

# ỨNG DỤNG WEB

Nguyễn Thị Mai Trang

1

## Chương 3

### CSS

2

## 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

3

## Nội dung

1. Giới thiệu CSS
2. Các loại CSS
3. Cú pháp CSS
4. Định dạng trang web với CSS

4

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)

- 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 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)

- Ví dụ:
 

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



8

8

## 3.2. Các loại CSS

- 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.

9

9

## Các loại 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  
`</TagName>`
  - Ví dụ:
 

```

<!DOCTYPE html>
<html>
<body>
<h1 style="color:blue;margin-left:30px;">This is a heading.</h1>
<p>This is a paragraph.</p>
</body>
</html>

```

**This is a heading.**

This is a paragraph.

10

10

## Các loại 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.

11

11

## Các loại 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>
```

12

12

## Các loại CSS (tt)

- Internal Style Sheet: ví dụ

```
<head>
<style>
body {
  background-color: yellow;
}
h1 {
  color: blue;
  margin-left: 40px;
}
</style>
</head>
```

**This is a heading**

This is a paragraph.

13

13

## Các loại CSS (tt)

- External Style Sheet (tt)

- 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"/>
```

14

14

## Các loại 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:

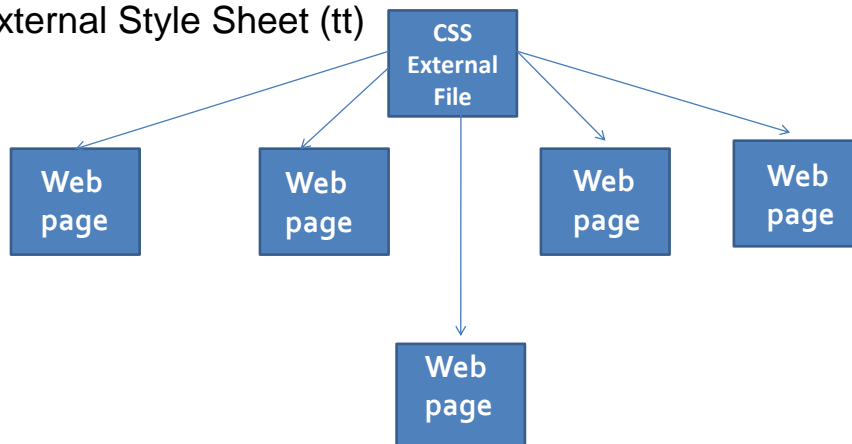
```
Selector { "property1: value1; property2: value2;...">
```

15

15

## Các loại CSS (tt)

- External Style Sheet (tt)



16

16



## Các loại CSS (tt)

- External Style Sheet (tt)
  - Ví dụ: file style.css lưu trong thư mục style, có cấu trúc như sau:
- Nội dung trang style.css

```
body {
    background-color: lightblue;
}
h1 {
    color: navy;
    margin-left: 20px;
}
```



17

17

## Các loại CSS (tt)

- External Style Sheet (tt)
  - Nội dung trang css.html: trong thẻ <head> khai báo như sau:

```
<head>
  <link rel="stylesheet" type="text/css" href=" style/style.css ">
</head>
<body>
  <h1>This is a heading</h1>
  <p>The style of this document is a combination of an external
  stylesheet, and internal style</p>
</body>
</html>
```

**This is a heading**

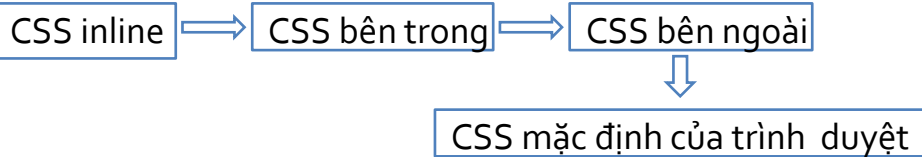
The style of this document is a combination of an external stylesheet, and internal style

18

18

## Các loại CSS (tt)

- Độ ưu tiên của CSS từ cao xuống thấp



- 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)

