

BÀI 4:

BOX MODEL VÀ CSS NÂNG CAO

Thiết kế web

Bộ môn Kỹ thuật và mạng máy

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

Nội dung bài học

CSS Box Model

1

CSS Borders

2

CSS margin

3

CSS padding

4

Nội dung bài học (tiếp)

CSS dimension

5

CSS link

6

CSS lists

7

CSS table

8

Nội dung bài học (tiếp)

CSS display

9

CSS position

10

CSS float và clear

11

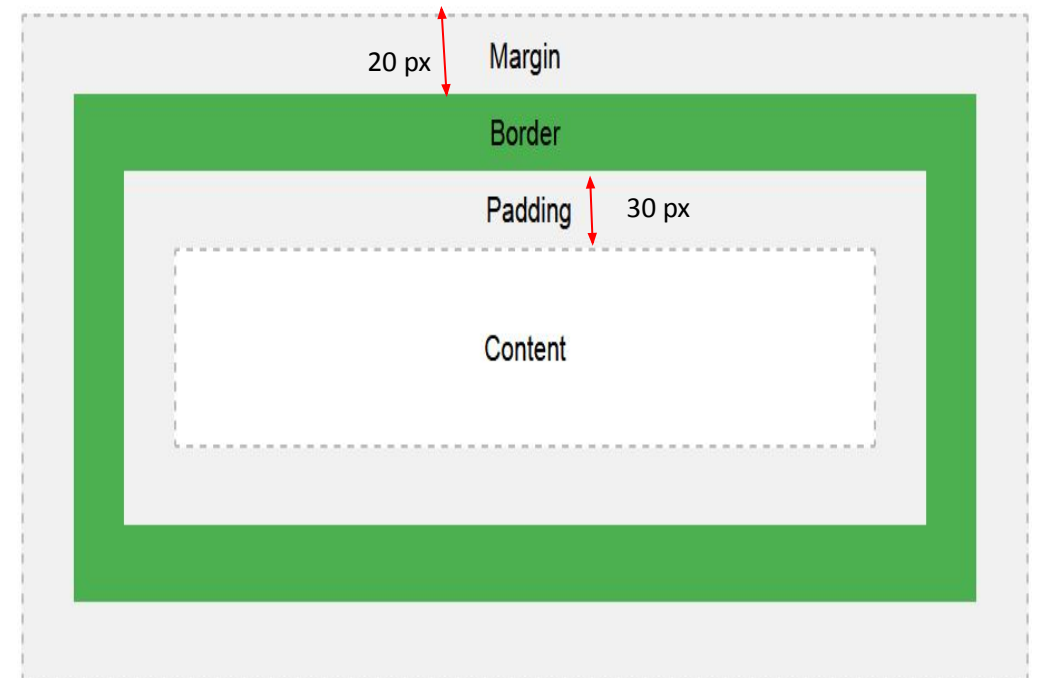
1. CSS Box Model

- Mỗi một phần tử HTML sẽ bao gồm các thành phần: margin, border, padding, và content (nội dung của phần tử).
- Mỗi phần tử HTML được coi là một Box với các thành phần margin, border, padding, và content được gọi là Box model.
- Box model cho phép chúng ta thêm đường viền xung quanh phần tử, xác định khoảng cách giữa các phần tử...

1. CSS Box Model (tiếp)

- Ví dụ

```
div { border: 1px solid black;
padding: 30px;
margin: 20px; }
```



2. CSS Borders

- CSS border dùng để thiết lập đường viền cho các thành phần html. Thuộc tính đường viền bao gồm: **border-style**, **border-width**, **border-color**.
- **border-style**: thiết lập kiểu đường viền cho một phần tử.

- Ví dụ.

```
p {  
    border-style: solid;  
}
```

- Một số giá trị khác là: none, dotted, dashed, double...

