

Phần II

Định dạng trang với CSS

GIỚI THIỆU

1. CSS là gì?

CSS (Cascading Style Sheets) được hiểu một cách đơn giản đó là cách mà chúng ta thêm các kiểu định dạng (font chữ, kích thước, màu sắc...) cho một tài liệu Web

2. Tại sao CSS?

CSS cung cấp hàng trăm thuộc tính trình bày dành cho các đối tượng với sự sáng tạo cao trong kết hợp các thuộc tính mang lại hiệu quả.

CSS giúp áp dụng một khuôn mẫu chuẩn từ một file CSS ở ngoài.

3. Học CSS cần những gì?

- ❖ Có kiến thức về HTML.
- ❖ Một trình soạn thảo.
- ❖ Phiên bản mới nhất của trình duyệt.
- ❖ Thực hành CSS sau mỗi buổi học.

3.2. Một số quy ước về cách viết CSS

3.2.1. Cú pháp của CSS

Ví dụ: Định dạng màu nền cho một trang web.

✓ Trong HTML `<body bgcolor ="#00BFF3"> /* Nền có màu xanh nhạt*/`

✓ Trong CSS: `body{ background-color:#00BFF3;} /* Nền có màu xanh nhạt*/`

→ Cú pháp của CSS được chia làm 3 phần. phần thẻ chọn (selector), phần thuộc tính (property), phần nhãn (value).

selector { property: value; }

a. Selector: Các đối tượng ta sẽ áp dụng các thuộc tính trình bày. Nó là các tag trong HTML, *class*, *id*

Ví dụ: *body*, *h2*, *p*, *img*, *#title*, *.username*.....

- Ngoài việc viết tên selector cụ thể, ta có thể dùng selector đại diện như *** để tác động lên tất cả các thành phần có trên trang web.

b. Property: Chính là các thuộc tính quy định trình bày

Ví dụ: background-color, font-family, color, padding, margin,...

Mỗi thuộc tính phải được gán một giá trị.

Một Selector có nhiều thuộc tính thì phải dùng dấu ; (chấm phẩy) để phân cách các thuộc tính

Ví dụ:

```
body{background-color:#3300CC;
```

```
color:#33FF33;
```

```
font: Arial, Helvetica, sans-serif;}
```

Ta có thể gom nhiều thành phần có cùng một số thuộc tính giống nhau

Ví dụ:

```
h1{color:#99FF00;  
    text-transform:uppercase;}
```

```
h2{color:#99FF00;  
    text-transform:uppercase;}
```

```
h3{color:#99FF00;  
    text-transform:uppercase;}
```

```
→ h1, h2, h3 {  
    color:#99FF00;  
    text-transform:uppercase;  
}
```

c. Value: Giá trị các thuộc tính.

Ví dụ: #FFF, uppercase, red,...

Nếu **giá trị** có nhiều từ ta nên đặt giá trị vào trong dấu “” (nháy kép)

Ví dụ: *p {font-family: "sans serif" }*

Đối với các giá trị là đơn vị đo, không nên đặt khoảng cách giữa số đo với đơn vị của nó.

Ví dụ: *width:100 px; /* Sai */*

Width:100px; / Đúng */*

d. Chú thích trong CSS

Chú thích được viết như sau: / Nội dung chú thích*/*

Ví dụ:

/ Màu chữ trang web*/*

body {

color:red

}

3.2.2. Vị trí đặt CSS vào trang Web

Có 3 cách cho phép chúng ta chèn định dạng CSS vào trong Website.

a. CSS được khai báo trong file riêng (Bên ngoài).

Toàn bộ mã CSS được chứa trong file riêng có phần mở rộng .css và được áp dụng cho nhiều trang khác nhau. Ta có thể thay đổi cách hiển thị của toàn bộ site mà chỉ cần thay đổi một file CSS

Trong cách này thì file CSS sẽ được chèn vào văn bản HTML thông qua thẻ <link>.. với thuộc tính href.

Cú pháp:

<head>

<link rel="stylesheet" type="text/css"

href="/mystyle.css" />

</head>

b. Chèn CSS trong tài liệu HTML(Bên trong)

- Chỉ giành riêng cho tài liệu HTML đó
- Khi bạn chèn trực tiếp thì đoạn mã CSS được đặt trong thẻ <style> và đặt trong phần <head>

~~Ví dụ~~ <head>

```
<style type="text/css">
```

```
hr {color: sienna}
```

```
p {margin-left: 20px}
```

```
body {background-image: url("images/back40.gif")}
```

```
</style>
```

```
</head>
```

c. Chèn trực tiếp vào thẻ của HTML(Nội tuyến)

Nội tuyến được sử dụng nhiều trong trường hợp một thẻ HTML nào đó cần có style riêng cho nó.

