

GIÁO TRÌNH LẬP TRÌNH HTML – CSS

MỤC LỤC

0. Giới thiệu.....	3
0.1 Giới thiệu tài liệu.....	3
0.2 Điều có thể làm sau khi đọc xong tài liệu	3
1. Hướng dẫn học và thực hành tài liệu.....	3
1.1 Phần mềm hỗ trợ lập trình.....	3
1.2 Hướng dẫn học	4
1.3 Hướng dẫn mở xem mã nguồn ví dụ.....	4
2. Các khái niệm cơ bản về html	7
2.1 Giới thiệu về Html.....	7
2.2 Cấu trúc tài liệu html	7
2.3 Thẻ (tag) html.....	7
3. Các thẻ html hay dùng.....	8
3.1 Thẻ div.....	8
3.2 Thẻ p.....	14
3.3 Thẻ ul.....	14
3.4 Thẻ h.....	15
3.5 Thẻ a	15
3.6 Thẻ span.....	16
3.7 Thẻ br.....	16
3.8. Thẻ fieldset.....	16
3.9. Thẻ table :.....	17
4 Thẻ Inline – Block.....	19
5. Tổng kết về HTML.....	20
Phần 2 CSS (Cascading Style Sheets).....	20
6. Khai báo CSS	21
7.Bộ chọn (CSS Selector).....	26
7.1 Khái niệm về CSS Selector	26
7.2 Khái niệm id và class.....	26
7.3 Các CSS Selector hay dùng :.....	28
8. Định dạng nội dung văn bản.....	29

9. Định dạng kích thước đối tượng với thuộc tính width , height.....	30
10. Mô hình hộp Box – Model	33
11. Float – clear trong css.....	36
11.1 Float.....	36
11.2 Clear	38
11.3 Chú ý thuộc khi sử dụng float và border	39
12 . Display.....	40
13. Position	43
14. Định dạng 1 vài thẻ html đặc biệt.....	45
14.1 Thẻ a	45
14.2 Thẻ ul.....	46
15. Giới thiệu về HTML5 , CSS 3.....	47
15.0 Giới thiệu Html5 – CSS 3	47
15.1 Các thẻ mới dùng trong thiết kế layout	48
15.2 Các thẻ media và đồ họa.....	48
15.3 Một vài thẻ khác	48
16. Thực hành thiết kế giao diện HTML – CSS.....	49
16.1 Giới thiệu.....	49
16.2 Xây dựng layout dùng thuộc tính float.....	51
16.3 Xây dựng layout dùng inline-block.....	54
16.4 Xây dựng layout dùng HTML 5.....	56
16.5. Định dạng menu website	57
16.6 Kết hợp ví dụ 16.4 - 16.5.....	59
17. Tổng kết.....	60

cuu duong than cong . com

0. Giới thiệu

0.1 Giới thiệu tài liệu

- Các bạn đang đọc tài liệu về hướng dẫn xây dựng , định dạng tài liệu HTML
- Html là ngôn ngữ đánh dấu văn bản chuẩn để truyền tải nội dung qua mạng Internet .
- Tất cả các công nghệ lập trình web của các hãng khác nhau mục đích cuối cùng đều sinh ra mã html để hiển thị lên trình duyệt web .
- Html dùng các thẻ để đánh dấu văn bản .Mỗi thẻ đánh dấu được qui ước hiển thị nội dung nhất định
- CSS là 1 ngôn ngữ dùng để trang trí và định dạng lại trong html , làm cho tài liệu html thêm sinh động
- Do đó nếu bạn muốn tự học xây dựng các ứng dụng web bằng các công nghệ như asp.net , php , jsp (hoặc các bạn chỉ là người đang sử dụng các phiên bản mã nguồn mở của các công nghệ này) thì các bạn đều cần kiến thức về html , CSS .Bởi Html , CSS là một nền tảng lập trình web không thể thiếu đối với lập trình viên web .
- Các phiên bản mới nhất của html và CSS là HTML 5 và CSS 3 vẫn chưa hoàn thành , và các trình duyệt web hiện tại vẫn chưa hỗ trợ đầy đủ hết hứa hẹn đưa ra những chuẩn lập trình web rất mới và hiện đại , và có thể là một cuộc cách mạng trong kỹ thuật lập trình web

0.2 Điều có thể làm sau khi đọc xong tài liệu

Khi đọc và thực hành xong tài liệu các bạn có thể :

- Dùng html và CSS để định dạng các trang html tĩnh .
- Nếu có kiến thức về các công nghệ lập trình web động như asp.net , php các bạn có thể định dạng các trang web động này .
- Hầu như các bạn có thể tạo và định dạng hiển thị nội dung tất cả các trang web theo sở thích và yêu cầu công việc

1. Hướng dẫn học và thực hành tài liệu

1.1 Phần mềm hỗ trợ lập trình

- Chúng ta thực hành html , css bằng công cụ Visual Studio 2012 (có thể xử dụng phiên bản 2010 , 2008 tuy nhiên không có 1 vài tính năng Intellicense cho CSS và html5)
- Ngoài ra chúng ta cũng nên sử dụng notepad ++ như 1 IDE song song .

- Môi trường VS của Microsoft hỗ trợ lập trình html , css ... rất mạnh mẽ .Chúng hỗ trợ tự động hóa hầu hết các đoạn code nhàm chán , giúp chúng ta chỉ quan tâm tới phát triển Logic ứng dụng hơn là tập chung tới cú pháp ngôn ngữ .

1.2 Hướng dẫn học

- Trong các bài hướng dẫn dưới đây các bạn hãy đọc kỹ nội dung , và xem mã nguồn gửi kèm tài liệu .



- Mỗi ví dụ tương ứng với tên file , thư mục mã nguồn .Ví dụ khi đọc Ví dụ 2 thì các bạn tìm tới file hay thư mục có tên là ViDu2 để xem mã nguồn và chạy thử

- Các bạn xem kỹ 1 vài ví dụ hướng dẫn thực hành tạo ví dụ trên môi trường phát triển Visual Studio và cách chạy các ví dụ mã nguồn đi kèm tài liệu để hiểu môi trường thực hành viết code

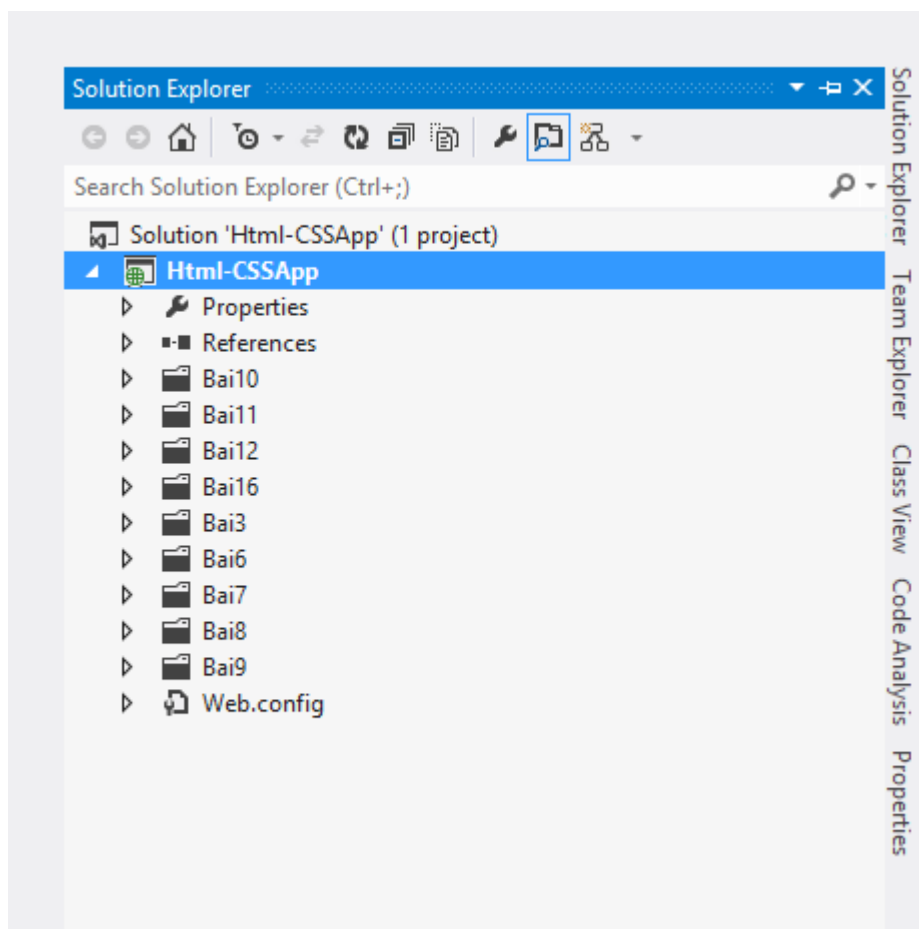
1.3 Hướng dẫn mở xem mã nguồn ví dụ

Mở bằng Visual Studio :

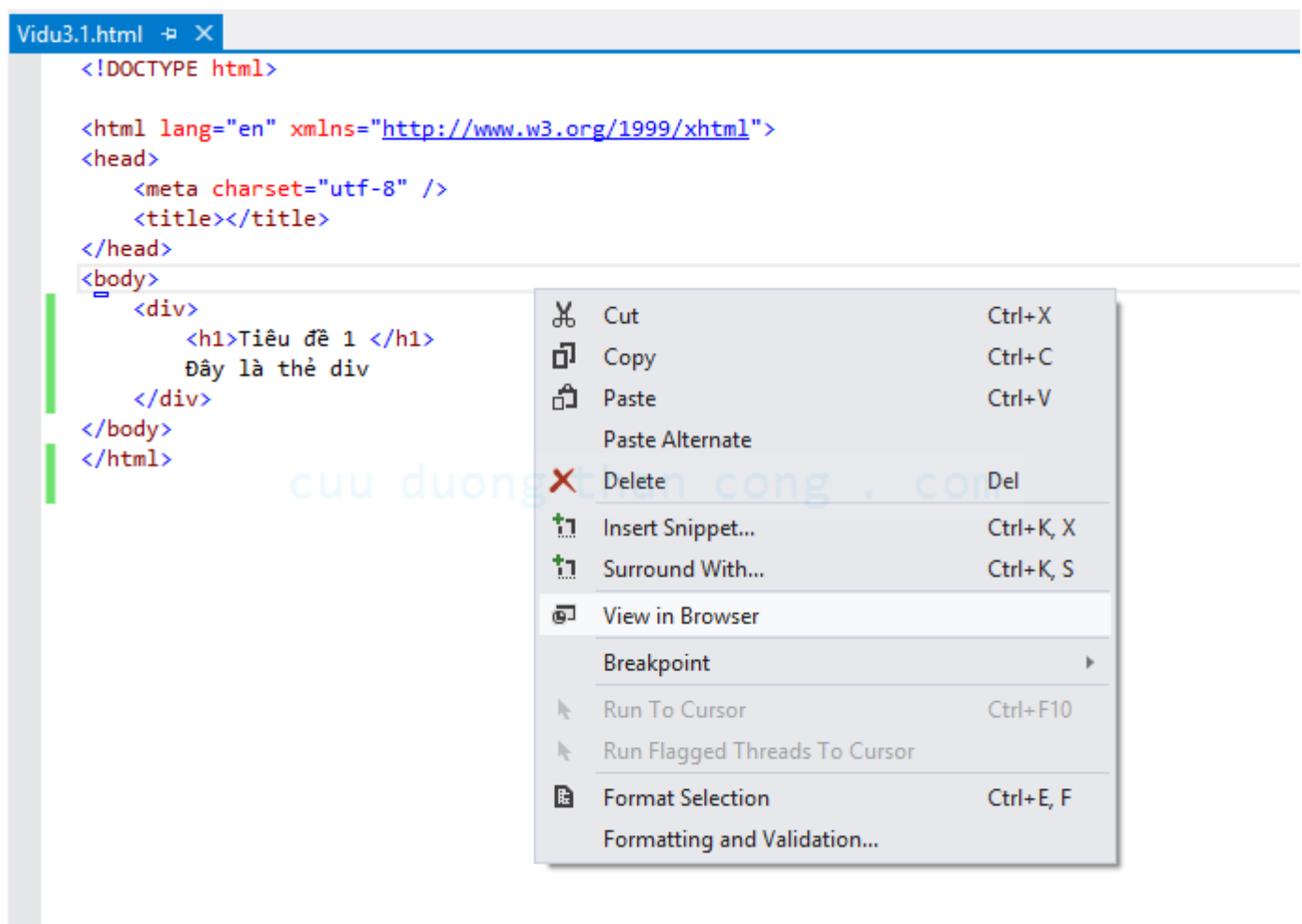
Click vào file Html-CSSApp.sln như hình dưới

Name	Date modified	Type	Size
 Html-CSSApp	11/5/2013 8:15 AM	File folder	
 Html-CSSApp	11/5/2013 8:09 AM	Microsoft Visual S...	1 KB

VS sẽ mở Project ví dụ lên được kết quả như hình dưới :



Giả sử ta muốn xem và chỉnh sửa file ví dụ của Bai3 ta tìm tới thư mục Bai3 và mở lên được giao diện chỉnh sửa code như hình vẽ

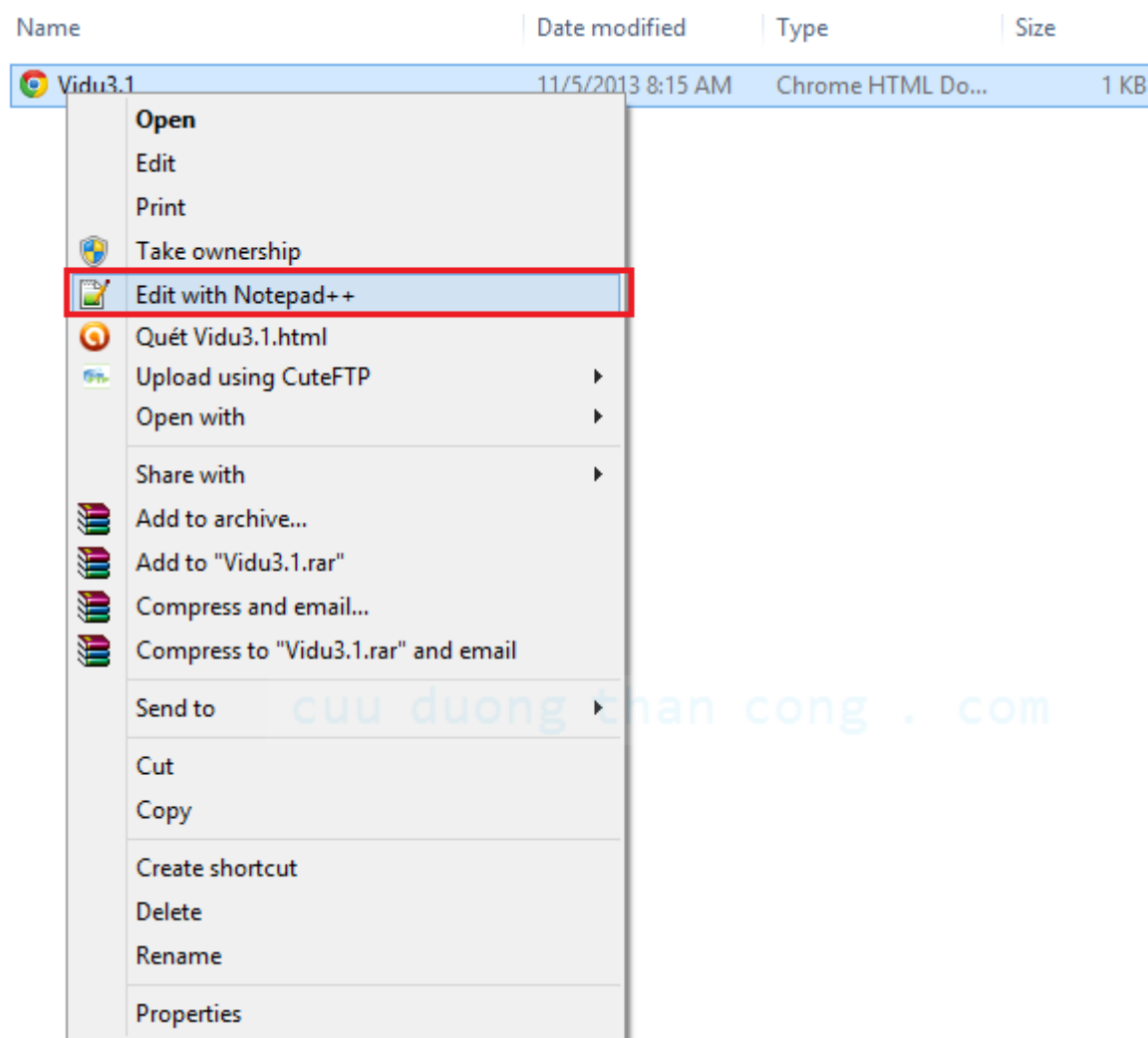


- ta có thể chạy thử hoặc thay đổi code theo ý thích

Mở bằng notepad ++ :

Ngoài ra để xem nhanh mã nguồn ví dụ có thể mở bằng notepad ++ như sau :

Chuột phải lên file ví dụ muốn xem code , giả sử ở đây muốn xem code file Vidu3.1.html , chọn chương trình mở là notepad++ như hình dưới :



Notepad++ sẽ mở nội dung code Vidu3.1.html như sau :

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title></title>
</head>
<body>
  <div>
    <h1>Tiêu đề 1 </h1>
    Đây là thẻ div
  </div>
</body>
</html>
```

2. Các khái niệm cơ bản về html

2.1 Giới thiệu về Html

- Bất kỳ ngôn ngữ hay kỹ thuật lập trình web nào cuối cùng đều sinh ra mã Html để hiển thị lên trình duyệt web .
- Do vậy dù bạn có phát triển ứng dụng web bằng asp.net , php , jsp thì cũng đều cần có hiểu biết về html (cũng như các công cụ bên cạnh nó là javascript , css) .
- Thông thường ta thiết kế 1 website thì chỉ cần tạo ra các tài liệu html , sau đó upload lên máy chủ web .Tuy nhiên ở đây tài liệu html lại khó thay đổi nội dung (phải chỉnh sửa nội dung trong mã nguồn) .Do đó ta gọi là các website hay tài liệu html tĩnh .
- Các tài liệu html được tự động sinh từ các ngôn ngữ lập trình như .net , php , java thì có thể tạo động các mã html (tức là có thể tạo giao diện thay đổi nội dung) .Ta gọi các website này là website động .
- Trong tài liệu này chúng ta chủ yếu tìm hiểu về html tĩnh , để hiểu cơ chế định dạng – trang trí – xử lý html .Việc xây dựng các trang html động chúng ta cần tham khảo các khóa học về asp , java , php

2.2 Cấu trúc tài liệu html

- Các tài liệu html tuân thủ theo cấu trúc sau

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Tiêu đề website</title>
</head>
<body>
  <!-- Code mã html tại đây -->
</body>
</html>
```

2.3 Thẻ (tag) html

- html là ngôn ngữ đánh dấu tài liệu dùng các thẻ (div , h , a , ul ...) , mỗi thẻ có ý nghĩa riêng , và được sử dụng tùy biến để mô tả tài liệu .
- Mỗi thẻ html có các **attribute** , để ta có thể dễ dàng thể hiện nội dung thẻ html cũng như truy xuất tới chúng 1 cách dễ dàng , ví dụ :

```
<input type="text" name="name1"/>
```