19

19

## Các loại 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

20

20

## Các loại 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>
```

21

21

## 3.3. Cú pháp CSS

- Một bộ chọn trong CSS:

Phần tử chọn

{thuộc tính: giá trị; thuộc tính: 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 **{thuộc tính: giá trị}**, nếu có nhiều thuộc tính thì các bộ cách nhau dấu chấm phẩy (;)
- Ví dụ:

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

22

22

## 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;
}
```

23

23

## 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 }
```

24

24

## 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 &lt;p &gt; </p>
  <p class="dong1"> Đoạn này áp dụng style của phần tử &lt;p &gt; và class dong1</p>
  <p class="dong2"> Đoạn này áp dụng style của phần tử &lt;p &gt; và class dong2</p>
</body>
```

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

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

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

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

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" ...>**

27

27

## 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 &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

28

28

## Các bộ chọn CSS

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

29

29

## Các bộ chọn CSS

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

30

30

## Các bộ chọn CSS

| Bộ chọn (Selector) | Mô tả  | Ví dụ   |
|--------------------|--|---|
| E + F              | Chọn phần tử F liền kề sau phần tử E         | <b>h1 + p {color: red;}</b><br>/* 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><br>/* 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><br>/* 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><br>/* thiết lập chữ in đậm cho các thẻ a có thuộc tính href="http://www.abc.com" */ |

31

31

## Các bộ chọn CSS

| 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><br>/* 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><br>/* 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><br>/* thiết lập màu chữ xanh dương cho các thẻ h1 mà thuộc tính id có ba ký tự cuối là "xyz" */             |

32

32



## Các bộ chọn CSS

| 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. | <code>div[id*="section"] {color: red;}</code><br>/* thiết lập màu chữ đỏ cho các thẻ div mà thuộc tính id chứa chuỗi "section" */   |
| :after<br>Hoặc ::after  | Thiết lập style áp dụng ngay sau phần tử chỉ định.                        | <code>div:after {content:url(banana.gif);}</code><br>Hoặc <code>div::after {content:url (banana.gif);}</code><br>/* chèn ảnh banana.gif ngay sau tất cả các thẻ div */            |
| :before<br>Hoặc :before | Thiết lập style áp dụng ngay trước phần tử chỉ định.                      | <code>div:before {content:url(banana.gif.gif);}</code><br>Hoặc <code>div::before {content:url(banana.gif.gif);}</code><br>/* chèn ảnh banana.gif ngay trước tất cả các thẻ div */ |

33

33

## Các bộ chọn CSS

| Bộ chọn (Selector) | Mô tả  | Ví dụ   |
|--------------------|--|---|
| :checked           | Chọn các phần tử có trạng thái được checked                                    | <code>:checked {color: red;}</code><br>/* 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 | <code>:default {background-color: red;}</code><br>/* 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)                      | <code>input:disabled {background-color: gray;}</code><br>/* thiết lập màu nền xám cho các thẻ input bị vô hiệu hóa */   |

34

34

## Các bộ chọn CSS

| Bộ chọn (Selector) | Mô tả  | Ví dụ  |
|--------------------|--|--|
| :empty             | Chọn phần tử rỗng                                      | <code>div:empty {display: none;}</code><br>/* 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><br>/* 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><br>/* thiết lập màu chữ đỏ cho các thẻ p đầu tiên nằm trong phần tử khác */        |
| :first-letter      | Chọn ký tử đầu trong nội dung của phần tử              | <code>p:first-letter {font-size: larger;}</code><br>/* thiết lập font chữ lớn (larger) cho các ký tự đầu trong các thẻ p */  |

35

35

## Các bộ chọn CSS

| Bộ chọn (Selector)               | Mô tả   | Ví dụ   |
|----------------------------------|---|---|
| :first-line<br>Hoặc ::first-line | Chọn dòng đầu tiên của phần tử  | <code>p:first-line {color: blue;}</code><br>Hoặc <code>p::first-line {color: red;}</code><br>/* 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><br>/* 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><br>/* thiết lập màu nền vàng cho phần tử input đang nhận focus */   |