Ví dụ:

```
<p style="color: sienna; margin-left: 20px">  
This is a paragraph  
</p>
```

3.2.3. Sự ưu tiên

CSS nội tuyến > CSS bên trong > CSS bên ngoài > CSS mặc định của trình duyệt

3.2.3. Sự ưu tiên (tt)

Ví dụ:

Trong một trang web có liên kết tới file style.css có nội dung như sau:

```
p {  
  color:#333;  
  text-align:left;  
  width:500px;  
}
```

Trong thẻ <style> giữa thẻ <head> cũng có một đoạn CSS liên quan:

```
p {  
  background-color:#FF00FF;  
  text-align:right;  
  width:100%;  
  height:150px;  
}
```

Trong phần nội dung trang web đó cũng có sử dụng CSS nội tuyến:

```
<p style="height:200px; text-align:center; border:1px solid  
#FF0000; color:#000" >
```

3.2.3. Sự ưu tiên (tt)

Khi thực thi CSS trình duyệt sẽ đọc hết tất cả các nguồn chứa style rồi sẽ tổng hợp lại vào một CSS ảo và nếu có sự trùng lặp các thuộc tính CSS thì nó sẽ lấy thuộc tính CSS có mức ưu tiên cao hơn. Như ví dụ trên chúng ta sẽ thấy CSS cuối cùng mà phần tử p nhận được là:

```
p {  
background-color:#FF00FF;  
width:100%;  
height:200px;  
text-align:center;  
border:1px solid #FF0000;  
color:#000;  
}
```


3.2.3. Sự ưu tiên (tt)

Để thay đổi độ ưu tiên cho một thuộc tính nào đó ta sử dụng cú pháp sau

selector { property:value !important }

Ví dụ:

```
p {  
  color:#333!important;  
  text-align:left !important;  
  width:500px;  
}
```

```
p {  
  background-color:#FF00FF;  
  text-align:right;  
  width:100%;  
  height:150px !important;  
}
```

```
<p style="height:200px; text-align:center; border:1px solid  
#FF0000; color:#000" }
```

3.2.3. Sự ưu tiên (tt)

Phần CSS sẽ tác động lên thuộc tính p là:

```
p {  
background-color:#FF0000;  
width:100%;  
height:150px !important;  
text-align:left !important;  
border:1px solid #FF0000;  
color:#333 !important;  
}
```

Lưu ý: Cùng một thuộc tính cho một **selector** thì nếu cả hai thuộc tính đều đặt **!important** thì cái sau được lấy

3.3. CSS cho Background

3.3.1. Màu nền (background-color)

```
body {  
Background-color:cygan;  
}
```

```
h1{  
Background-color:Red;  
}
```

```
h2{  
Background-color:Blue;  
}
```


3.3.2. Ảnh nền (background-image)

```
body {  
  Background-image:url(logo.gif);  
}
```

- url: đường dẫn chỉ đến bức ảnh
- Nếu bức ảnh cùng với thư mục đặt file .css ta chỉ cần ghi tên file ảnh.
- Nếu nằm ở thư mục image thì ghi: *background-image:url(img/abc.gif)*
- Nếu không chắc ta nên dùng đường dẫn tuyệt đối.

3.3.3. Lặp lại ảnh nền (background-repeat)

- repeat-x: Chỉ lặp lại ảnh theo phương ngang.
- repeat-y: Chỉ lặp lại ảnh theo phương dọc.
- repeat: Lặp lại ảnh theo cả 2 phương, đây là giá trị mặc định.
- no-repeat: Không lặp lại ảnh.

3.3.4. Khóa **ảnh nền** (*background-attachment*)

scroll: Ảnh nền sẽ cuộn cùng nội dung trang web, đây là giá trị mặc định.

fixed: Cố định ảnh nền so với nội dung trang web.

3.3.5. Định vị **ảnh nền** (*thuộc tính background-position*)

Giá trị	Ý nghĩa
Background-position:5cm 2cm	Ảnh được định vị 5cm từ trái qua và 2cm từ trên xuống.
Background-position:20% 30%	Ảnh được định vị 20% từ trái qua và 30% từ trên xuống.
Background-position:bottom left	Ảnh được định vị ở góc trái phía dưới

3.3.6. Rút gọn *thuộc tính background*

Rút gọn các thuộc tính cho background tạo ra CSS đơn giản dễ quản lý

Ví dụ: Có thể nhóm lại đoạn CSS sau:

```
background-color: #0000FF;  
background-image: url(logo.png);  
background-repeat: no-repeat;  
background-attachment: fixed;  
background-position: right bottom;
```

Thành một dòng ngắn gọn:

```
background: #0000FF url(logo.png) no-repeat fixed  
right bottom;
```


3.4. CSS cho Font

3.4.1. Font-family:

Định nghĩa một danh sách ưu tiên các font sẽ được dùng để hiển thị một thành phần trang web.

Có hai loại tên font được dùng để chỉ định trong font-family: family-names và generic families.

- Family-names: Tên cụ thể của một font.

Ví dụ: Arial, Verdana, Tohama,...

- Generic families: Tên của một họ gồm nhiều font.

Ví dụ: sans-serif, serif,...

Ví dụ:

```
body { font-family: "Times New Roman",Tohama,sans-serif }
```

```
h1, h2, h3 { font-family:arial,verdana,serif }
```

3.4.2. Font-style:

Font-Style: Định nghĩa việc áp dụng các kiểu in thường (normal), in nghiêng (italic) hay xiên (oblique) lên các thành phần trang web

Ví dụ:

```
h1 {  
    font-style:italic;  
}  
h2 {  
    font-style:oblique;  
}
```

3.4.3. Font-variant:

Thuộc tính font-variant được dùng để chọn giữa chế độ bình thường và small-caps của một font chữ.

