



KHOA CÔNG NGHỆ THÔNG TIN

NGÔN NGỮ CSS



ThS. PHẠM ĐÀO MINH VŨ

vupdm@itc.edu.vn



PHẦN 1. CSS CƠ BẢN

PHẦN 2. CSS NÂNG CAO

PHẦN 3. CSS 3

PHẦN 1. CSS CƠ BẢN

- ❑ **Giới thiệu CSS**
- ❑ Các thuộc tính định dạng CSS



Giới thiệu CSS

✓ CSS là gì?

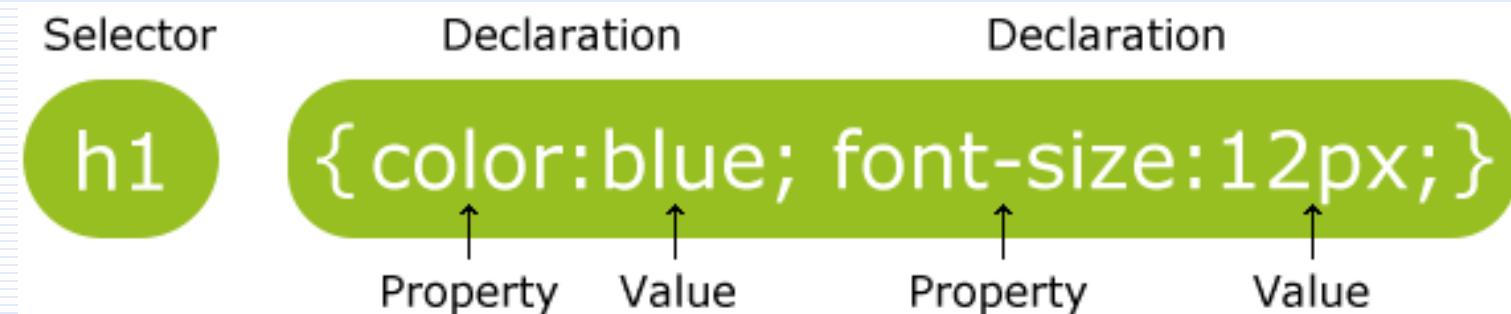
- CSS : viết tắt của **Cascading Style Sheet**
- CSS là tập hợp các **định dạng** để hiển thị & trang trí trang HTML
- CSS được lưu dưới dạng file text có đuôi .css

✓ Lợi ích của CSS ?

- Tách phần định dạng ra khỏi HTML (Loại bỏ thẻ ****, **<i>**, **<u>**, ****...)
- Tăng tốc việc phát triển web. Việc lập trình nội dung trang web và việc layout giao diện có thể được làm song song
- Tiêu tốn ít không gian lưu trữ. Có hiệu quả đồng bộ.
- Được hỗ trợ bởi tất cả các trình duyệt
- Thay đổi giao diện dễ dàng, chỉ cần thay đổi file CSS mới

CSS syntax

- ✓ CSS bao gồm các luật định dạng
- ✓ Mỗi luật bao gồm 2 phần :



- Selector : xác định các thẻ HTML sẽ được định dạng
- Declaration gồm 1 property và 1 value. Property là thuộc tính định dạng mà ta cần thay đổi, value là giá trị của thuộc tính đó

Trong ví dụ trên ta định dạng tất cả thẻ `<h1>` sẽ có `color = blue` và `font-size = 12 px`

CSS example

```
<head>
  <style>
    h1{
      color: red;
      text-align: center;
    }
  </style>
</head>
<body>
  <h1>Hello ITC</h1>
  <h1>My name is VU</h1>
</body>
```



CSS comment

- ✓ Giống C++, ta sử dụng // và /* */ để comment các ghi chú và các luật không xài.

```
/*This is a comment*/  
p  
{  
text-align:center;  
/*This is another comment*/  
color:black;  
font-family:arial;  
}
```

- ✓ Lưu ý:

- Các declaration trong 1 luật CSS phải được bao lại bởi { và }
- Mỗi declaration nên để 1 dòng để dễ đọc và dễ bảo trì

CSS id & class

- ✓ Để xác định các thẻ cần định dạng, CSS có 3 cách :
 - Sử dụng tên thẻ
 - Sử dụng id và class
 - Hỗn hợp



Tag selector

Ta có thể định dạng CSS cho nhiều thẻ bằng cách
dùng tên thẻ :

```
p      // Tất cả thẻ p  
{  
    background-color : #FFFFFF;
```

```
}
```

```
h1 p // Tất cả thẻ p nằm trong thẻ h1
```

```
{
```

```
    color : red;  
    font-weight:bold;
```

```
}
```



Id selector

- ✓ Id selector dùng để áp dụng định dạng CSS cho 1 thành phần HTML duy nhất
- ✓ Id selector = “#” + giá trị thuộc tính id của thành phần HTML
- ✓ Ví dụ luật dưới đây áp dụng cho các thành phần có id = “para1”

```
#para1{  
    color: red;  
    text-align: center;  
}
```



Ví dụ

```
<head>
<style>
  #para1{
    color: red;
    text-align: center;
  }
</style>
</head>
<body>
  <h1 id="para1">Hello ITC</h1>
  <h1>My name is VU</h1>
```

← → ⌂ http://127.0.0.1:3003/test.html

THIẾT KẾ
Hello ITC
My name is VU

Class selector

- ✓ Class selector dùng để định dạng 1 nhóm các HTML element thuộc cùng 1 lớp (class)
- ✓ Class selector = “.” + giá trị thuộc tính class trong thành phần HTML
- ✓ Ví dụ

```
.center{ text-align: center; }
```

```
h1.center { text-align: center; }
```



Ví dụ

```
<style>
  .center-red{
    color: red;
    text-align: center;
  }
</style>
</head>
<body>
  <h1 class="center-red">Hello ITC</h1>
  <h1>My name is VU</h1>
</body>
```

← → ⏪ <http://127.0.0.1:3003/test.html>

Hello ITC

My name is VU

Ví dụ

```
<head>
<style>
  h1.center-green{
    color: green;
    text-align: center;
  }
</style>
</head>
<body>
  <h1 class="center-green">Hello ITC</h1>
  <h2 class="center-green">My name is VU</h2>
</body>
```

← → ⏪ http://127.0.0.1:3003/test.html

Hello ITC

My name is VU

Tổng hợp bộ chọn (selectors)

Bộ chọn	Ví dụ	Mô tả
*	*	Chọn tất cả các thành phần html.
tag	p	Chọn tất cả các thành phần cùng loại.
tag01 tag02	div p	Chọn tất cả thành phần <tag02> bên trong thành phần <tag01>.
tag01,tag02	div,p	Chọn tất cả thành phần <tag01> và <tag02>.
tag01+tag02	div+p	Chọn tất cả thành phần <tag02> được đặt kế và sau thành phần <tag01>.
tag01>tag02	div>p	Chọn tất cả thành phần <tag02> có thành phần cha là <tag01>.
.class	.className	Chọn tất cả các thành phần có cùng tên class.
#id	#idName	Chọn tất cả các thành phần có cùng tên id.

Sử dụng CSS

✓ Có 3 cách chèn CSS vào HTML :

- Dùng file CSS riêng
- Chèn CSS vào thẻ `<head>`
- Chèn CSS dạng inline vào các thẻ HTML



Dùng file CSS riêng

- ✓ Thích hợp cho việc định dạng nhiều trang cùng 1 lúc
- ✓ Các trang HTML phải liên kết đến file CSS bằng thẻ `<link>`

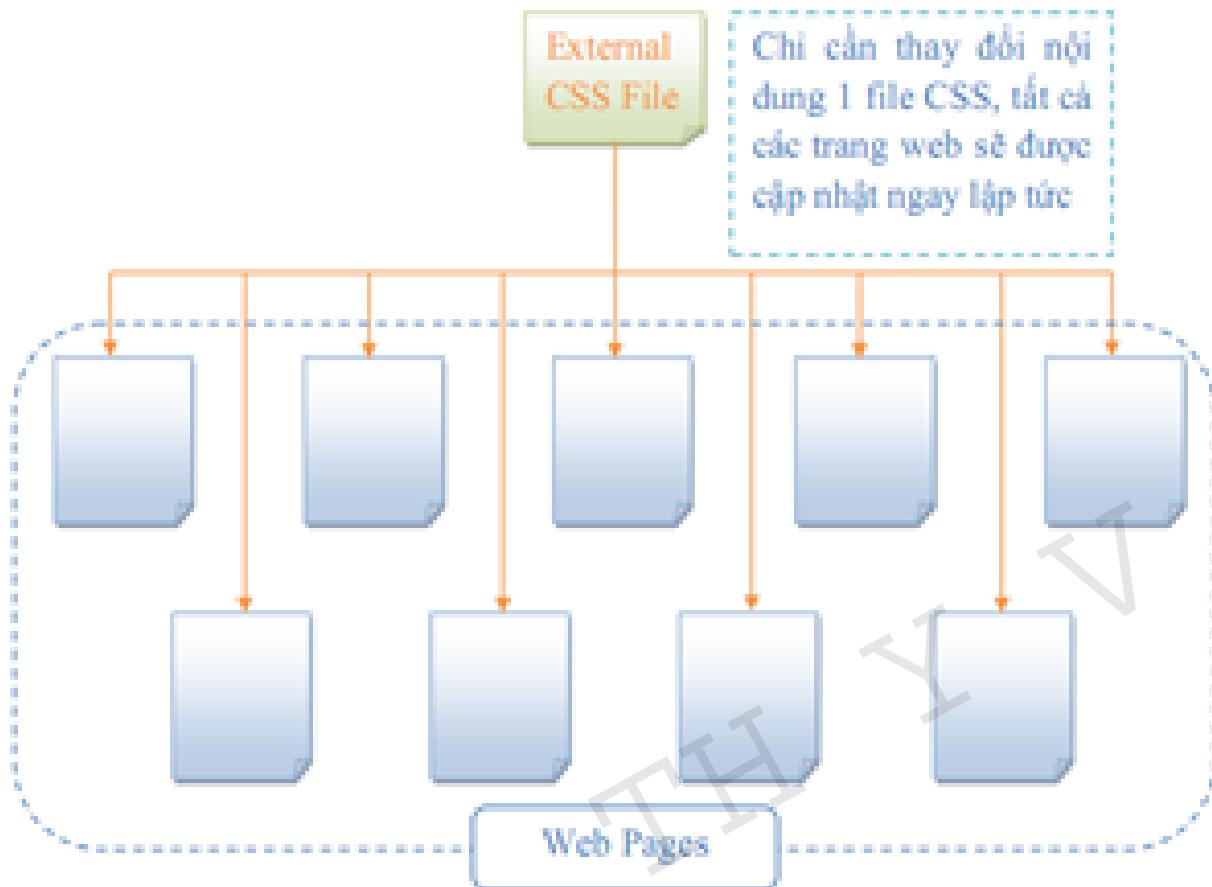
```
<head>
```

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

- ✓ Việc lưu file CSS riêng sẽ giúp ta dễ dàng đổi giao diện trang web bằng cách thay file CSS
- ✓ File CSS lúc này chỉ bao gồm toàn các luật CSS :

```
h1{text-align: center;}  
p.canhgiua{text-align: center;}  
h2.indam{font-weight: bold;}
```

Dùng file CSS riêng



Chèn CSS vào thẻ <head>

- ✓ Thay vì để CSS riêng ra 1 file, ta có thể nhúng vào trang HTML ở trong phần <head>
- ✓ Các luật CSS phải nằm trong thẻ <style>

```
<head>
<style>
  h1.center-green{
    color: green;
    text-align: center;
  }
</style>
</head>
```



Chèn CSS dạng inline

- ✓ Các định dạng CSS sẽ được chèn trực tiếp vào thuộc tính style của các thẻ HTML. Lúc này ta không cần đến selector
- ✓ Sử dụng dạng inline sẽ xóa bỏ ưu điểm tách biệt nội dung và giao diện của CSS

```
<h1 style="color: blue; text-align: center;">Hello </h1>
```

- ✓ Nếu trang HTML sử dụng tất cả các kiểu chèn CSS trên thì **thứ tự định dạng** sẽ là:
 1. Các định dạng inline
 2. Các định dạng trong thẻ `<head>`
 3. Các định dạng từ file CSS
 4. Các định dạng mặc định của trình duyệt



Các thuộc tính định dạng CSS

1. **CSS Background**
2. CSS Border
3. CSS margin/padding
4. CSS width/height
5. CSS Text
6. CSS Font
7. CSS Link
8. CSS List
9. CSS Table
10. CSS Box Model



1. Background

- ✓ Các thuộc tính background dùng để xác định hiệu ứng màu nền + hình nền cho các thành phần HTML
- ✓ Thuộc tính này bao gồm :
 - background-color : màu nền
 - background-image : hình nền
 - background-repeat : cách lặp lại hình nền
 - background-position : vị trí của hình nền



Background color

- ✓ Thuộc tính background-color giúp ta đặt màu nền cho thành phần HTML
- ✓ Cú pháp : background-color=#<Hexa color>;
- ✓ Ví dụ :

```
h1{ background-color: green; }
```

```
p { background-color: #FF0000; }
```

Ví dụ

```
<style>
  h1{ background-color: red; }
  h2 { background-color: #FF00FF; }
</style>
</head>
<body>
  <h1 >Hello ITC</h1>
  <h2>My name is VU</h2>
</body>
```



Background image

- ✓ Thuộc tính background-image dùng để đặt hình nền cho thành phần HTML
- ✓ Mặc định hình nền sẽ tô đầy kích thước của thành phần HTML
- ✓ Cú pháp :

background-image : url(<Đường dẫn hình nền>);



Ví dụ

```
<head>
<style>
body{ background-image: url(images/bg.jpg); }
</style>
</head>
<body>
  <h1>Hello ITC</h1>
  <h2>My name is VŨ</h2>
</body>
```



Background repeat

- ✓ Mặc định background-image sẽ lặp lại hình nền cho cả 2 chiều dọc và ngang
- ✓ Một số hình chỉ cần repeat ngang hoặc dọc nếu không sẽ rất xấu.
- ✓ Thuộc tính background-repeat qui định việc lặp lại này
- ✓ Cú pháp

background-repeat : <Giá trị repeat>;

Giá trị repeat gồm :

- repeat-x
- repeat-y
- repeat
- no-repeat



Ví dụ

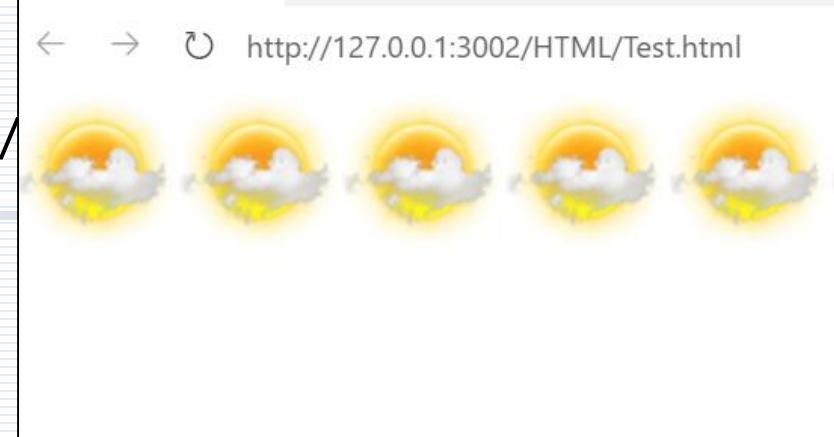
```
<head>
<style>
body{ background-image: url(..../Images/30.png) }
</style>
</head>
<body>

</body>
```



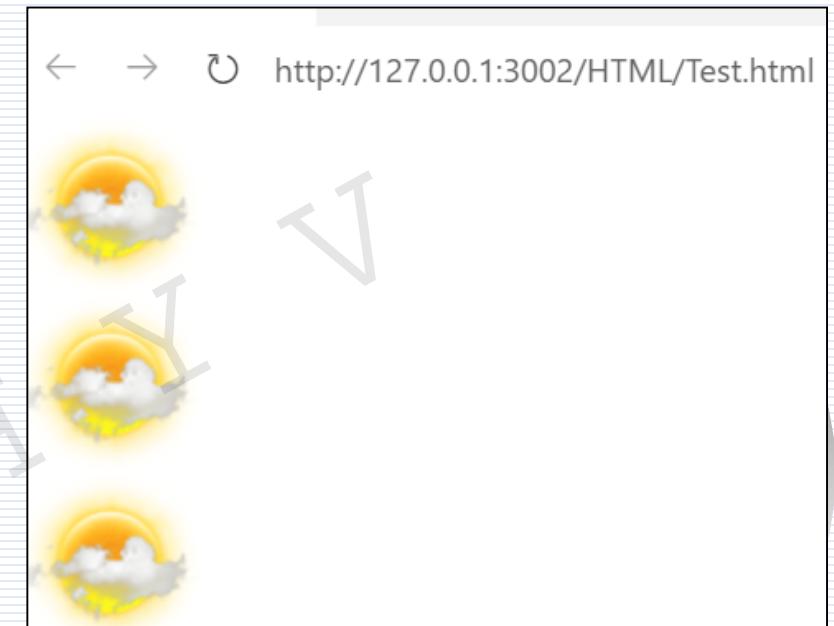
```
body{
```

```
background-image: url(..../Images/  
background-repeat: repeat-x; }
```



```
body{
```

```
background-image: url(..../Images/30.png);  
background-repeat: repeat-y; }
```