36

36

## Các bộ chọn CSS

| Bộ chọn (Selector) | Mô tả  | Ví dụ  |
|--------------------|--|--|
| :hover             | Chọn phần tử đang nhận con trỏ chuột   | <b>p:hover {background-color: yellow;}</b><br>/* 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"  | <b>*:lang(fr) {color: blue;}</b><br>/* 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.                                | <b>p:last-child {font-size: small;}</b><br>/* 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. | <b>p:last-of-type {background-color: red;}</b><br>/* thiết lập màu nền đỏ cho các thẻ p cuối cùng trong mỗi phần tử khác */            |

37

37

## Các bộ chọn CSS

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

38

38

## Các bộ chọn CSS

| 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.           | <b>p:nth-last-of-type(2) {color:red;}</b><br>/* 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   | <b>h1:only-child {color: blue;}</b><br>/* 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 | <b>p:only-of-type {font-weight: bold;}</b><br>/*thiết lập font chữ in đậm cho thẻ p duy nhất nằm trong phần tử khác */   |

39

39

## Các bộ chọn CSS

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

40

40

## 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 */
```

41

41

## 3.4. Định dạng trang web với CSS

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

42

42

- Các dạng màu trong CSS
  - 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",...

43

43

- 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

44

44

## Màu và nền (tt)

- background-color: màu

### 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 < div >

Văn bản trong thẻ < body >

```
<head> <style>
  body{background-color: yellow;}
  h1 {background-color: #6495ed;}
  p {background-color: #e0fff;}
  div {background-color: #b0c4de;}
</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>
    Văn bản trong thẻ &lt; div &gt;
  </div>
  Văn bản trong thẻ &lt; body &gt;
</body>
```

45

45

## Màu và nền (tt)

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

```
body {
  background-image: url("paper.gif");
}
```

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

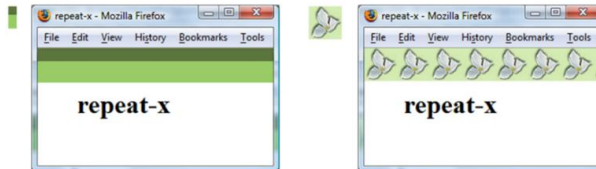
46

46

## Màu và nền (tt)

- **background-repeat:**

- repeat-x: lặp lại ảnh theo hướng ngang



```
body {
  background-image: url("bg.png");
  background-repeat: repeat-x;
}
```

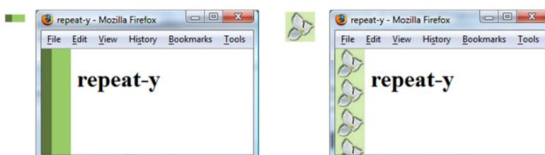
47

47

## Màu và nền (tt)

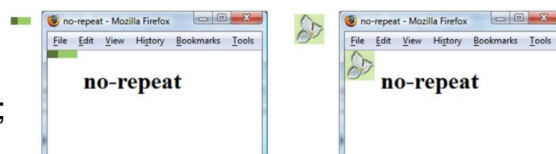
- **background-repeat (tt):**

- repeat-y: lặp lại ảnh theo hướng dọc



- no-repeat: không lặp

- background-image: url(bg.jpg);



48

48



## Màu và nền (tt)

- **background-position**: định vị trí cho phần tử