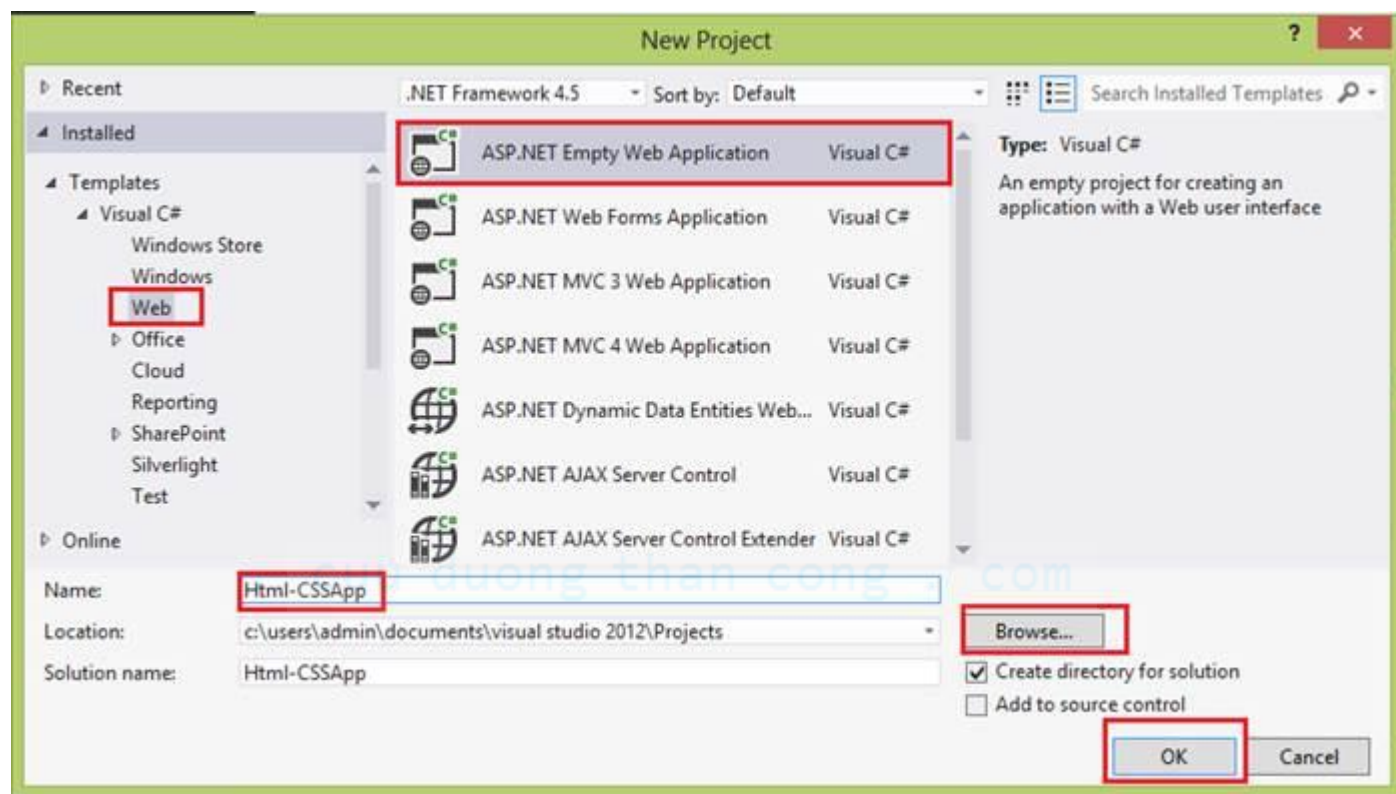
// thẻ input là thẻ dùng tạo các Control nhập liệu trong html , ở đây hiển thị 1 ô textBox dùng để nhập chữ (attribute type) và đặt tên cho thẻ là name1 (thông qua Attribute name)

- Các Attribute tương đương được dùng rất nhiều đó là : name . id , class

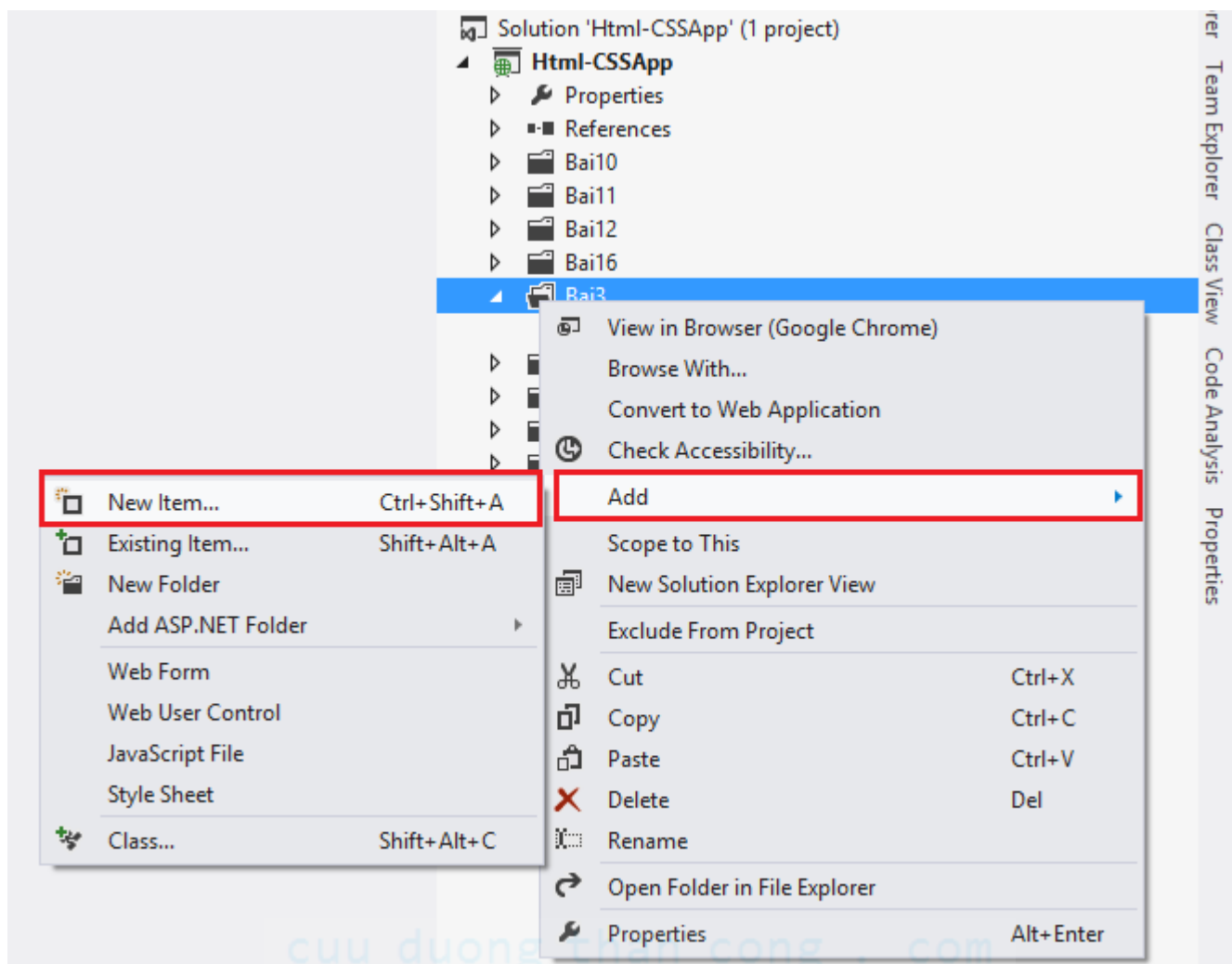
3. Các thẻ html hay dùng

3.1 Thẻ div

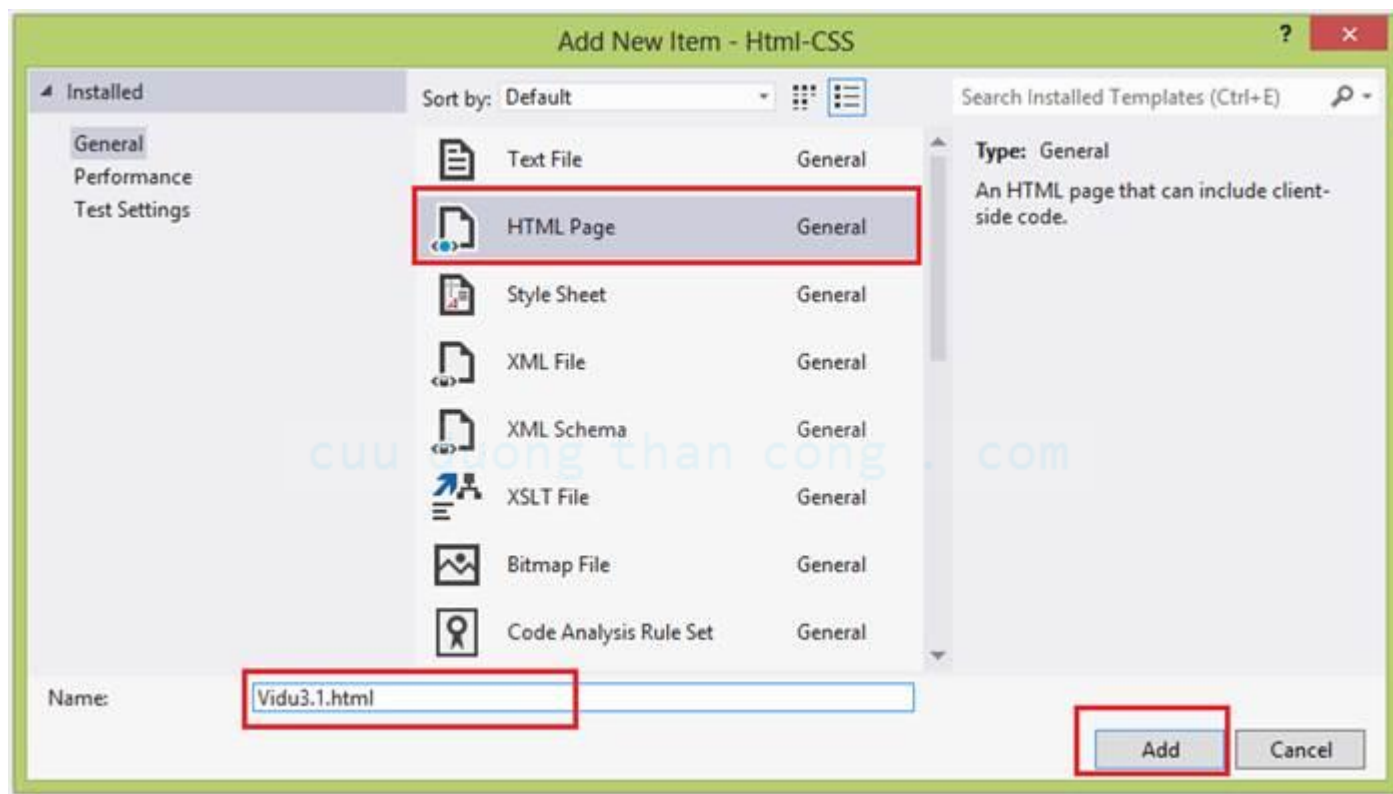
- Thẻ div dùng để nhóm khối phần tử html
- Hướng dẫn dùng Visual Studio thực hành viết mã html
- Mở Visual Studio (ở đây chúng tôi hướng dẫn trên phiên bản VS 2012). Chọn mẫu Template Project theo hình sau :



- Nhấn Ok , mở Solution right lên Project chọn theo hình sau để tạo thư mục trong Project



chọn mẫu html như hình dưới :



Ví dụ 3.1 (Xem file ViDu3.1.html trong mã nguồn kèm theo giáo trình)

// Với các thẻ khác các bạn cũng có thể tự tạo file html tương tự file ViDu1.html để xem ý nghĩa và phân tích được khi nào cần dùng thẻ tương ứng cho hợp lý

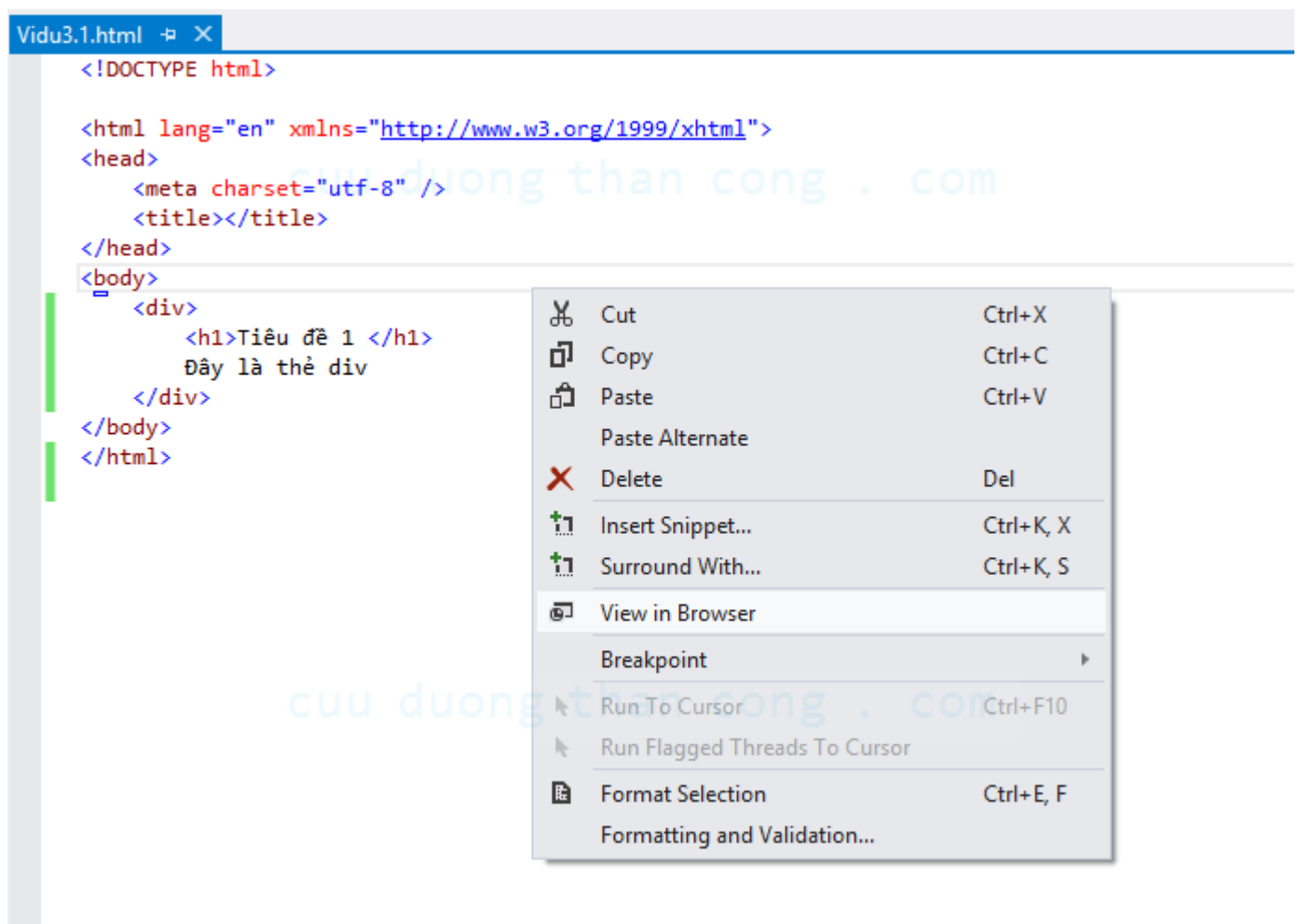
```
<div>
  <h1>Tiêu đề 1 </h1>
  Đây là thẻ div
</div>
```

Mã nguồn hoàn chỉnh

```
<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta charset="utf-8" />
  <title></title>
</head>
<body>
  <div>
    <h1>Tiêu đề 1 </h1>
    Đây là thẻ div
  </div>
</body>
</html>
```

// chạy ứng dụng bằng cách chuột phải chọn View In Browser như hình sau :



được kết quả như hình sau :



Tiêu đề 1

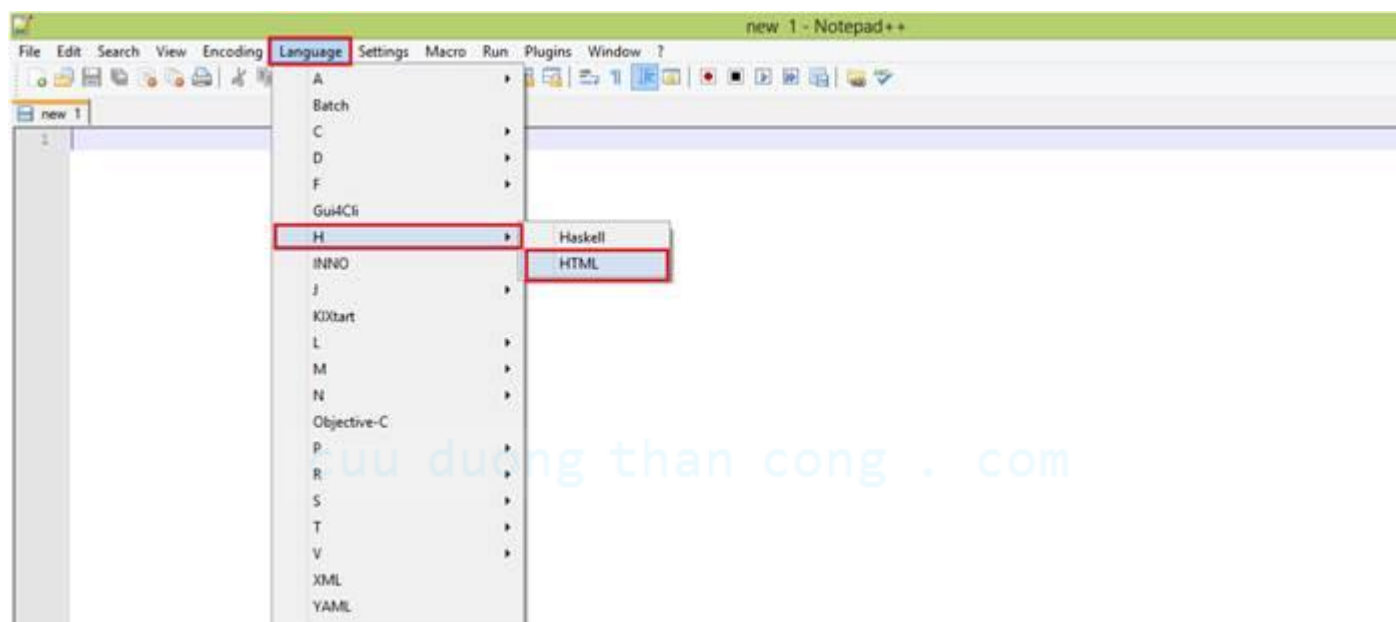
Đây là thẻ div

Sau này các ví dụ ta cũng thực hành trên môi trường Visual Studio làm tương tự

Hướng dẫn thực hành bằng notepad ++ :

// Mở notepad ++

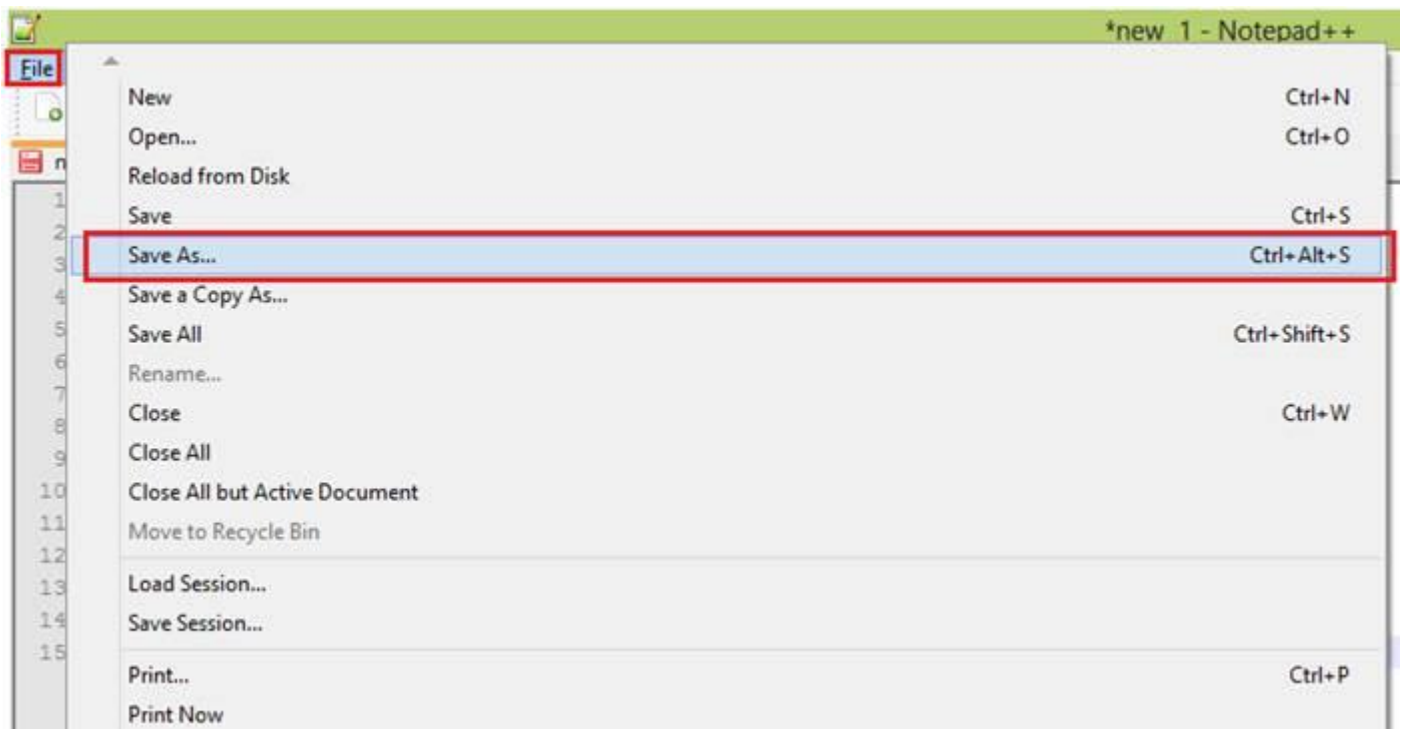
// Language viết code như hình vẽ :