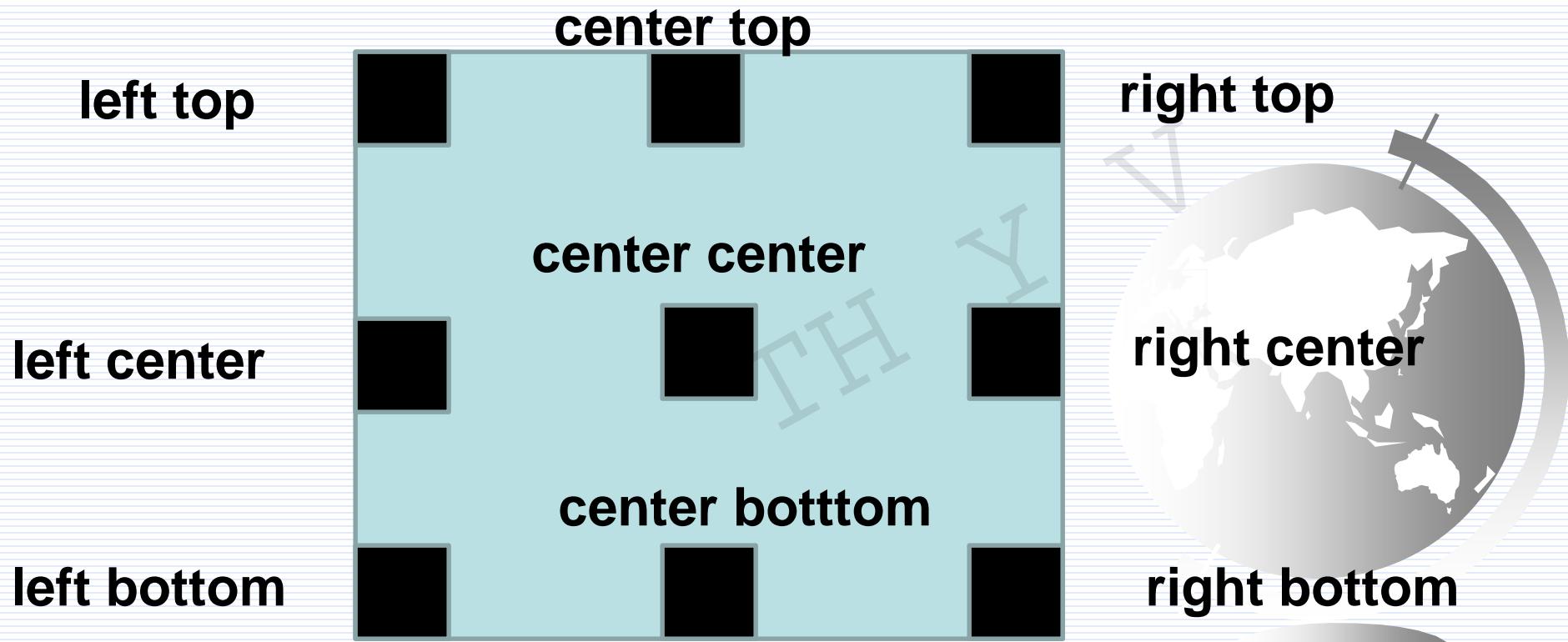
```
background-repeat: no-repeat;
```

```
← → ⌂ http://127.0.0.1:3002/HTML/Test.html
```



Background position

- ✓ Thuộc tính background-position chỉ hiện hình nền tại 1 vị trí nhất định (đi kèm với background-repeat: no-repeat;)
- ✓ Cú pháp : background-position : <Vị trí>;



Ví dụ

Hello World!

W3Schools background no-repeat, set position example.

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

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

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

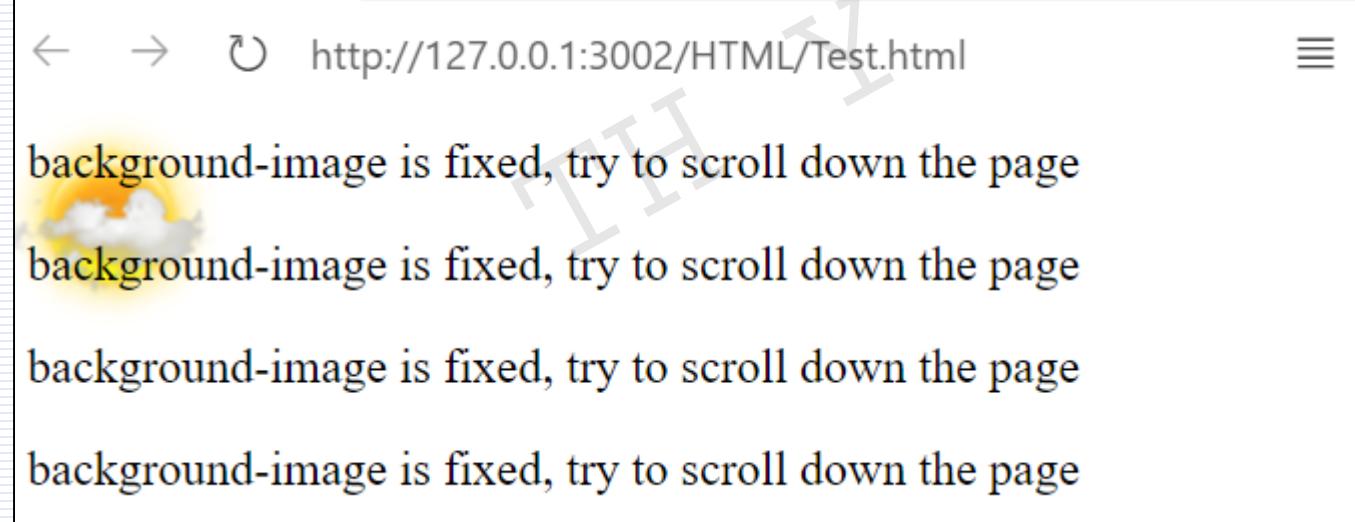


Fixed background

- ✓ Thuộc tính **background-attachment** :
fixed dùng để cố định vị trí của hình nền.
- ✓ Ví dụ ta muốn có 1 hình nền ở góc trên
bên trái trang web. Hình nền sẽ luôn luôn
ở vị trí góc trên bên trái dù người dùng có
cuộn trang web.



```
<head>
<style>
body{
    background:#ffffff url('../Images/30.png') no-repeat ;
}
</style>
</head>
<body>
<p>background-image is fixed, try to scroll down the page</p>
<p>background-image is fixed, try to scroll down the page</p>
</body>
```



Background short hand

- ✓ Để làm ngắn code CSS, ta có thể gom tắt cả các thuộc tính (background-color, background-image, background-repeat, background-position) vào thuộc tính background :

```
background:#ffffff url('img_tree.png')  
no-repeat right top;
```

Property	Description	Values	CSS
<u>background</u>	Đặt màu nền hình nền	<i>background-color , background-image , background-repeat , background-attachment , background-position, inherit</i>	1
<u>background-attachment</u>	Đặt vị trí cố định cho hình nền	scroll , fixed ,inherit	1
<u>background-color</u>	Đặt màu nền	<i>color-rgb , color-hex , color-name transparent , inherit</i>	1
<u>background-image</u>	Đặt hình nền	url(<i>URL</i>) none inherit	1
<u>background-position</u>	Đặt vị trí cho hình nền	left top , left center , left bottom , right top , right center , right bottom , center top ,center center , center bottom , x% y% , xpos ypos , inherit	1
<u>background-repeat</u>	Qui định cách lặp lại hình	repeat , repeat-x , repeat-y , no-repeat , inherit	1

2. CSS border

✓ border-style : qui định kiểu border :

dotted: Defines a dotted border



dashed: Defines a dashed border

solid: Defines a solid border

double: Defines two borders. The width of the two borders are the same as the border-width value

groove: Defines a 3D grooved border. The effect depends on the border-color value

ridge: Defines a 3D ridged border. The effect depends on the border-color value

inset: Defines a 3D inset border. The effect depends on the border-color value

outset: Defines a 3D outset border. The effect depends on the border-color value

CSS border (2)

- ✓ border-width : qui định chiều dày của border

```
p{ border-style: solid;  
border-width: 5px;  
}
```

Some text.

CSS border(3)

- ✓ border-color : qui định màu cho border

```
p.one{ border-style: solid;  
border-color: red;  
}
```

```
p.two{ border-style: solid;  
border-color: green;  
}
```

A solid red border

A solid green border

CSS border (4)

- ✓ Tiếp đầu ngữ : border-left , border-top, border-right, border-bottom dùng kèm với width, color, style để qui định độ dày, màu sắc, kiểu của border trái, trên, phải, dưới.

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

[2 different border styles.]

CSS border (5)

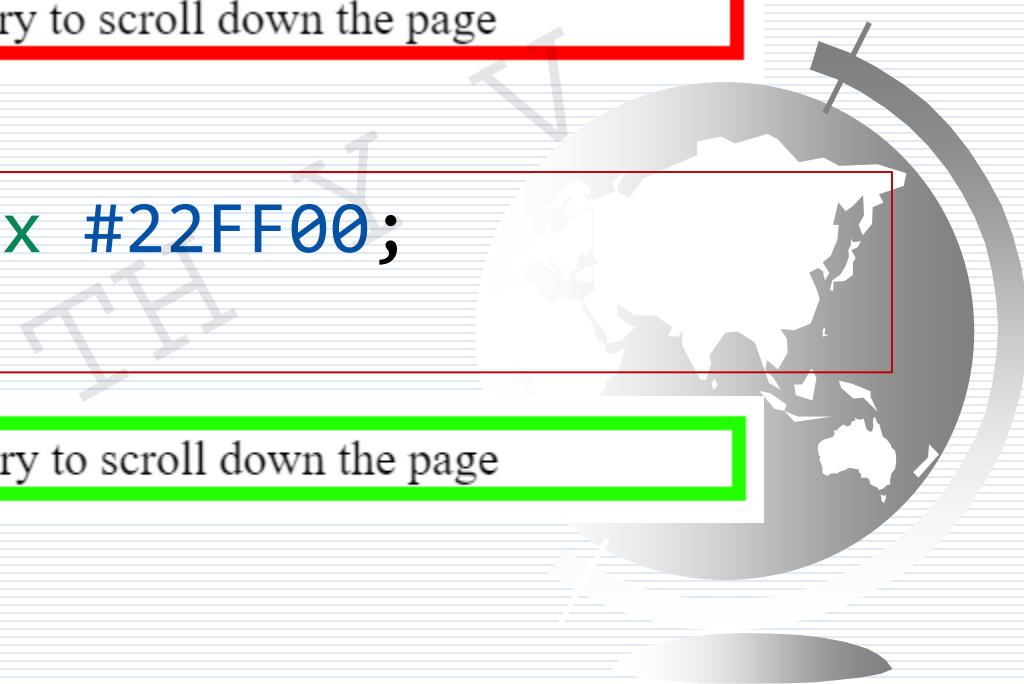
✓ Ghi tắt

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

background-image is fixed, try to scroll down the page

```
p{ border:solid 5px #22FF00;  
}
```

background-image is fixed, try to scroll down the page



3. CSS margin

- ✓ margin-left , margin-top,margin-right, margin-bottom : dùng để qui định margin của bên trái, trên, phải và dưới
- ✓ Giá trị của margin :
 - auto : trình duyệt tự động quyết định
 - px, em : độ dài bằng pixel hay em
 - % độ dài bằng phần trăm



CSS margin(2)

```
p{  
background-color: yellow;  
}
```

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

This is a paragraph with no specified margins.



This is a paragraph with specified margins.



CSS margin(3) – Viết tắt

- ✓ `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



4. CSS padding

- ✓ padding-left , padding-top, padding-right, padding-bottom : dùng để qui định padding của bên trái, trên, phải và dưới
- ✓ Giá trị của padding:
 - px, em : độ dài bằng pixel hay em
 - % độ dài bằng phần trăm



CSS padding (2)

```
p{  
background-color: yellow;  
}
```

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

This is a paragraph with no specified padding.

This is a paragraph with specified paddings.

CSS padding (3)

✓ `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



5.1 Width / Height

- ✓ Width: thiết lập chiều rộng cho đối tượng
- ✓ Height: thiết lập chiều cao cho đối tượng
- ✓ Ví dụ :
 - width**: 500px; //chiều rộng cố định
 - width**: 50%; //chiều rộng tương đối
 - height**: 30em;
- ✓ *Chú ý: Chúng ta chỉ dùng width/height cho các đối tượng là block*

5.2 max-width

✓ Max-width: thiết lập chiều rộng tối đa, nếu kích thước màn hình nhỏ hơn thì đối tượng tự động co lại

✓ Ví dụ : **Max-width: 400px;**

– Màn hình lớn

This div element has width: 500px;

This div element has max-width: 500px;

– Màn hình nhỏ

This div element has width: 500px;

This div element has max-width: 500px;



6. Block và inline element

- ✓ **Block element** là 1 element khi hiển thị sẽ chiếm toàn bộ chiều ngang của cha nó.
 - Ví dụ block element : <h1>,<h2>,<p>,<div>
- ✓ **Inline element** là 1 element khi hiển thị chỉ chiếm đủ chiều ngang để hiển thị nội dung
 - Ví dụ inline element : <a>,

Phần tử là inline thì chúng ta không thể thay đổi chiều rộng và chiều cao của nó. Ngược lại, block thì thay đổi được.



Ví dụ

```
4      background:red;
5
6      background:yellow;
7
8      background:green;
9
10     background:gold;
11
12     background:cyan;
```

Heading 1

Heading 2

I am span

Link 1 Link 2 Link 3



I am p

- Item 1
- Item 2

Display

✓ Ta có thể thay đổi cách hiển thị mặc định của 1 element sử dụng display:

- **display: inline;**
- **display: block;**
- **display: inline-block;** // 1 phần tử vừa là inline vừa là block (có nghĩa xuất hiện vừa đủ không gian nó chiếm, đồng thời cũng có thể điều chỉnh width và height)

```
a{  
    display: inline-block;  
    width: 120px;  
    background-color: yellow;  
    line-height: 35px;  
    text-align: center;  
}
```

[liên kết](http://127.0.0.1:3002/)

7. Text : color

✓ Thuộc tính color dùng để đặt màu chữ cho văn bản

✓ Cú pháp :

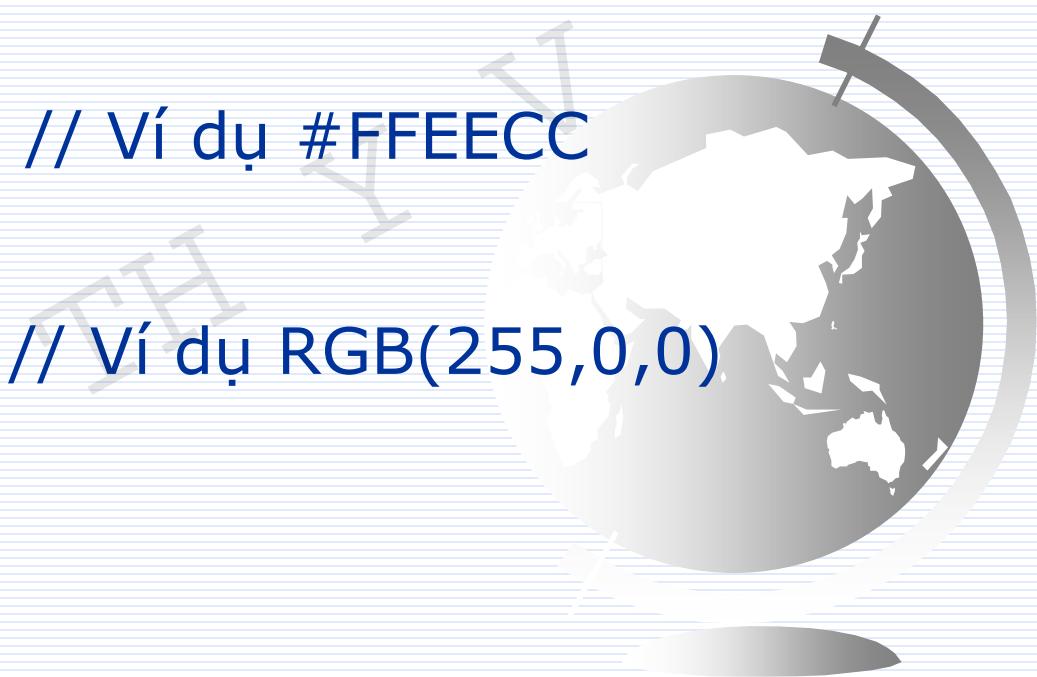
color : <Tên màu>; // Ví dụ red

hoặc

color : <Màu hexa>; // Ví dụ #FFEECC

hoặc

color : <Màu RGB>; // Ví dụ RGB(255,0,0)



```
<html>
<head>
<style type="text/css">
body {color:red;}
h1 {color:#00ff00;}
p.ex {color:rgb(0, 0, 255);}
</style>
</head>

<body>
<h1>This is heading 1</h1>
<p>This is an ordinary paragraph. Notice that this
text is red. The default text-color for a page is
defined in the body selector.</p>
<p class="ex">This is a paragraph with class="ex".
This text is blue.</p>
</body>
</html>
```

This is heading 1

This is an ordinary paragraph. Notice that this text is red. The default text-color for a page is defined in the body selector.

This is a paragraph with class="ex". This text is blue.