3.4.4. Font-weight:

Font-weight mô tả cách thức thể hiện của font chữ là ở dạng bình thường (normal) hay in đậm (bold).

```
p {
```

```
  font-weight: bold;
```

```
}
```

3.4.5. Font-size:

Kích thước của một font được định bởi thuộc tính font-size.

```
body {
```

```
  font-size: 20px
```

```
}
```

```
h1 {
```

```
  font-size: 3em
```

```
}
```

```
h2 {
```

```
  font-size: 2em
```

```
}
```

1 em = 20px

3.4.6. Font rút gọn:

Ví dụ:

```
h1 {  
    font-style: italic;  
    font-variant: small-caps;  
    font-weight: bold;  
    font-size: 35px;  
    font-family: arial, verdana, sans-serif;  
}
```

Được rút gọn thành:

```
h1 {  
    font: italic bold 35px arial, verdana, sans-serif;  
}
```

3.5. CSS cho Text

3.5.1. Màu chữ (color)

```
body {  
    color:#000  
}
```

```
h1 {  
    color:#0000F  
}
```

```
h2 {  
    color:#00FF  
}
```

3.5.2. Text-indent

Text-indent cung cấp khả năng tạo ra khoảng thụt đầu dòng cho dòng đầu tiên trong đoạn văn bản

Ví dụ:

```
p {  
    text-indent:30px;  
}
```

3.5.3. Text-align

Text-align giúp bạn thêm các canh chỉnh văn bản cho các thành phần trong trang web.

```
h1, h2 {  
    text-align:right;  
}
```

```
p {  
    text-align:justify;  
}
```

3.5.4. Letter-spacing

Letter-spacing được dùng để định khoảng cách giữa các ký tự trong một đoạn văn bản.

```
h1, h2 {  
    letter-spacing: 7px;  
}  
p { letter-spacing: 5px;  
}
```

3.5.5. Text-decoration

Text-decoration giúp ta thêm các hiệu ứng gạch chân (underline), gạch xiên (line-through), gạch đầu (overline), và một hiệu ứng đặc biệt là văn bản nhấp nháy (blink).

Ví dụ

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

3.5.6. Text-transform

Text-transform qui định chế độ in hoa hay in thường của văn bản mà không phụ thuộc vào văn bản gốc trên HTML.

Có tất cả 4 giá trị: uppercase (in hoa), lowercase (in thường), capitalize (in hoa ở ký tự đầu tiên trong mỗi từ) và none (không áp dụng hiệu ứng – mặc định).

Ví dụ

```
h1 {  
    text-transform:uppercase;  
}  
h2 {  
    text-transform:capitalize;  
}
```

3.6. Pseudo-classes for links

Pseudo-classes cho phép bạn xác định các hiệu ứng định dạng cho một đối tượng liên kết ở một trạng thái xác định như khi liên kết chưa được thăm (**a:link**), khi rê chuột lên liên kết (**a:hover**), khi liên kết được thăm (**a:visited**) hay khi liên kết đang được kích hoạt – đang giữ nhấn chuột (**a:active**).

Ví dụ 1: Ví dụ này chúng ta sẽ áp dụng 4 màu sắc khác nhau cho từng trạng thái liên kết: các liên kết chưa thăm có màu xanh lá; các liên kết mouse over sẽ có màu đỏ tươi; các liên kết đã thăm sẽ có màu đỏ và các liên kết đang kích hoạt có màu tím

```
a:link {  
color:#00FF00  
}
```

```
a:hover {  
color:#FF00FF  
}
```

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

```
a:active {  
color:# 662D91  
}
```

Ví dụ 2: Tạo các hiệu ứng tương ứng với trình trạng liên kết: các liên chưa thăm có màu xanh lá, kích cỡ font 14px; liên kết mouse over có màu đỏ tươi, kích cỡ font 1.2em, hiệu ứng nhấp nháy; liên kết đã thăm sẽ có màu xanh da trời, không có đường gạch chân; các liên kết đang kích hoạt có màu tím và font dạng small-caps.

```
a:link {  
color:#00FF00; font-size:14px  
}  
a:hover {  
color:#FF00FF; font-size:1.2em; text-decoration:blink  
}  
a:visited {  
color:#FF0000; text-decoration:none  
}  
a:active {  
color:# 662D91; font-variant:small-caps  
}
```


Ví dụ 3: ví dụ này cũng tạo cho liên kết hiệu ứng màu sắc giống ví dụ 2 nhưng sẽ có thêm 1 số hiệu ứng: các liên kết sẽ có khung viền màu đen, kích cỡ font 14px; liên kết mouse over có nền light cyan; các liên kết đã thăm có nền light yellow.

```
a {  
border:1px solid #000; font-size:14px  
}  
a:link {  
color:#00FF00;  
}  
a:hover {  
background-color:#00BFF3;  
color:#FF00FF;  
font-size:1.2em;  
text-decoration:blink  
}  
a:visited {  
background-color:#FFF568;  
color:#FF0000;  
text-decoration:none  
}  
a:active { color:#662D91; font-variant:small-caps }
```

