời gian, tốc độ, chế độ lặp lại của chuyển động...

154



**Giới thiệu Bootstrap**

- Các thiết bị kích thước và độ phân giải màn hình khác nhau: máy tính, máy tính bảng (Tablet), điện thoại di động (Mobile).
- Giao diện website cần phải tương thích với các thiết bị: tính đáp ứng (Responsive):
- Bootstrap: framework miễn phí, mã nguồn mở, dựa trên HTML, CSS, Javascript → giúp giao diện website đạt tính đáp ứng.
- Bao gồm các mẫu thiết kế dựa trên HTML, CSS để định dạng kiểu chữ, biểu mẫu (forms), nút (button), bảng (table), hình ảnh,... và hỗ trợ thêm khá nhiều plugin khác, như JavaScript.
- Được sử dụng phổ biến trên thế giới.

156

## Giới thiệu Bootstrap (tt)

- Ưu điểm:
  - Dễ sử dụng: vì Bootstrap được xây dựng trên HTML, CSS và Javascript.
  - Tính đáp ứng: Bootstrap đã xây dựng sẵn các "Responsive CSS" tương thích với các thiết bị khác nhau → chỉ cần tìm hiểu cú pháp sử dụng. Tính năng này giúp tiết kiệm rất nhiều thời gian để tạo giao diện website.
  - Tương thích với các trình duyệt: Chrome, Firefox, Internet Explorer, Safari, Opera..

157

157

## Giới thiệu Bootstrap (tt)

- Một số Bootstrap Components
  - container class
  - page-header class
  - Bootstrap Wells
  - Navigation Menu
  - Dropdown Menus
  - carousel slide
  - Các hộp thông báo (alerts)
  - Các nút lệnh (buttons)

159

159

## Giới thiệu Bootstrap (tt)

- Cách sử dụng:
  - Cách 1: tham chiếu trực tuyến đến Bootstrap từ CDN
  - Cách 2: Download Bootstrap từ [getbootstrap.com](https://getbootstrap.com)
- Hệ thống lưới trong Bootstrap:
  - Bootstrap Grid System cho phép thiết lập tối đa 12 cột trên trang.
  - Có bốn lớp tương thích các loại thiết bị như sau:
    - xs (dành cho điện thoại): chiều rộng màn hình < 768px.
    - sm (dành cho máy tính bảng): chiều rộng màn hình >= 768px.
    - md (dành cho máy tính xách tay nhỏ): chiều rộng màn hình >= 992px.
    - lg (dành cho máy tính để bàn và máy tính xách tay lớn): chiều rộng màn hình >= 1200px.

158

158