Text alignment

- ✓ Thuộc tính text-align qui định cách canh lề cho văn bản
- ✓ Cú pháp :
text-align : <Cách canh lề>
- ✓ Cách canh lề :
 - left : canh lề trái
 - right : canh lề phải
 - center : canh giữa
 - justify : canh đều 2 bên

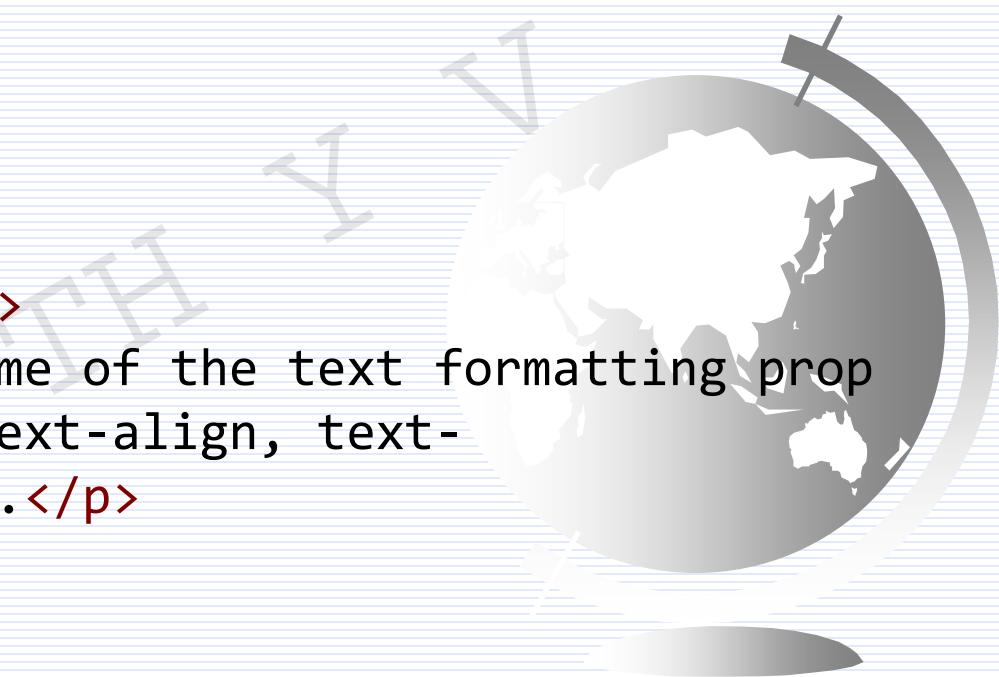


TITLE

24 Sep 2022

This text is styled with some of the text formatting properties. The heading uses the text-align, text-transform, and color properties. The paragraph is indented,

```
<head>
<style>
h1{ text-align: center; }
p{ text-align: justify; }
p.date { text-align: right; }
</style>
</head>
<body>
<h1>TITLE</h1>
<p class="date">24 Sep 2022</p>
<p>This text is styled with some of the text formatting properties. The heading uses the text-align, text-transform, and color properties.</p>
</body>
```



Text decoration

- ✓ Thuộc tính text-decoration dùng để đặt hoặc xóa các trang trí cho văn bản
- ✓ Cú pháp
 - text-decoration : <Cách trang trí> ;
- ✓ Cách trang trí :
 - overline : đường gạch ngang phía trên văn bản
 - underline : đường gạch dưới
 - line-through : đường gạch giữa văn bản
 - blink : nhấp nháy
 - none : không trang trí

Ví dụ

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

This is heading 1

~~This is heading 2~~

This is heading 3

Note: The "blink" value is not supported in IE, Chrome, or Safari.

Text Transform

- ✓ Thuộc tính text-transform dùng để chuyển chữ hoa ,thường cho văn bản
- ✓ Cú pháp
 - text-transform : <Kiểu chữ>
- ✓ Kiểu chữ :
 - lowercase : Chuyển tất cả kí tự sang chữ thường
 - uppercase : Chuyển tất cả kí tự sang chữ hoa
 - capitalize : Ghi hoa các kí tự đầu của chữ

Ví dụ

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

THIS IS SOME TEXT.

this is some text.

This Is Some Text.



Các thuộc tính khác

- ✓ text-indent: qui định cách lề cho dòng đầu tiên (bằng pixel).
- ✓ letter-spacing : khoảng cách giữa các kí tự
- ✓ word-spacing : khoảng cách giữa các từ
- ✓ line-height: chiều cao của 1 dòng (pixel hoặc %)
- ✓ direction : hướng viết văn bản (rtl : phải qua trái)
- ✓ white-space : nowrap ; // Không wrap văn bản
- ✓ vertical-align: đặt canh lề dọc (baseline, sub, super, top, text-top, middle, bottom, text-bottom).

```
h1{ letter-spacing: 3px; }
```

```
h2{ letter-spacing: -3px; }
```

This is heading 1

This is heading 2

```
p{ word-spacing: 30px; }
```

This is some text. This is some text.

```
p{ line-height: 90%; }
```

```
p{ line-height: 200%; }
```

This is a paragraph with a standard line-height. The default line height in most browsers is about 110% to 120%.

This is a paragraph with a smaller line-height. This is a paragraph with a smaller line-height.

This is a paragraph with a bigger line-height. This is a paragraph with a bigger line-height.



Ví dụ

```
img.top{ vertical-align: text-top; }  
img.bottom{ vertical-align: text-bottom; }
```



9. CSS Font

- ✓ Các thuộc tính CSS font cho phép thay đổi font, kích thước font, kiểu font, độ đậm của font.
- ✓ Trong CSS có 2 loại font :
 - Loại tổng quát : serif , sans-serif, monospace
 - Loại cụ thể : 1 font cụ thể

Loại tổng quát

Serif

Sans-serif

Monospace

Font cụ thể

Times New Roman
Georgia

Arial
Verdana

Courier New
Lucida Console



Font Family

- ✓ Thuộc tính font-family qui định loại font cho văn bản
- ✓ Nên sử dụng nhiều font, để nếu không có font này, trình duyệt sẽ dùng font khác
- ✓ Cú pháp :

```
p{ font-family: 'Courier New', Courier, monospace; }
```



Font style & font weight

- ✓ Thuộc tính **font-style** dùng để in nghiêng văn bản
- ✓ Giá trị :
 - *normal* : bình thường
 - *italic* : in nghiêng
- ✓ Thuộc tính **font-weight** dùng để in đậm văn bản
- ✓ Giá trị
 - *normal*
 - *bold*
 - *bolder*
 - *lighter*
 - *100, 200, 300, ... 900*



Font size

- ✓ Thuộc tính **font-size** qui định kích thước văn bản
- ✓ Có 2 loại kích thước :
 - Tuyệt đối : là giá trị tính bằng pixel
 - Tương đối là giá trị tính bằng “**em**”, “**rem**” (1 em = kích thước font mặc định của trình duyệt. Mặc định = 16px)

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

```
h1{ font-size: 2.5em; } /*2.5 x 16 = 40px */  
h2{ font-size: 1.875em; } /*1.875 x 16 = 30px */  
p { font-size: 1em; } /*1 x 16 = 16px */
```

10. Link

- ✓ Liên kết có thể được trang trí bởi các thuộc tính : color, background-color, background-image, font ...
- ✓ Một liên kết có 4 trạng thái sau :
 - ❑ a:link** – 1 liên kết bình thường, chưa được truy cập
 - ❑ a:hover** – 1 liên kết khi chuột đang ở trên nó
 - ❑ a:active** – 1 liên kết đang được nhấn
 - ❑ a:visited** – 1 liên kết đã được người dùng truy cập

Ví dụ 1

```
a:link{ color: red; }
```

This is a link

```
a:hover{ color: pink; }
```

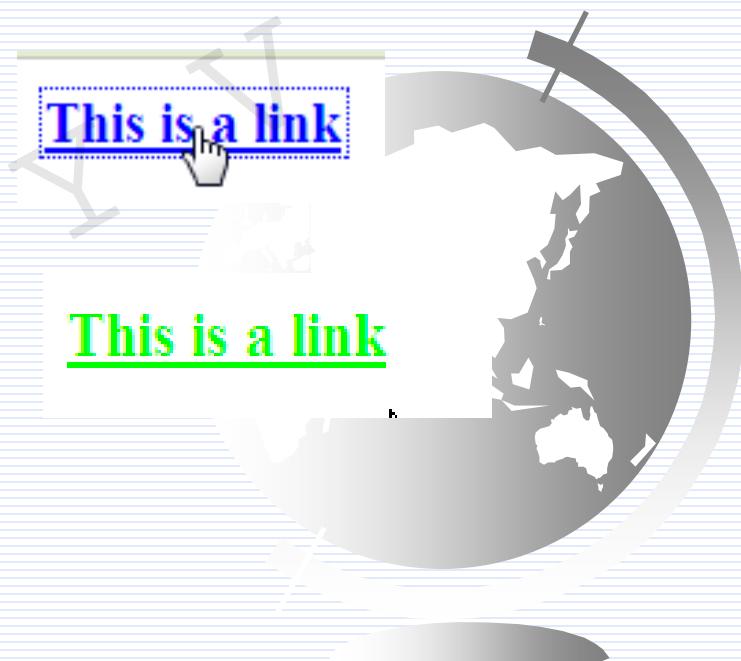
This is a link

```
a:active{ color: blue; }
```

This is a link

```
a:visited{ color: green; }
```

This is a link



11. List

- ✓ Với danh sách, ta có thể dùng CSS để thay đổi :
 - Bullet trong danh sách không thứ tự hoặc cách đánh số trong danh sách có thứ tự (list-style-type)
 - Thay thế bullet bằng hình ảnh (list-style-image)
- ✓ Giá trị thuộc tính :
 - **list-style-type** :
 - none, disc, circle, square
 - decimal, lower-roman, upper-alpha, upper-roman
 - **list-style-image** : url (image_path)



Ví dụ 1

```
ul.a{list-style-type: circle;}
```

```
ul.a{list-style-type: square;}
```

```
ol.a{list-style-type: upper-roman;}
```

```
ol.a{list-style-type: lower-alpha;}
```

Example of unordered lists:

- Coffee
- Tea
- Coca Cola

- Coffee
- Tea
- Coca Cola

Example of ordered lists:

- I. Coffee
- II. Tea
- III. Coca Cola

- a. Coffee
- b. Tea
- c. Coca Cola

Ví dụ 2

✓ `ul{`

`list-style-image: url('bullet.png');`

`}`

- Coffee
- Tea
- Coca Cola

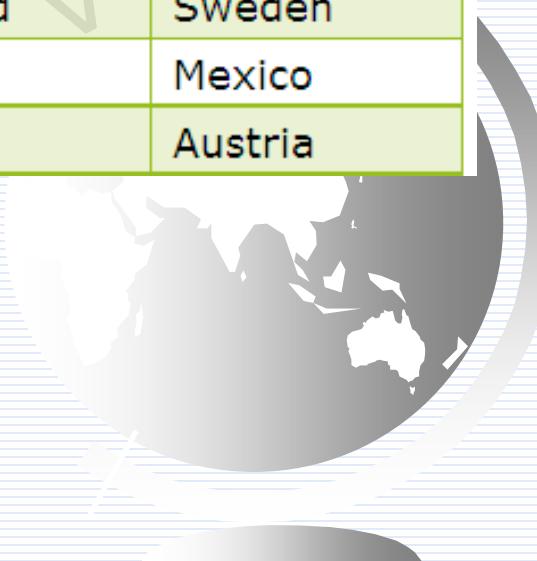


12. CSS Table

- ✓ Sử dụng CSS ta có thể trang trí bảng rất đẹp.
- ✓ Ví dụ như :

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

THIẾT KẾ WEBSITE



CSS Table

- ✓ border-collapse :
 - collapse: border các ô sẽ dính sát nhau.
 - separate : border các ô sẽ được tách ra.
- ✓ border : thuộc tính border qui định kiểu, độ dày và màu sắc đường viền
- ✓ width, height: qui định chiều dài, chiều cao của bảng, ô hoặc dòng
- ✓ text-align (left, right, justify) : canh lề văn bản
- ✓ vertical-align(bottom, middle, top) : canh lề dọc
- ✓ padding : kích thước đệm mỗi ô

Ví dụ

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

Firstname	Lastname
Peter	Griffin
Lois	Griffin

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

Firstname	Lastname
Peter	Griffin
Lois	Griffin

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

Firstname	Lastname
Peter	Griffin
Lois	Griffin

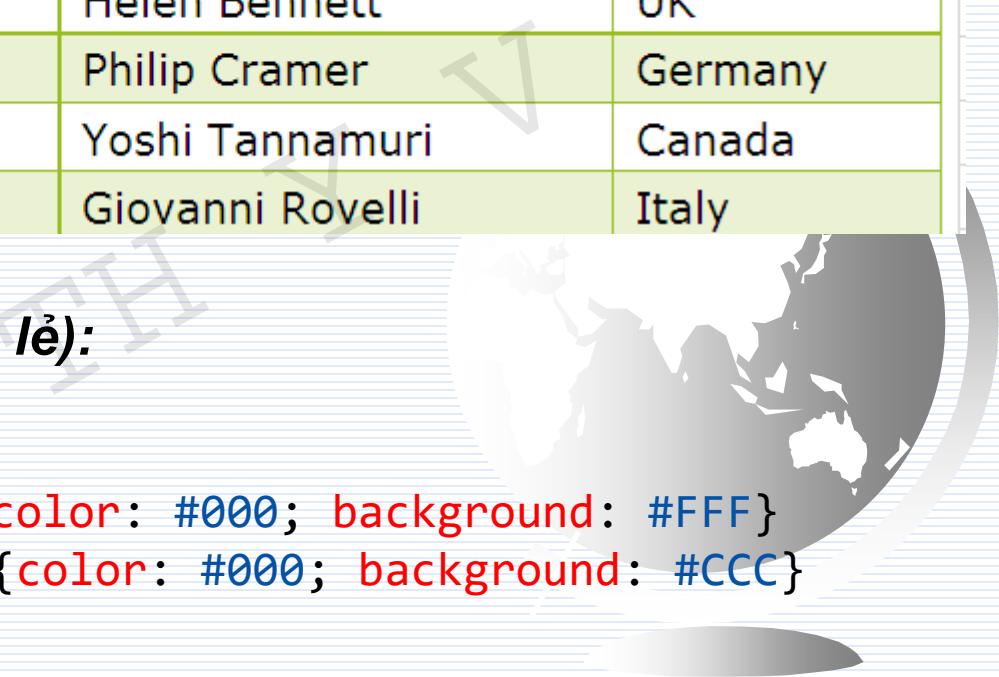
Table

- ✓ Demo dùng CSS để tạo bảng giống hình bên dưới

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Berglunds snabbköp	Christina Berglund	Sweden
Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK
Königlich Essen	Philip Cramer	Germany
Laughing Bacchus Winecellars	Yoshi Tannamuri	Canada
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy

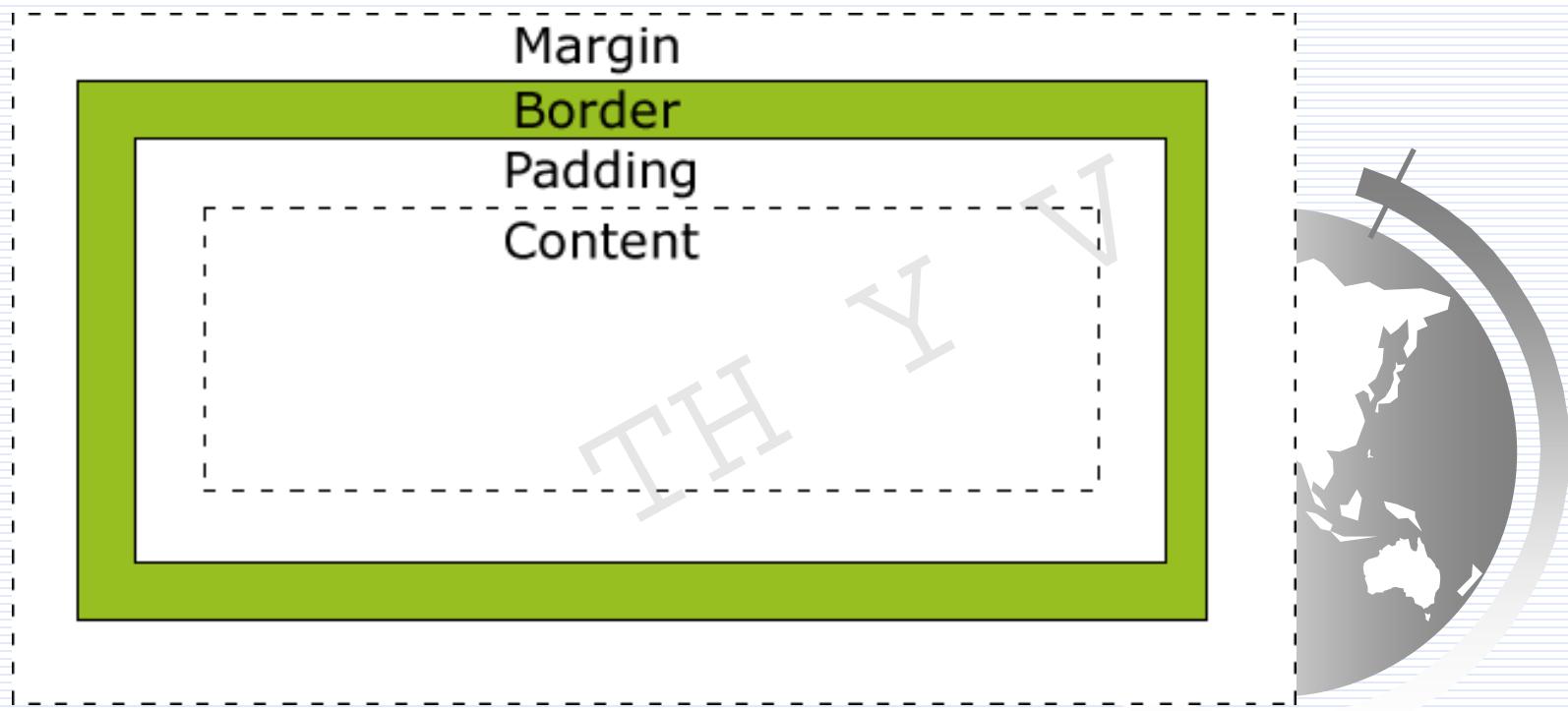
Gợi ý (tô màu dòng chẵn dòng lẻ):

```
<style type="text/css">
  table tr:nth-child(odd) {color: #000; background: #FFF}
  table tr:nth-child(even) {color: #000; background: #CCC}
</style>
```