```
body {
  background-image: url("img_tree.png");
  background-repeat: no-repeat;
  background-position: right top;
}
```

### Hello World!

W3Schools background no-repeat, set position example.

Now the background image is only shown once, and positioned away from the text.

In this example we have also added a margin on the right side, so the background image will never disturb the text.



49

49

## Màu và nền (tt)

- **background-position**: các giá trị

| Giá trị | Ý nghĩa   |
|---------|---|
| x% y%   | Tỷ lệ phần trăm theo trục x (ngang) và y (trục dọc) |
| x y     | Chiều rộng, chiều cao theo đơn vị pixel             |
| left    | Nằm ở lề trái của trang hoặc phần tử chứa nó        |
| right   | Nằm ở lề phải của trang hoặc phần tử chứa nó        |
| center  | Nằm ở giữa trang hoặc phần tử chứa nó               |
| top     | Nằm ở lề trên của trang hoặc phần tử chứa nó        |
| bottom  | Nằm ở lề dưới của trang hoặc phần tử chứa nó        |

50

50

## 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("w3css.gif");
  background-repeat: no-repeat;
  background-attachment: fixed;
}
```

51

51

## Màu và nền (tt)

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

```
body {
  background: #ffffff url("img_tree.png") no-repeat right top;
}
```

- Thứ tự các thuộc tính:
 

background:<background-color> | <background-image> |  
 <background-repeat> | <background-attachment> | <background-position>

52

52

### 3.4. 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

53

53

### 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

54

54

## Text và Font (tt)

- Thiết lập màu chữ: color

```
<style>
  body {color:#0000ff; background-color:#ffff00;}
  span.color {color:#FF0000;}
</style>
```

**color**

Nền trang web có màu vàng

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

55

55

## 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>
```

56

56

## 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 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.

57

57

## 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>
.sub {vertical-align:sub;}
.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%;}
.pixel10 {vertical-align:10px;}
</style>
```

58

58

## 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

59

59

## 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

60

60

## Text và Font (tt)

- Định khoảng cách các dòng: **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;
}
```

61

61

## 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.

62

62

- Tạo bóng cho chữ: **text-shadow**

```
<style>
  h1 { text-shadow: 2px 2px #FF0000; }
</style>
```

**Text-shadow effect**

63

63

- 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 “”
- 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; }
```

64

64



## 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 %        |

65

65

## 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.

66

66

## Text và Font (tt)

- **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

67

67

## Text và Font (tt)

- **font-size:** kết hợp giữa em và %

```
body {
  font-size: 100%;
}
h1 { font-size: 2.5em; }
h2 { font-size: 1.875em; }
p { font-size: 0.875em; }
```

**This is heading 1**

**This is heading 2**

This is a paragraph.

Specifying the font-size in percent and em displays the same size to resize the text!

68

68

## 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.**

69

69

## 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)

```
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.*

70

70

## Text và Font (tt)

- Thiết lập kiểu chữ in hoa hay thường: **font-variant**
  - Gồm các giá trị: normal, small-caps, initial, inherit

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

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

CHANGE FONT VARIANT BY CLICKING THE RADIOBUTTONS

71

71

## 3.4. Định dạng trang web với CSS

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

72

72

## 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

73

73

## 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:active {color:#0033ff;text-decoration:underline;}
a:hover {
    background-color:#ffff00;
    text-decoration:underline;
    font-size: 1.5em;
}
  
```

### 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

74

74

### 3.4. 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

75

### 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:

- Coffee
- Tea
- Coca Cola
- Coffee
- Tea
- Coca Cola

Example of ordered lists:

- I. Coffee
- II. Tea
- III. Coca Cola
- a. Coffee
- b. Tea
- c. Coca Cola

76

76

## Định dạng danh sách

- Các thuộc tính định dạng danh sách trong CSS

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

77

77

## 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           |
|------------------------|------------------------|
| <b>none</b>            | Không có               |
| <b>disc (mặc định)</b> | Hình tròn có tô nền    |
| <b>circle</b>          | Hình tròn không tô nền |
| <b>square</b>          | Hình vuông có tô nền   |

78

78

## 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,... |

79

79

## 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

5. Ký hiệu là số thứ tự thường  
 f Ký hiệu là chữ cái thường  
 G. Ký hiệu là chữ cái in hoa  
 viii. Ký hiệu là số la mã thường  
 IX. Ký hiệu là số 1a mã in hoa

80

80



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

Outside:

- Coffee
- Tea
- Coca-cola

Inside:

- Coffee
- Tea
- Coca-cola

81

81

## Định dạng danh sách (tt)

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

```
ul {
  list-style-image: url('bulleted.png');
}
```

- Coffee
- Tea
- Coca Cola

- Cách viết trên không hiển thị đúng trong tất cả trình duyệt → có thể thay thế bằng giải pháp crossbrowser

82

82

## Định dạng danh sách (tt)

- **list-style-image**

- Đối với tag <ul>:
  - Thiết lập list-style-type là none
  - Thiết lập hai thuộc tính padding và margin = 0px
- Đối với các tag <li> trong <ul>:
  - Thiết lập background-image chỉ hiển thị một lần (no-repeat)
  - Thiết lập background-position phù hợp (left: 0px , vertical: center)
  - Thiết lập vị trí van bản (padding-left) phù hợp (lớn hơn chiều rộng hình)

```
ul {
  list-style-type: none;
  padding: 0px;
  margin: 0px;
}
ul li {
  background-image: url(bulleted.png);
  background-repeat: no-repeat;
  background-position: 0px center;
  padding-left: 15px;
}
```

83

83

## Định dạng danh sách (tt)

- **Thuộc tính** list-style có thể viết ngắn gọn hơn, ví dụ:

```
ul {
  list-style: square inside url("bulleted.png");
}
```

- Thứ tự các thuộc tính phải được sắp xếp theo trình tự:  
**list-style-type**, **list-style-position**, **list-style-image**

84

84

### 3.4. Định dạng trang web với CSS

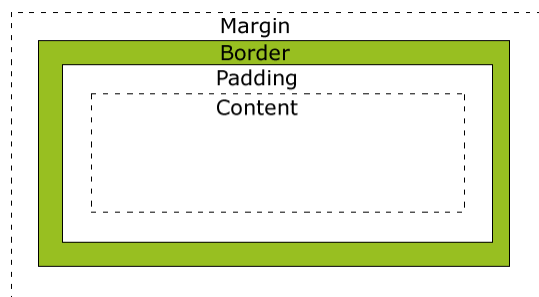
- Màu và nền
- Text và Font
- Liên kết
- Danh sách
- Box Model
- Bảng
- CSS Display
- CSS Position
- CSS Float & Clear
- Layers
- Giới thiệu CSS3

85

85

### 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



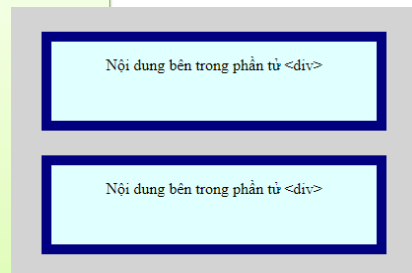
86

86

## 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 &lt;div>&lt;/div>
<div>Nội dung bên trong phần tử &lt;div>&lt;/div>
</body>
```



87

87

## 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 dãy 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;}**

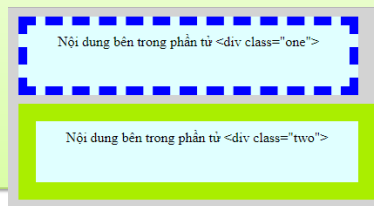
88

88

## 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.one{ border-style: dashed; border-width: 10px; border-color: blue; }
  div.two{border: 20px solid #aeee00;}
</style>
</head>
<body>
  <div class="one">...</div>
  <div class="two">...</div>
</body>
```



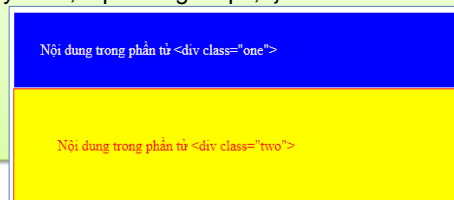
89

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.one{ color: white; background: blue; padding: 30px; }
div.two{ color: red; background: yellow; padding: 50px; }
</style></head>
<body>
  <div class="one"> ... </div>
  <div class="two"> ... </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, bottom)
    - margin: 25px 50px; (top, right)
    - margin: 25px; (áp dụng cho tất cả các chiều)
  - 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