3.7. Class & ID

3.7.1. Class

Ví dụ tạo đoạn mã HTML như sau:

```
<p>Danh sách các môn học</p>  
<ul>  
  <li>Lập trình web ASPX</li>  
  <li>Thiết kế web</li>  
  <li>Phân tích thiết kế</li>  
  <li>Vật lý</li>  
  <li>Toán A1</li>  
  <li>Tin học Đại cương</li>  
  <li>Flash</li>  
</ul>
```

Yêu cầu đặt ra là phân ra thành 2 loại **Các môn học đại cương, các môn học chuyên ngành**:

<p>Danh sách các môn học</p>

<li class =“cn”>Lập trình web ASPX

<li class=“cn”>Thiết kế web

<li class=“cn”>Phân tích thiết kế

<li class=“dc”>Vật lý

<li clas=“dc”>Toán A1

<li class=“dc”>Tin học Đại cương

<li class=“cn”>Flash

Thiết kế CSS như sau để phân biệt

li .cn

{color:FF0000 }

li .dc

{ color:0000FF }

Lưu ý: Không nên đặt tên class với ký tự đầu là chữ số, nó sẽ không làm việc cho Firefox.

3.7.2. Id

Cũng với đoạn HTML như ví dụ về class. Nhưng yêu cầu đặt ra là Lập trình sẽ có màu đỏ sậm, Thiết kế web màu đỏ, Flash màu đỏ tươi, Phân tích thiết kế màu vàng còn các môn đại cương màu xanh da trời. Để giải quyết vấn đề này chúng ta sẽ sử dụng thuộc tính HTML là Id để nhận dạng các môn chuyên ngành và dùng class để nhóm các môn đại cương. Đoạn HTML của chúng ta bây giờ sẽ là :

<p>Danh sách các môn học</p>

<li id =“tlw”>Lập trình web ASPX

<li id=“tkw”>Thiết kế web

<li id=“pttk”>Phân tích thiết kế

<li class=“dc”>Vật lý

<li clas=“dc”>Toán A1

<li class=“dc”>Tin học Đại cương

<li id=“fl”>Flash

Và đoạn CSS cần dùng sẽ là

#ltw

{ color:# 790000 }

#tkw

{ color:#FF0000 }

#pttk

{ color:#FF00FF }

#fl

{ color:"#FFFF00" }

.dc

{ color:#0000FF }

Lưu ý: Không nên đặt tên Id với ký tự đầu là chữ số, nó sẽ không làm việc cho Firefox.

Trải qua hai ví dụ trên ta thấy:

- Class được đặt tên với ký tự đầu tiên là dấu chấm (.)
- Class dùng để nhóm các đối tượng có cùng thuộc tính, do tính chất đó nó có thể được sử dụng nhiều lần.
- Id được đặt tên với ký tự đầu tiên là dấu thăng (#)
- Id dùng để nhận dạng một đối tượng đặc trưng, id có tính duy nhất.

3.8. Span & Div

3.8.1. Thẻ :

Thẻ trong HTML thật ra là một thẻ trung hòa, nó không thêm hay bớt bất cứ một thứ gì vào một tài liệu HTML cả. Nhưng chính nhờ tính chất trung hòa này mà nó lại là một công cụ đánh dấu tuyệt vời để qua đó chúng ta có thể viết CSS định dạng cho các phần tử mong muốn.

Ví dụ: Chúng ta có đoạn HTML sau:

<p>Không có gì quý hơn **độc lập, tự do.**</p>

Yêu cầu ở đây là chúng ta hãy dùng CSS tô đậm 2 từ **độc lập, tự do.**

Để giải quyết vấn đề này, chúng ta sẽ thêm thẻ vào đoạn HTML như sau:

<p>Không có gì quý hơn **độc lập**, **tự do**</p>.

Và bây giờ chúng ta có thể viết CSS cho yêu cầu
trên: **.nhanmanh**
{ font-weight:bold }

3.8.2. Thẻ <Div>

Cũng như , <div> cũng là một thẻ trung hòa và được thêm vào tài liệu HTML với mục đích định dạng bằng CSS. Tuy nhiên, điểm khác biệt là dùng để nhóm một khối phần tử trong khi đó <div> có thể nhóm một hoặc nhiều khối phần tử.

Trở lại ví dụ về danh sách các môn học trong phần class trước ta sẽ giải quyết vấn đề bằng cách nhóm các phần tử với <div> như sau:

<p>Danh sách các môn học</p>

<div id =“cn”>

Lập trình web ASPX

Thiết kế web

Phân tích thiết kế

Flash

</div>

<div class =“dc”>

Vật lý

Toán A1

Tin học Đại cương

</div>

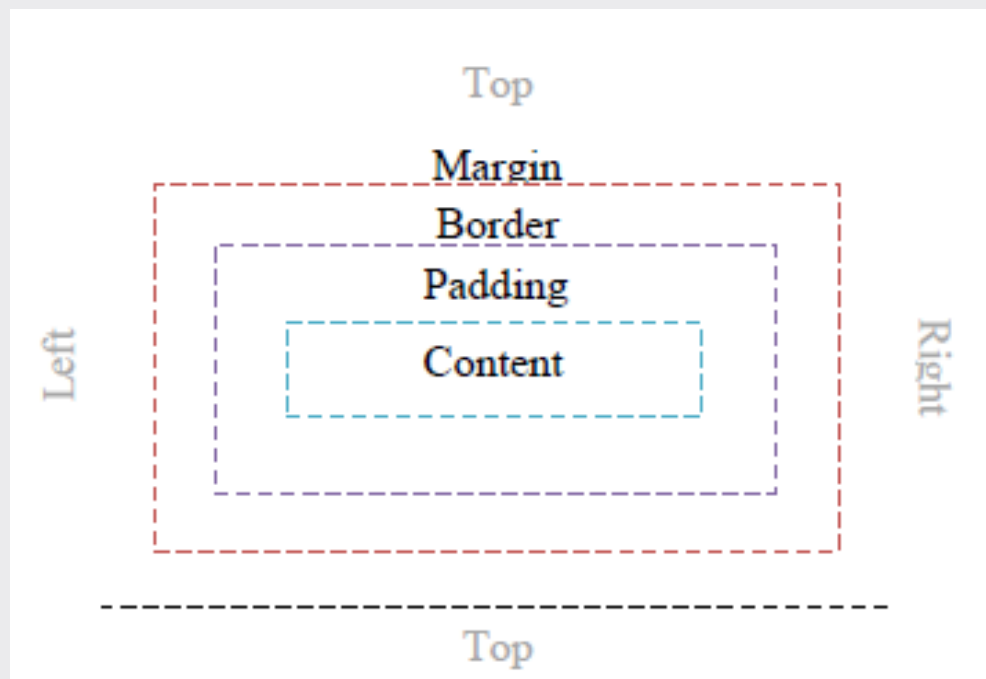
Và đoạn CSS sẽ là:

#cn { color:#FF0000 }

.dc { color:0000FF }

3.9. Box Model

Box model (mô hình hộp) mô tả cách mà CSS định dạng khối không gian bao quanh một thành phần. Nó bao gồm padding (vùng đệm), border (viền) và margin (canh lề) và các tùy chọn. Hình bên dưới mô tả cấu trúc minh họa mô hình hộp cho một thành phần web.



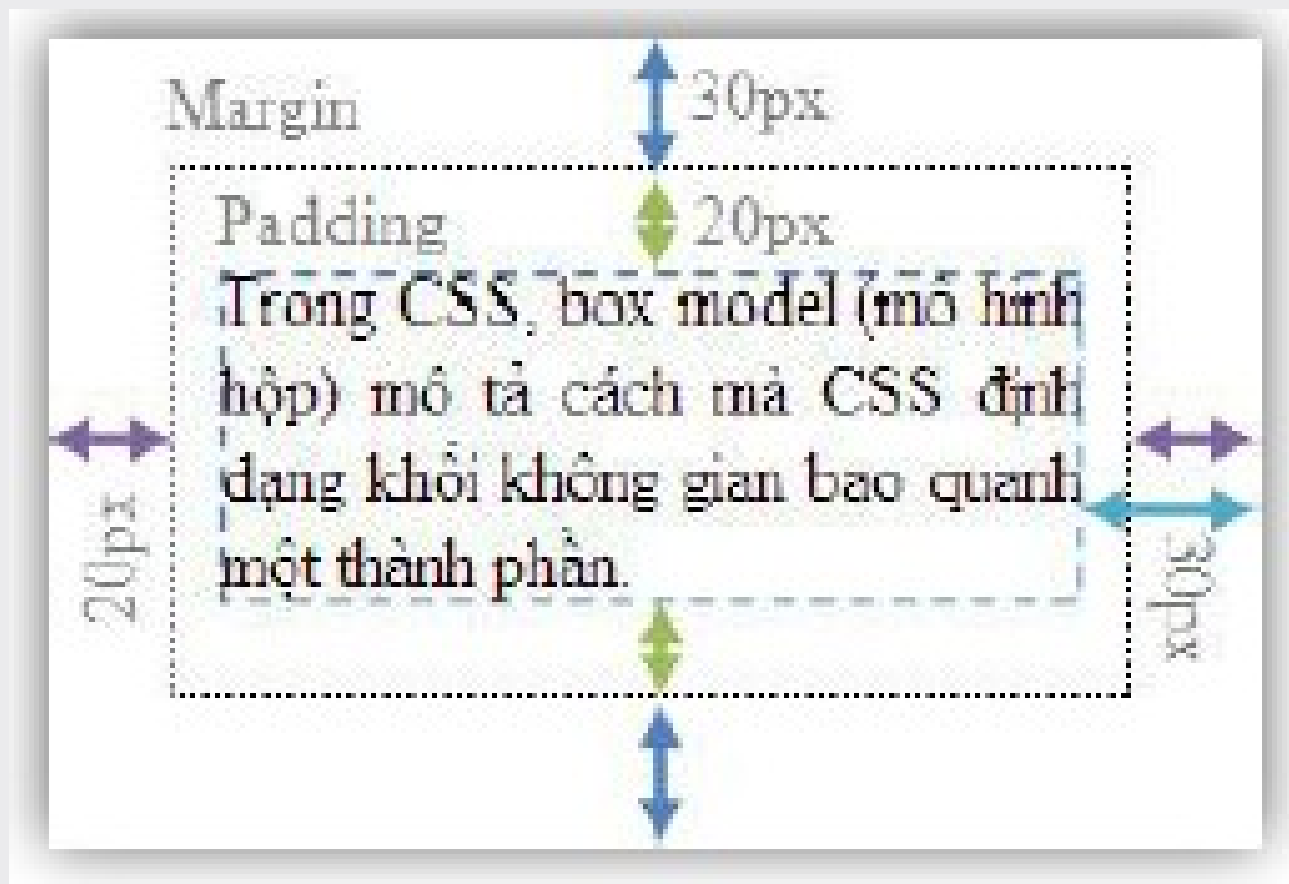
Ví dụ: Chúng ta có một đoạn HTML sau:

<p> Trong CSS, box model (mô hình hộp) mô tả cách mà CSS định dạng khối không gian bao quanh một thành phần. </p>

Phần CSS cho đoạn HTML trên:

```
p
{
width:200px;
margin:30px 20px;
padding:20px 10px;
border:1px solid #000;
text-align:justify
}
```


Ta được mô hình hộp như sau:



3.10. Margin & Padding

3.10.1. Margin

Thuộc tính ***Margin*** trong CSS cũng được dùng để canh lề cho cả trang web hay một thành phần web này với các thành phần web khác hay với viền trang.

Ví dụ sau sẽ chỉ cho chúng ta biết cách canh lề cho một trang web.

```
body
{
    margin-top:80px;
    margin-bottom:40px;
    margin-left:50px;
    margin-right:30px;
    border:1px dotted #FF0000
}
```

Hoặc gọn hơn:

body

{

margin:80px 30px 40px 50px;

border:1px dotted #FF0000

}

Cách viết tắt như sau:

margin:<margin-top> | <margin-right> | <margin-bottom> | <margin-left>

Các ví dụ:

Body

```
{ margin:80px 30px 40px 50px;  
border:1px solid #FF0000  
}
```

#box1

```
{ margin:50px 30px 20px 40px;  
border:1px solid #00FF00  
}
```

#box2

```
{ margin:50px 30px 20px 40px;  
border:1px solid #0000FF  
}
```


3.10.1. Padding

Padding có thể hiểu như là một thuộc tính đệm. Padding không ảnh hưởng tới khoảng cách giữa các đối tượng như margin mà nó chỉ quy định khoảng cách giữa phần nội dung và viền của một đối tượng

Cú pháp: Tương tự margin.

Padding:<padding-top> | <padding-right> | <padding-bottom> | <padding-left>

3.11 Border

3.11.1. Thuộc tính border-width

Border-width quy định độ rộng cho viền của một đối tượng web. Thuộc tính này có các giá trị: **thin** (mảnh), **medium** (vừa), **thick** (dày), hay là một giá trị đo cụ thể như pixels

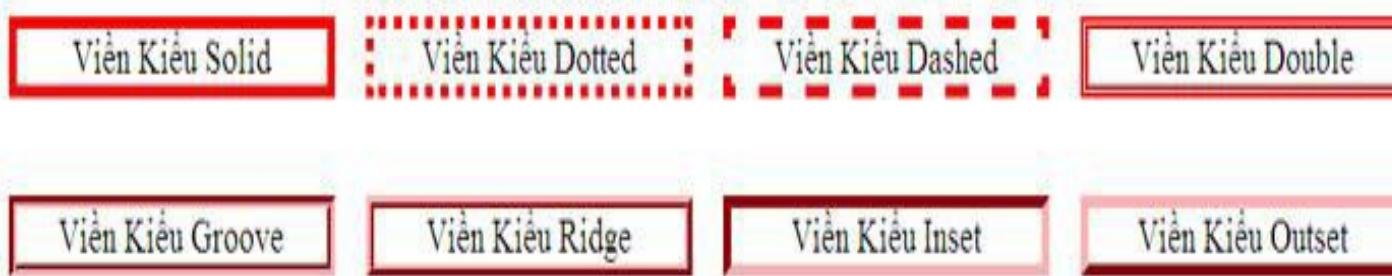
3.11.2. Thuộc tính border-color

Border-color quy định màu viền cho một đối tượng web. Thuộc tính này nhận tất cả đơn vị màu CSS hỗ trợ.

3.11.3. Thuộc tính border-style

Border-style quy định kiểu viền thể hiện của một đối tượng web. CSS cung cấp tất cả 8 kiểu viền tương ứng với 8 giá trị: **dotted**, **dashed**, **solid**, **double**, **groove**, **ridge**, **inset** và **outset**. Ngoài ra, hai giá trị **none** hay **hidden** dùng để ẩn đường viền.

Các kiểu đường viền



Ví dụ:

h1

```
{ border-width:thin;  
  border-color:#FF0000;  
  border-style:solid  
}
```

h2

```
{ border-width:thick;  
  border-color:#CCC;  
  border-style:dotted  
}
```

p

```
{ border-width:5px;  
  border-color:#FF00FF;  
  border-style:double  
}
```


Ngoài ra, chúng ta cũng có thể dùng riêng các thuộc tính ***border-top***, ***border-right***, ***border-bottom*** hay ***border-left*** để chỉ định viền riêng cho các đối tượng.