13. Box model

- ✓ Mô tả cách CSS định dạng khối không gian bao xung quanh 1 element. Mỗi HTML element khi hiển thị trên trình duyệt sẽ là 1 cái hộp như sau :



CSS Box model part

- ✓ **margin**: là khoảng trống nằm phía ngoài border, margin sẽ không có màu nền. Luôn luôn vô hình
- ✓ **border** : là đường biên bao quanh padding và content. Một border bao gồm kiểu, độ dày, và màu
- ✓ **padding** : là khoảng đệm giữa border và content
- ✓ **content** : là nơi mà văn bản, hình ảnh nội dung của HTML element xuất hiện



Width, height of an element

- ✓ Khi chúng ta đặt thuộc tính width, height cho 1 HTML element trong CSS, thực chất ta chỉ đặt width và height cho content của nó mà thôi
- ✓ Muốn có width , height thật của box, ta phải cộng thêm margin, border, padding
- ✓ **Total element width =**
$$\text{width} +$$

$$(\text{left padding} + \text{right padding}) +$$

$$(\text{left border} + \text{right border}) +$$

$$(\text{left margin} + \text{right margin})$$



Ví dụ

```
width:250px};  
padding:10px;  
border:5px solid gray;  
margin:10px;
```

Let's do the math:

250px (width)
+ 20px (left and right padding)
+ 10px (left and right border)
+ 20px (left and right margin)
= 300px



```
* {  
  box-sizing: border-box;  
}
```

CSS box-sizing

- ✓ Để tránh trường hợp quên tính kích thước của padding, border khi thiết lập chiều rộng cho đối tượng -> **box-sizing**
- ✓ **Box-sizing: border-box** : sẽ bao gồm luôn padding và border khi thiết lập chiều rộng

```
.div1 {  
  width: 300px; height: 100px;  
  border: 1px solid blue;  
}
```

```
.div2 {  
  width: 300px; height: 100px;  
  padding: 30px;  
  border: 1px solid red;  
  box-sizing: border-box;  
}
```

Width=300px, no padding and border



Width=300px, padding=30px, border=1px

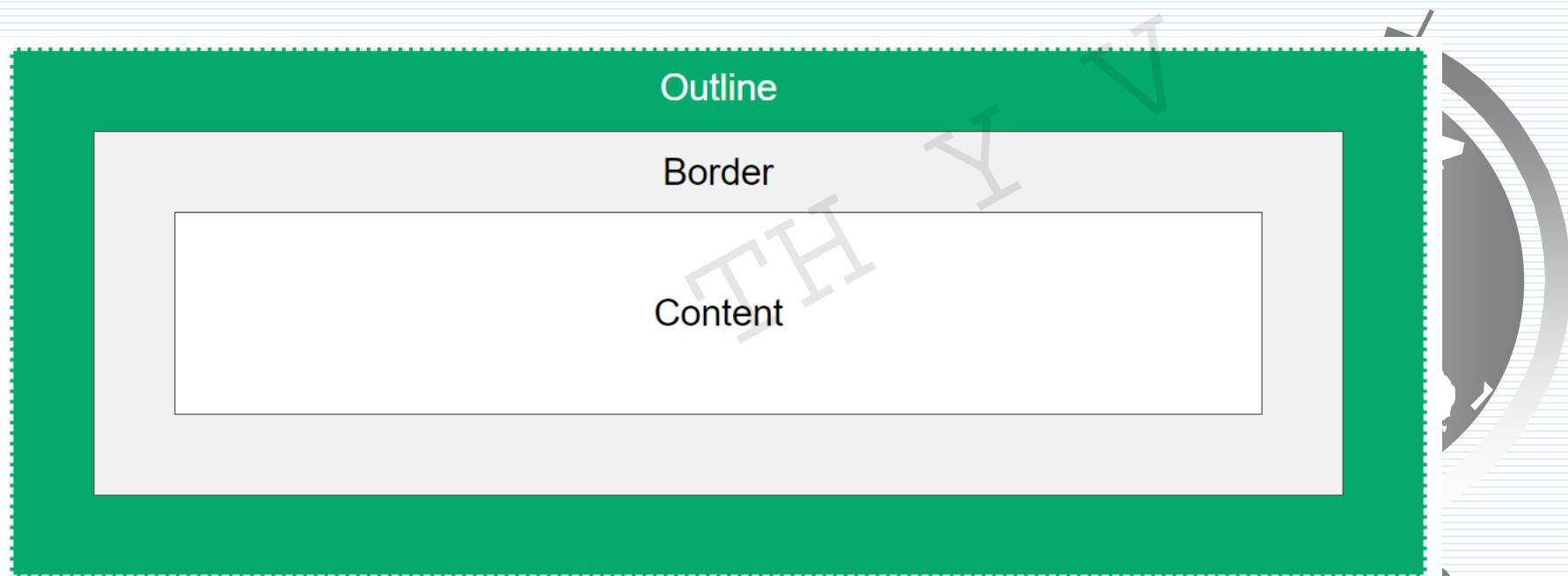
Nội dung

- ❑ Giới thiệu CSS
- ❑ CSS Background
- ❑ CSS Text
- ❑ CSS Font
- ❑ CSS Link
- ❑ CSS List
- ❑ CSS Table
- ❑ CSS Box Model
- ❑ Các thuộc tính khác**



CSS Outline

- ✓ Là dạng đường viền nằm bên ngoài đối tượng.
- ✓ Outline không làm thay đổi kích thước của đối tượng.
 - Nếu đối tượng có Margin: đè lên vùng margin
 - Nếu đối tượng không có Margin: đè lên đối tượng khác



CSS Outline

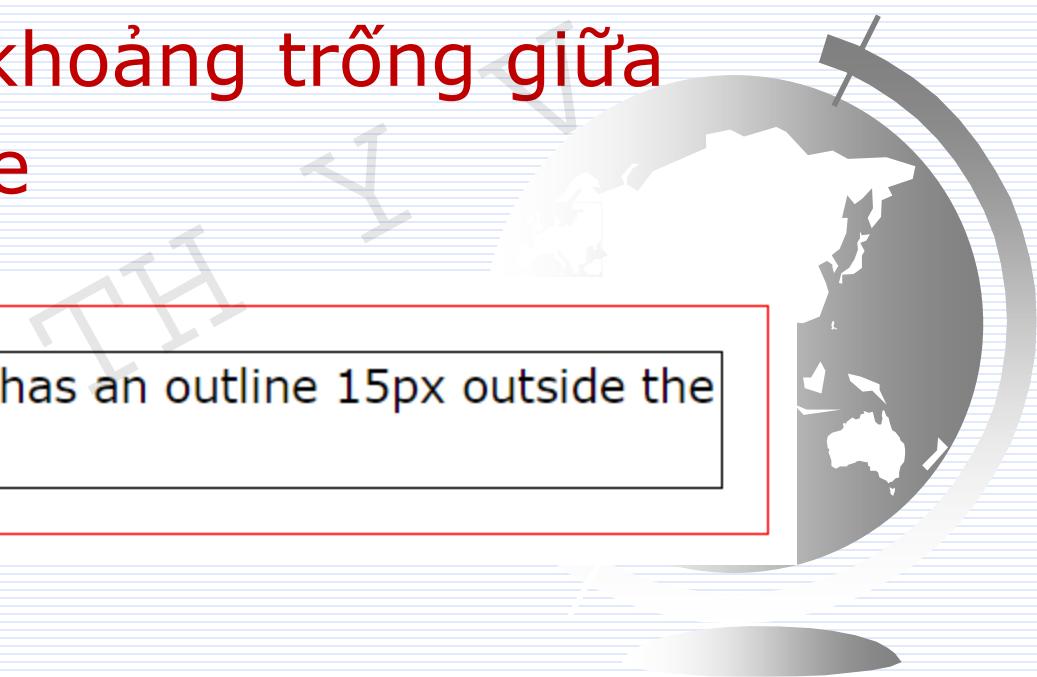
```
p {  
    border: 2px solid black;  
    outline: #4CAF50 solid 10px;  
    margin: 0px;  
    padding: 20px;  
    text-align: center;  
}
```

This element has a black border and a green outline with a width of 10px.

CSS Outline

- ✓ Các thuộc tính khác của Outline (giống Border):
 - outline-style : kiểu viền
 - outline-color : màu viền
 - outline-width : độ rộng
 - outline-offset : khoảng trống giữa border và outline

This paragraph has an outline 15px outside the border edge.

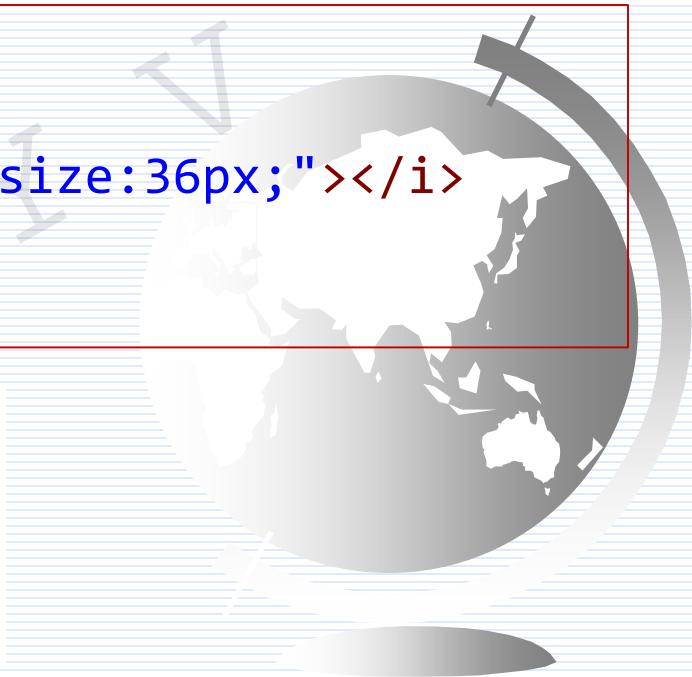


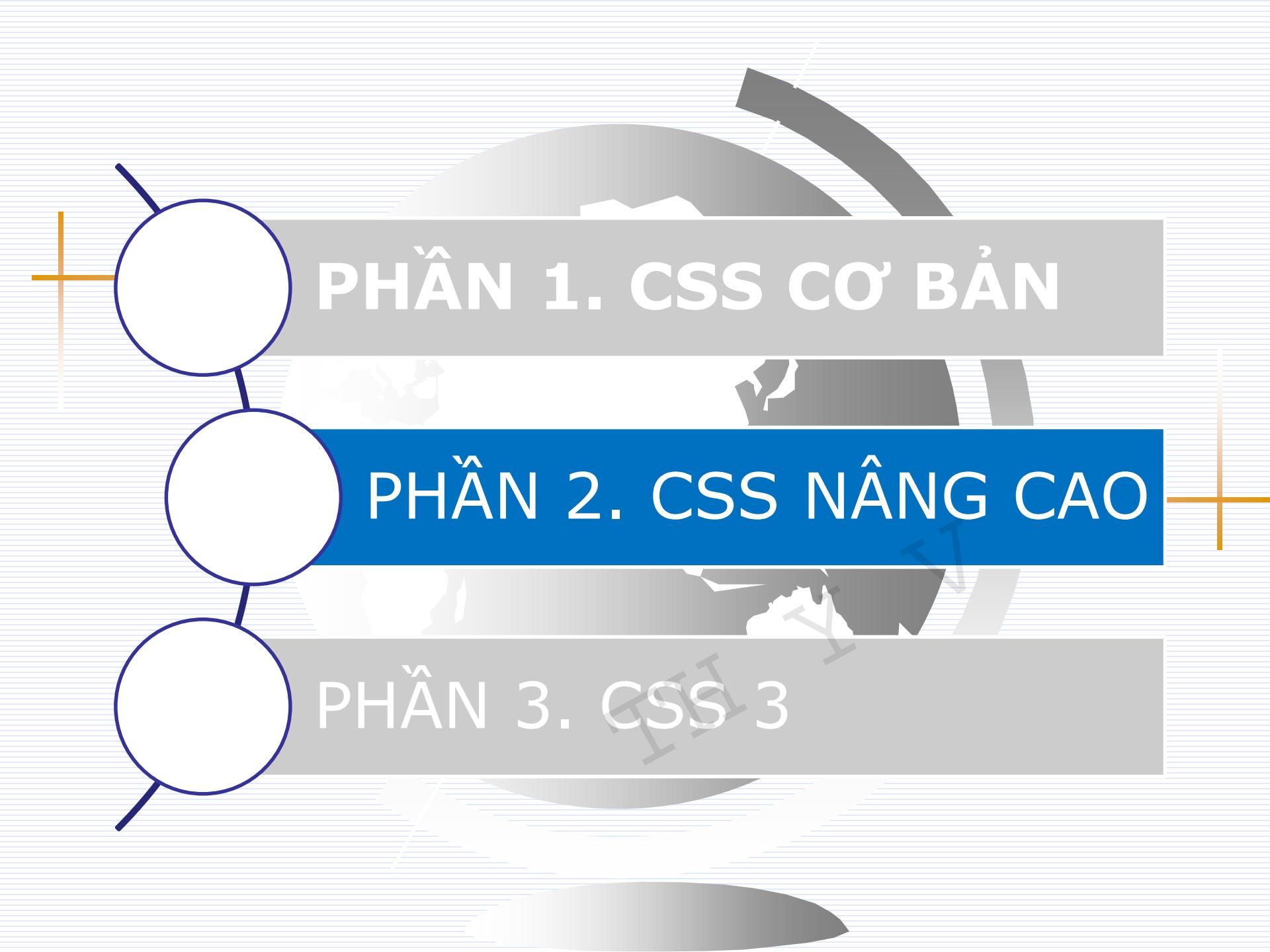
CSS ICON

- ✓ Sử dụng thư viện ICON của fontawesome.com.
- ✓ Thêm đường link này vào vùng thẻ `<head>`
- ✓ `<script src="https://kit.fontawesome.com/0e966e8647.js" crossorigin="anonymous"></script>`

```
<p>Some Font Awesome icons:</p>
  <i class="fas fa-cloud"></i>
  <i class="fas fa-heart" style="font-size:36px;"></i>
  <i class="fas fa-car" style="font-size:48px;color:red;"></i>
```

Some Font Awesome icons:





PHẦN 1. CSS CƠ BẢN

PHẦN 2. CSS NÂNG CAO

PHẦN 3. CSS 3

Nội dung

- Ẩn phần tử
- Flexbox
- CSS Position
- CSS Float
- CSS Align
- CSS Navigation Bar
- CSS Image Gallery
- CSS Image Sprite
- CSS Image Opacity



1. Ẩn element

- ✓ Ta có thể ẩn 1 element bằng cách sử dụng :
 - **display : none;**
 - **visibility : hidden;**
- ✓ `display:none;` element sẽ không chiếm giữ bất kỳ khoảng không nào. Trình duyệt sẽ hiển thị như không có element này.
- ✓ `visibility:hidden;` ẩn element nhưng khoảng không tại vị trí element vẫn được giữ nguyên

Demo ẩn element

✓ DisplayHidden.html

```
1  <html>
2  <head>
3  <style type="text/css">
4  #first{           [
5  /*visibility:hidden;*/
6  display:none;
7  }
8  </style>
9  </head>
10
11 <body>
12 
13 
14 
15 </body>
16
17 </html>
```



Demo ẩn element



Hình 1



Hình 2



Hình 3

Hình 1



Hình 2



Hình 3

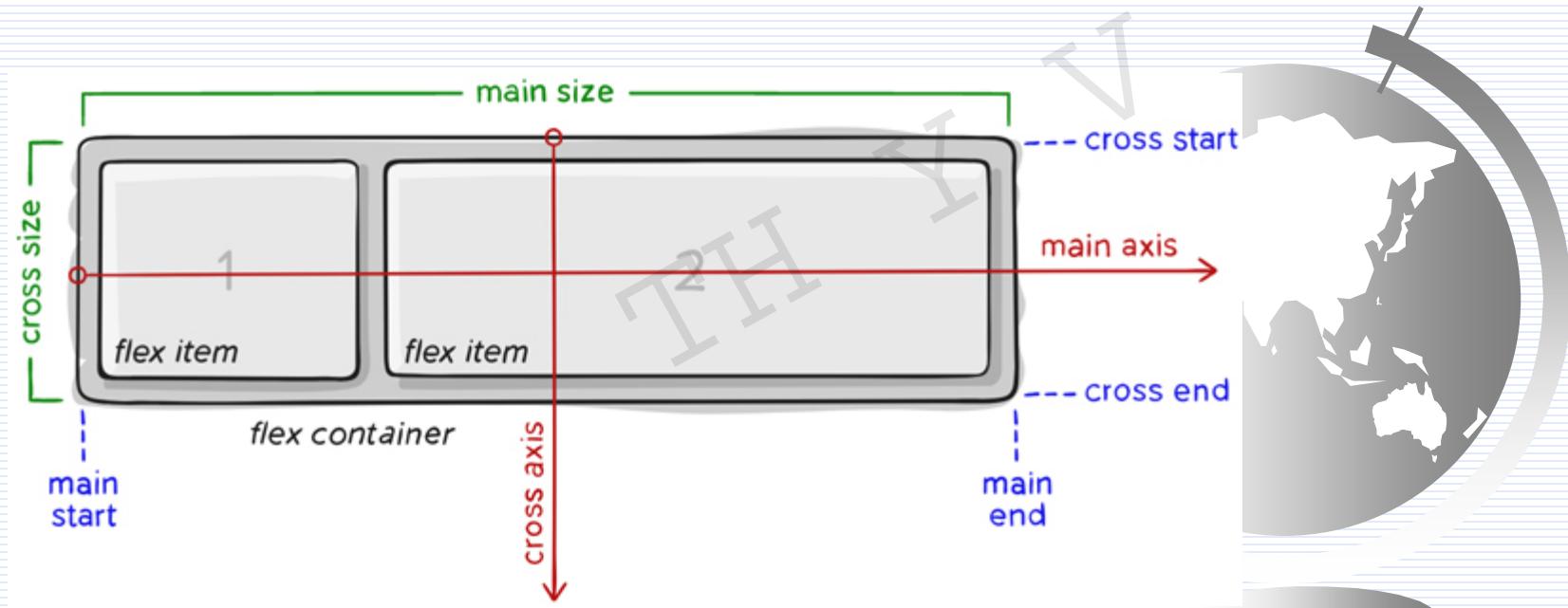
2. Flexbox

- ✓ **Flexbox Layout (hay còn gọi là Flexible Box)** là một kiểu layout có khả năng tự cân đối kích thước, thay đổi chiều rộng/chiều cao, thứ tự và vị trí phần tử bên trong để phù hợp với tất cả các loại thiết bị hiển thị và kích thước màn hình
- ✓ Flexbox giúp cho việc thiết kế layout nhanh và dễ hơn rất nhiều mà không cần Position hoặc Float.
- ✓ **Lưu ý:** Flexbox Layout phù hợp nhất để thiết lập bố cục ở quy mô nhỏ, còn thiết lập bố cục với phạm vi lớn hơn thì vẫn nên sử dụng kiểu thông thường là dàn trang theo dạng lưới (grid layout).