2. CSS Borders (tiếp)

- **border-width**: thiết lập độ dày của đường viền
- **border-color** : thiết lập lại màu của đường viền

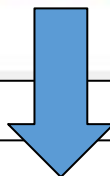
■ Ví dụ

```
p {  
  border-width: 1px;  
}  
  
p {  
  border-color: blue;  
}
```


2. CSS Borders (tiếp)

- Chúng ta cũng có thể xác định thuộc tính cho từng đường viền: trên, dưới, trái, phải.

```
p {
    border-top-style: dotted;
    border-right-style: solid;
    border-bottom-style: dotted;
    border-left-style: solid;
}
```



```
p {
    border-style: dotted solid;
}
```

2. CSS Borders (tiếp)

```
p {  
  background-color: lightcyan;  
  border-top-color: #0000FF;  
  border-right-color: #FF0000;  
  border-bottom-color: #FF00FF;  
  border-left-color: #FFFF00;  
  border-style: double;  
  border-width: 20px;  
}
```

2. CSS Borders (tiếp)

```
img {  
    border-style: dashed;  
    border-color: #CC00FF;  
    border-top-width: 20px;  
    border-left-width: thick;  
    border-right-width: thin ;  
}
```

2. CSS Borders (tiếp)

- Chúng ta có thể xác định kiểu đường viền cho cả 4 cạnh như sau.

- **border-style: dotted solid double dashed;**

- top border is dotted
- right border is solid
- bottom border is double
- left border is dashed

- **border-style: dotted solid double;**

- top border is dotted
- right and left borders are solid
- bottom border is double

- **border-style: dotted solid;**

- top and bottom borders are dotted
- right and left borders are solid

- **border-style: dotted;**

- all four borders are dotted

2. CSS Borders (tiếp)

- Chúng ta có thể viết tắt cho cả 3 thuộc tính đường viền như sau.

```
p {  
    border: 5px solid red;  
}
```

3. CSS margin

- Thuộc tính **margin** dùng để xác định khoảng trống nằm xung quanh đường viền. Giá trị của thuộc tính margin có thể là:
 - **auto**: Trình duyệt từ xác định lề
 - **length**: Xác định lề là px, cm, pt, ... (giá trị mặc định là 0px).
 - **%**: Xác định lề là bao nhiêu % chiều rộng của đối tượng định đặt lề.

3. CSS margin (tiếp)

- Chúng ta có thể đặt lề cho riêng từng bên: trên, dưới, trái phải

```
p {  
    margin-top: 100px;  
    margin-bottom: 100px;  
    margin-right: 150px;  
    margin-left: 50px;  
}
```

3. CSS margin (tiếp)

- Chúng ta có thể xác định lề cho cả 4 phía như sau.

- **margin: 25px 50px 75px 100px;**

- top margin is 25px
- right margin is 50px
- bottom margin is 75px
- left margin is 100px