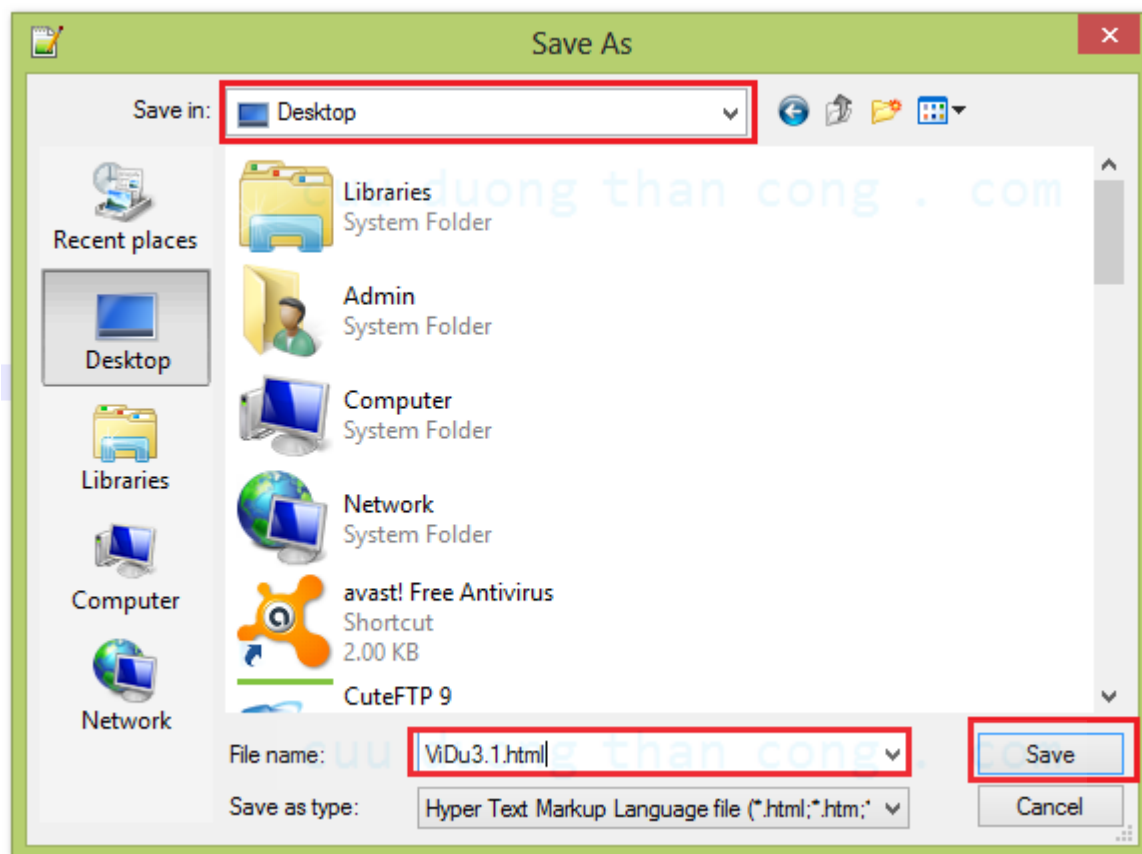
Soạn nội dung cho ví dụ 3.1 như sau :

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title></title>
</head>
<body>
  <div>
    <h1>Tiêu đề 1 </h1>
    Đây là thẻ div
  </div>
</body>
</html>
```

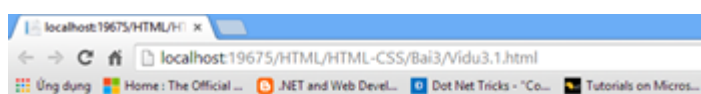
Lưu lại theo hình dưới



// Đặt tên là ViDu3.1.html và chọn nơi lưu như hình dưới :



// vào thư mục lưu file ViDu3.1.html mở bằng trình duyệt được kết quả như chạy bằng VS



Tiêu đề 1

Đây là thẻ div

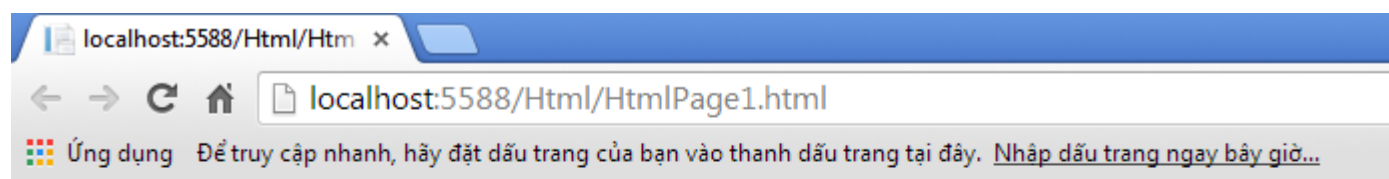
3.2 Thẻ p

// tương tự như với ví dụ về thẻ div , các bạn tự code các thẻ dưới đây để xem kết quả chạy trên trình duyệt .

p là thẻ mô tả đoạn văn bản (paragraph)

```
<p>
    Đoạn văn 1
</p>
<p>
    Đoạn văn 2
</p>
```

Kết quả :



Đoạn văn 1

Đoạn văn 2

cuu duong than cong . com

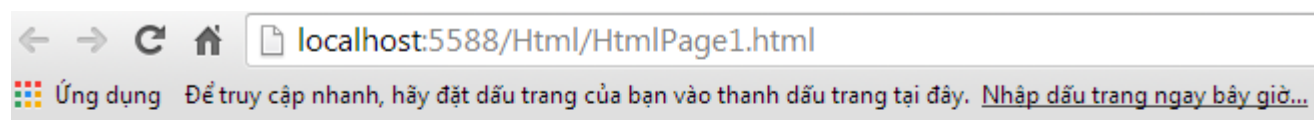
Không như thẻ div thành phần bên trong thẻ p không bao gồm các thẻ chứa nội dung như : thẻ h , thẻ div mà chỉ chứa các control html như input , button , label ...

3.3 Thẻ ul

- thẻ ul đi đôi với thẻ li dùng để liệt kê danh sách

```
<ul>
    <li>Danh sách 1</li>
    <li>Danh sách 2</li>
</ul>
```

Kết quả :



- Danh sách 1
- Danh sách 2

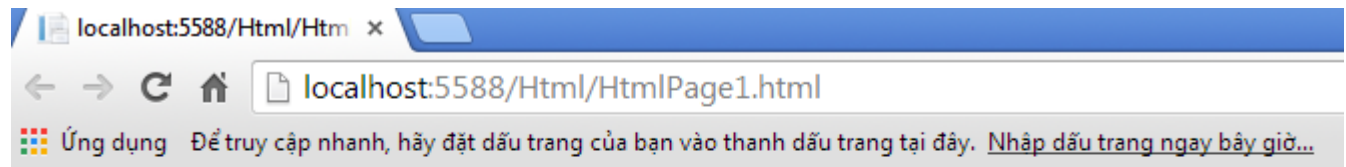
cuu duong than cong . com

3.4 Thẻ h

h (header) là thẻ dùng mô tả tiêu đề đối tượng , mỗi thẻ h có kích cỡ từ nhỏ tới lớn theo thứ tự từ h1 tới h6

```
<h1>Tiêu đề 1 </h1>  
<h2>Tiêu đề 2</h2>
```

Kết quả :



Tiêu đề 1

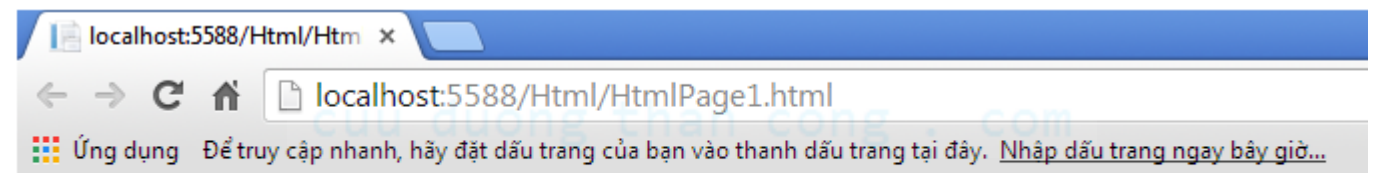
Tiêu đề 2

3.5 Thẻ a

- Thẻ a dùng để định nghĩa 1 link liên kết

```
<a href="http://google.com">Tìm Kiếm Google</a>
```

Kết quả :



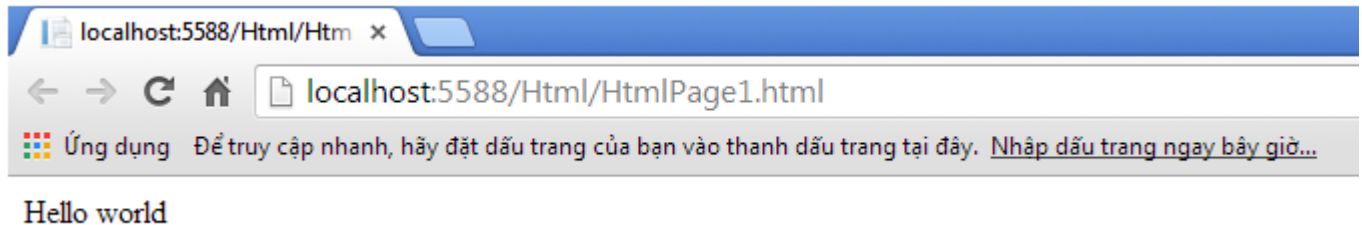
[Tìm Kiếm Google](http://google.com)

3.6 Thẻ span

- Thẻ span dùng để nhóm phần tử html , khác với thẻ div dùng để nhóm khối phần tử .Tức là trong thẻ div có thể có nhiều thẻ khác nhưng với thẻ span thì không .

```
<span>Hello world</span>
```

Kết quả :



- thẻ span có thể đặt trong thẻ p

3.7 Thẻ br

- thẻ br dùng để xuống dòng .Đây là 1 thẻ trung gian , không chứa nội dung

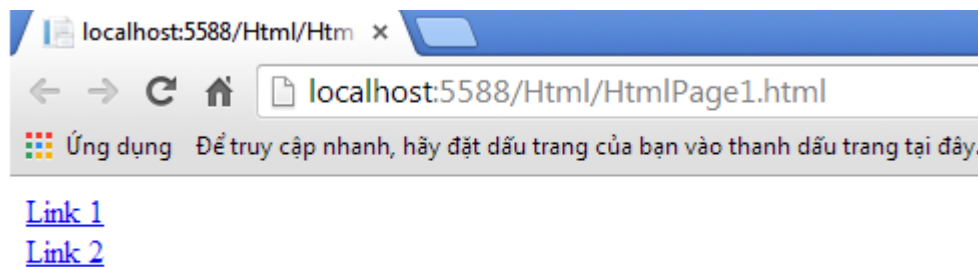
```
<a href="#">Link 1</a>
```

```
<br />
```

```
<a href="#">Link 2</a>
```

- Khi không có thẻ br thì Link 1 và Link 2 được đặt trên cùng 1 dòng

Kết quả :



3.8. Thẻ fieldset

- Thẻ fieldset dùng để bao quanh và nhấn mạnh 1 nội dung

```
<fieldset>
```

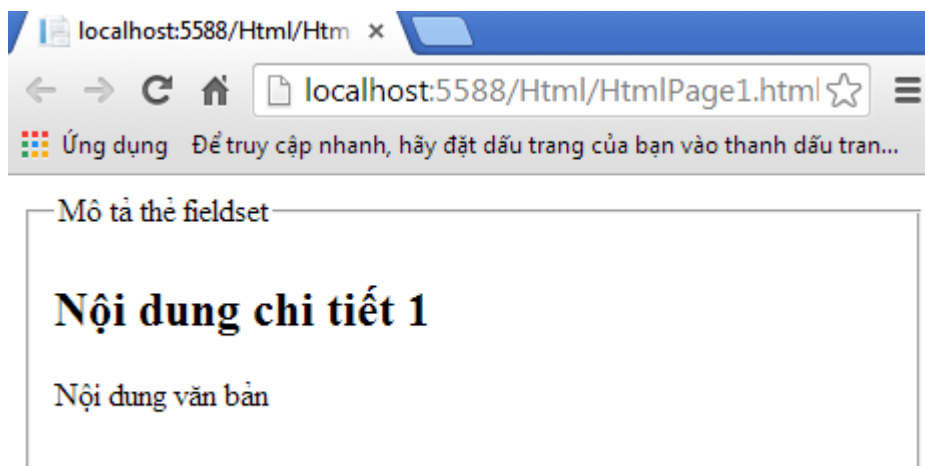
```
<legend>Mô tả thẻ fieldset</legend>
```

```
<h2>Nội dung chi tiết 1</h2>
```

```
<p>Nội dung văn bản</p>
```

```
</fieldset>
```

Kết quả :



3.9. Thẻ table :

- thẻ table được cấu thành từ các thẻ tr (table – row : dòng) và td (table – column : cột) , thẻ td phải đặt trong thẻ tr .

- 1 thẻ table đầy đủ chia làm 3 phần chính là :

Phần tiêu đề : thead

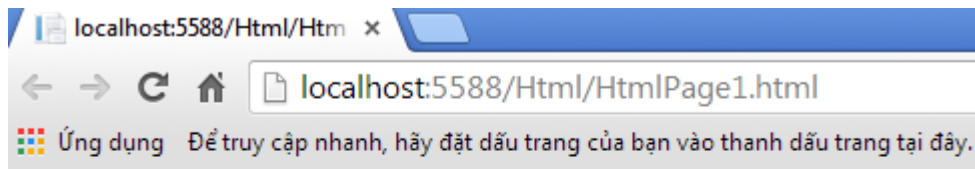
Phần nội dung chính : tbody

Phần chân tiêu đề : tfoot

- Cấu trúc thẻ table đầy đủ như sau :

```
<table>
  <thead>
    <tr>
      <td>Tiêu đề 1</td>
      <td>Tiêu đề 2</td>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Dòng 1 , cột 1</td>
      <td>Dòng 1 , cột 2</td>
    </tr>
    <tr>
      <td>Dòng 2 , cột 1</td>
      <td>Dòng 2 , cột 2</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td>Chân Tiêu đề 1</td>
      <td>Chân Tiêu đề 2</td>
    </tr>
  </tfoot>
</table>
```

Kết quả :

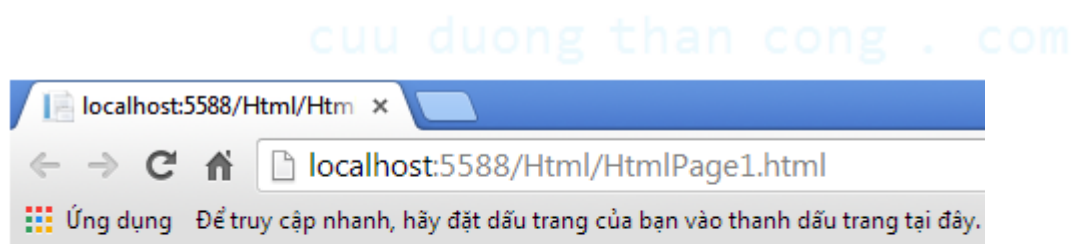


Tiêu đề 1	Tiêu đề 2
Dòng 1 , cột 1	Dòng 1 , cột 2
Dòng 2 , cột 1	Dòng 2 , cột 2
Chân Tiêu đề 1	Chân Tiêu đề 2

- Trong nhiều trường hợp không cần phân biệt ý nghĩa của dữ liệu ta có thể chỉ cần khai báo trực tiếp table bằng các thẻ tr và td

```
<table>  
  <tr>  
    <td>Dòng 1 , cột 1</td>  
    <td>Dòng 1 , cột 2</td>  
  </tr>  
  <tr>  
    <td>Dòng 2 , cột 1</td>  
    <td>Dòng 2 , cột 2</td>  
  </tr>  
</table>
```

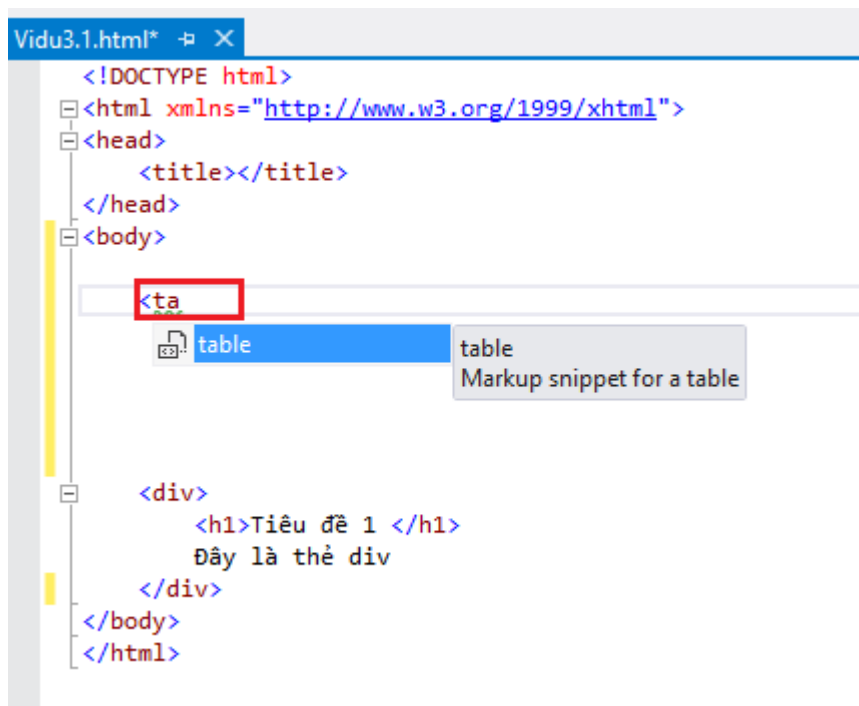
Kết quả :



Dòng 1 , cột 1	Dòng 1 , cột 2
Dòng 2 , cột 1	Dòng 2 , cột 2

Mẹo nhỏ :

- Trong môi trường VisualStudio hỗ trợ các Snippets các ngôn ngữ lập trình (html , css , C# ...) để gõ nhanh code .
- Giả sử muốn tạo nhanh thẻ table ta gõ <ta VS sẽ hiện Intellisense tới thẻ table , nhấn phím tab 2 lần VS sẽ tự động tạo thẻ table với các thẻ tr , td như hình sau :



// kết quả nhấn tab 2 lần

```
<table>
  <tr>
    <td></td>
  </tr>
</table>
```

- Một tính năng mạnh mẽ nữa đó là chúng ta có thể tạo các đoạn gõ tắt Snippets cho riêng mình .

4 Thẻ Inline – Block

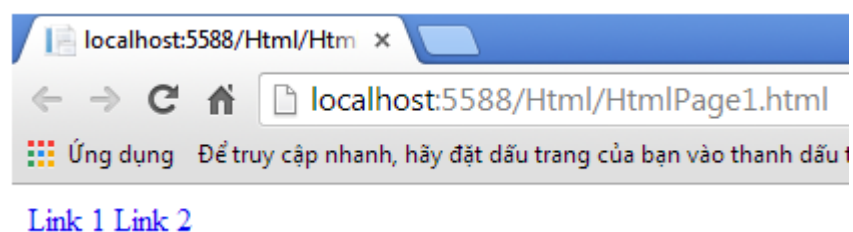
- Xét về việc sắp xếp các thẻ (trên cùng 1 hàng hay xuống hàng) các thẻ html chia ra làm 2 loại : Inline và Block

- thẻ Inline là các thẻ html mà các thẻ sau xếp cùng trên 1 hàng so với thẻ trước , khi không gian còn trống. Các loại thẻ Inline thường gặp như : , <a>, ,

Ví dụ :