Flexbox

- ✓ Flexbox layout bao gồm 2 thành phần chính:
 - **container**: là thành phần lớn bao các phần tử bên trong, các item bên trong sẽ hiển thị dựa trên thiết lập của container này.
 - **item**: là phần tử con của container, ta có thể điều chỉnh vị trí, kích thước hoặc thiết lập thứ tự hiển thị của nó.
- ✓ Cú pháp:

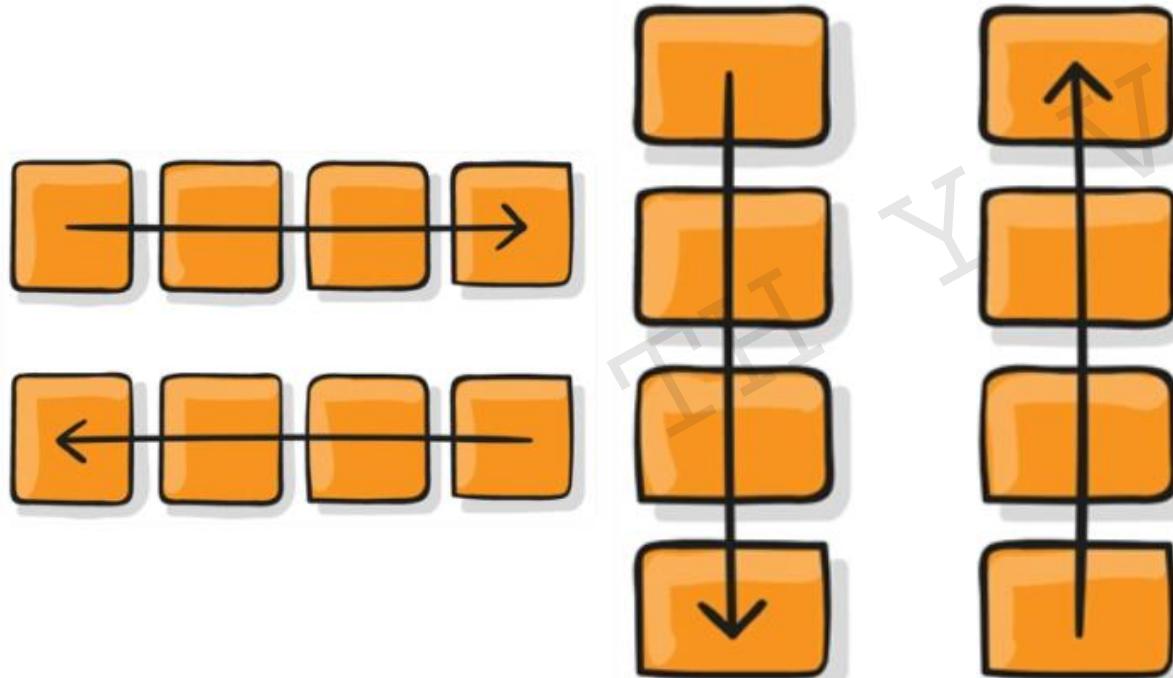
```
.container { display: flex; }
```



Flexbox: các thuộc tính

- ✓ **flex-direction:** Sắp xếp items theo hàng hoặc cột

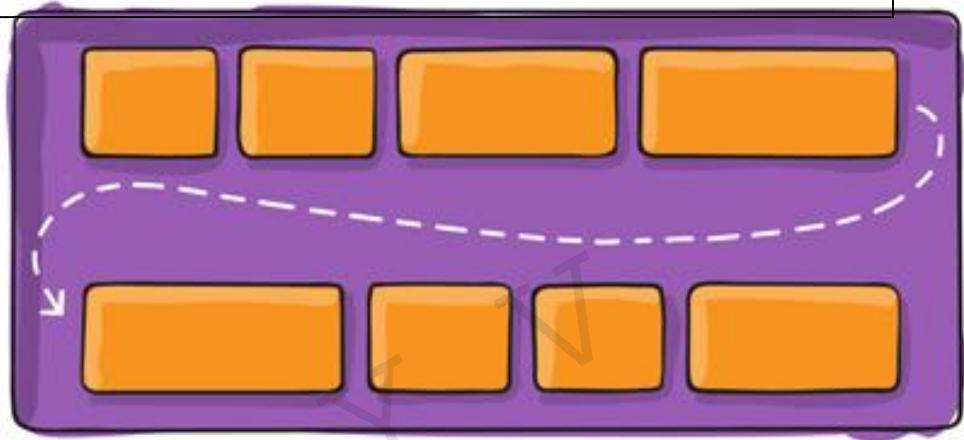
```
.container {  
  flex-direction: row | row-reverse | column |  
  column-reverse;  
}
```



Flexbox: các thuộc tính

- ✓ **flex-wrap:** cho phép items xuống dòng

```
.container{  
  flex-wrap: nowrap | wrap | wrap-reverse;  
}
```



- ✓ **flex-flow:** sử dụng để gộp chung hai thuộc tính flex-direction và flex-wrap

```
.flex-container {  
  display: flex;  
  flex-flow: row wrap;  
}
```

Flexbox: các thuộc tính

- ✓ **justify-content:** Điều chỉnh vị trí các items theo chiều ngang

```
.flex-container {  
  display: flex;  
  justify-content: flex-start;  
}
```

copy



```
.flex-container {  
  display: flex;  
  justify-content: flex-end;  
}
```

copy



```
.flex-container {  
  display: flex;  
  justify-content: center;  
}
```

copy



Flexbox: các thuộc tính

```
.flex-container {  
  display: flex;  
  justify-content: space-between;  
}
```

copy



```
.flex-container {  
  display: flex;  
  justify-content: space-evenly;  
}
```

copy



```
.flex-container {  
  display: flex;  
  justify-content: space-around;  
}
```

copy



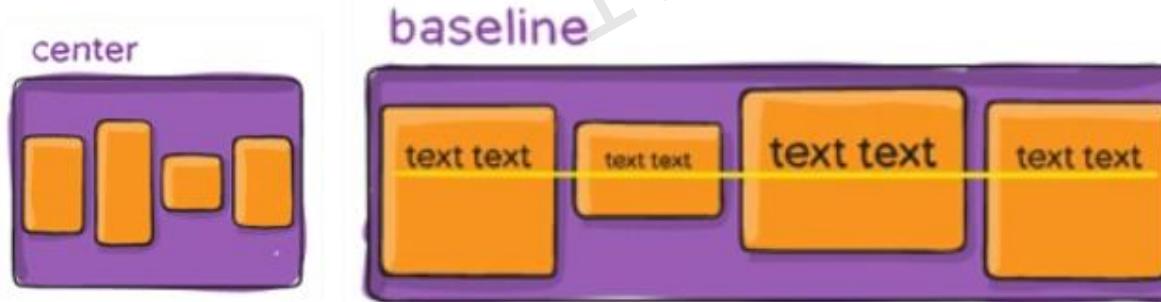
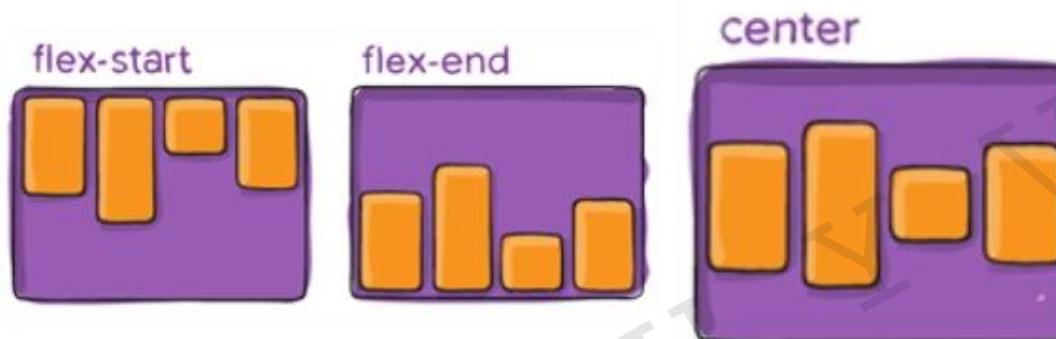
THIENHUY



Flexbox: các thuộc tính

- ✓ **align-items:** Canh chỉnh vị trí Items theo chiều đứng

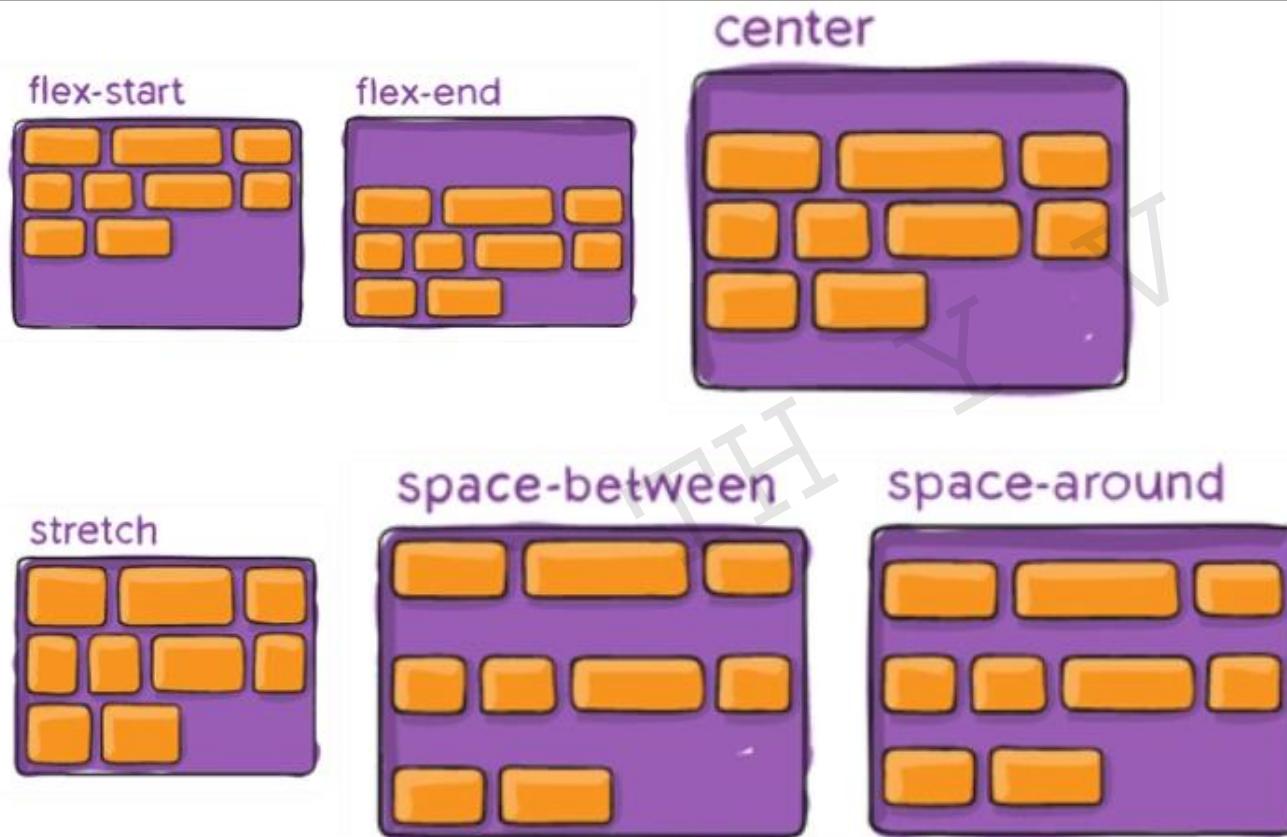
```
.container {  
  align-items: stretch | flex-start | flex-end  
  | center | baseline;  
}
```



Flexbox: các thuộc tính

- ✓ **align-content:** Canh chỉnh khoảng cách các items theo chiều đứng trong container

```
.container {  
    align-content: flex-start | flex-end | center | space-  
    between | space-around | stretch;  
}
```



Flexbox: các thuộc tính

- ✓ **order:** sắp xếp vị trí của các item

```
<div class="flex-container">  
  <div style="order: 3">1</div>  
  <div style="order: 2">2</div>  
  <div style="order: 4">3</div>  
  <div style="order: 1">4</div>  
</div>
```

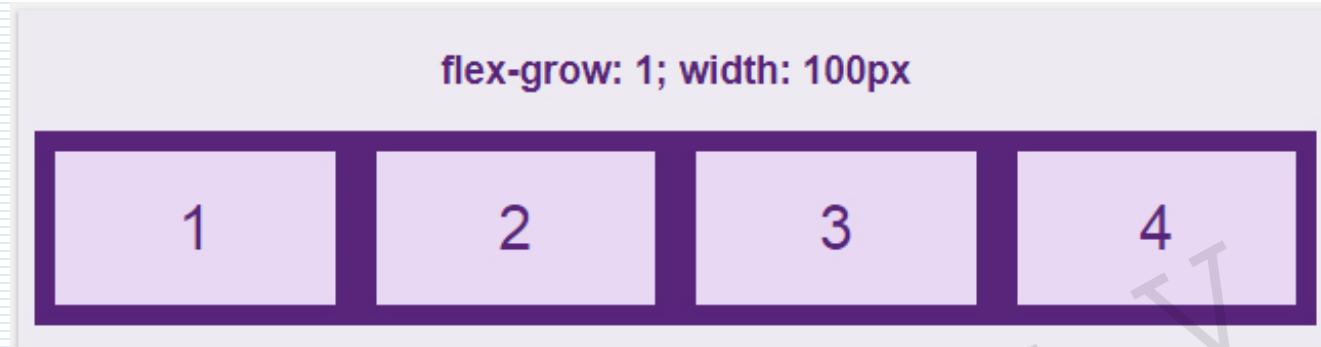
copy



Flexbox: các thuộc tính

✓ **flex-grow**: cho phép các phần tử giãn theo độ rộng của container.

- **flex-grow = 0**: Không giãn (mặc định)
- **flex-grow = 1**: giãn đều ra theo độ rộng container



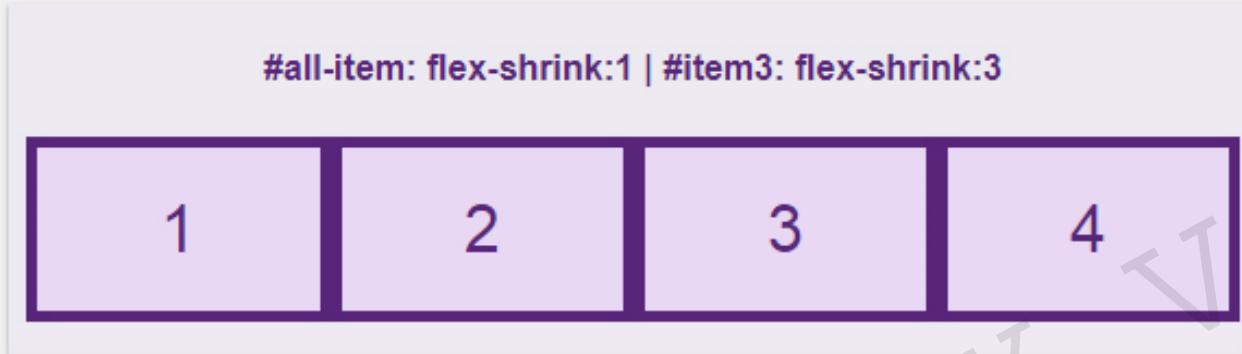
- **flex-grow = 2,3,4 ...** : giãn theo tỉ lệ tương ứng



Flexbox: các thuộc tính

✓ **flex-shrink**: cho phép các phần tử co lại theo độ rộng của container.