- **margin: 25px 50px 75px;**

- top margin is 25px
- right and left margins are 50px
- bottom margin is 75px

- **margin: 25px 50px;**

- top and bottom margins are 25px
- right and left margins are 50px

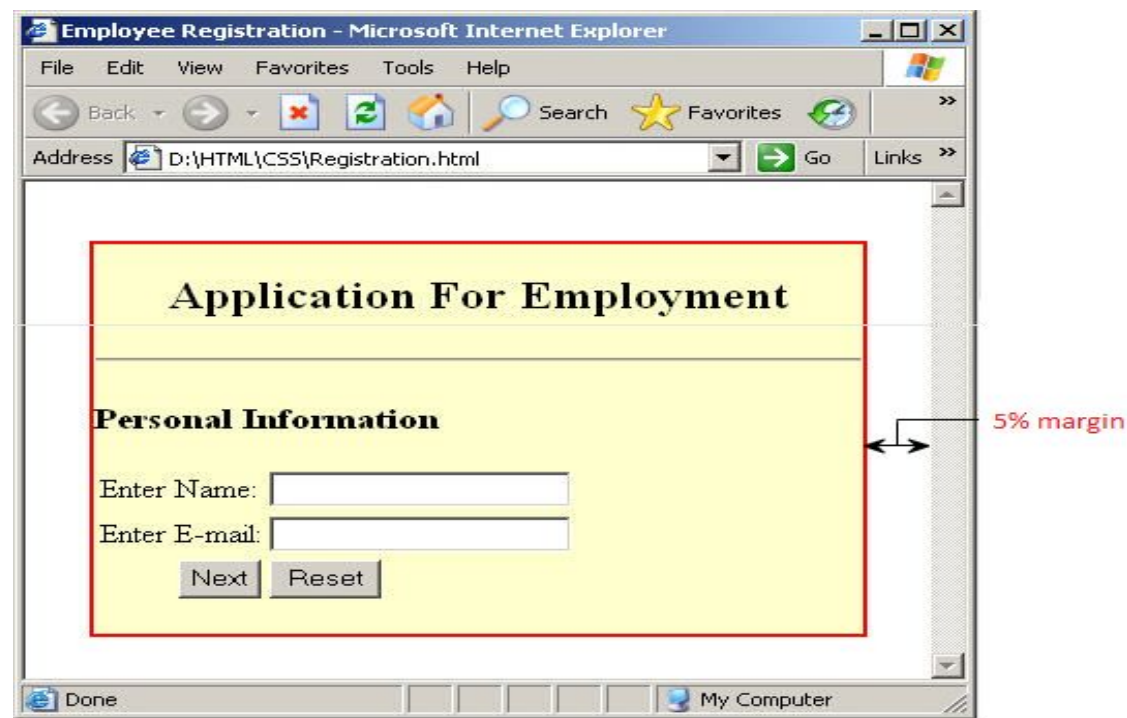
- **margin: 25px;**

- all four margins are 25px

3. CSS margin (tiếp)

- Ví dụ

```
.format
{
    background-color: #FFFFCC;
    border: solid thin red;
    margin-top: 5%;
    margin-left: 5%;
    margin-right: 5%;
}
```



4. CSS padding

- Thuộc tính **padding** dùng để xác định khoảng trống nằm giữa đường viền và nội dung. Giá trị của thuộc tính này có thể là:
 - **length**: Định nghĩa padding cố định (px, pt, cm, em, ...).
 - **%**: Xác định % chiều dài của đối tượng cần padding.

- Ví dụ

```
p {
    padding-top: 25px;
    padding-right: 50px;
    padding-bottom: 25px;
    padding-left: 50px;
}
```