```
<a href="#">Link 1</a>
<a href="#">Link 2</a>
```

Kết quả :



Code tạo 2 link nằm trên cùng 1 dòng .

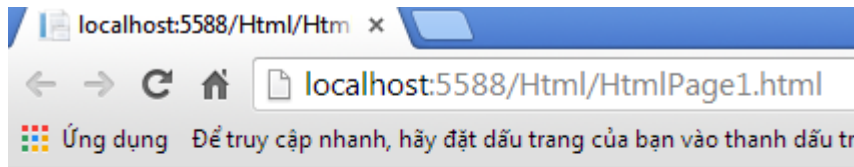
- Thẻ block là các thẻ html mà các thẻ theo sau nó tự động xuống hàng (không cùng 1 dòng)

Các loại thẻ block thường gặp như : <h1>, <p>, , <table> <div />

Ví dụ :

```
<h1>Tiêu đề 1 </h1>
<h2>Tiêu đề 2</h2>
```

Kết quả :



Tiêu đề 1

Tiêu đề 2

- Ta tạm hiểu đơn giản là các thẻ Inline nằm trên cùng 1 dòng , các thẻ block nằm trên các dòng riêng biệt .

- Ta có thể dùng CSS để thay đổi giá trị Inline hoặc block của bất kỳ thẻ html nào .

block	inline
div	a
p	em
h1 → h6	span
li ; dt ; dd (các thẻ danh sách)	br
table	img
blockquote	
pre	
form	

5. Tổng kết về HTML

- Ngôn ngữ html còn rất nhiều thẻ cũng như thuộc tính .Tuy nhiên chúng ta chỉ cần nắm vững các thẻ cũng như thuộc tính cơ bản trên là có thể bắt tay vào xây dựng nên các ứng dụng web

- Khi cần dùng 1 thuộc tính nào đó , hay tìm hiểu 1 thẻ chúng ta có thể xem ý nghĩa và ví dụ tại website : <http://www.w3schools.com/>

Phần 2 CSS (Cascading Style Sheets)

- Chúng ta đã biết qua về các thẻ html , và thể hiện nội dung html từ phần 1 .

Tuy nhiên chỉ với các thẻ html thuần thì chúng ta không thể tạo ra được các bố cục trang trí như màu sắc , vị trí ... (như kiểu ta định dạng tài liệu trên word vậy)

Và chúng ta sẽ dùng CSS để làm điều đó .Việc định dạng tài liệu html dùng CSS cũng như ta định dạng tài liệu word thông thường thôi .Có điều với word ta định dạng tài liệu một cách trực quan , còn với CSS ta phải dùng code để định dạng , và dĩ nhiên dùng code tuy khó hơn nhưng lại mang tính mềm dẻo và làm được nhiều thứ hơn .

- Html đánh dấu văn bản , còn CSS sẽ truy vấn các thành phần html để định dạng về màu mè , tọa độ (vị trí) , kiểu dáng ...

- Còn nếu các bạn muốn định dạng các yếu tố html động thì chỉ cần kết hợp với javascript thì có thể tạo nên nhiều ứng dụng rất đẹp và chuyên nghiệp

- Dĩ nhiên kiến thức về html còn nhiều và chúng ta cần đi xa hơn rất nhiều , nhưng chúng tôi muốn nhấn mạnh tới các bạn rằng chỉ đó thôi cũng đủ làm nền tảng để các bạn thiết kế web và đi xa hơn nữa về lập trình web .

- Chúng tôi không muốn giới thiệu đầy đủ kiến thức về 1 nền tảng sẽ khiến các bạn mới lập trình trở nên chóng mặt với đống kiến thức , không biết học xong ứng dụng được vào đâu và như thế nào .Mà phần lớn tập chung vào cơ sở lý thuyết rồi đi vào xây dựng ứng dụng thực tế trên nền tảng kiến thức đã học .

- Mục tiêu của chúng tôi là cung cấp các kiến thức nền tảng để các bạn có thể viết lên được các ứng dụng thực tế .Khi xây dựng các ứng dụng phức tạp , các bạn dựa vào nền tảng đã học để có thể tự học và tiếp thu các công nghệ mới .Trong thế giới lập trình IT thì việc thay đổi công nghệ và nền tảng là việc thường xuyên .Vì thế khả năng tự học đóng vai trò quan trọng hơn là đi chi tiết vào 1 nội dung.Chúng ta cần phải thường xuyên cập nhật công nghệ để giải quyết các vấn đề mắc phải trong phát triển ứng dụng .

6. Khai báo CSS

- Các thuộc tính có thể được khai báo trong 1 file .css riêng biệt , trong thẻ <style> của html hoặc khai báo trực tiếp trong Attribute Style của tag Html .

- Ví dụ dưới đây định dạng màu nền cho 1 thẻ div :

- Thuộc tính background dùng để định màu nền cho 1 đối tượng .Chúng ta sẽ học về các thuộc tính hay dùng để định dạng 1 website sau .

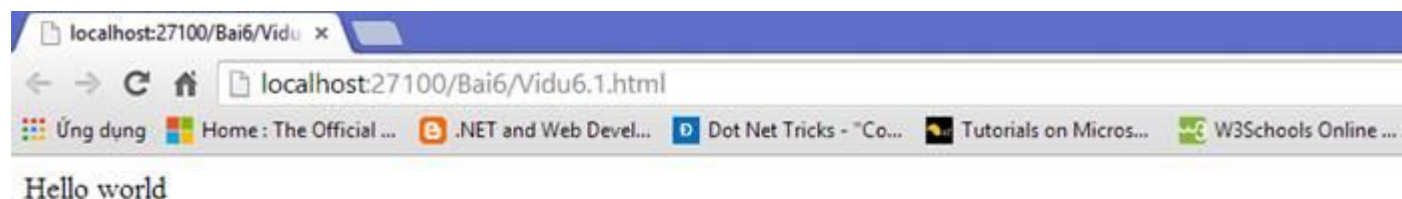
Cách 1 : Khai báo tại Attribute thẻ div cần định dạng

Ví dụ 6.1 : (Xem mã nguồn ViDu6.1.html

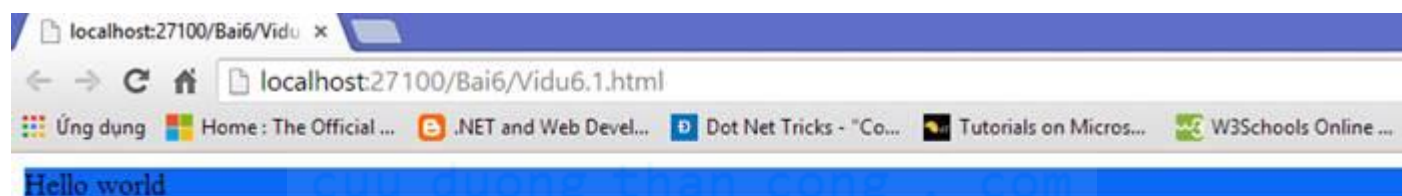
```
<body>
  <div style="background: #0e68f7">
    Hello world
  </div>
</body>
```

Chạy ví dụ 6.1 :

Không CSS :



Có CSS



Cách 2 : Khai báo trong thẻ Style của html

Ví dụ 6.2

- với cách này chúng ta cần phải dùng đến CSS selector (sẽ học phần sau) .Ở đây ta dùng id để đánh dấu thẻ div .

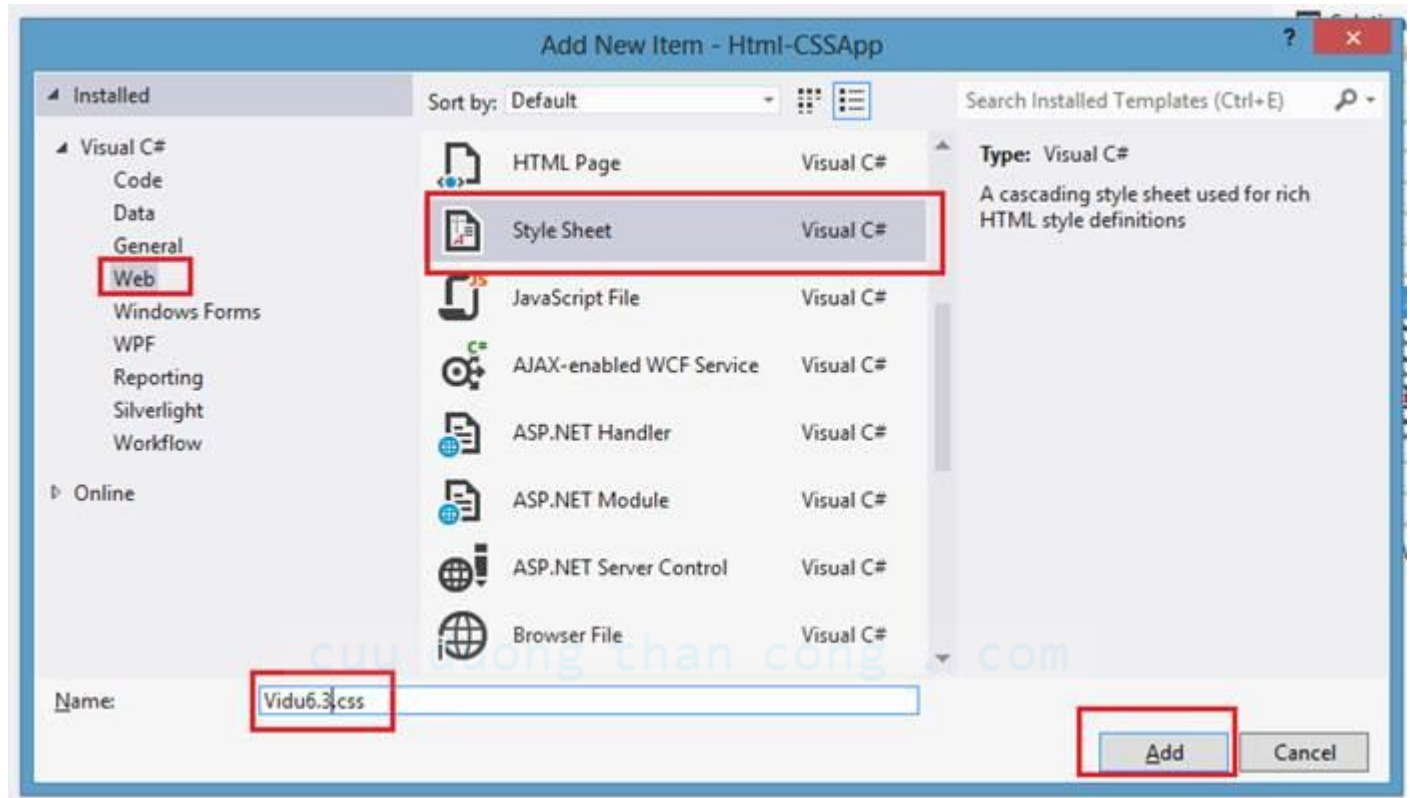
```
<head>
  <title></title>
  <style>
    #div1 {
      background: #0e68f7;
    }
  </style>
</head>
<body>
  <div id="div1">
    Hello world
  </div>
</body>
```

Cách 3 : khai báo thuộc tính CSS trong 1 file .css riêng biệt

Ví dụ 6.3

- tương tự cách 2 , tuy nhiên ta khai báo thuộc tính trong 1 file css riêng biệt .

Bước 1 : Tạo mới 1 file StyleSheet đặt tên là : Vidu6.3.css (ở đây đặt cùng thư mục file html) .Trong VS ta chuột phải , chọn Add / New item chọn template là Style Sheet (theo hình dưới đây)



Bước 2 : Viết mã file CSS như sau :

```
#div1 {  
    background: #0e68f7;  
}
```

Bước 3 : Tham chiếu tới file css trong tài liệu html :

- Trong môi trường VS có thể kéo thả file CSS trực tiếp vào code file html , để VS tự động khai báo tham chiếu

```
<head>  
    <title></title>  
    <link href="Vidu6.3.css" rel="stylesheet" />  
</head>  
<body>  
    <div id="div1">  
        Hello world  
    </div>  
</body>
```

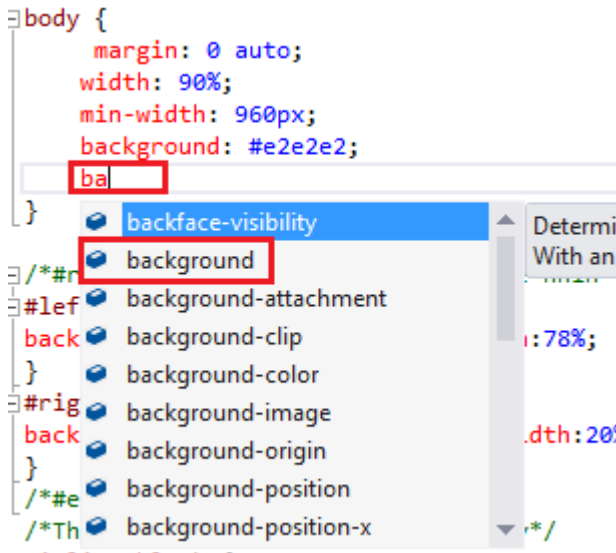
- Ta có 3 cách để code mã CSS .Tùy từng trường hợp ta chọn phương án để tối ưu mã nguồn
- Các ví dụ thực hành dưới đây chủ yếu khai báo theo cách thứ 2 để đơn giản và dễ đọc code ví dụ .

- Các bạn nào đã từng học asp.net theo kỹ thuật kéo thả thì thấy cách sinh code định dạng CSS theo cách 1

// chạy cả 3 file html ở các ví dụ 6.1 , 6.2 , 6.3 ta được các kết quả như nhau

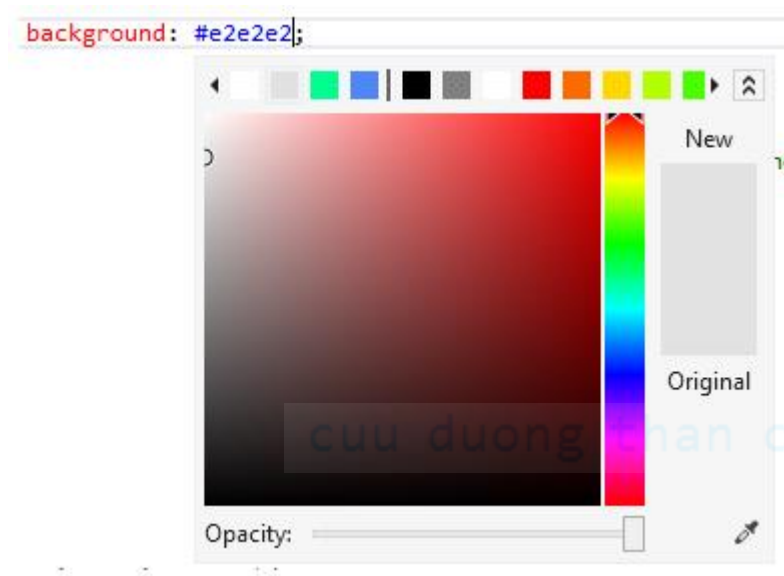
Mẹo nhỏ :

- Trong môi trường VS (đặc biệt là VS 2012 hỗ trợ rất tốt viết mã CSS (kể cả CSS 3)
- giả sử nhập vào từ khóa **ba** VS đã hiện intellisence hỗ trợ chọn thuộc tính như hình sau :



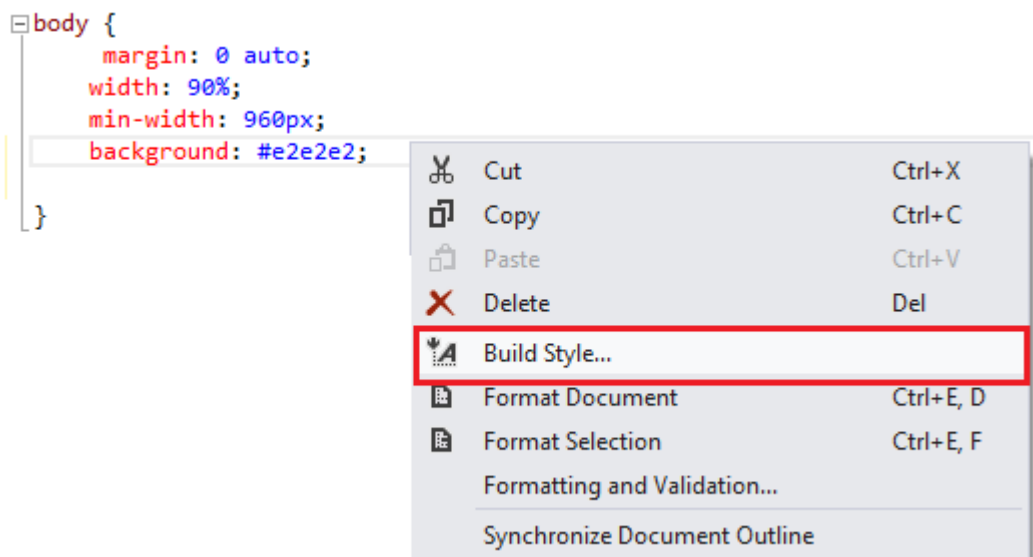
(Trường hợp chưa hiện danh mục chọn các bạn có thể nhấn phím Ctrl + Space)

- Hoặc khi chọn màu , chỉ cần nhấn phím # VS sẽ hiện ra hộp màu để chúng ta lựa chọn như hình dưới

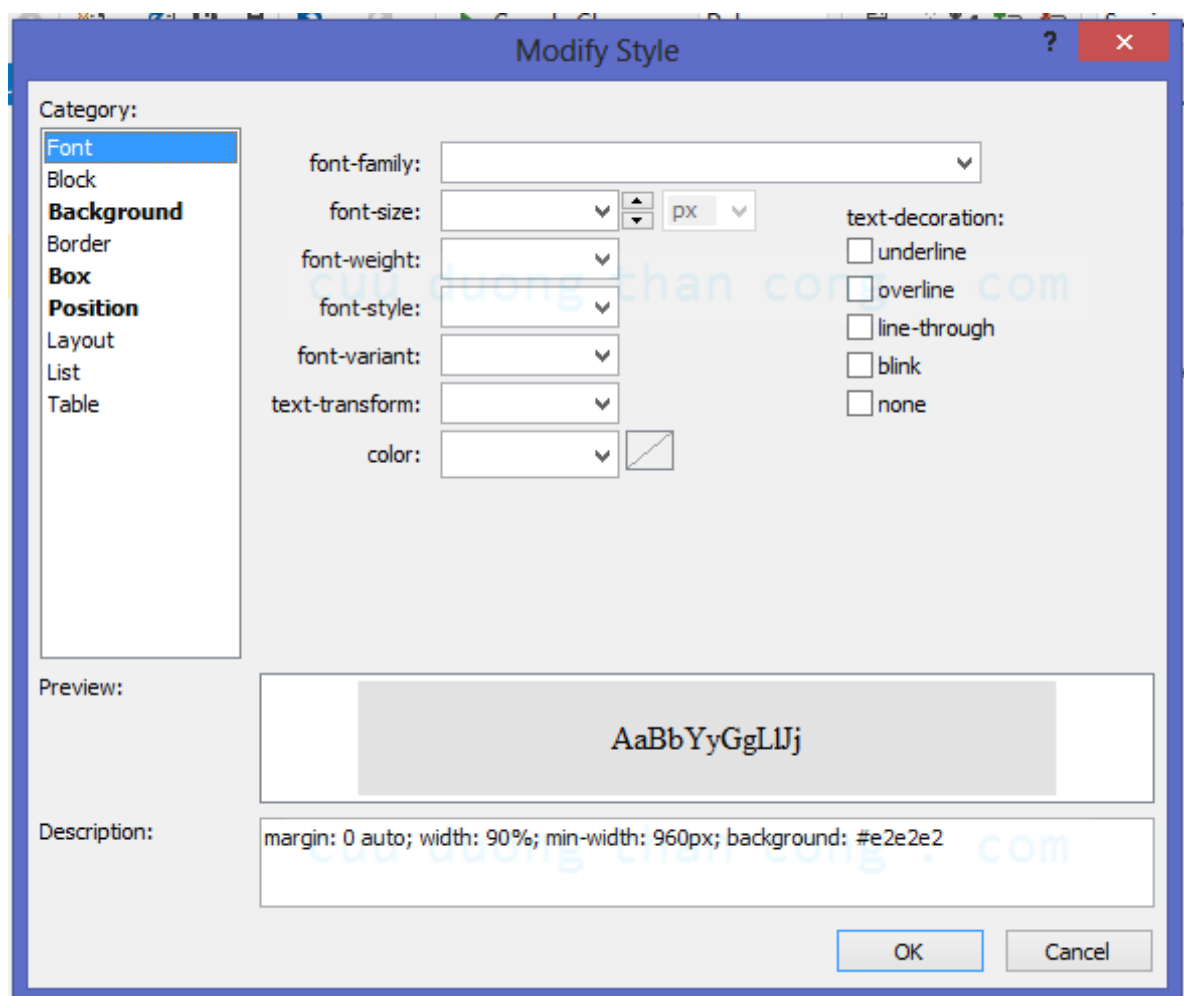


- Ở trên chúng tôi hướng dẫn các bạn code các thuộc tính CSS bằng mã CSS trực tiếp .Môi trường VS cũng hỗ trợ tự tạo các mã CSS bằng giao diện bằng cách :

Chuột phải vào thuộc tính muốn định dạng CSS , chọn build Style như hình dưới :



- ta đến được giao diện định dạng các thuộc tính CSS như sau :



- Tại đây ta có thể tùy chọn các thông số về font chữ , background , Border ...

- Tuy nhiên dù định dạng bằng code hay giao diện thì các bạn cũng cần hiểu bản chất các thuộc tính đối tượng CSS .Vì thế khi mới học chúng tôi khuyên bạn nên xây dựng các thuộc tính bằng code .Trong 1 vài trường hợp các bạn có thể dùng giao diện VS để sinh code và xem rõ 1 thuộc tính nào đó được khai báo và sử dụng như thế nào

7. Bộ chọn (CSS Selector)

7.1 Khái niệm về CSS Selector

- CSS Selector ta tạm hiểu là cách CSS chọn các phần tử HTML để định dạng
- Khi các bạn học được về CSS selector , thì sẽ học được luôn jQuery Selector (jQuery là 1 thư viện của JavaScript) . Vì CSS Selector và jQuery Selector gần giống nhau .
- Tại ví dụ của phần 4 , khi khai báo CSS tại file riêng hoặc trong thẻ Style ta đã sử dụng tới CSS Selector để chọn ra thẻ div cần định dạng

7.2 Khái niệm id và class

- id và class là 2 attribute của tag html .

id :

- id có giá trị duy nhất trong tài liệu html . Ví dụ sau khai báo id Attribute cho 1 thẻ div :

```
<div id="div1">
    Hello world
</div>
```

- khi này không 1 thẻ html nào có attribute giá trị là div1 nữa . Giá sử khai báo sau sẽ báo lỗi .