- **flex-shrink = 0**: Không co (mặc định)
- **flex-shrink = 1**: co lại đều theo theo độ rộng container



- **flex-shrink = 2,3,4 ...**: co lại theo tỉ lệ tương ứng



Flexbox: các thuộc tính

- ✓ **flex-basic:** sử dụng để xác định độ dài ban đầu của một item

```
.item {  
  flex-basis: <length> | auto; /* mặc định là auto */  
}
```

#item3: flex-basis: 250px



- ✓ **Short-hand:**

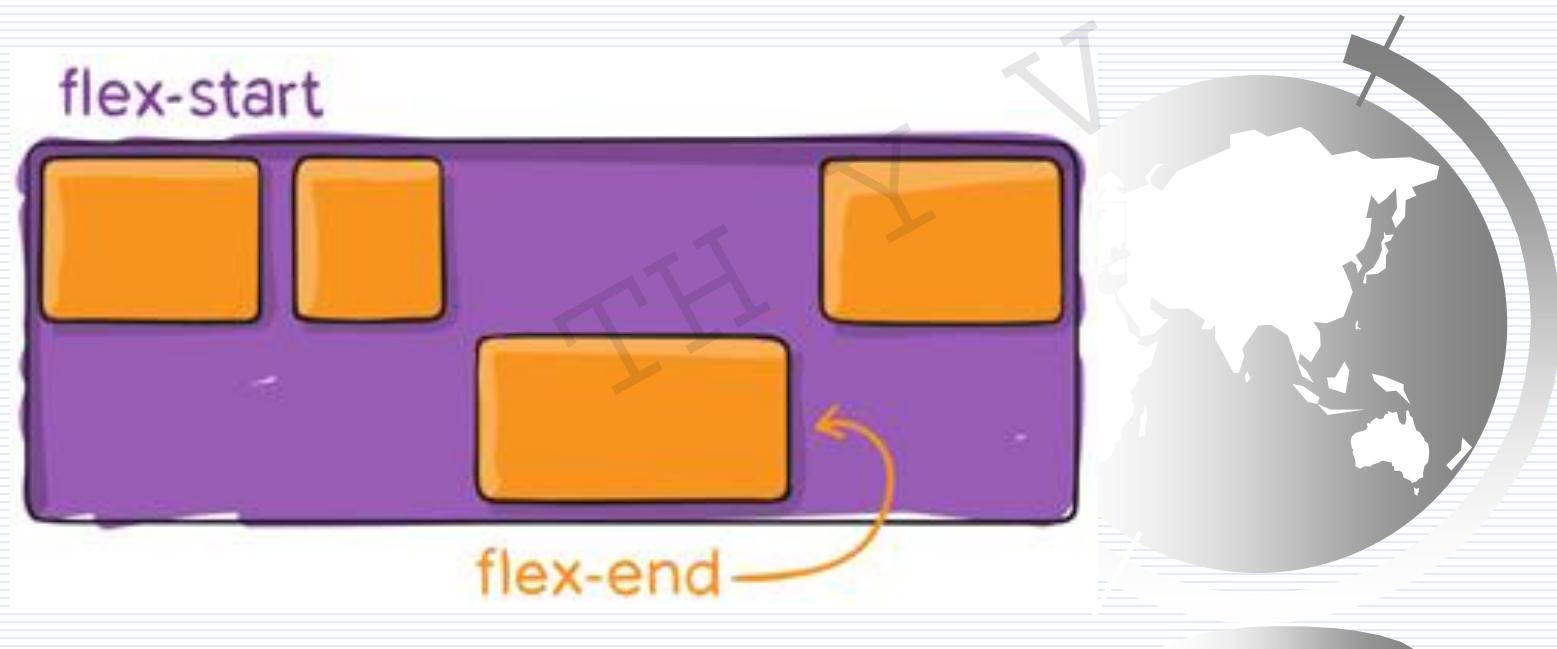
```
.item {  
  flex: <'flex-grow'> <'flex-shrink'> || <'flex-basis'>  
}
```

- ✓ **Ví dụ:**

flex: 1 3 250px;

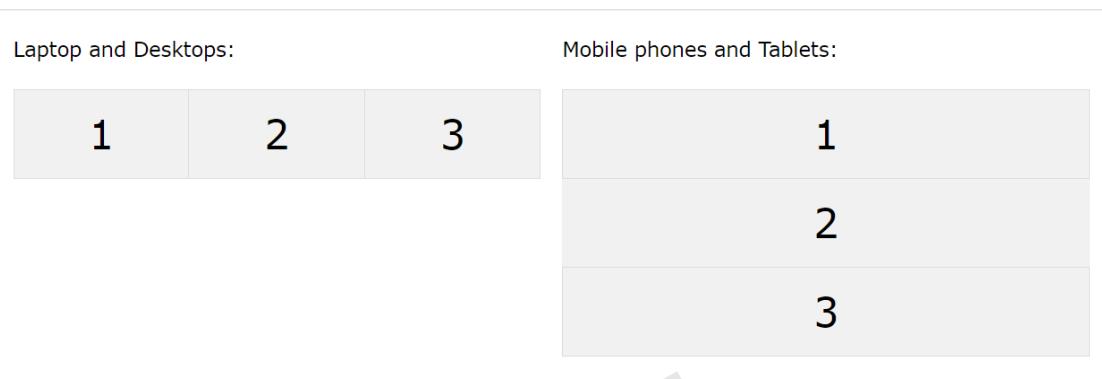
Flexbox: các thuộc tính

- ✓ **align-self:** có tác dụng tương tự như align-items của container nhưng **sử dụng riêng cho từng item**, ta có thể dùng nó để đặt lại vị trí cho một số item mà align-items đã quy định
- ✓ align-self cũng có các giá trị giống như align-items đó là: **flex-start, flex-end, center, stretch** và **baseline**



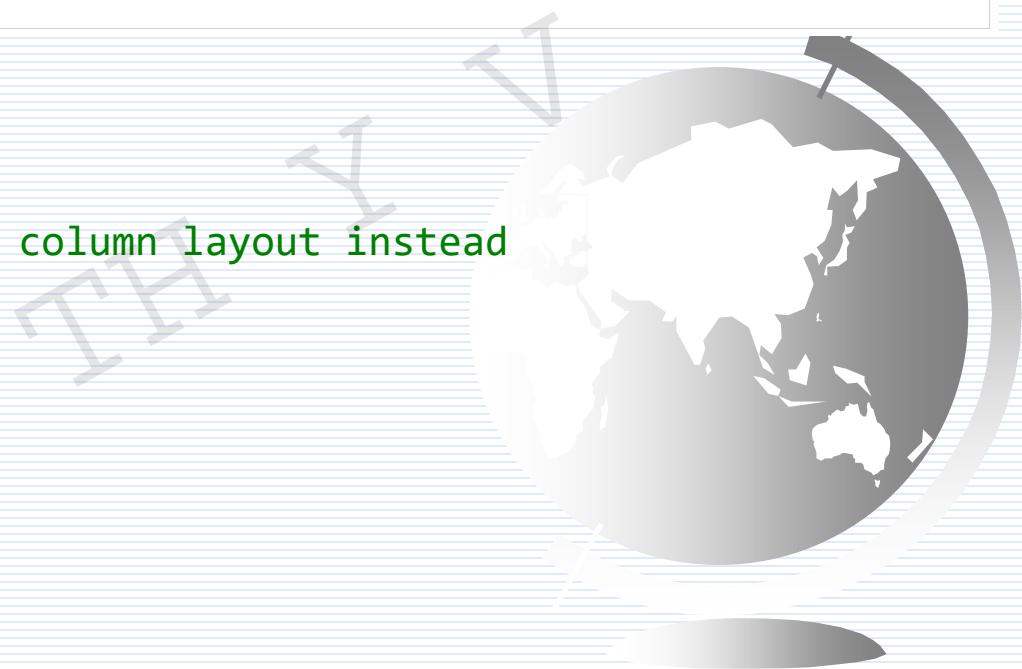
Responsive Flexbox

- ✓ Sử dụng **flex-direction: row/column**
- ✓ hoặc **flex-wrap: wrap**



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

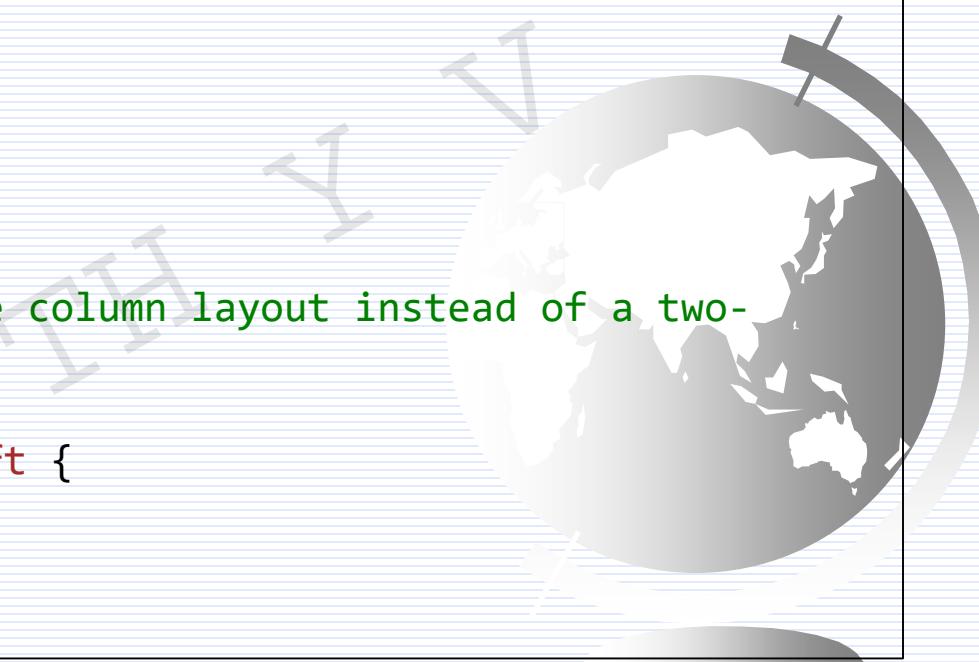
```
/* Responsive layout - makes a one column layout instead  
of a two-column layout */  
@media (max-width: 800px) {  
  .flex-container {  
    flex-direction: column;  
  }  
}
```



Responsive Flexbox

✓ hoặc **flex-wrap: wrap**

```
.flex-container {  
  display: flex;  
  flex-wrap: wrap;  
}  
  
.flex-item-left {  
  flex: 50%;  
}  
  
.flex-item-right {  
  flex: 50%;  
}  
  
/* Responsive layout - makes a one column layout instead of a two-  
column layout */  
@media (max-width: 800px) {  
  .flex-item-right, .flex-item-left {  
    flex: 100%;  
  }  
}
```



3. CSS position

✓ Các thuộc tính CSS position cho phép :

- Qui định vị trí của các Element. Ta có thể đặt vị trí xuất hiện ở bất cứ nơi nào tùy thích.
- Thậm chí đặt 1 element phía trên hay phía dưới 1 element khác.
- Hoặc qui định cách hiển thị khi nội dung element lớn vượt quá kích thước của element
- Thuộc tính Position có 4 kiểu : position-static, position-fixed, position-relative, position-absolute

Position-static

- **position:static** : là kiểu hiển thị mặc định của trình duyệt. Các element sẽ hiển thị theo kiểu bình thường. Element nào xuất hiện trước sẽ ở phía trên, rồi đến các element kế tiếp

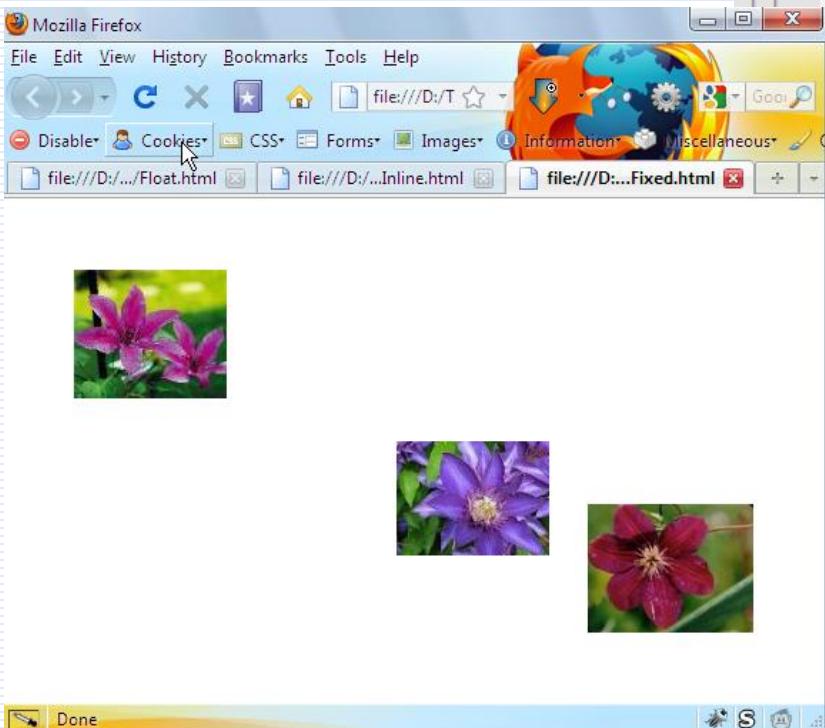


Fixed position

- **position:fixed;** => element sẽ có vị trí tương đối so với cửa sổ trình duyệt. Nó thậm chí không bị di chuyển khi cửa sổ bị cuộn
- ✓ Lúc này ta kết hợp với các thuộc tính **left, right, top, bottom** để qui định các element sẽ cách bên trái, phải, trên hoặc dưới của cửa sổ bao nhiêu pixel
 - **Lưu ý :** IE chỉ hỗ trợ fixed positioned khi HTML có thẻ **!DOCTYPE**
 - fixed element sẽ bị loại khỏi cách hiển thị thông thường. Trình duyệt sẽ hiển thị vị trí các element khác như không có fixed element

Demo fixed element

✓ PositionFixed.h



```
6   <style type="text/css">
7     #first{
8       position:fixed;
9       left:50px;
10      top:50px;
11    }
12
13   #second{
14     position:fixed;
15     top:48%;
16     left:48%;
17   }
18
19   #third{
20     position:fixed;
21     right:50px;
22     bottom:50px;
23   }
24
25 </style>
26 </head>
27
28
29
<body>



</body>
```

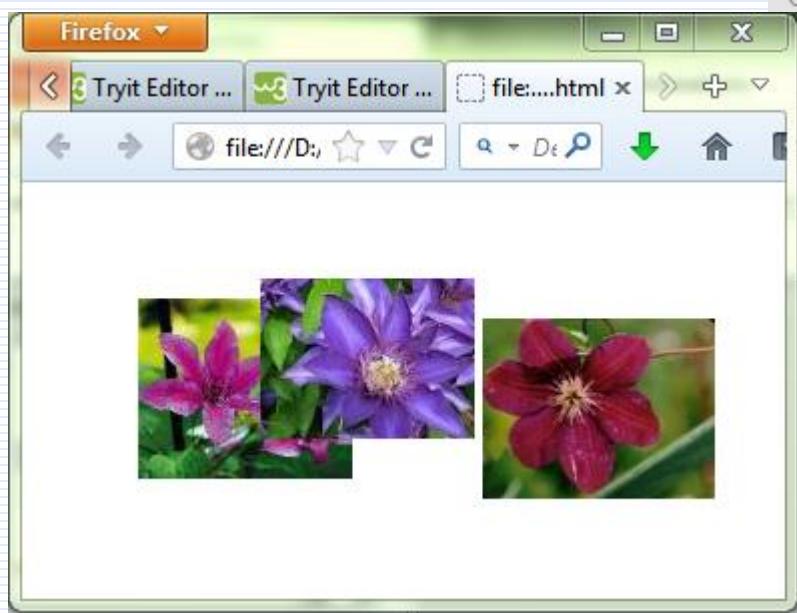
Relative position

- ✓ **position: relative;** => element sẽ có vị trí tương đối so với vị trí của nó lúc hiển thị bình thường (static)
- ✓ Nội dung của relative position có thể di chuyển đè lên các element khác (tùy thuộc vào thuộc tính left, right, top, bottom) nhưng khoảng không dành cho element này vẫn được giữ lại.