4. CSS padding (tiếp)

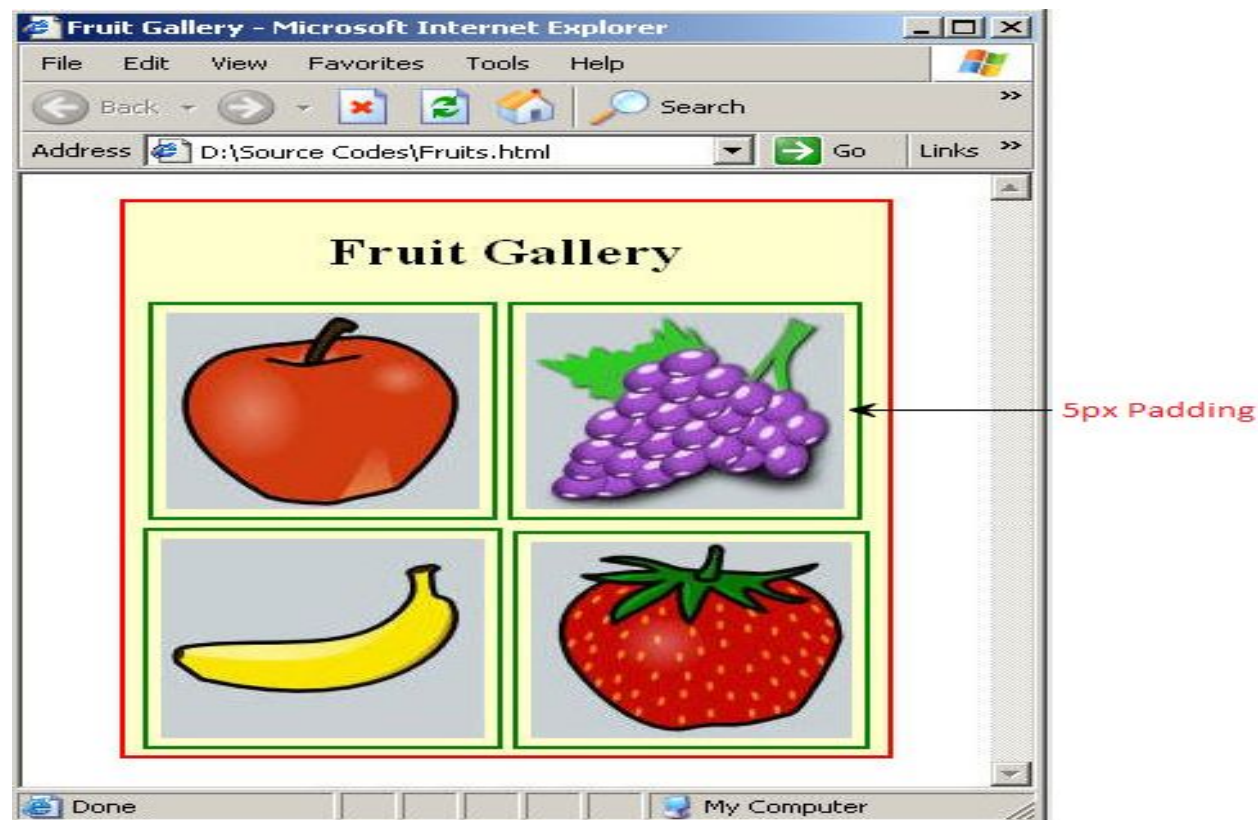
- Chúng ta có thể xác định vùng khoảng cách vùng đệm cho cả 4 phía như sau.
 - **padding: 25px 50px 75px 100px;**
 - top padding is 25px
 - right padding is 50px
 - bottom padding is 75px
 - left padding is 100px
 - **padding: 25px 50px 75px;**
 - top padding is 25px
 - right and left paddings are 50px
 - bottom padding is 75px
 - **padding: 25px 50px;**
 - top and bottom paddings are 25px
 - right and left paddings are 50px
 - **padding: 25px;**
 - all four paddings are 25px

4. CSS padding (tiếp)

- Ví dụ

```

BODY
{
    margin-left: 10%;
    margin-right: 10%;
}
.fruits
{
    padding-left: 5px;
    padding-top: 5px;
    padding-right: 5px;
    padding-bottom: 5px;
    border: solid thin green;
}
  
```



5. CSS dimension

- CSS **dimension** dùng để xác định chiều dài, chiều cao, chiều dài, cao tối thiểu và tối đa cho các thành phần html.
 - Thuộc tính **width** dùng để thiết lập chiều rộng cho phần nội dung của phần tử.
 - Thuộc tính **height** dùng để thiết lập chiều cao cho phần nội dung của phần tử.
 - Thuộc tính **max-width**, **min-width** được dùng để thiết lập chiều rộng tối đa và tối thiểu cho phần tử.
 - Thuộc tính **max-height**, **min-height** dùng để thiết lập chiều cao tối đa và tối thiểu cho phần tử.

5. CSS dimension (tiếp)

- Lưu ý: Chiều rộng và chiều cao không bao gồm phần padding, border, margin.