```
<div id="div1">
    Hello world 1
</div>
<div id="div2">
    Hello world 2
</div>
```

class :

- với attribute class thì nhiều thẻ html có thể có chung 1 class , ví dụ :

```
<div class="class1">
    Hello world 1
</div>
<div class="class1">
    Hello world 2
</div>
```

Sử dụng id và class :

- khi ta muốn định dạng riêng cho một đối tượng thì dùng id
- khi muốn nhiều đối tượng có chung 1 định dạng thì dùng class

Ví dụ :

Ví dụ sau định màu xanh cho các div1 , div2 có class là class1

Và màu đỏ cho div3 có id là div3

Ví dụ 7.2 : (xem mã nguồn Vidu7.1.html)

HTML :

```

<div class="class1">
    Nội dung thẻ div1
</div>
<div class="class1">
    Nội dung thẻ div2
</div>
<div id="div3">
    Nội dung thẻ div 3
</div>

```

CSS :

```

.class1 {
    background:#0094ff;
}
#div3 {
    background:#f00;
}

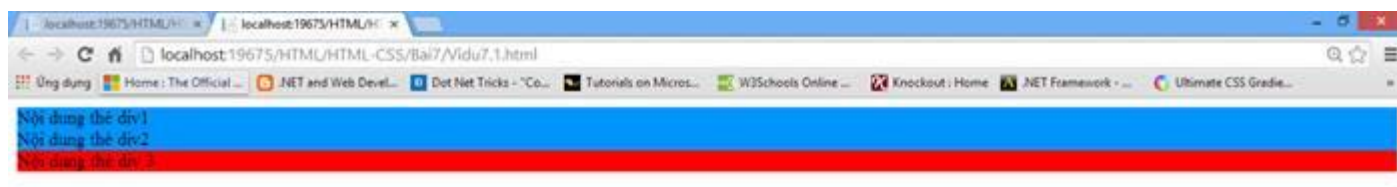
```

-chạy Ví dụ 7.2 :

Không CSS :



Có CSS



- Hai khái niệm id và class rất quan trọng trong CSS , ở trên ta đã dùng CSS Selector để truy vấn các tài liệu html thông qua attribute là id và class .

Với id ta truy vấn theo cấu trúc :

#<Tên id>

Với class ta truy vấn theo cấu trúc :

.<tên class>

Ngoài ra chúng ta còn dùng rất nhiều Selector khác nữa và được nêu ở phần sau :

7.3 Các CSS Selector hay dùng :

(1) Selector thường

div"	Chọn tất cả thẻ div , có thể chọn bất kỳ thẻ html nào
#id1"	Chọn thẻ có id ="id1"
.class1	chọn các thẻ có class =class1
div#VinaDevelop ul li	truy xuất lồng
#id1>a	Chọn phần tử con .Chọn các thẻ a trong div có id là id1
#id1:not(a)	Truy vấn loại trừ phần tử , không chọn các thẻ a
*{ }	Chọn tất cả các phần tử html
html { }	Toàn bộ trang html
body { }	Toàn bộ trang chứa nội dung

(2) pseudo Selector

// thêm vào sau dấu " : "

Selector	Ví dụ	Mô tả
:link	a:link	chọn tất cả các thẻ a
:visited	a:visited	chọn tất cả các thẻ a đã xem qua
:active	a:active	Chọn ra thẻ a đang được kích hoạt
:hover	a:hover	chọn ra thẻ a được di chuột qua

Ngoài ra còn có 1 số selector ít dùng hơn là :

[:focus](#) ; [:first-letter](#) ; [:first-line](#) ; [:first-child](#) ; [:before](#) ...

8. Định dạng nội dung văn bản

- Khi đã lấy được nội dung cần định dạng thông qua CSS Selector , vấn đề còn lại là chúng ta dùng các thuộc tính CSS để định dạng tài liệu html .

- Trong phần này chúng ta sẽ học cách định dạng văn bản như : font , cỡ chữ , màu sắc , màu nền

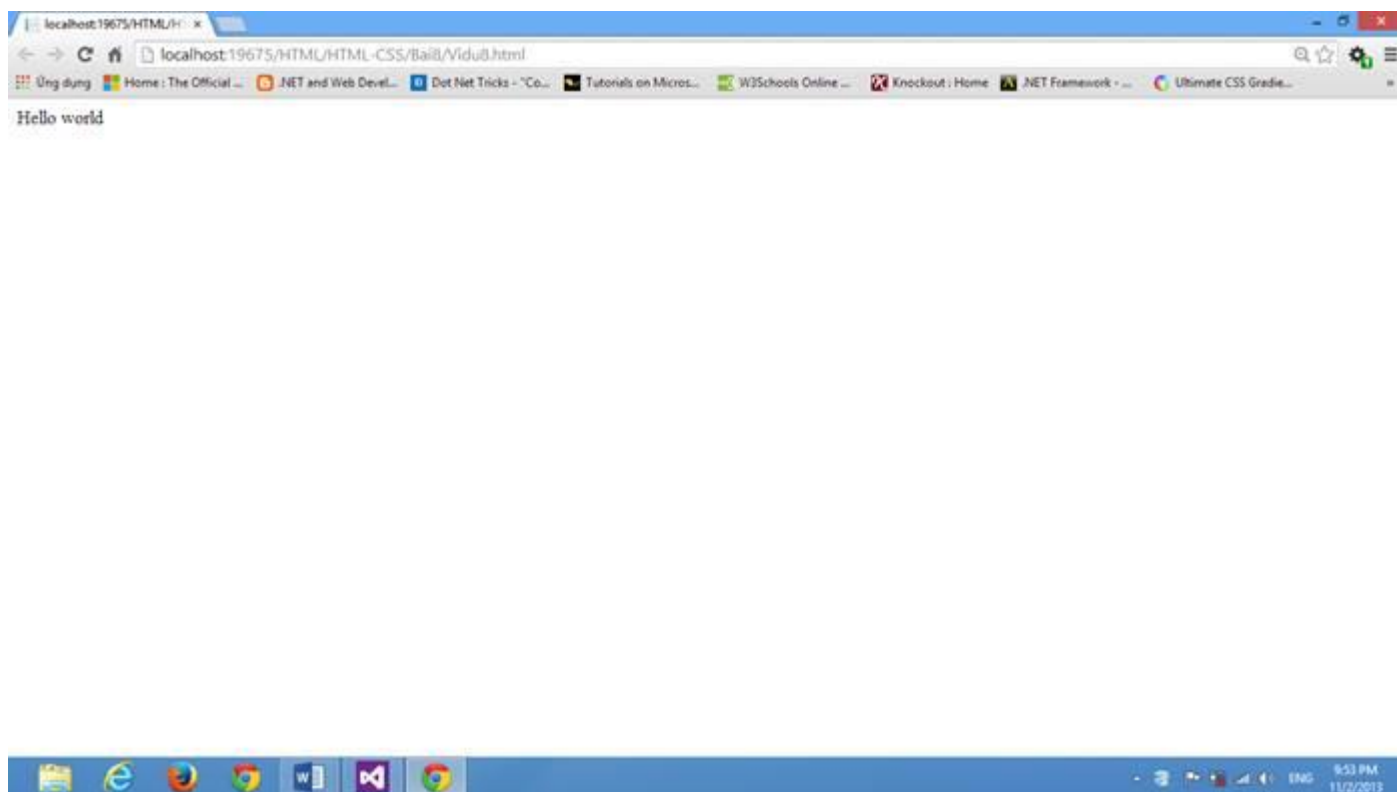
Ở đây chúng ta khai báo một vài thuộc tính cơ bản thường dùng để định dạng text .Các bạn có thể dùng VS để tự tạo mã máy để học các thuộc tính khác

Ví dụ 8

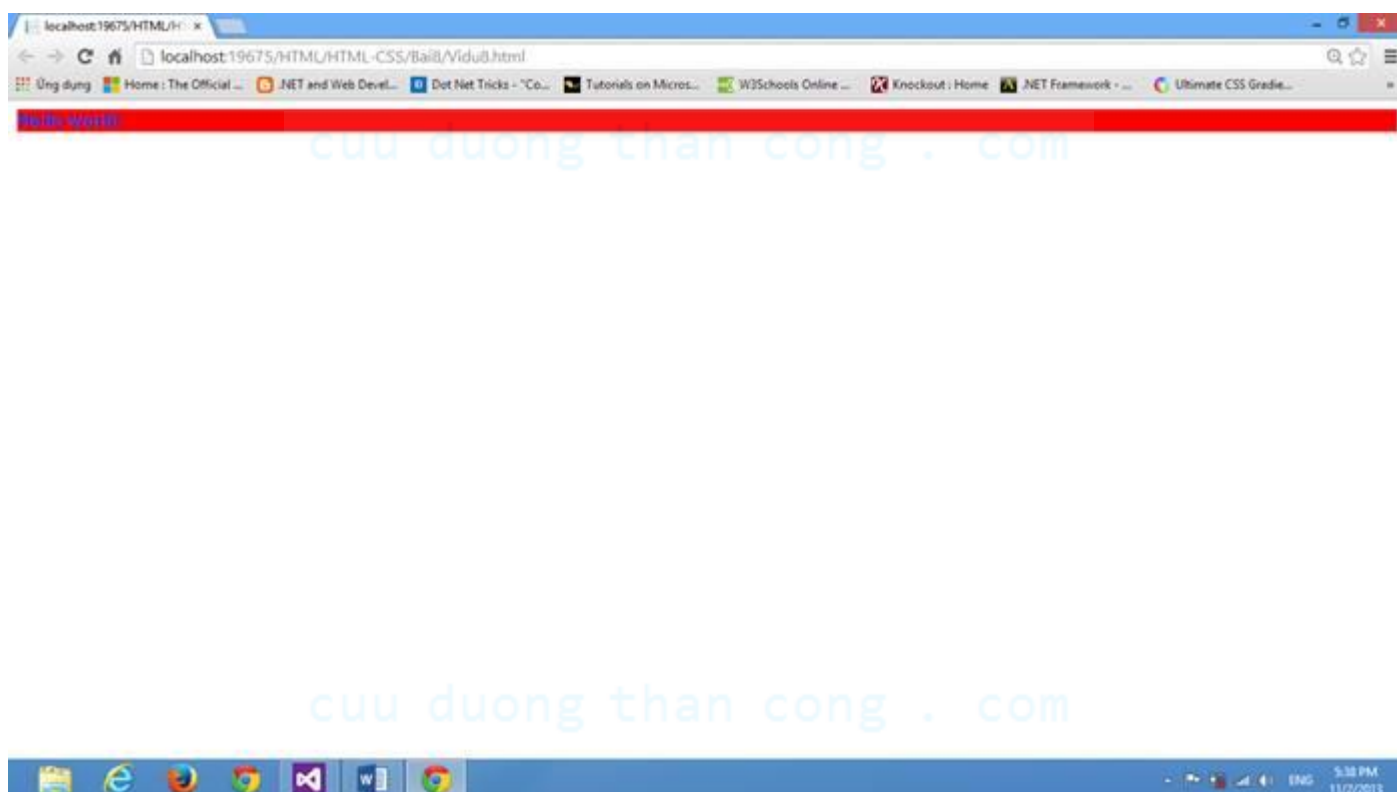
```
#div3 {
    /*màu nền*/
    background: #f00;
    /*Font chữ*/
    font-family:Arial;
    /*Cỡ chữ*/
    font-size:16px;
    /*Kiểu chữ in đậm*/
    font-weight:bold;
    /*màu chữ*/
    color:#044bf6;
    /*gạch chân - không gạch chân*/
    text-decoration:none;
}
```

- chạy ta được kết quả như hình sau :

Không CSS



Có CSS :



9. Định dạng kích thước đối tượng với thuộc tính width , height

- chúng ta có thể định dạng kích thước 1 đối tượng bằng thuộc tính width , height
- Ngoài ra còn có thêm thuộc tính min-width ; max-width , trong trường hợp đối tượng để kích thước có thể co giãn theo độ phân giải màn hình

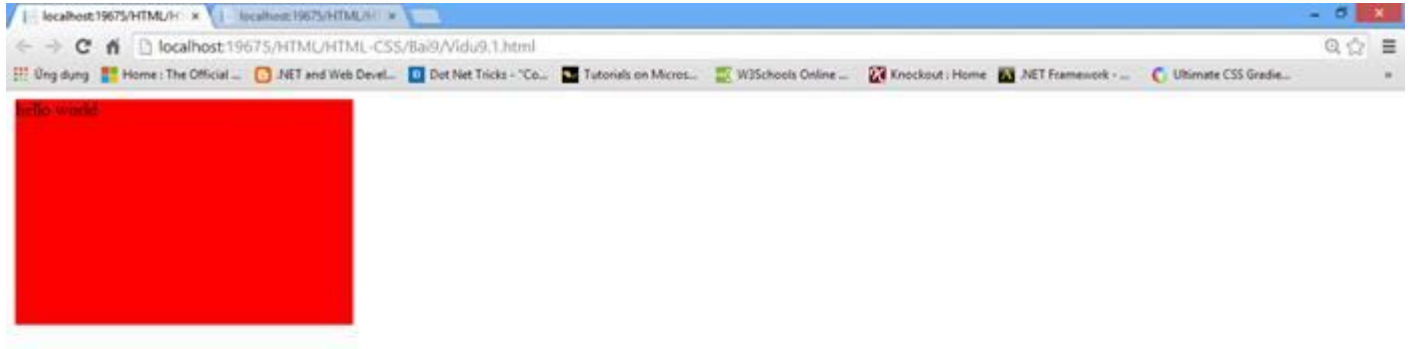
Ví dụ 9.1

```
#div3 {  
    /*thuộc tính màu nền để thấy rõ kích thước đối tượng*/  
    background: #f00;  
    width:300px;  
    height:200px;  
}
```

- Khai báo các thuộc tính min-width , min-height , max-width , max-height tương tự

```
min-width:300px;  
min-height:200px;
```

- Chạy ứng dụng 9.1 được kết quả như hình sau :



- Đi theo thuộc tính width , height là thuộc tính overflow .Đây là thuộc tính qui định nội dung sẽ hiển thị như thế nào nếu kích thước đối tượng không đủ chứa nội dung

- Ví dụ sau sẽ tạo 1 thanh cuộn để người dùng có thể kéo xem nội dung , khi nội dung tràn kích cỡ đối tượng :

Ví dụ 9.2

html :

```
<div id="div3">  
    Nội dung thẻ div 3  
    Nội dung thẻ div 3  
    Nội dung thẻ div 3  
    Nội dung thẻ div 3  
    Nội dung thẻ div 3  
</div>
```

CSS :

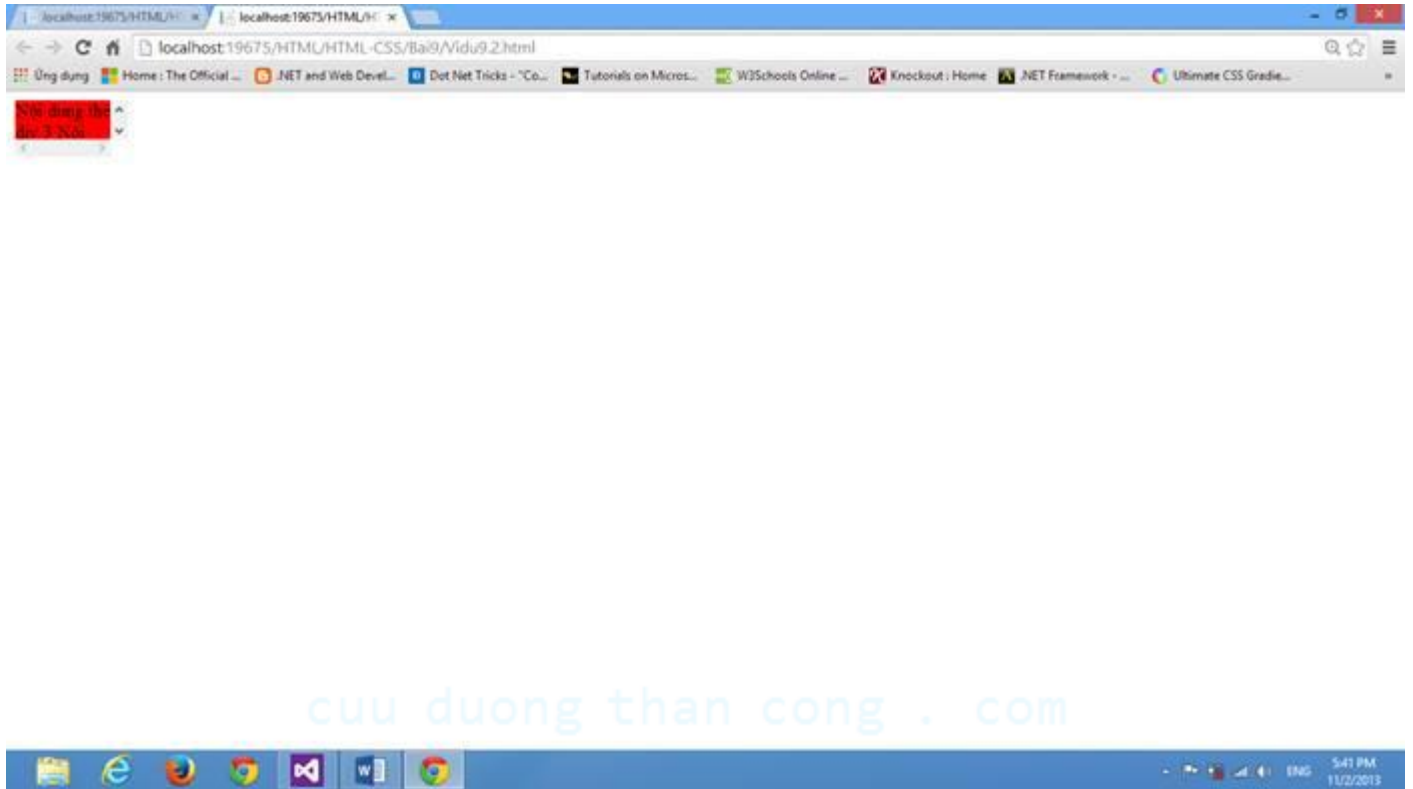
```
#div3 {  
    /*thuộc tính màu nền để thấy rõ kích thước đối tượng*/
```