91

91

## CSS Box Model (tt)

- **Thuộc tính margin:** ví dụ

```
<head> <style>
  div{ border: solid 1px; }
  div.two{margin-top: 80px; margin-bottom: 40px;}
</style> </head>
<body>
  <div > ... </div>
  <div class="two"> ... </div>
  <div > ... </div>
</body>
```

Nội dung trong phần tử <div>

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

Nội dung trong phần tử <div>

92

92

## 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

93

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

94

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

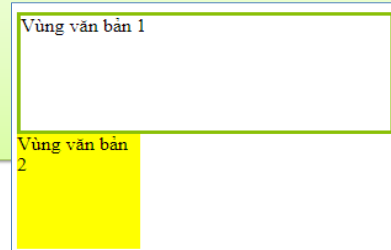
95

95

## CSS Box Model (tt)

- **Chiều rộng, chiều cao hộp**

```
<head> <style>
  div.one{ width: 300px; height: 100px; border: 3px solid #8AC007; }
  div.two{ max-width: 100px; min-height: 100px; background-color: yellow; }
</style></head>
<body>
  <div class="one">Vùng văn bản 1</div>
  <div class="two">Vùng văn bản 2</div>
</body>
```



96

96



### 3.4. 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

97

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

98

98

## CSS Tables (tt)

- Thiết lập màu sắc cho đường biên, nền

```
table, td, th {
  border: 1px solid green;
}
th {
  background-color: green;
  color: white;
}
```

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

99

99

## 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.one {border-spacing:15px;}
table.two {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 |

100

100

## CSS Tables (tt)

- **border-spacing: ví dụ**

```
table.one {border-spacing:15px;}
table.two {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

101

## CSS Tables (tt)

- Thuộc tính **empty-cells**:

- Quy định ô trống 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 |                      |

102

102

## CSS Tables (tt)

- **Thuộc tính border-collapse:**

- Cho/không cho phép thu gọn đường viền của bảng thành một đường đơn.

- Có hai giá trị:

- collapse
- separate (mặc định).

**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 |

103

103

## CSS Tables

- **border-collapse:** ví dụ

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

| Firstname | Lastname |
|-----------|----------|
| Peter     | Griffin  |
| Lois      | Griffin  |

104

104

## 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   |

105

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   |

106

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   |

107

107

## 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   |

108

108

## 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

109

109

## 3.4. 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

110

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](#), [list-item](#), [flex](#), [grid](#)

111

111

## 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>
```

112

112



**CSS Display (tt)**

`div { display: inline-block; border: 1px solid; width: 100px; height: 50px; }`

• Ví dụ

Thiết lập thuộc tính display như trong code

li {display: inline;}

span {display: block;}

nếu không sử dụng thuộc tính display

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

Đây là nội dung trong phần tử <p>

113

113

**3.4. 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

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.

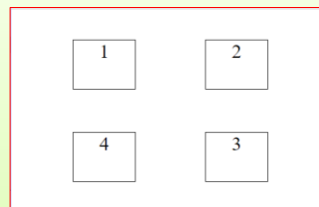
115

115

## CSS Position (tt)

### • position:absolute

```
<head> <style>
  div{ position: absolute; width: 100px; height: 80px;
    border: 1px solid; font-size: 2em; text-align: center;
  }
  #d1 { top: 50px; left: 100px;}
  #d2 { top: 50px; right: 100px;}
  #d3 { bottom: 50px; right: 100px;}
  #d4 { bottom: 50px; left: 100px;}
</style></head>
<body>
  <div id="d1">1</div> <div id="d2"> 2 </div><div id="d3">3</div> <div id="d4"> 4 </div>
</body>
```