5. CSS dimension (tiếp)

- Ví dụ

```
p{           p{  
  height: 50px;  max-width: 300px;  
  width: 300px;  min-height: 50px;  
}
```

6.CSS link

- Link có thể được định dạng bằng bất kỳ thuộc tính CSS nào (**color, font-family, background-color,...**).
- Ví dụ:

```
a {  
    color: #FF0000;  
}
```


6. CSS link (tiếp)

- Ngoài ra, ta còn định dạng cho 4 trạng thái của link:
 - **a:link** – liên kết bình thường chưa được kích
 - **a:visited** – Liên kết đã được kích
 - **a:hover** – Liên kết khi người sử dụng di chuyển chuột vào
 - **a:active** – Liên kết tại thời điểm kích chuột

6. CSS link (tiếp)

- Ví dụ

```
/* unvisited link */  
a:link {  
    color: #FF0000;  
}
```

```
/* visited link */  
a:visited {  
    color: #00FF00;  
}
```

```
/* mouse over link */  
a:hover {  
    color: #FF00FF;  
}
```

```
/* selected link */  
a:active {  
    color: #0000FF;  
}
```

6. CSS link (tiếp)

- Một số định dạng chung cho link:
 - text-decoration – Thường dùng để gỡ bỏ gạch chân của link.

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

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

```
a:hover {  
    text-decoration: underline;  
}
```

```
a:active {  
    text-decoration: underline;  
}
```

6. CSS link (tiếp)

- **background-color**: Dùng để xác định màu nền cho link

```
a:link {  
    background-color: #B2FF99;  
}
```

```
a:visited {  
    background-color: #FFFF85;  
}
```

```
a:hover {  
    background-color: #FF704D;  
}
```

```
a:active {  
    background-color: #FF704D;  
}
```

7. CSS lists

- CSS list dùng để:
 - Thiết lập các bullet và numbering khác nhau cho danh sách không có thứ tự và có thứ tự.
 - Thiết lập hình ảnh cho bullet
- Để thiết lập bullet hay numbering cho danh sách ta sử dụng thuộc tính **list-style-type**.
- Để thiết lập ảnh cho bullet ta sử dụng thuộc tính **list-style-image**.
- Để thiết lập vị trí hiển thị của bullet hay numbering ta sử dụng thuộc tính **list-style-position**.

7. CSS lists (tiếp)

- Ví dụ `list-style-type`:

```
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;  
}
```

7. CSS lists (tiếp)

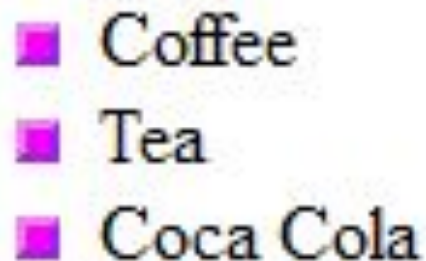
- Ví dụ `list-style-image`:

```
ul {
    list-style-image: url('sqpurple.gif');
}
```

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
    list-style-image: url('sqpurple.gif');
}
</style>
</head>
<body>

<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>

</body>
</html>
```



7. CSS lists (tiếp)

- Ví dụ `list-style-position`:

```
ul {
    list-style-position: inside;
}
```

Outside:

- Coffee
- Tea
- Coca-cola

Inside:

- Coffee
- Tea
- Coca-cola

7. CSS lists (tiếp)

- Ví dụ viết vắn tắt cho thuộc tính của list là: **list-style**:

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

8. CSS table

- Dùng thuộc tính **border** để xác định đường viền cho bảng và các ô dữ liệu.
 - Ví dụ:

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

8. CSS table (tiếp)

- Để collapse (gộp) đường viền của bảng ta sử dụng thuộc tính **border-collapse**
 - Ví dụ:

```
table { border-collapse: collapse; }
```

8. CSS table (tiếp)

- Thuộc tính **width** và **height** dùng để thiết lập chiều rộng của bảng và chiều cao của dòng.

- Ví dụ