Demo relative element

✓ PositionRelative.html



```
6  <style type="text/css">
7      #first{
8          position: relative;
9          top: 50px;
10         left: 50px;
11     }
12     /* 
13      #second{
14          position: relative;
15          top: 30px;
16      }
17      #third{
18          position: relative;
19          top: 60px;
20      }
21  </style>
22  </head>
23
24  <body>
25      
26      
27      
28  </body>
```

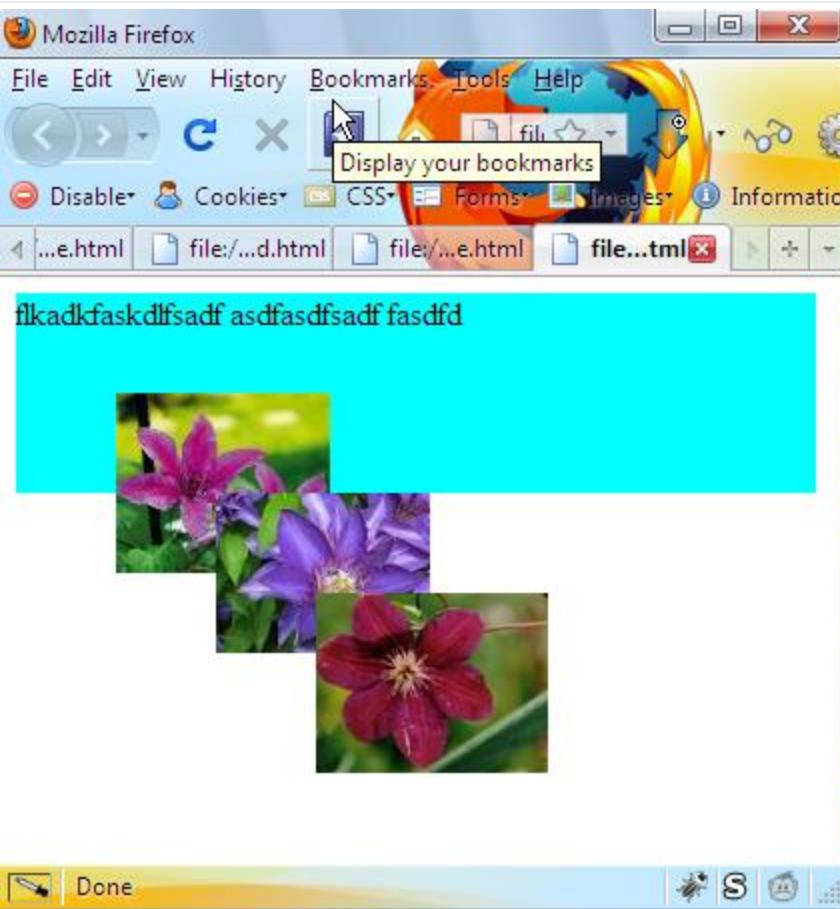
Absolute position

- ✓ **position: absolute;** => Element sẽ nhận giá trị tuyệt đối và có vị trí không đổi khi thu giảm hoặc tăng kích thước trình duyệt
- ✓ các thành phần được định vị không để lại bất cứ một khoảng trống nào trong tài liệu



Demo absolute position

✓ AbsolutePosition.html



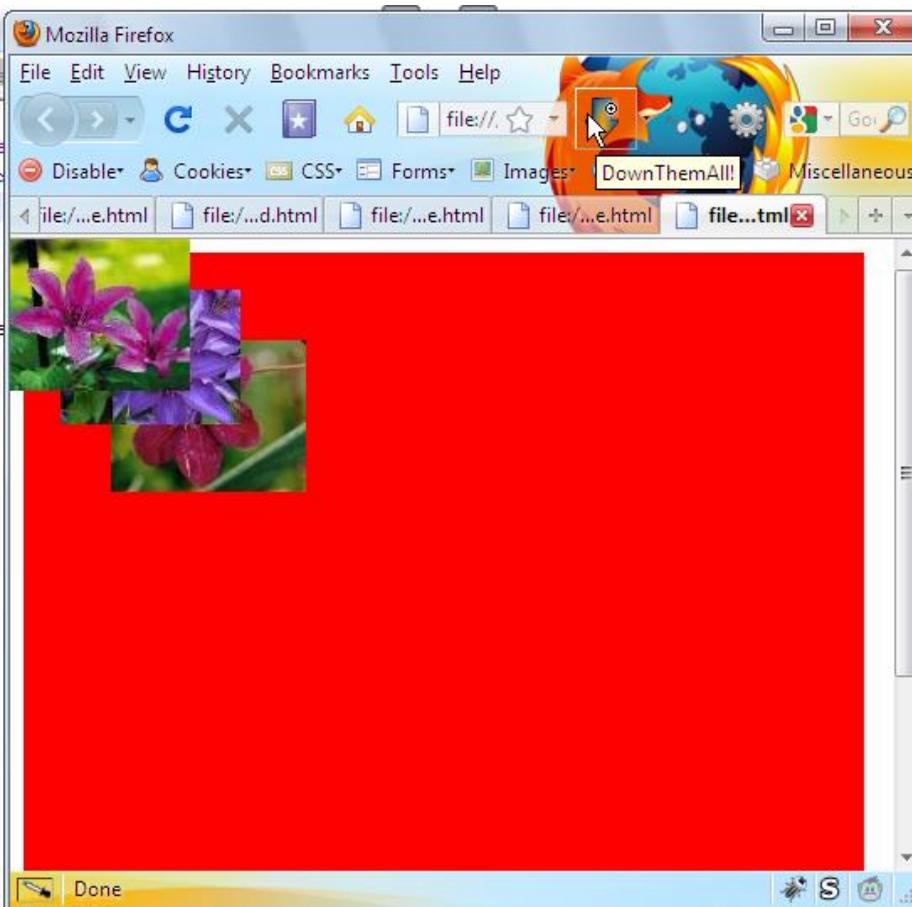
```
6  <style type="text/css">
7  div{
8      position: relative;
9      width: 400px;
10     height: 100px;
11     background-color: cyan;
12 }
13 #first{
14     position: absolute;
15     left: 50px;
16     top: 50px;
17 }
18 #second{
19     position: absolute;
20     left: 100px;
21     top: 100px;
22 }
23 #third{
24     position: absolute;
25     left: 150px;
26     top: 150px;
27 }
28 </style></head>
29 <body>
30 <div>
31     
32     
33     
34     flkadkfaskdlfsadfasdfsadffasdfd
```

Overlap element

- ✓ Nếu 2 element bị chồng lên nhau (overlap) khi hiển thị thì ta có thể sử dụng thuộc tính z-index để qui định element nào sẽ được nằm trên
- ✓ Giá trị z-index là 1 số nguyên (có thể âm)
- ✓ z-index càng cao thì sẽ nằm trên.



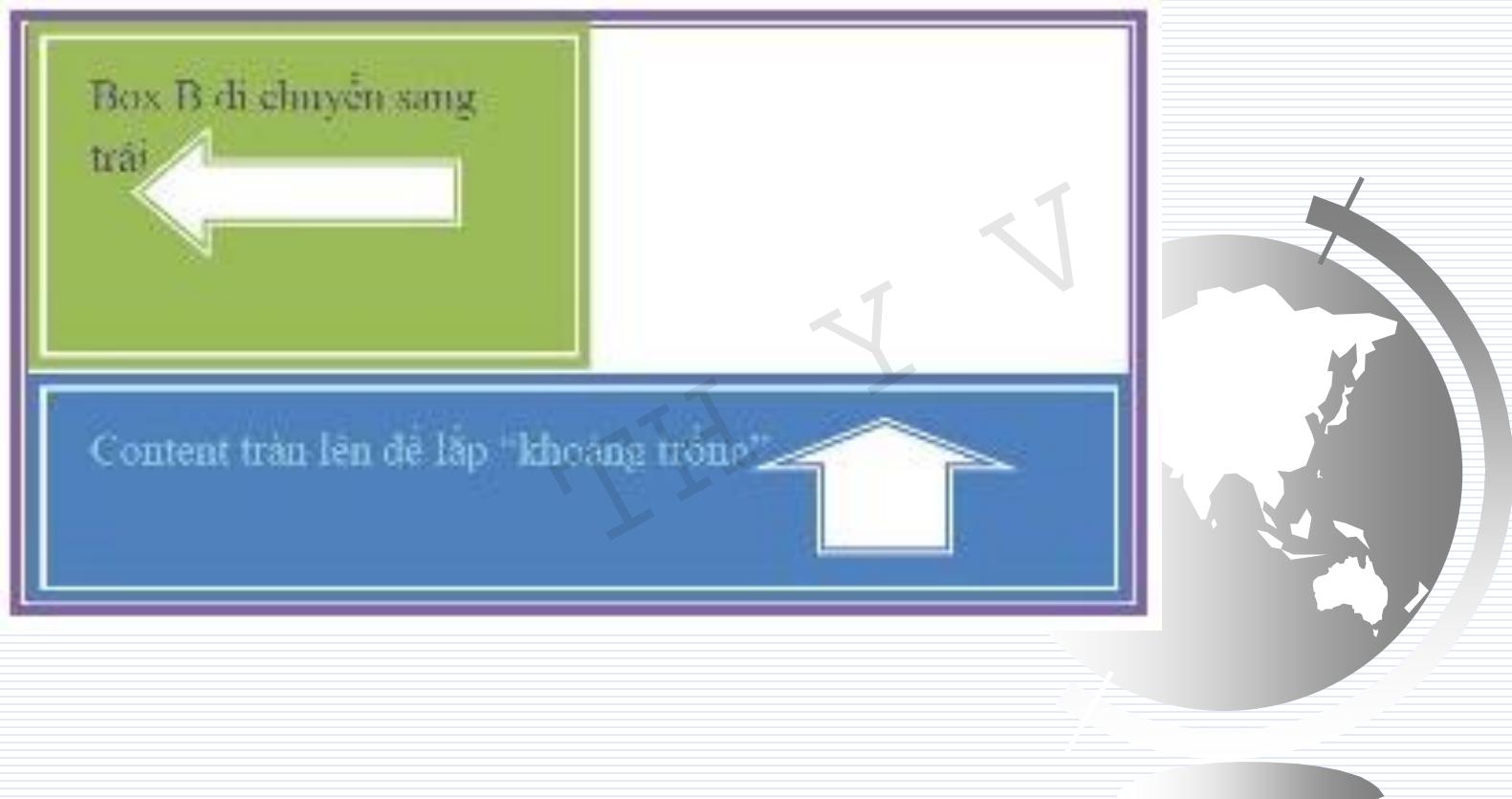
✓ PositionOverlap.htm



```
6  <style type="text/css">
7  div{
8      display: relative;
9      width: 500px;
10     height: 500px;
11     background-color: red;
12 }
13 #first{
14     position: absolute;
15     left: 0px;
16     top: 0px;
17     z-index: 3;
18 }
19 #second{
20     position: absolute;
21     left: 30px;
22     top: 30px;
23     z-index: 2;
24 }
25 #third{
26     position: absolute;
27     left: 60px;
28     top: 60px;
29     z-index: 1;
30 }
31 </style></head>
32 <body>
33 <div>
34 
35 
36 
37 </div>
38 </body>
```

4. CSS Float

- ✓ Với CSS float, 1 element có thể trôi về bên trái hoặc bên phải để cho các element khác có thể nằm bao quanh nó



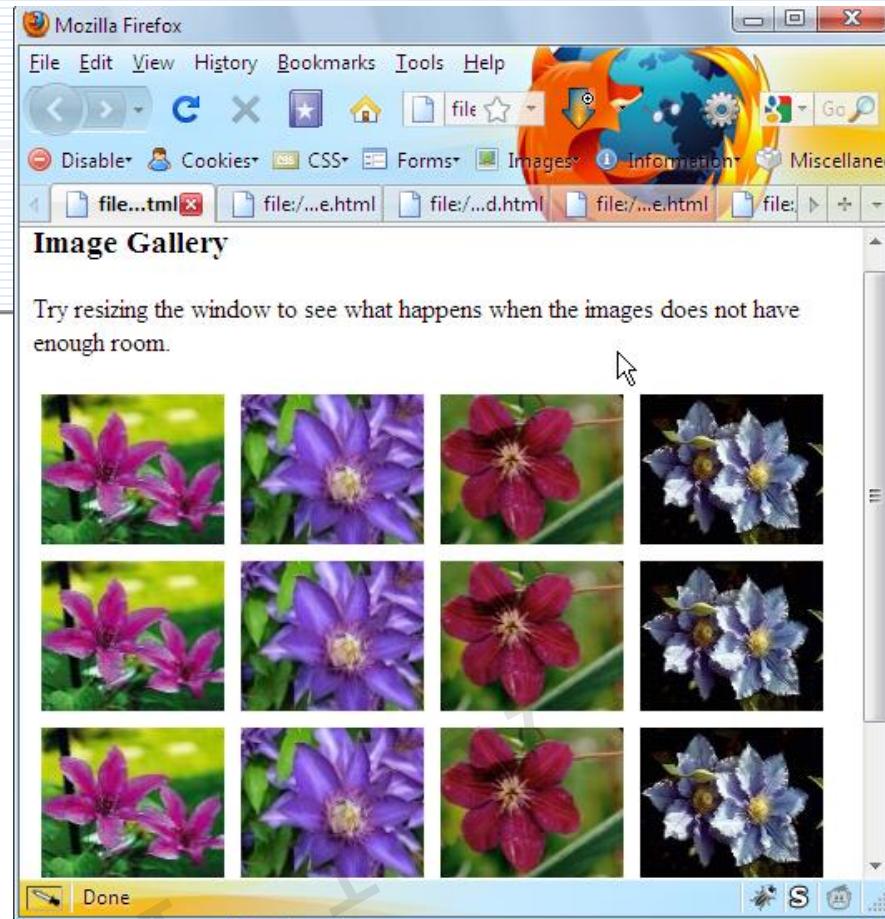
CSS Float – Đặc điểm

- ✓ CSS float rất hữu ích khi hiển thị ảnh hoặc làm layout
- ✓ 1 float element sẽ trôi về bên trái hoặc phải cho đến khi đụng biên hoặc 1 element khác
- ✓ Các element phía sau 1 float element sẽ nằm bao quanh nó
- ✓ Các element phía trước 1 float element sẽ không bị ảnh hưởng



✓ Float.html

```
1  <html>
2  <head>
3  <style type="text/css">
4  .thumbnail
5  {   I
6  float:left;
7  width:110px;
8  height:90px;
9  margin:5px;
10 }
11 </style>
12 </head>
13 <body>
14 <h3>Image Gallery</h3>
15 <p>Try resizing the window to see what happens
16 when the images does not have enough room.</p>
17 
18 
19 
20 
21 
```



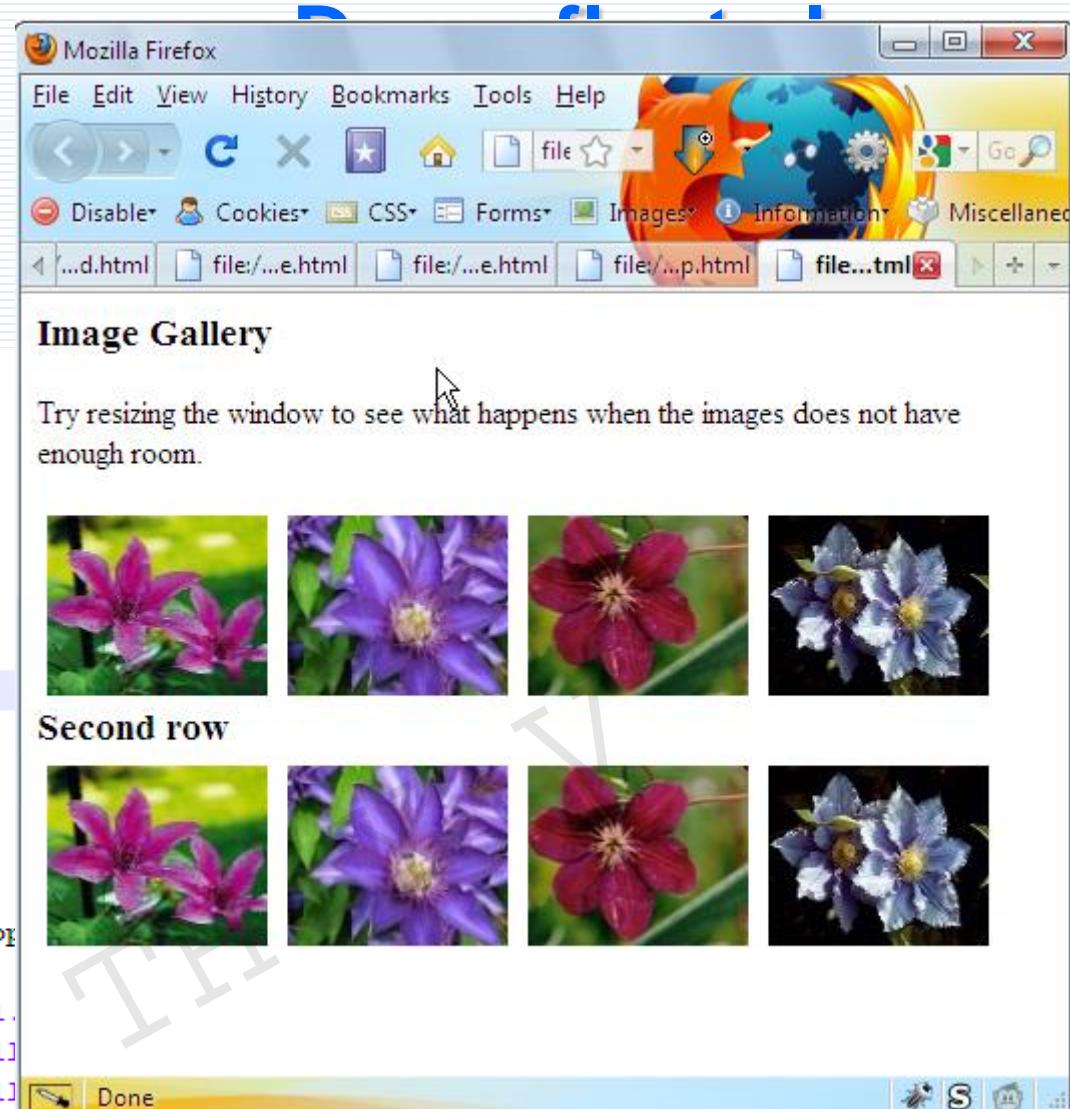
Tắt float

- ✓ Các element phía sau sẽ tự động bao quanh float element. Để tránh chuyện này xảy ra ta có thể thêm thuộc tính clear
 - clear:left; => không cho float bên trái
 - clear:right; => không cho float bên phải
 - clear:both; => không cho float cả 2 bên