```

background: #f00;
width: 100px;
height: 50px;
overflow: scroll;
}

```

Chạy ví dụ 9.2 được kết quả như hình sau :



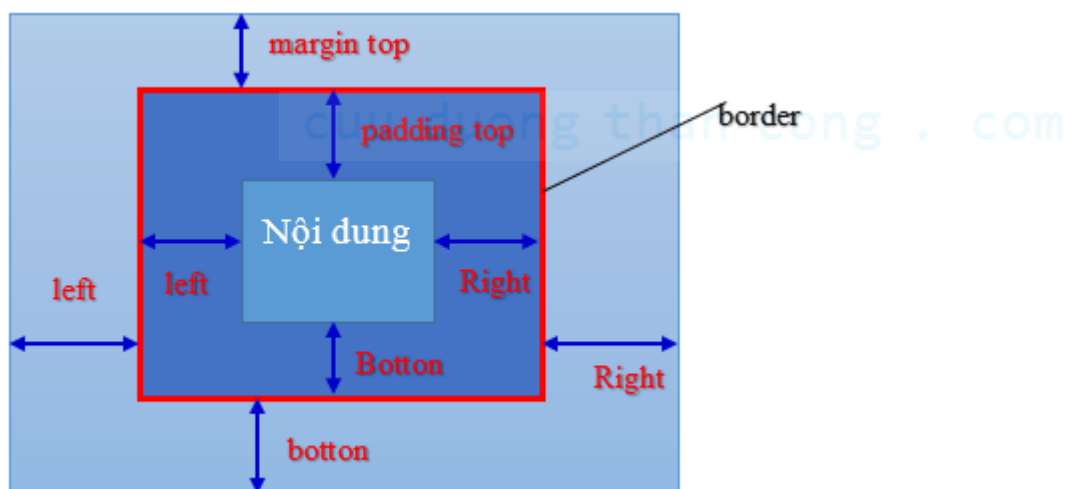
overflow có nhiều giá trị để xử lý đối tượng tràn như : ẩn , để tràn , tạo thanh cuộn như sau :

Thuộc tính	giá trị	Ví dụ	Mô tả
<u>overflow</u>	<u>visible</u>	overflow: visible;	Khi chiều cao của box không đủ chứa text, thì text vẫn hiển thị tràn qua box, đây là mặc định.
	<u>hidden</u>	overflow: hidden;	Khi chiều cao của box không đủ chứa text, thì text bị tràn sẽ được dấu đi.
	<u>scroll</u>	overflow: scroll;	_ Khi chiều cao của box không đủ chứa text, thì text bị tràn sẽ được dấu đi và xuất hiện thanh scroll, khi cuộn sẽ hiển thị text. _ Khi sử dụng thành phần này sẽ xuất hiện cả thanh scroll ngang và dọc.
	<u>auto</u>	overflow: auto;	_ Khi chiều cao của box không đủ chứa text, thì thanh scroll sẽ tự động hiển thị.

			_ Khi sử dụng thành phần này sẽ xuất hiện thanh scroll dọc.
	inherit	overflow: inherit;	Xác định thừa hưởng thuộc tính từ thành phần cha (thành phần bao ngoài).

10. Mô hình hộp Box – Model

- Mô hình Box Model mô tả các thuộc tính border , margin , padding là các thuộc tính rất quan trọng trong việc bố trí giao diện website
- Mô hình được mô tả bằng hình vẽ sau :



- Mỗi thuộc tính padding , margin đều có 4 thuộc tính con là :top,right,bottom,left .
- Border là đường viền bao quanh đối tượng

- padding là khoảng trống tính từ phần nội dung tới viền bao quang đối tượng
- margin là khoảng trắng của đối tượng so với các đối tượng bên cạnh nó

Ví dụ 10 :

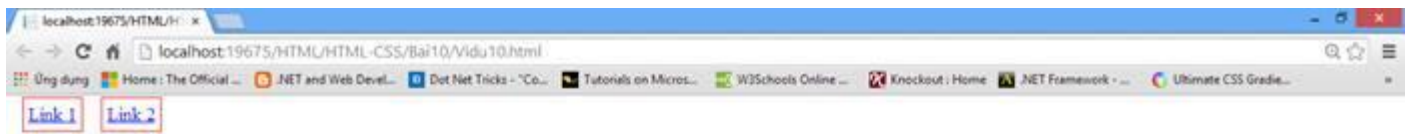
HTML :

```
<a href="#" class="class1">Link 1</a>
<a href="#" class="class1">Link 2</a>
```

CSS :

```
.class1 {
    padding:5px;
    margin:6px;
    border:solid 1px #f00;
}
```

Chạy ví dụ 10 được kết quả như hình sau :



cuu duong than cong . com



Chú ý khai báo :

- thuộc tính border ta để kiểu đường liền , màu đỏ , độ lớn viền là 1px .
- Khi khai báo margin và padding trong trường hợp trên thì tất cả các giá trị top,right,botton,left đều bằng nhau .Có thể khai báo từng thuộc tính riêng như là : `margin-top:1px;`
- Và có 1 quy tắc là : 1 đối tượng nào không được khai báo thì sẽ lấy giá trị đối xứng của nó .Ví dụ nếu giá trị left không được khai báo thì sẽ lấy giá trị right .Nếu giá trị đối xứng không được khai báo thì lấy giá trị là 0px .Ví dụ trên chỉ khai báo `margin-top:1px;` thì giá trị botton sẽ là 1px các giá trị left, right sẽ là 0px .
- ta có thể khai báo tắt theo quy tắc như sau : tính theo chiều kim đồng hồ , từ vị trí top
top – right – botton – left

Ví dụ : `margin:6px 3px 5px 7px ;`

cuu duong than cong . com

cuu duong than cong . com

11. Float – clear trong css

11.1 Float

Vi dụ 11.1

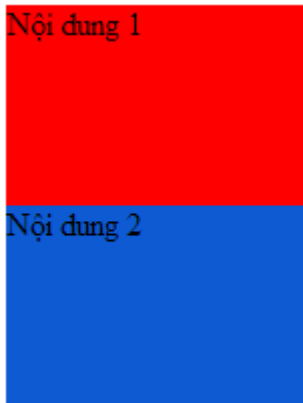
- float là thuộc tính ép đối tượng sau trôi theo đối tượng trước khi không gian đối tượng trước vẫn còn .

- giả sử ta có 2 thẻ div mã html như sau :

```
<div id="div1">Nội dung 1</div>  
<div id="div2">Nội dung 2</div>
```

- Ta có 1 chút mã CSS đơn giản để dễ dàng quan sát kích thước mỗi thẻ div như sau :

```
#div1 {  
    background: #f00;  
    width: 150px;  
    height: 100px;  
}  
#div2 {  
    background: #0d5ad3;  
    width: 150px;  
    height: 100px;  
}
```



Kết quả :

- Một vấn đề được đặt ra là vùng không gian bên phải div1 trống rất nhiều , ta muốn div2 sẽ hiển thị lên trên cùng hàng với div1 .Thuộc tính float sẽ cho ta làm điều này :

- Thêm thuộc tính float là left code như sau :

```
#div1 {  
    background: #f00;  
    width: 150px;  
    height: 100px;  
    float:left;  
}  
#div2 {  
    background: #0d5ad3;  
    width: 150px;  
    height: 100px;  
    float:left;  
}
```

cuu duong than cong . com

cuu duong than cong . com

Kết quả :

- Nếu ta để div2 float bên phải :

```
#div2 {  
    background: #0d5ad3;  
    width: 150px;  
    height: 100px;  
    float:right;  
}
```

Ta được kết quả như sau :

cuu duong than cong . com

11.2 Clear

- Thuộc tính clear thường đi theo phần tử mà đã được gán float
- Clear ngăn 1 đối tượng với float với đối tượng trước nó đặt float, và xác định có cho nó float tiếp theo đối tượng
- Chúng có các giá trị như: both (không tràn nữa), left, right

Trong ví dụ trên giả sử ta có thêm 1 div3 với yêu cầu là : div3 float sang bên phải nhưng lại khác hàng với div1 , div2 .Thì ta cần dùng thuộc tính clear để ngăn ảnh hưởng float từ div1 và div2 như sau :

Ví dụ 11.2

cuu duong than cong . com

HTML :

```
<div id="div1">Nội dung 1</div>  
<div id="div2">Nội dung 2</div>  
<div id="div3">Nội dung 3</div>
```

Mã CSS khi chưa dùng thuộc tính clear :

```
#div1 {  
    background: #f00;  
    width: 150px;  
    height: 100px;  
    float: left;  
}  
  
#div2 {
```

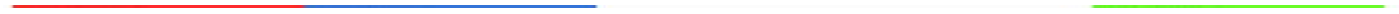
```

background: #0d5ad3;
width: 150px;
height: 100px;
float: left;
}

#div3 {
background: #4cff00;
width: 150px;
height: 100px;
float: right;
}

```

Kết quả :



Mã CSS dùng thuộc tính clear :

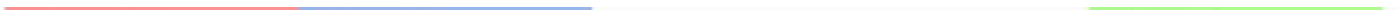
Thêm thuộc tính clear cho div3

```

#div3 {
background: #4cff00;
width: 150px;
height: 100px;
float: right;
clear: both;
}

```

Kết quả :



Chú ý : Trong nhiều trường hợp sử dụng float , muốn không cho nội dung động bị trôi theo float , ta thường tạo 1 thẻ tring gian (không chứa nội dung) có thuộc tính clear là both để ngắt ảnh hưởng float của đối tượng cha .

11.3 Chú ý thuộc khi sử dụng float và boder

- Float 1 đối tượng thì đối tượng đó coi không gian phía float là vô tận (hết miền thì xuống dòng) .
- Nên không làm đường boder mà không có đối tượng bạn đầu nhận float và cuối cùng

- Giả sử có div cha A chứa 4 div con là : A1,A2,A3,A4 .
- Cả 4 đối tượng A1,A2,A3,A4 đều để float là left, thì ta không thể tạo được đường border cho div A.Để tạo đường border cần thêm 1 thẻ bất kỳ,1 nội dung bất kỳ mà không nhận float
- Nếu thêm thẻ thì phải gán nội dung cho thẻ

12 . Display

- Thuộc tính display dùng để qui định cách hiển thị 1 thẻ html như là : ẩn , hiện kiểu inline , block

Ví dụ 12.1 :

Html :

```
<div id="div1">Nội dung 1</div>
```

CSS :

```
#div1 {
    display:none;
    /*Có thể thay bằng thuộc tính*/
    /*visibility:hidden;*/
}
```

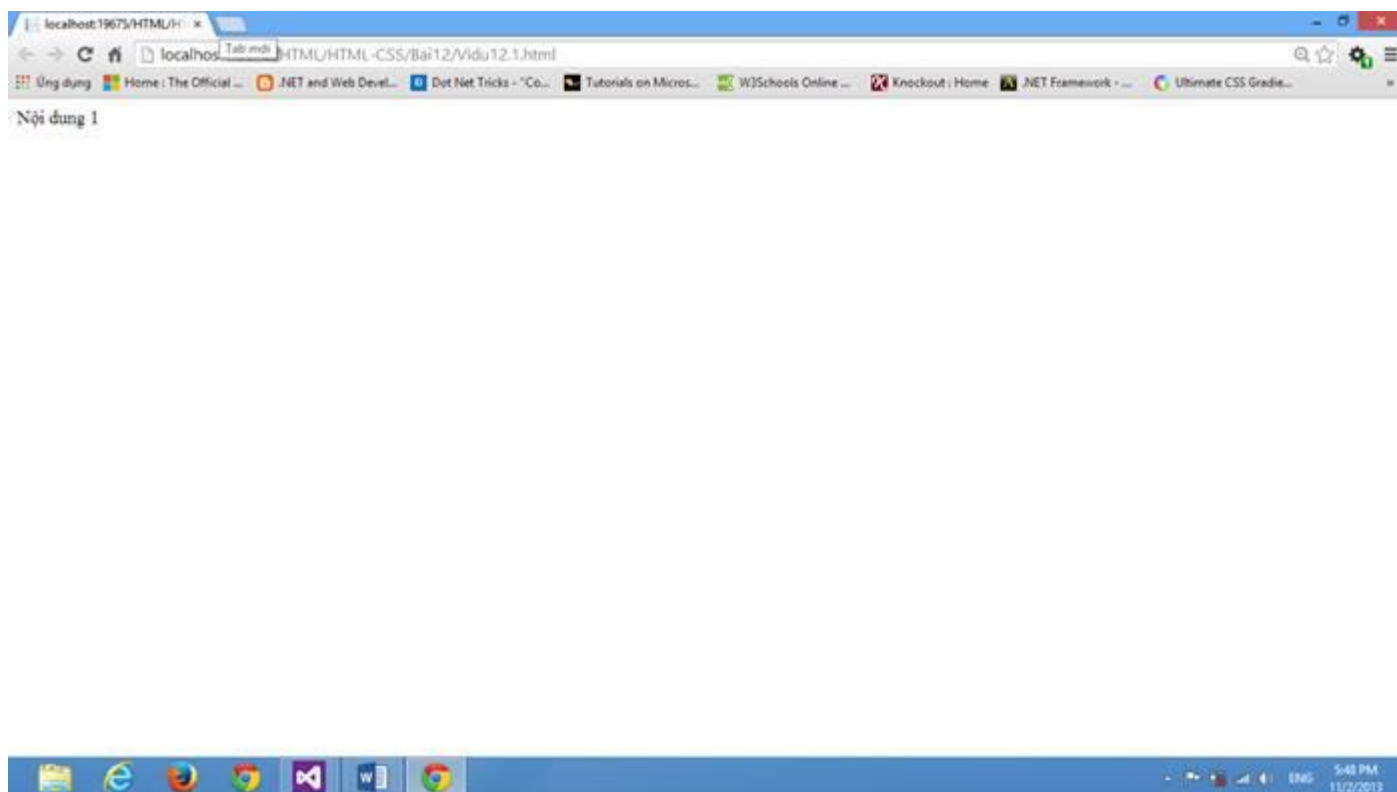
- khi này div1 sẽ bị ẩn đi

Chạy Ví dụ 12 .1 :

// Không có CSS :



// Có CSS :



Ví dụ 12.2 :

- thẻ a là 1 thẻ dạng inline , ta có thể dùng thuộc tính display để thẻ a có thể trở thành block như sau :

HTML :

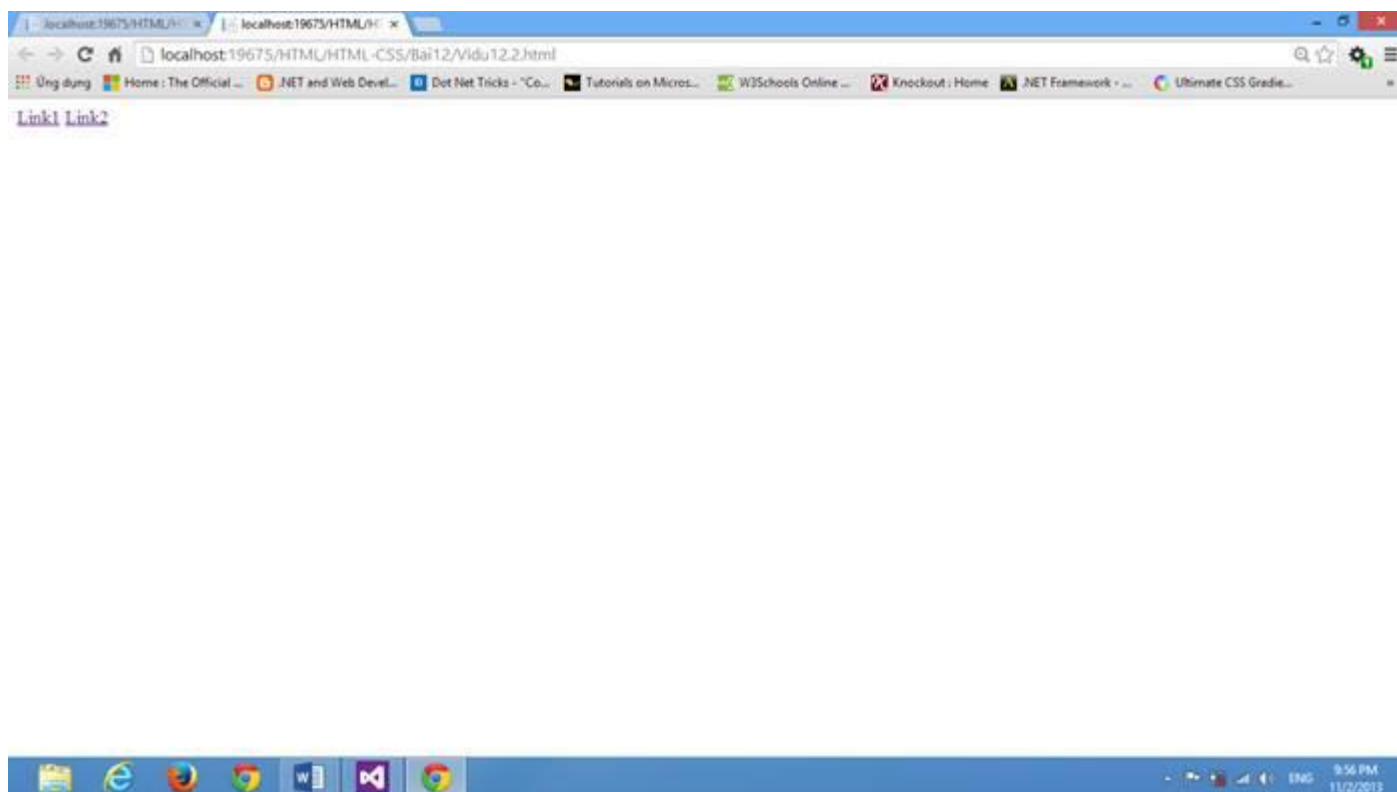
```
<a href="#" class="class1">Link1</a>  
<a href="#" class="class1">Link2</a>
```

CSS :

```
.class1 {  
    display: block;  
}
```

Kết quả :

Không CSS



Có CSS



- Hoặc với thẻ div là kiểu block , ta có thể biến div thành kiểu Inline như sau :

HTML :

```
<div class="class1">Nội dung 1</div>  
<div class="class1">Nội dung 2</div>
```

CSS :

```
.class1 {  
    display: inline;  
}
```

Trường hợp muốn dùng các thuộc tính width . height thì cần đặt giá trị là inline-block như sau:

```
.class1 {  
    width:50px; height:100px;  
    display: inline-block;  
}
```

Dùng thuộc tính display hoặc float ta có thể thiết kế layout website bằng thẻ div thay thế layout bằng html .Các bạn sẽ được hướng dẫn cụ thể trong bài thực hành phần cuối về thiết kế layout.

13. Position

- Position cho định dạng cụ thể 1 đối tượng dựa vào tọa độ của nó
- Position được định dạng ,thì các thuộc tính margin và padding mất tác dụng

Định vị tuyệt đối Absolute

position:absolute; top:50px; left:70px ;

- Ở trên xác định chính xác đối tượng cách lề trên 50 px, lề trái 70px
- Định dạng này sẽ gặp khó khăn khi chưa biết chiều cao,độ rộng đối tượng (do lập trình web chưa trả về nội dung)

Định vị tương đối Relative

- Định vị tương đối là so sánh vị trí đối tượng với đối tượng cha .Khi đối tượng không có đối tượng cha thì vị trí được định vị như absolute

```
position: relative;  
top: 30px;  
left: 30px;
```

Nhận xét : Relative dùng khi định vị cho đối tượng con theo đối tượng cha .

Ví dụ :

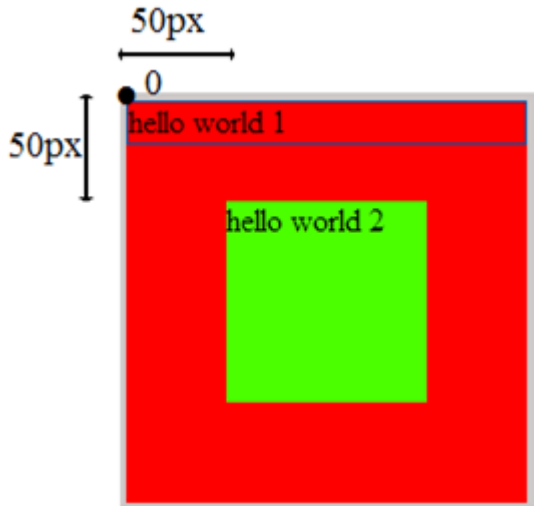
// giả sử có 2 div ,div1 là cha div2