Ví dụ:

```
h1 { border-top-width:thin;  
border-top-color:#FF0000;  
border-top-style:solid;  
border-right-width:thick;  
border-right-color:#AFAFAF;  
border-right-style:dotted;  
border-bottom-width:2px;  
border-bottom-color:blue;  
border-bottom-style:double;  
border-left-width:5px;  
border-left-color:violet;  
border-left-style:groove }
```


Ta có thể sử dụng cấu trúc rút gọn của CSS như sau:
Border:<border-width> |<border-color> |<border-style>

3.12 Height & Width

3.12.1. Thuộc tính width

Width dùng để quy định chiều rộng cho một thành phần web.

Ví dụ.

```
p  
{ width:700px; }
```

3.12.2. Thuộc tính max-width:

Max-width quy định chiều rộng tối đa cho một thành phần web.

3. 12.3. Thuộc tính min-width:

Min-width quy định chiều rộng tối thiểu cho một thành phần web.

3.12.4. Thuộc tính height:

Height là một thuộc tính CSS dùng để quy định chiều cao cho một thành phần web.

Ví dụ:

```
p { height:300px }
```

3.12.5. Thuộc tính max-height:

Max-height quy định chiều cao tối đa cho một thành phần web.

3.12.6. Thuộc tính min-height:

Min-height quy định chiều cao tối thiểu cho một thành phần web.

Lưu ý:

- Thông thường chiều cao một thành phần web do văn bản trong trang web đó quyết định. Việc định chiều cao chính xác cho một thành phần sẽ tạo thành cuộn văn bản nếu chiều cao văn bản lớn hơn chiều cao đã định.

- Các thuộc tính **max/min-width/height** được sử dụng trong những trường hợp ta không chắc giá trị chính xác cho **width**, **height** cả một thành phần.

Ví dụ, Vùng chứa bài post của một forum có bề ngang 500px, bạn có thể định **max-width:500px** cho phần hình ảnh trong phần đó để tránh những ảnh lớn bị lệch ra ngoài.

3.13. Float & Clear

3.13.1 Float:

Float (theo như nghĩa tiếng Việt là thả trôi) dùng để cố định một thành phần web về bên trái hay bên phải không gian bao quanh nó.

Đây là một thuộc tính rất cần thiết khi dàn trang (như tạo các trang web 2 column layout hay 3 column layout), hiển thị văn bản thành cột, hay thực hiện việc định vị trí ảnh và text.

Thuộc tính float có 3 giá trị:

- + Left: Cố định phần tử về bên trái.
- + Right: Cố định phần tử về bên phải.
- + None: Bình thường.

Box B di chuyển sang
trái



Content tràn lên để lấp “khoảng trống”



Nhìn vào hình minh họa trên chúng ta thấy là ban đầu trong box lớn có hai thành phần là Box B và phần Content. Lúc đầu Box B nằm bên trên và Content nằm bên dưới, nhưng khi chúng ta đặt thuộc tính float cho Box B thì Box B bị cố định về bên trái và chừa lại khoảng trống bên trái nó. Còn phần Content thì vốn nằm bên dưới sẽ tự động tràn lên để lấp đầy khoảng trống do Box B tạo ra.

Ví dụ sau, chúng ta sẽ thực hiện float ảnh logo sang trái để phần nội dung bên dưới tràn lên nằm cạnh logo.

```
#logo  
{  
  float:left;  
}
```

Ví dụ: Chúng ta sẽ thử dùng float để chia 2 cột văn bản.

```
.column1, .column2  
{ width:45%; float:left;  
  text-align:justify;  
  padding:0 20px;  
}
```

```
.column1 {  
  border-right:1px solid #000  
}
```

3.13.2 Clear:

Đi cùng với thuộc tính float, trong CSS còn có một thuộc tính là Clear.

Thuộc tính clear là một thuộc tính thường được gán vào các phần tử liên quan tới phần tử đã được float để quyết định hướng xử sự của phần tử này.