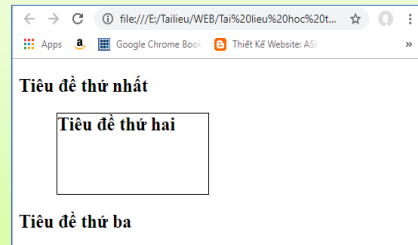
116

116

## CSS Position (tt)

- **position:relative**

```
<head>
<style>
  .relative { position:relative; border:1px solid black; height:100px;
              left: 50px;  max-width: 200px;
            }
</style>
</head>
<body>
  <h2>Tiêu đề thứ nhất</h2>
  <h2 class="relative">Tiêu đề thứ hai</h2>
  <h2>Tiêu đề thứ ba</h2>
</body>
```



117

117

## 3.4. 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

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.

119

119

## CSS Float và Clear

- **float**: ví dụ

```
img {
  float: right;
  margin: 0 0 10px 10px;
}
```

In this example, the image will float to the right in the paragraph, and the text in the paragraph will wrap around the image.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.



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 #000000; padding:5px; margin:5px;
  }
  .clear{ 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 class="clear">4</div>
</body>
```

Ví dụ về thuộc tính float - clear



121

121

## CSS Float và Clear (tt)

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

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

Có sử dụng thuộc tính overflow: auto



Không sử dụng thuộc tính overflow: auto



122

122

### 3.4. 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

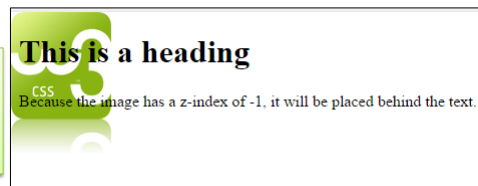
123

123

### 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 là static.

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



124

124

### 3.4. 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

### 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)

- 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

## 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-gradient</a> | • <a href="#">box-sizing</a>        | • <a href="#">resize</a>             |                                    |
| • <a href="#">background</a>          | • <a href="#">box-shadow</a>        | • <a href="#">text-justify</a>       |                                    |
| • <a href="#">border-image</a>        | • <a href="#">column</a>            | • <a href="#">text-overflow</a>      |                                    |
| • <a href="#">border-radius</a>       | • <a href="#">@font-face</a>        | • <a href="#">text-shadow</a>        |                                    |

128

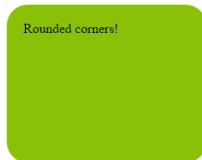
128



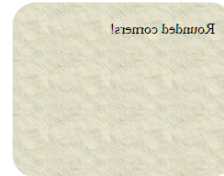
## CSS3 Rounded Corners

Tthuộc tính [border-radius](#): tạo đường biên bo tròn các góc

```
#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

129

## CSS3 Border Images

- Thuộc tính [border-image](#): đường viền là hình ảnh

```
#borderimg {
  border: 10px solid transparent;
  padding: 15px;
  -webkit-border-image: url(border.png) 30 round; /* Safari 3.1-5 */
  -o-border-image: url(border.png) 30 round; /* Opera 11-12.1 */
  border-image: url(border.png) 30 round;
}
```

130

130

## CSS3 Border Images (tt)

- Các thuộc tính khác:

- background-size
- background-origin
- background-clip

Original background-image:



Resized background-image:

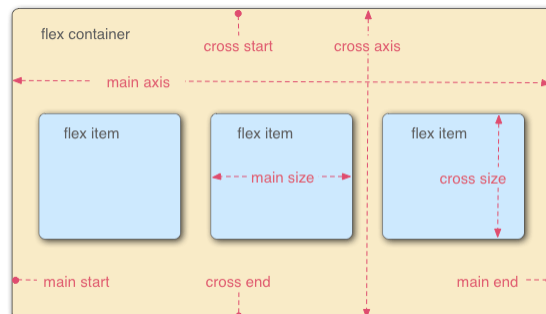


131

131

## 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)

132

132

## Flexbox

- **container:**
  - Chứa các phần tử bên trong
  - Được thiết lập kiểu hiển thị theo chiều ngang hoặc chiều dọc
  - Các item bên trong sẽ hiển thị theo thiết lập của container.
- **item:**
  - Các phần tử bên trong container
  - Có thể thiết lập cho phép item sử dụng bao nhiêu cột trong một container, hoặc thiết lập thứ tự hiển thị của nó.