```
<div id="div1">  
    hello world 1  
    <div id="div2">  
        hello world 2  
    </div>  
</div>
```

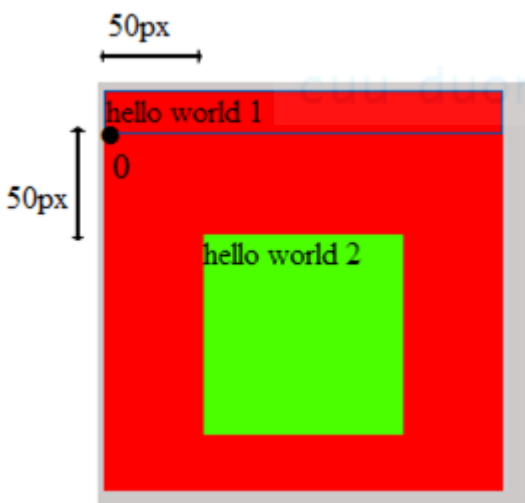
TH1 : div1 để absolute :

- div2 đặt absolute được kết quả :

div2 canh top,left với gốc (0,0) là vị trí div1



- div2 đặt relative :



TH2: div1 để relative

- div2 để absolute : cho kết quả tương tự div1 : absolute,div2 : absolute

- div2 để relative : cho kết quả tương tự div1 : absolute,div2 : relative

(3) z-index :

- khi đặt thuộc tính position là absolute thì đối tượng có thêm thuộc tính z-index dùng để xác định đối tượng nào hiển thị trước .

14. Định dạng 1 vài thẻ html đặc biệt

14.1 Thẻ a

Với thẻ a ta dùng các selector để định dạng như sau :

Thẻ a cung cấp các định dạng sau :

- a:link khi link ở trạng thái thường
- a:hover khi rê chuột qua
- a:visited link đã từng được click (được thăm)
- a:active link đang được kích hoạt

Ví dụ :

HTML :

```
<a href="#">Link 1</a>
<a href="#">Link 2</a>
<a href="#">Link 3</a>
<a href="#">Link 4</a>
<a href="#">Link 5</a>
```

CSS :

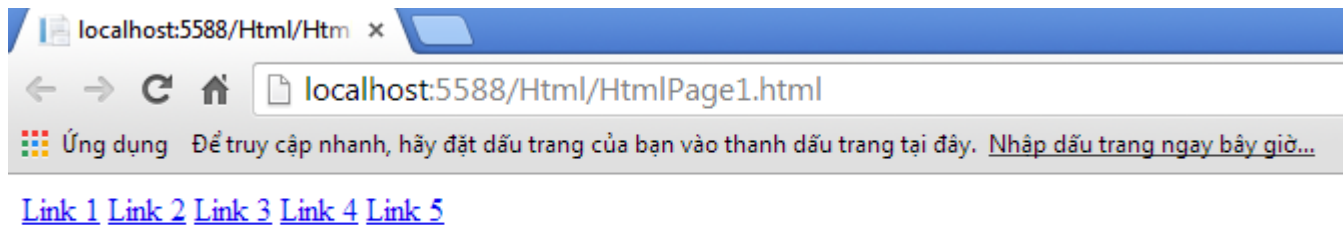
```
a:link {
    background:#0094ff;padding:5px;
    color:#fff;
}
a:hover {
    color:#808080;
    font-weight:bold;
}
```

Code hoàn chỉnh :

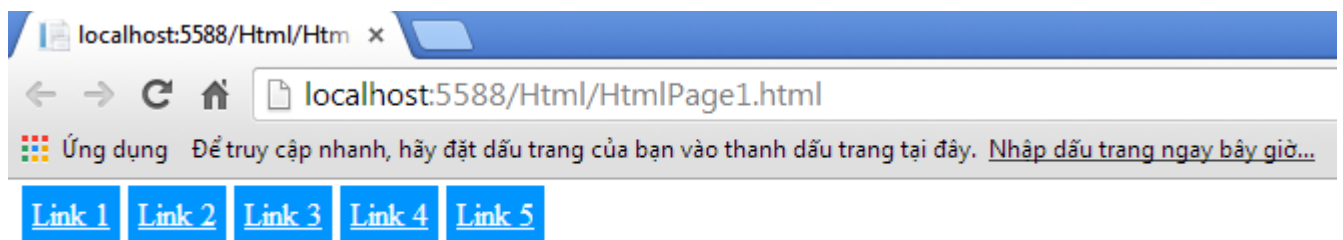
```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title></title>
    <style>
        a:link {
            background:#0094ff;padding:5px;
            color:#fff;
        }
        a:hover {
            color:#808080;
            font-weight:bold;
        }
    </style>
</head>
<body>
    <a href="#">Link 1</a>
    <a href="#">Link 2</a>
    <a href="#">Link 3</a>
    <a href="#">Link 4</a>
    <a href="#">Link 5</a>
</body>
</html>
```

Chạy ví dụ ta được kết quả như hình dưới :

Không CSS :



Có CSS :

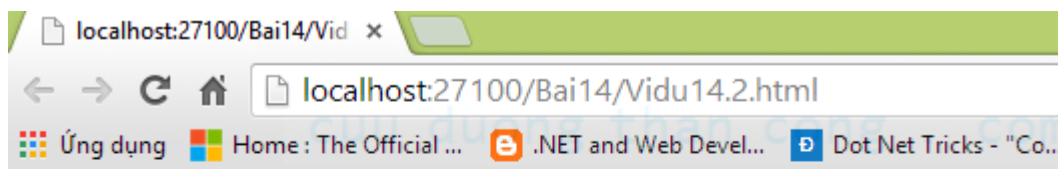


14.2 Thẻ ul

- thẻ ul /li khi dùng bị ảnh hưởng bởi margin và padding của đối tượng phía trước và phía sau nó .
- Nên nội dung thẻ li thường không được căn lề theo ý muốn .
- Giả sử ta có đoạn html như sau :

```
<div>
  Danh Mục 1
  <ul>
    <li><a href="#">Danh Sách 1</a></li>
    <li><a href="#">Danh Sách 1</a></li>
    <li><a href="#">Danh Sách 1</a></li>
  </ul>
</div>
```

Kết quả chạy trên trình duyệt đoạn code trên :



Danh Mục 1

- [Danh Sách 1](#)
- [Danh Sách 1](#)
- [Danh Sách 1](#)

Trong định dạng menu , danh sách ta muốn tất cả lề đều canh trái .Do dùng thẻ ul lên có 1 khoảng trắng giữa nội dung và lề

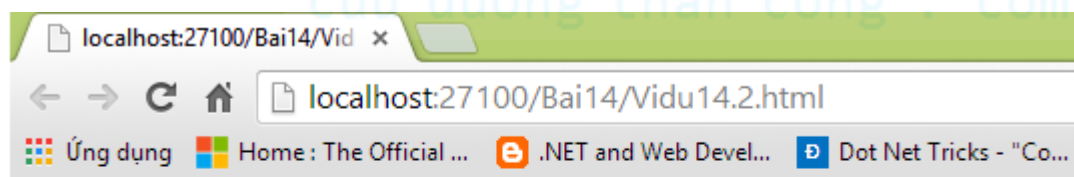
Giả sử muốn căn lề trái thẻ li css như sau :

```
<style>
    ul {
        margin:0px; padding:0px;
    }
</style>
```

Mã nguồn hoàn chỉnh :

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title></title>
    <style>
        ul {
            margin:0px; padding:0px;
        }
    </style>
</head>
<body>
    <div>
        Danh Mục 1
        <ul id="NormalMenu">
            <li><a href="#">Danh Sách 1</a></li>
            <li><a href="#">Danh Sách 1</a></li>
            <li><a href="#">Danh Sách 1</a></li>
        </ul>
    </div>
</body>
</html>
```

Chạy ví dụ được kết quả như hình sau :



Danh Mục 1
[Danh Sách 1](#)
[Danh Sách 1](#)
[Danh Sách 1](#)

Chú ý : ở đây định dạng chung cho các thẻ ul , do đó các thẻ ul khác mà ta không muốn canh nội dung bên trái cũng bị ảnh hưởng , do đó có thể tạo các class , hay id để định dạng riêng khi cần cho các thẻ ul canh lề trái .

15. Giới thiệu về HTML5 , CSS 3

15.0 Giới thiệu Html5 – CSS 3

- HTML 5 và CSS mới ra đời cung cấp nhiều thuộc tính mô tả và xử lý để việc lập trình bớt phụ thuộc vào javascript .
- HTML 5 còn cung cấp nền tảng để lập trình giao diện đồ họa mạnh mẽ .Là một sự thay thế đáng kể về kỹ thuật lập trình web so với kỹ thuật dùng flash hay Silverlight trước đó .

- Trong khuôn khổ tài liệu này chúng ta sẽ tìm hiểu một chút về thiết kế giao diện sử dụng HTML 5 , CSS3 với các tính năng mới trong phần thực hành thiết kế giao diện web
- Các tính năng chuyên sâu về lập trình dữ liệu với html 5 , giao diện đồ họa sẽ được học trong các khóa học sau .

15.1 Các thẻ mới dùng trong thiết kế layout

header	Thẻ hiện nội dung trên cùng
footer	Thẻ hiện nội dung cuối cùng
nav	chứa danh sách link chính website (menu ngang)
aside	Chứa phần cố định,nội dung chính (menu dọc)
article	nội dung độc lập ,có thể chứa nhiều section và ngược lại
section	một nội dung cụ thể

Xem sử dụng các thẻ này để thiết kế layout trong bài thực hành thiết kế layout

15.2 Các thẻ media và đồ họa

<audio> : Định nghĩa nội dung âm thanh

<video> : Định nghĩa cho nội dung là video

<source>: Định nghĩa nhiều nguồn media cho các thẻ <video> và <audio>

<embed>:Định nghĩa cho một ứng dụng được nhúng hoặc các nội dung tương tác (giống như một plugin)

<track> : Định nghĩa tracks cho các thẻ <video> <audio>

<canvas>: Dùng cho việc vẽ, xử lý đồ họa, thông qua mã kịch bản (Javascrpts)

15.3 Một vài thẻ khác

<hgroup>	Nhóm một nhóm các phần tử từ <h1> đến <h6> khi một heading có nhiều cấp độ khác nhau
<progress>	Xác định tiến trình của nhiệm vụ (trạng thái loading)
<datalist>	Xác định một danh sách các lựa chọn được định nghĩa trước
<keygen>	Quy định một trường truy xuất key
<output>	Hiển thị kết quả một phép tính
<bdi>	Cô lập một phần của văn bản có thể được định dạng theo một hướng khác từ các văn bản khác bên ngoài nó
<command>	Định nghĩa một nút lệnh mà người dùng có thể gọi
<details>	Xác định các chi tiết khác mà người dùng có thể xem hoặc ẩn

<dialog>	Hiển thị dialog thông báo
<summary>	Xác định phần đầu có thể nhìn thấy trong thẻ <details>
<figure>	Xác định nội dung trong nội dung như minh họa, biểu đồ, ảnh, mã nguồn
<figcaption>	Xác định caption mô tả cho thẻ <figure>
<mark>	Đánh dấu, làm nổi bật văn bản nào đó
<meter>	Xác định kích thước trong khoảng đã biết
<ruby>	Xác định kiểu chú thích màu đỏ (Kiểu viết ở khu vực Đông Á)
<rt>	Xác định kiểu phát âm, giải thích các ký tự (Kiểu chữ viết Đông Á)
<rp>	Xác định kiểu hiển thị trong các trình duyệt không hỗ trợ bằng các dòng chú thích màu đỏ.
<time>	Xác định thời gian và ngày
<wbr>	Xác định xuống dòng, ngắt dòng

16. Thực hành thiết kế giao diện HTML – CSS

16.1 Giới thiệu

- Với nền tảng đã học về CSS và html ở trên chúng ta đã có thể xây dựng nên bố cục và định dạng website của mình .
- Nếu muốn giao diện thêm hấp dẫn và có tính tương tác với người dùng nhiều hơn chúng ta có thể dùng javascript , tuy tốt về mặt kỹ thuật lập trình , tuy nhiên cũng cần xem xét với ứng dụng web , tốc độ luôn là vấn đề được ưu tiên hàng đầu
- Dưới đây chúng ta sẽ được hướng dẫn về xây dựng layout (bố cục) website bằng thẻ div kết hợp CSS , một sự thay thế cho layout dùng table như thông thường .
- Ý nghĩa xây dựng layout dùng thẻ div thay table như là tăng tốc độ , đơn giản mã nguồn (thẻ table cần khai báo thêm rất nhiều thẻ tr , td)
- Dưới đây chúng ta sẽ đi xây dựng layout dùng thẻ div bằng 2 cách là dùng float và inline-block . Với CSS 3 chúng ta có thêm các khác nữa , sẽ được hướng dẫn cụ thể trong tài liệu về HTML 5 – CSS 3 .
- Ngoài ra chúng tôi cũng chú ý tới kỹ thuật lập trình HTML , CSS để áp dụng vào thực tế xây dựng cả 1 website .

Bố cục layout :

Một layout thường chia làm 3 phần chính đó là :

- Đầu trang (header)

- Thân trang (body)
- Cuối trang (footer)

Trong mỗi phần có thể chia làm các phần nhỏ hơn . Ví dụ header thường chứa logo , tên website , các thẻ đăng nhập . Thân trang (body) thường chia làm phần nội dung chính , nội dung nổi bật , các quảng cáo ...

Layout đơn giản có giao diện như sau

cuu duong than cong . com

cuu duong than cong . com

Giả sử cần thiết kế layout trên ta thiết kế các thẻ div như sau :

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title></title>
</head>
<body>
  <div>Đầu trang</div>
  <div>
    <div>Nội dung chính</div>
    <div>Nội dung nổi bật</div>
  </div>
  <div>Cuối trang</div>
</body>
</html>
```

- Ta sẽ dùng CSS để tạo mẫu giao diện như hình vẽ

16.2 Xây dựng layout dùng thuộc tính float

- tạo file Vidu16.2.css và Vidu16.2.html có cấu trúc html như trên
- chú ý tham chiếu file css vào trong file html .File html có mã như sau :

```
<link href="Vidu16.1.css" rel="stylesheet" />
<div>Đầu trang</div>
<div>
  <div>Nội dung chính</div>
  <div>Nội dung nổi bật</div>
</div>
<div>Cuối trang</div>
```

Định dạng chung :

- Trước tiên ta định dạng bộ khung cho thẻ layout bằng khai báo CSS sau :

```
body {
  margin: 0 auto;
  width: 90%;
  min-width: 960px;
  background: #e2e2e2;
}
```

margin để 0, auto như trên để canh nội dung vào giữa màn hình .Chiều ngang để 90% kích thước màn hình người dùng .Đặt giá trị min là 960 để khi người dùng co cửa sổ nhỏ hơn 960 px thì giao diện không bị vỡ .Các bạn có thể đặt giá trị min khác , hoặc không để min để xem tác dụng của chúng

- Do các thuộc tính float , display rất hay dùng trong thiết kế web nên ta tạo sẵn các lớp này trong file CSS , để khi nào dùng chỉ cần khai báo trong html

```
.float-left {  
float:left;  
}  
.float-right {  
float:right;  
}  
.clear-both {  
clear:both;  
}
```

Ta khai báo mã html như sau :

```
<link href="Vidu16.1.css" rel="stylesheet" />  
<div>Đầu trang</div>  
<div>  
    <div class="float-left">Nội dung chính</div>  
    <div class="float-right">Nội dung nổi bật</div>  
</div>  
<div class="clear-both">Cuối trang</div>
```

- chạy thử và ta đã đạt kết quả như mong muốn .

- Khi lập trình thiết kế web động thì nội dung được tạo động và thuộc tính chiều cao (height) sẽ tự động theo nội dung .Ta chỉ quan tâm tới phần chiều ngang (width) .

- Trong ví dụ này (nội dung ít) nên để dễ nhìn ta đặt thuộc tính height của mỗi đối tượng như sau :

CSS

```
#left-body {  
background:#fff; height:400px; width:80%;  
}  
#right-body {  
background:#00ff90; height:400px; width:20%;  
}
```

Html :

```
<link href="Vidu16.1.css" rel="stylesheet" />  
<div>Đầu trang</div>  
<div>  
    <div class="float-left" id="left-body">Nội dung chính</div>  
    <div class="float-right" id="right-body">Nội dung nổi bật</div>  
</div>  
<div class="clear-both">Cuối trang</div>
```

Mã nguồn hoàn chỉnh :

CSS :

```
/*#region Định dạng khung website */  
body {  
    margin: 0 auto;  
    width: 90%;  
    min-width: 960px;  
    background: #e2e2e2;
```

```

}
/*#endregion */

/*#region Các thuộc tính thêm cho dễ nhìn trong ví dụ ( không cần trong thực tế) */
#left-body {
background:#fff; height:400px; width:80%;
}
#right-body {
background:#00ff90; height:400px; width:20%;
}
/*#endregion */
/*#region Các class hay dùng */
.float-left {
float:left;
}
.float-right {
float:right;
}
.clear-both {
clear:both;
}
/*#endregion */

```

Html :

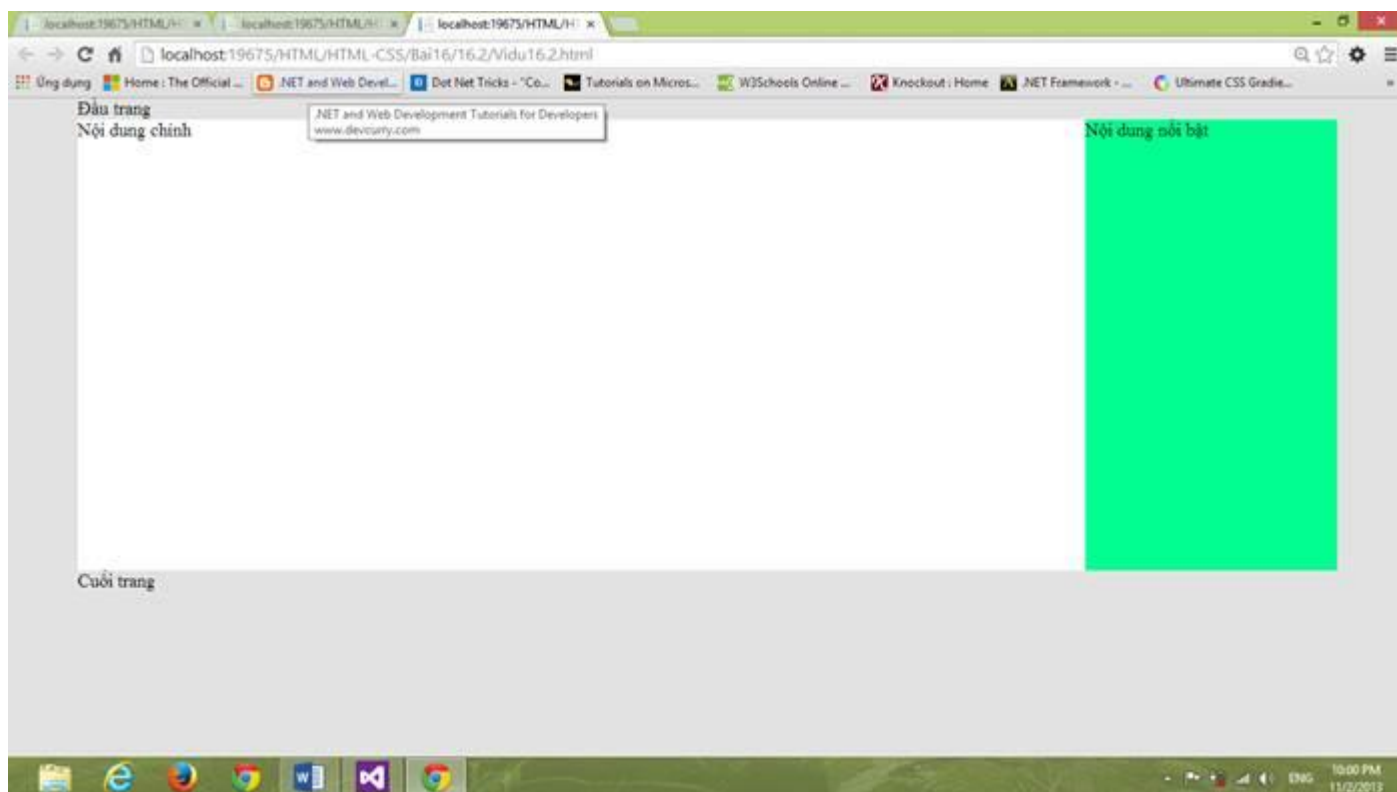
```