```
table {  
  width: 100%;  
}  
  
th {  
  height: 50px;  
}
```

8. CSS table (tiếp)

- Thuộc tính **text-align** và **vertical-align** dùng để căn lề dữ liệu cho các ô trong bảng.

- Ví dụ:

```
th {  
    text-align: left;  
}  
td {  
    vertical-align: middle;  
}
```

9. CSS display

- **CSS display** dùng để xác định một thành phần html hiển thị như thế nào. Giá trị mặc định của thuộc tính này là: **inline** và **block**.
- Các phần tử mức **block**: Luôn bắt đầu ở dòng mới và chiếm toàn bộ độ rộng còn lại. Một số phần tử mức block là: `<div>`, `<h1>` đến `<h6>`, `<p>`, `<form>`, ``, ...
- Các phần tử **inline**: Không bắt đầu ở dòng mới và chỉ chiếm độ rộng cần thiết. Một số phần tử inline như: ``, `<a>`, ``.

9. CSS display (tiếp)

- Ngoài giá trị **block** và **inline** còn có một số thuộc tính hay sử dụng khác như: **none**, **inline-block**.
- Chúng ta có thể thay giá trị mặc định cho thuộc tính **display**

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

10. CSS position

- **CSS position** dùng để xác định phương thức xác định vị trí cho các phần tử html. Giá trị của thuộc tính **position** là:
 - **static**: Đây là giá trị mặc định của thuộc tính **position**, với giá trị này phần tử html hiển thị bình thường không có gì đặc biệt.
 - **relative**: Xác định vị trí hiển thị của chúng có quan hệ với vị trí bình thường. Thông qua thiết lập các thuộc tính kèm theo: **left, right, top, bottom** (giữa hai thuộc tính top & bottom các bạn chỉ có thể sử dụng một thuộc tính, giữa hai thuộc tính left & right cũng tương tự như vậy.)

10. CSS position (tiếp)

- **fixed**: Thiết lập kiểu hiển thị là cố định ngay cả khi ta cuộn trang web. Để thiết lập vị trí ta dùng kết hợp với các thuộc tính **left, right, top, bottom**
- **absolute**: Xác định vị trí hiển thị của chúng có quan hệ với vị trí của thẻ chứa chúng. Thông qua thiết lập các thuộc tính kèm theo: **left, right, top, bottom**.

11. CSS float và clear

- **CSS float** dùng để xác định liệu một thành phần html có float (trôi) hay không. Giá trị của thuộc tính này là:
 - **none**: phần tử không float (trôi) đi đâu cả đây là giá trị mặc định.
 - **left**: phần tử sẽ float (trôi) sang trái.
 - **right**: phần tử sẽ float (trôi) sang phải.

11. CSS float và clear (tiếp)

- Hình ảnh dưới đây được float về bên phải

Example

```
img {
  float: right;
}
```

Try it yourself »

e have added an image with style **float:right**. The float to the right in the paragraph.

me text. This is some text. This is
t. This is some text. This is some
is some text. This is some text.
me text. This is some text. This is
t. This is some text. This is some
is some text. This is some text.
me text. This is some text. This is



12. CSS float và clear (tiếp)

- **CSS clear** dùng để xác định phía nào của phần tử mà thành phần float không được phép float. Giá trị của thuộc tính này thường là:
 - **none**: bị ảnh hưởng bởi thuộc tính float từ phần tử phía trước nó
 - **left**: không bị ảnh hưởng bởi thuộc tính float (với giá trị left) từ phần tử phía trước nó
 - **right**: không bị ảnh hưởng bởi thuộc tính float (với giá trị right) từ phần tử phía trước nó
 - **both**: không bị ảnh hưởng bởi thuộc tính float từ phần tử phía trước nó

Xin cảm ơn