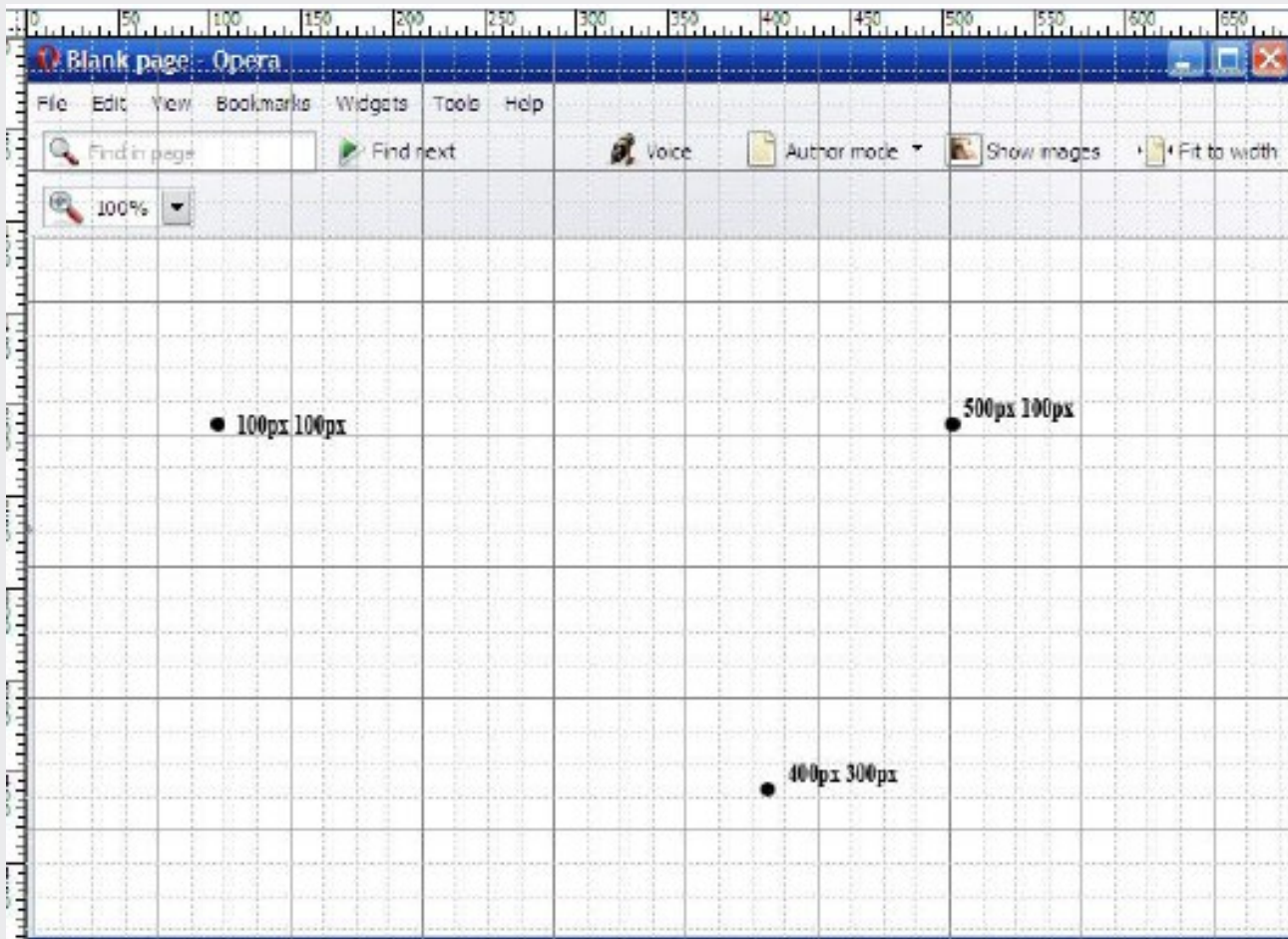
Ở ví dụ trên, khi chúng ta float tấm ảnh qua trái thì mặc nhiên văn bản sẽ được tràn lên để lấp vào chỗ trống. Nhưng khi chúng ta đặt vào văn bản thuộc tính clear thì chúng ta có quyền quyết định xem phần văn bản đó có được tràn lên hay không.

Thuộc tính clear có tất cả 4 thuộc tính:

- left (tràn bên trái),
- right (tràn bên phải),
- both (không tràn)
- none.

3.14: Position

Hãy tưởng tượng cửa sổ trình duyệt của ta giống như một hệ tọa độ và với position ta có thể đặt một đối tượng web ở bất cứ vị trí nào trên hệ tọa độ này.



Ví dụ: Chúng ta muốn định vị một ảnh ở vị trí 70px cách đỉnh và 90px từ bên trái tài liệu, chúng ta sẽ viết CSS như sau:

```
img
{
    position:absolute; top:70px; left:90px
}
```

3.14.1 Absolute position:

Định vị tuyệt đối là sự định vị mà trong đó các thành phần được định vị không để lại bất cứ một khoảng trống nào trong tài liệu. Một thành phần được định vị tuyệt đối sẽ nhận giá trị position là absolute. Các đối tượng đã định vị tuyệt đối sẽ dùng kết hợp với các thuộc tính top, left, right, bottom để xác định tọa độ.

Ví dụ sau sẽ chỉ cho chúng ta cách đặt bốn ảnh ở bốn góc tài liệu bằng định vị tuyệt đối.

#logo1

```
{ position:absolute; top:50px; left:70px }
```

#logo2

```
{ position:absolute; top:0; right:0 }
```

#logo3

```
{ position:absolute; bottom:0; left:0 }
```

#logo4

```
{ position:absolute; bottom:70px; right:50px }
```

3.14.1 Relative position:

Sự định vị tương đối cho một thành phần là sự định vị được tính từ vị trí gốc trong tài liệu. Các thành phần đã được định vị tương đối sẽ để lại khoảng không trong tài liệu. Các thành phần được định vị tương đối sẽ nhận giá trị **position là relative**.

3.15 Layers

Layers là cách bạn đặt một thành phần này lên trên một thành phần khác. Với thuộc tính z-index

Ví dụ sau chúng ta sẽ đặt 5 bức ảnh logo ở 5 lớp.

#logo1

```
{ position:absolute; top:70px; left:50px; z-index:1  
}
```

#logo2

```
{ position:absolute; top:140px; left:100px; z-index:2  
}
```

#logo3

```
{ position:absolute; top:210px; left:150px; z-index:3  
}
```

#logo4

```
{ position:absolute; top:280px; left:200px; z-index:4  
}
```

#logo5

```
{ position:absolute; top:350px; left:250px; z-index:5  
}
```




The end