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>
</head>
<body>
<link href="Vidu16.1.css" rel="stylesheet" />
<div>Đầu trang</div>
<div>
<div class="float-left" id="left-body">Nội dung chính</div>
<div class="float-right" id="right-body">Nội dung nổi bật</div>
</div>
<div class="clear-both">Cuối trang</div>
</body>
</html>

```

Các bạn có thể xem toàn bộ mã nguồn trong file Vidu16.2.html và Vidu16.2.css

Chạy ví dụ 16.2 được kết quả như hình sau :



16.3 Xây dựng layout dùng inline-block

- dùng float thường cần dùng thêm thuộc tính clear , trong nhiều trường hợp không có thể khai báo clear thì cần khai báo thẻ phụ .
- CSS hỗ trợ thuộc tính hiển thị layout kiểu block (các phần tử sắp ngang trên cùng 1 hàng) .Khi này code html và css gọn gàng hơn .
- Tạo file Vidu16.3.html và Vidu16.3.css có mã tương tự như ví dụ 16.2 code như sau :

CSS :

thay thuộc tính float

```
/*Thay thuộc tính float bằng display*/
.inline-block {
    display: inline-block;
    vertical-align: top;
}
```

- do inline – block có tạo khoảng cách giữa các đối tượng nên cần chỉnh lại width không đủ 100% .Đổi lại như sau :

```
#left-body {
    background:#fff; height:400px; width:78%;
}
```

Html :

```
<link href="Vidu16.2.css" rel="stylesheet" />
<div>Đầu trang</div>
<div>
    <div class="inline-block" id="left-body">Nội dung chính</div>
    <div class="inline-block" id="right-body">Nội dung nổi bật</div>
</div>
```

```
<div>Cuối trang</div>
```

// Ta thấy code đơn giản và dễ sử dụng hơn float rất nhiều

// Một chú ý là thuộc tính display : inline-block không hoạt động tốt trên IE7 , tuy nhiên cũng có cách tạo mã hack để chạy tốt trên đó .Tuy nhiên với các trình duyệt thông dụng hiện nay đều hỗ trợ tốt thuộc tính inline-block

Mã nguồn hoàn chỉnh :

CSS :

```
body {
    margin: 0 auto;
    width: 90%;
    min-width: 960px;
    background: #e2e2e2;
}

/*#region Các thuộc tính thêm cho dễ nhìn trong ví dụ ( không cần trong thực tế) */
#left-body {
    background:#fff; height:400px; width:78%;
}
#right-body {
    background:#00ff90; height:400px; width:20%;
}
/*#endregion */
/*Thay thuộc tính float bằng display*/
.inline-block {
    display: inline-block;
    vertical-align: top;
}
```

cuu duong than cong . com

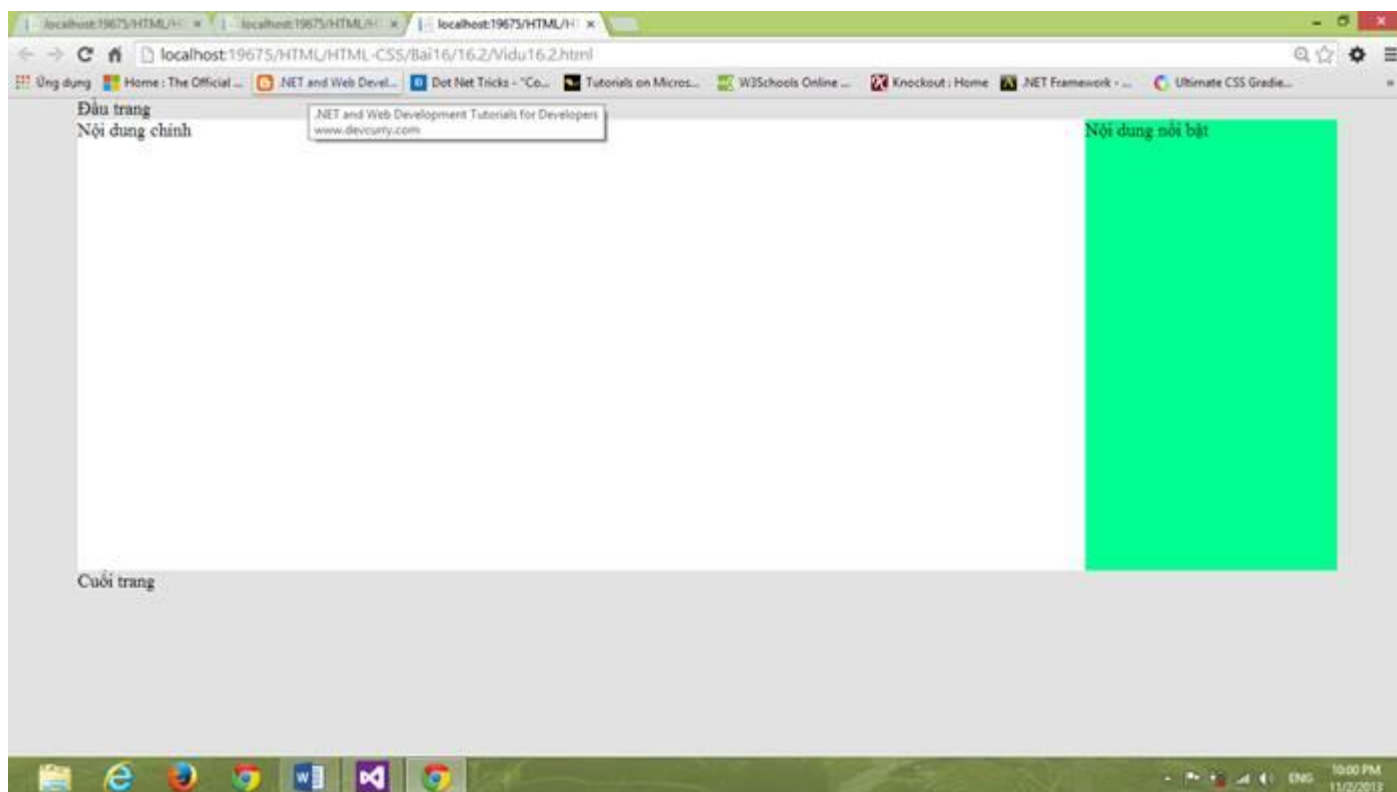
Html :

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title></title>
</head>
<body>
    <link href="Vidu16.2.css" rel="stylesheet" />
    <div>Đầu trang</div>
    <div>
        <div class="inline-block" id="left-body">Nội dung chính</div>
        <div class="inline-block" id="right-body">Nội dung nổi bật</div>
    </div>
    <div>Cuối trang</div>
</body>
</html>
```

cuu duong than cong . com

Các bạn có thể xem toàn bộ mã nguồn trong file Vidu16.3.html và Vidu16.3.css

Chạy ví dụ 16.3 được kết quả như ví dụ 16.2 như hình sau :



16.4 Xây dựng layout dùng HTML 5

- trong việc xây dựng layout ví dụ trên ta dùng các thẻ div .Các thẻ này không nói nên bản chất nội dung chúng chứa .
- Html 5 cung cấp 1 số thẻ hỗ trợ tốt việc thiết kế layout , từ các thẻ này ta phân biệt được đâu là phần đầu trang , phần nội dung , phần cuối trang , phần menu chính ...
- Việc thiết kế các thẻ layout theo chuẩn html 5 sẽ hỗ trợ tốt hơn trong việc SEO website
- Các thẻ dùng trong thiết kế layout như là :

header	Thẻ hiện nội dung trên cùng
footer	Thẻ hiện nội dung cuối cùng
nav	chứa danh sách link chính website (menu ngang)
aside	Chứa phần cố định,nội dung chính (menu dọc)
article	nội dung độc lập ,có thể chứa nhiều section và ngược lại
section	một nội dung cụ thể

// Có thể xem qua tính năng các thẻ trong layout sau :

Trở lại ví dụ 16.2 và 16.3 ta code lại mã html như sau :

HTML :

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title></title>
</head>
<body>
    <link href="Vidu16.4.css" rel="stylesheet" />
    <header>Đầu trang</header>
    <div>
        <article class="inline-block" id="left-body">Nội dung chính</article>
        <aside class="inline-block" id="right-body">Nội dung nổi bật</aside>
    </div>
    <footer>Cuối trang</footer>
</body>
</html>
```

CSS :

- Giữ nguyên file CSS , giống file CSS của Ví dụ 16.3
 - Có thể xem code html và css hoàn chỉnh trong các file Vidu16.4.css và Vidu16.4.html
 - Chúng ta chỉ cần đổi mã html , ở đây chúng ta hãy chú ý kỹ thuật viết code css làm sao để kế thừa và sử dụng được ở nhiều nơi mà không cần code lại hay copy code .Trong thiết kế web động các style các trang thường tương tự nhau , chỉ có trang html là thay đổi .Các bạn hãy ứng dụng kỹ thuật này để code cho mình một framework CSS cho riêng mình
 - Các kỹ thuật này sẽ được thực hành trong các giáo trình về lập trình web Asp.net MVC , PHP
- Chạy ví dụ 16.4 cũng được kết quả như ví dụ 16.3 và 16.2

16.5. Định dạng menu website

- menu chính là thanh điều hướng website , là 1 thành phần rất hay gặp mà chúng ta cần định dạng
- Ở ví dụ này chúng ta sẽ định dạng 1 menu 1 cấp dùng CSS3 và HTML5
- Việc xây dựng và xử lý menu đa cấp được đề cập tới trong tập sau của giáo trình về lập trình dynamic html – css
- Với lập trình web động , menu thường được lấy nội dung từ database , tuy nhiên cách định dạng thì tương tự .Việc làm menu động lấy nội dung dưới CSDL được học trong các giáo trình về thiết kế web Asp.net MVC

Code hoàn chỉnh

HTML :

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title></title>
</head>
<body>
    <link href="Vidu16.5.css" rel="stylesheet" />
    <nav class="mainMenu">
        <a href="#">Trang Chủ</a>
        <a href="#">Giới Thiệu</a>
        <a href="#">Tin Tức</a>
        <a href="#">Liên Hệ</a>
        <a href="#">Sản Phẩm</a>
    </nav>
</body>
</html>
```

CSS :

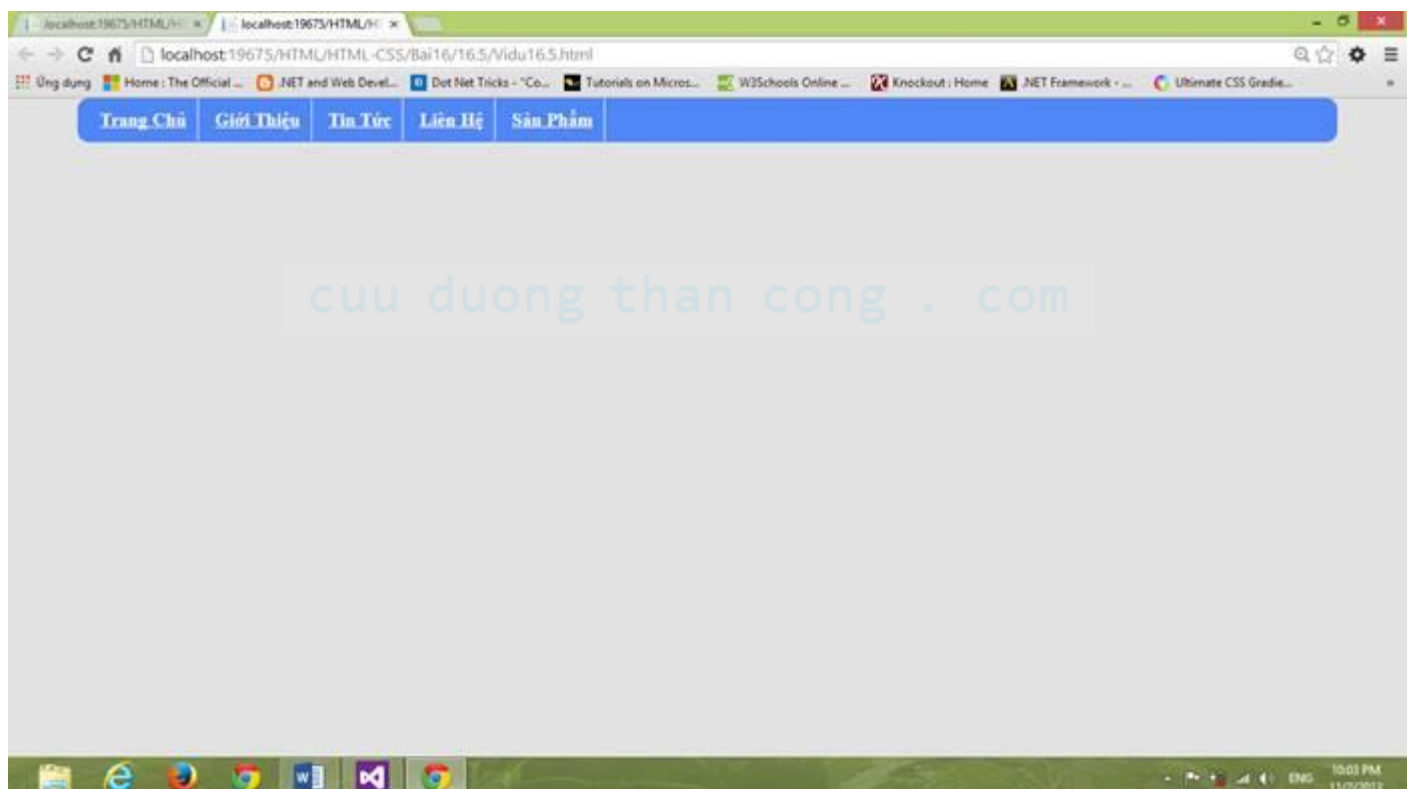
```
.mainMenu {
    background: #4f86f8;
    border-radius: 10px;
    padding: 10px;
}

.mainMenu a {
    font-size: 16px;
    color: #FFFFFF;
    font-weight: bold;
    padding: 10px;
    border-right: 1px solid #fff;
}
```

- Thuộc tính border-radius dùng để bo góc đường viền .Đây là thuộc tính của CSS3

Có thể xem ví dụ hoàn chỉnh trong file Vidu16.5.css và Vidu16.5.html

- kết quả khi chạy ví dụ 16.5



16.6 Kết hợp ví dụ 16.4 - 16.5

- Ta có thể đặt menu vào layout trong phần tiêu đề website
- kết hợp ví dụ 16.4 và 16.5 ta được layout như sau

HTML :

```
<link href="Vidu16.6.css" rel="stylesheet" />
<header>
  <h1>Tiêu đề website</h1>
  <nav class="mainMenu">
    <a href="#">Trang Chủ</a>
    <a href="#">Giới Thiệu</a>
    <a href="#">Tin Tức</a>
    <a href="#">Liên Hệ</a>
    <a href="#">Sản Phẩm</a>
  </nav>
</header>
<div>
  <article class="inline-block" id="left-body">Nội dung chính</article>
  <aside class="inline-block" id="right-body">Nội dung nổi bật</aside>
</div>
<footer>Cuối trang</footer>
```

CSS :

```
body {
  margin: 0 auto;
  width: 90%;
  min-width: 960px;
  background: #e2e2e2;
}

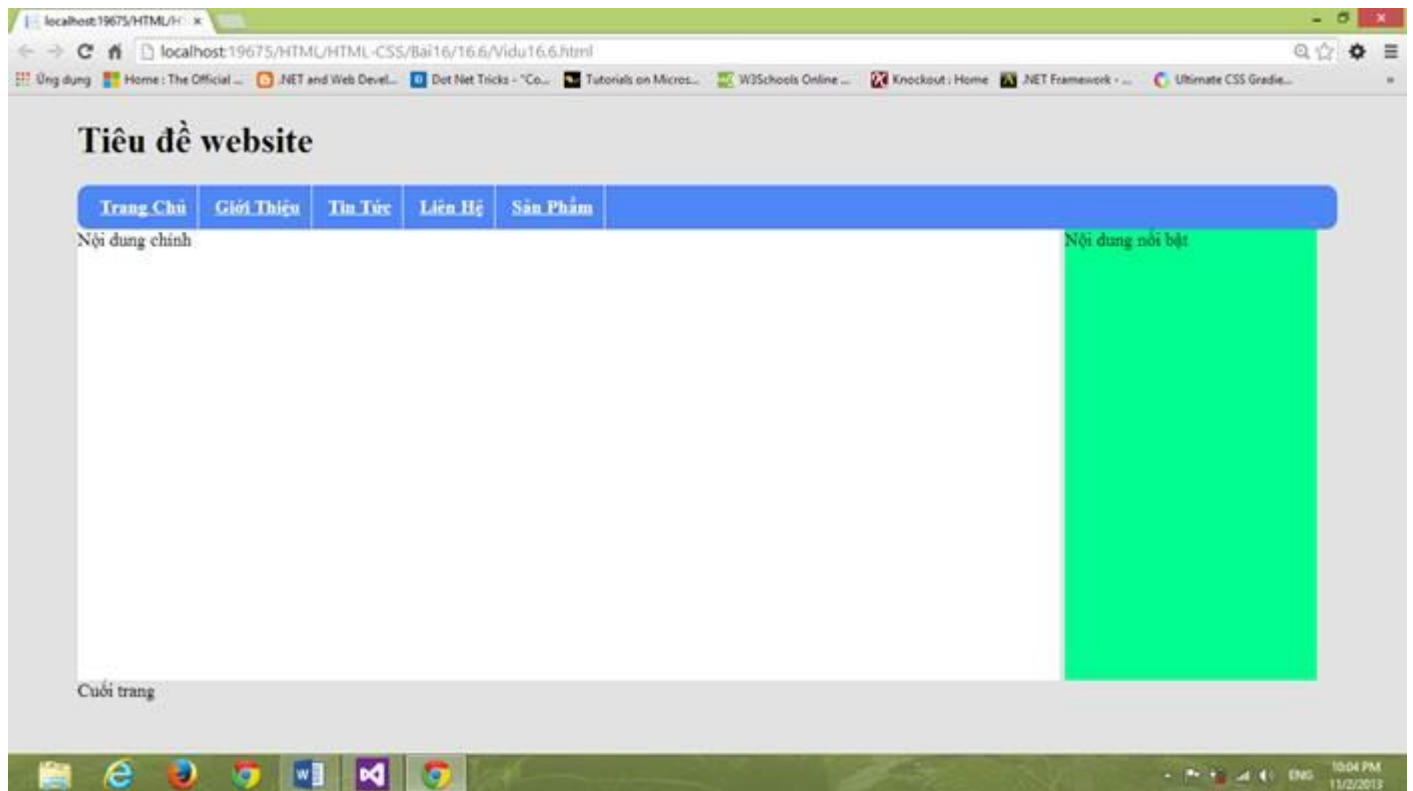
/*#region Các thuộc tính thêm cho dễ nhìn trong ví dụ ( không cần trong thực tế) */
#left-body {
  background:#fff; height:400px; width:78%;
}
#right-body {
  background:#00ff90; height:400px; width:20%;
}
/*#endregion */
/*Thay thuộc tính float bằng display*/
.inline-block {
  display: inline-block;
  vertical-align: top;
}
/*#region Main Menu */
.mainMenu {
  background: #4f86f8;
  border-radius: 10px;
  padding: 10px;
}

.mainMenu a {
  font-size: 16px;
  color: #FFFFFF;
  font-weight: bold;
  padding: 10px;
  border-right: 1px solid #fff;
}

/*#endregion */
```

Có thể xem ví dụ hoàn chỉnh trong file Vidu16.6.css và Vidu16.6.html

- kết quả khi chạy ví dụ 16.6 :



17. Tổng kết

- Chúng ta đã học được những khái niệm cơ bản nhất về ngôn ngữ html và định dạng html bằng CSS . Đây là các kiến thức nền tảng để các bạn có thể theo học các khóa lập trình ứng dụng MVC của chúng tôi .

- Đây là các khái niệm của lập trình web dù cho bạn có học công nghệ asp , php , jsp cũng đều cần đến nó

- Tất nhiên còn các kiến thức chuyên sâu và nhiều kỹ thuật nữa .Tuy nhiên điều chúng tôi muốn là các bạn có được nền tảng và đi vào xây dựng các ứng dụng thực tiễn như asp , php , jsp hay xử lý opensource để có thể thực hành chúng luôn hơn là học chuyên sâu về nó .

- Đó là lý do vì sao chỉ cần mất một vài buổi đọc vài chục trang tài liệu là các bạn có thể tự mình xử lý html , css .Điều mà rất kiêng kỵ trong lập trình ứng dụng đó là học thật nhiều nhưng không dùng hết .

- Điều chúng tôi muốn hướng tới đó là các bạn có kiến thức nền tảng (chỉ mất khoảng vài ngày để học nó) .Kết hợp với lập trình ứng dụng thực tế để dần nâng cao tay nghề hơn là mình đi học chi tiết một tay nghề rồi công nghệ mới ra đời ta lại bỏ đi những kỹ thuật cũ kỹ đã học không cần thiết với xu thế công nghệ .