133

133

## Flexbox

- **main start, main end:** điểm bắt đầu và điểm kết thúc của container
- **cross start, cross end:** luôn 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
  - Các item sẽ hiển thị theo chiều của main axis.
  - 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 dựa theo trục main axis
- **cross size:** kích thước của mỗi item dựa theo trục cross axis.

134

134

## Flexbox

- **Các thuộc tính CSS3 được sử dụng trong Flexbox:**

- display: flex
- flex-direction
- flex-wrap
- flex-grow
- flex-flow (kết hợp flex-direction và flex-wrap)

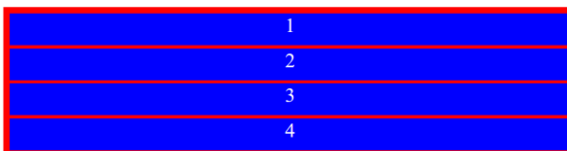
135

135

## 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;
}
```

136

136



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

## Flexbox

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

1 2 3 4

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

4 3 2 1

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

1

2

3

4

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


4

3

2

1

137



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

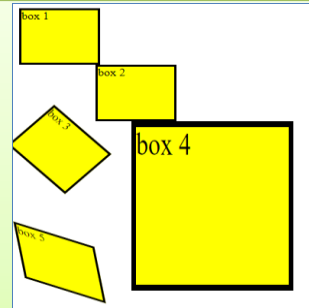
## CSS3 2D Transforms

- 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.

138

## CSS3 2D Transforms (tt)

```
<head> <style>
  #box1, #box2, #box3, #box4, #box5{
    width: 100px; height: 80px;
    border: 3px solid black; background-color: yellow;
  }
  #box2 { transform: translate(100px, 0px); }
  #box3 { transform: rotate(45deg); }
  #box4 { transform: scale(2, 3) translate(100px, 0px); }
  #box5 { transform: skew(10deg, 20deg); }
</style></head>
<body>
  <div id = "box1">...</div> <div id = "box2">...</div> <div id = "box3">...</div>
  <div id = "box4">...</div> <div id = "box5">...</div>
</body>
```



139

139

## Hoạt hình

- **Thuộc tính transition:** ép 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→nhanh→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.

140

140

## Hoạt hình (tt)

- Thuộc tính transition: [ví dụ](#)

```
<head>
<style>
  h1{ color: black; transition:all 1s ease-in; }
  h1:hover {color: red; font-size: 500%;}
</style>
</head>
<body>
  <h1>Transition Demo</h1>
</body>
```

Transition Demo

Bình thường

khí di chuyển con trỏ chuột lên trên chữ

**Transition Demo**

141

141

## Hoạt hình (tt)

- Các bước tạo hoạt hình cho phần tử:

1. Tạo một tập hợp các khung hình chính: mỗi khung hình chính chỉ định một trạng thái của đối tượng trong trình duyệt (một cảnh trong hoạt hình).
2. Thiết lập giá trị phần trăm cho mỗi khung hình chính: khung hình chính đầu tiên là 0% (phần đầu của hoạt ảnh) và khung hình cuối cùng phải là 100% (phần cuối của hoạt ảnh), và các khung hình trung gian với các tỷ lệ % tăng dần.
3. Thiết lập style cho mỗi khung hình chính.
4. Áp dụng chuyển động cho phần tử.
5. Thiết lập các tham số chuyển động như thời gian, tốc độ, chế độ lặp lại của chuyển động...

142

142

## 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>
```

143

143

## 3.5 Giới thiệu Bootstrap

- Các thiết bị có 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.

144

144



## 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..

145

145

## 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.

146

146

- 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)