✓ FloatClear.html

```
3 <style type="text/css">
4   .thumbnail
5   {   float:left;
6     width:110px;
7     height:90px;
8     margin:5px;
9   }
10  .text_line
11  {   clear:both;
12    margin-bottom:2px;
13  }
14 </style>
15 </head><body>
16 <h3>Image Gallery</h3>
17 <p>Try resizing the window to see what happens when the images does not have enough room.</p>
18 
19 
20 
21 
22 <h3 class="text_line">Second row</h3>
23 
24 
25 
26 
```



Làm 1 gallery

Album Ảnh 1 - 20 trên tổng số 23 | Trở lại Album ảnh của Van Tran

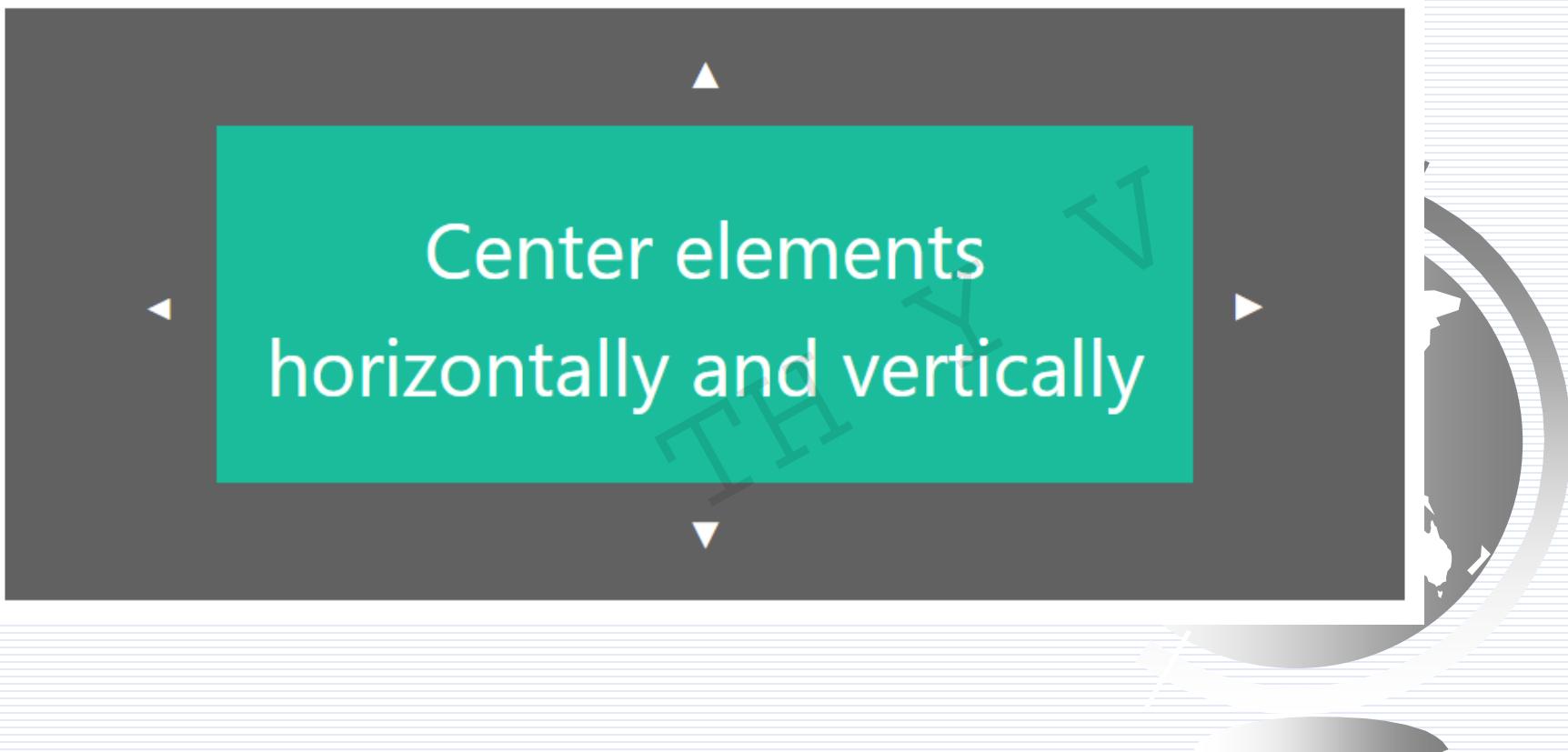
12 Tiếp

5



5. Align

- ✓ Horizontally : chiều ngang
- ✓ Vertically : chiều đứng



Horizontally Align

✓ Canh chữ : text-align

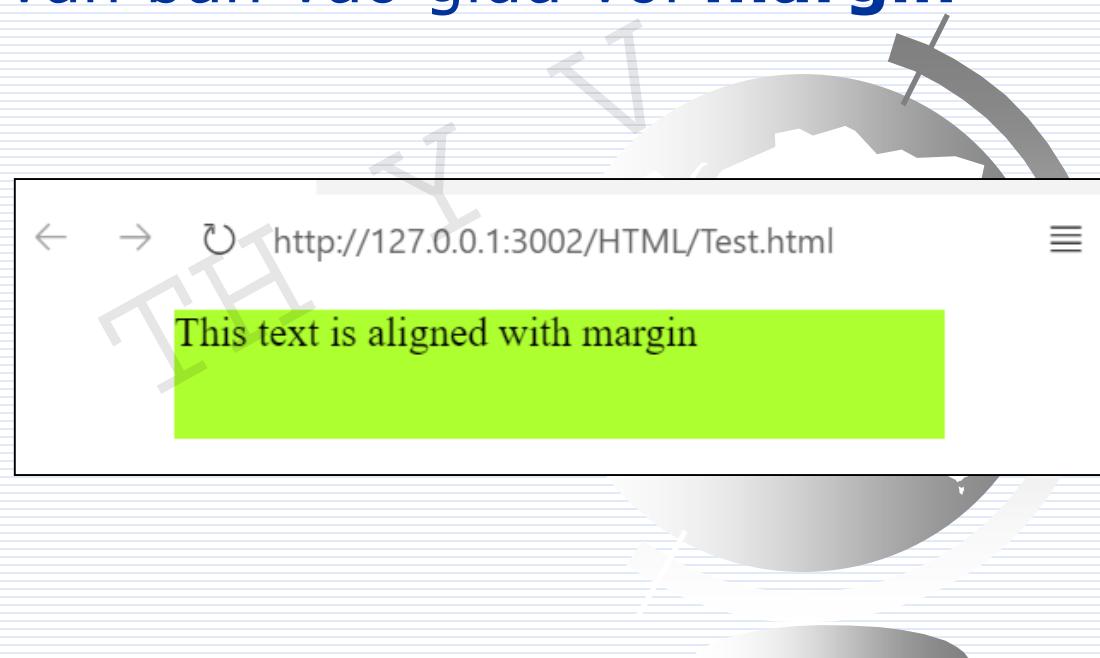
✓ Canh 1 block :

- margin

- Position

✓ Ví dụ: Canh 1 đoạn văn bản vào giữa với **margin**

```
p{  
    width: 300px;  
    height: 50px;  
    background-  
    color: greenyellow;  
    margin: auto;  
}
```

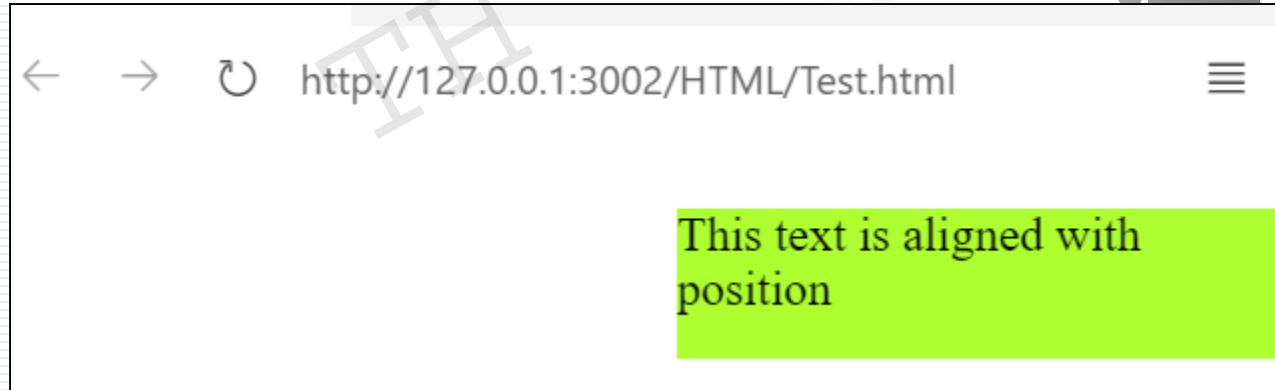


Horizontally Align

✓ Sử dụng **Position**:

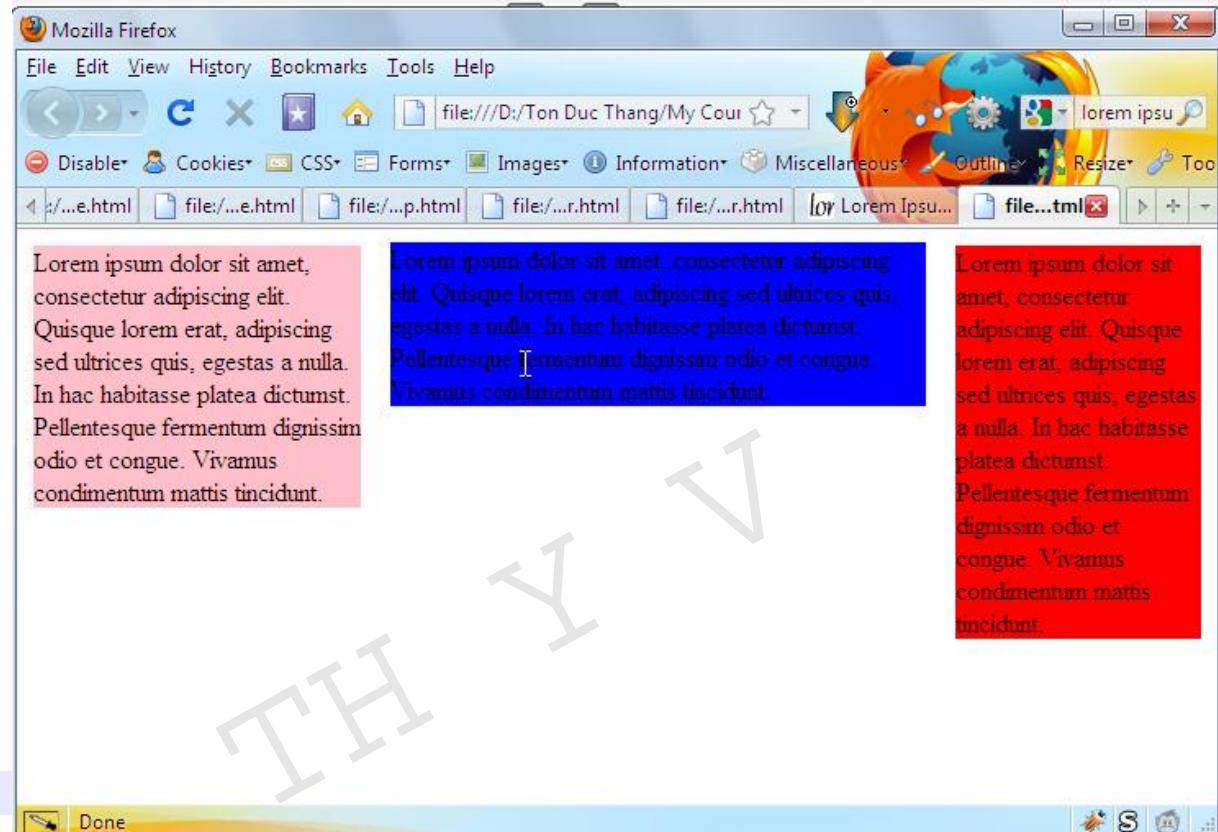
- Canh trái : Đặt position: absolute; left: 0px
- Canh phải : Đặt position: absolute; right: 0px

```
p{  
    width: 200px;  
    height: 50px;  
    background-color: greenyellow;  
    position: absolute;  
    right: 0px;  
}
```



Demo align block left,right,center

```
6 <style type="text/css">
7
8     margin:0px;
9     padding:0px;
10
11 .left{
12     position:absolute;
13     top:10px;
14     left:10px;
15     width:200px;
16     margin:0px auto;
17     background-color:pink;
18 }
19 .center{
20     margin-left:220px;
21     margin-right:170px;
22     background-color:blue;
23 }
24 .right{
25     position:absolute;
26     top:10px;
27     right:10px;
28     width:150px;
29     background-color:red;
30 }
31 </style></head>
```



Vertically Align

✓ Canh chữ : line-height

```
p{  
    width: 200px;  
    height: 50px;  
    background-color: red;  
    line-height: 50px;  
}
```

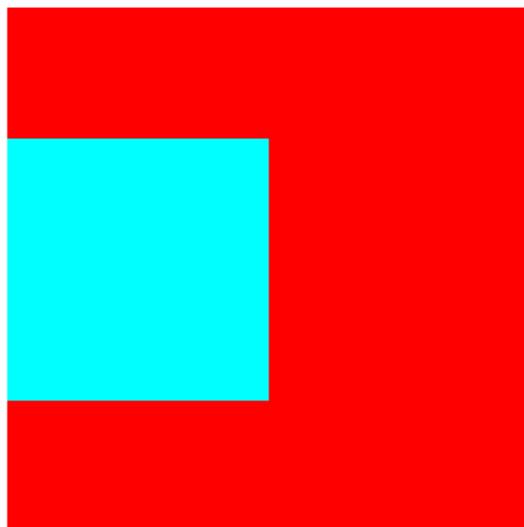
This text is aligned



Vertically Align

- ✓ Canh 1 block :
 - padding
 - flexbox

← → ⏪ http://127.0.0.1:300



```
<style>
.vertical{
    width: 200px;
    height: 150px;
    background-color: red;
    padding-top: 50px;
}
.inside{
    width: 100px;
    height: 100px;
    background-color: aqua;
}
</style>
</head>
<body>
<div class="vertical">
    <div class="inside"></div>
</div>
</body>
```

6. Image sprite

- ✓ Khi 1 trang web sử dụng nhiều ảnh thì cần nhiều yêu cầu gửi tới server để xử lý => làm sao để giảm số lần gửi yêu cầu ảnh về server???
- ✓ Image sprite : Là cách nhóm các ảnh đơn lại với nhau để tạo thành 1 ảnh duy nhất.
- ✓ Khi đó chỉ cần 1 yêu cầu là có thể đáp ứng được yêu cầu

Demo image sprite

✓ ImageSprite.html



7. Opacity

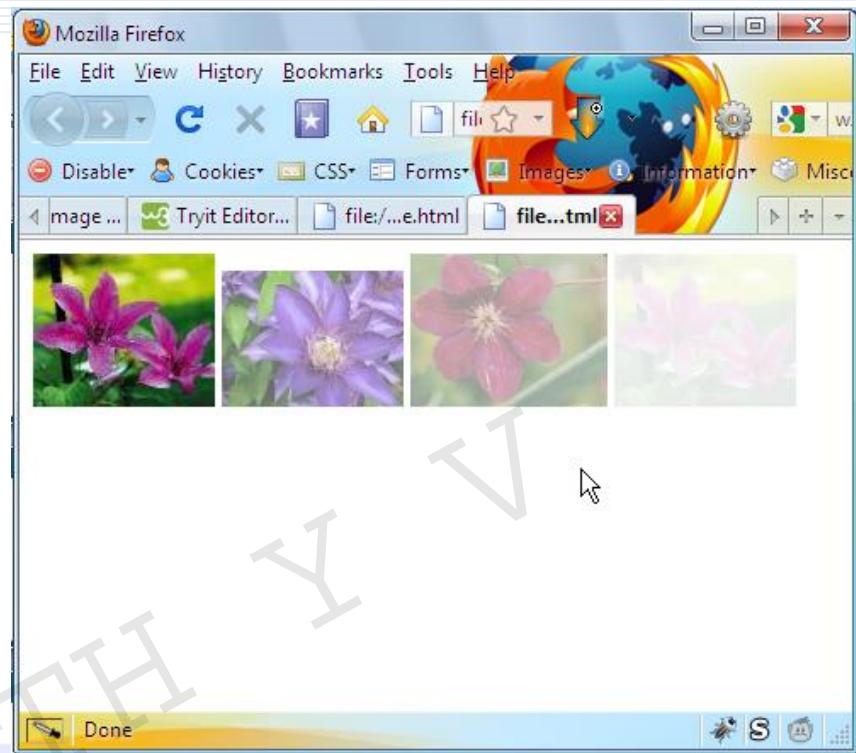
- ✓ Ta có thể làm mờ đi 1 hình ảnh bằng cách sử dụng thuộc tính :
 - opacity :1.0; (cho FF)
 - filter:alpha(opacity=100); (cho IE)
- ✓ 1.0 (100) : Không trong suốt
- ✓ 0.0 (0) : Trong suốt hoàn toàn (không thể nhìn thấy)

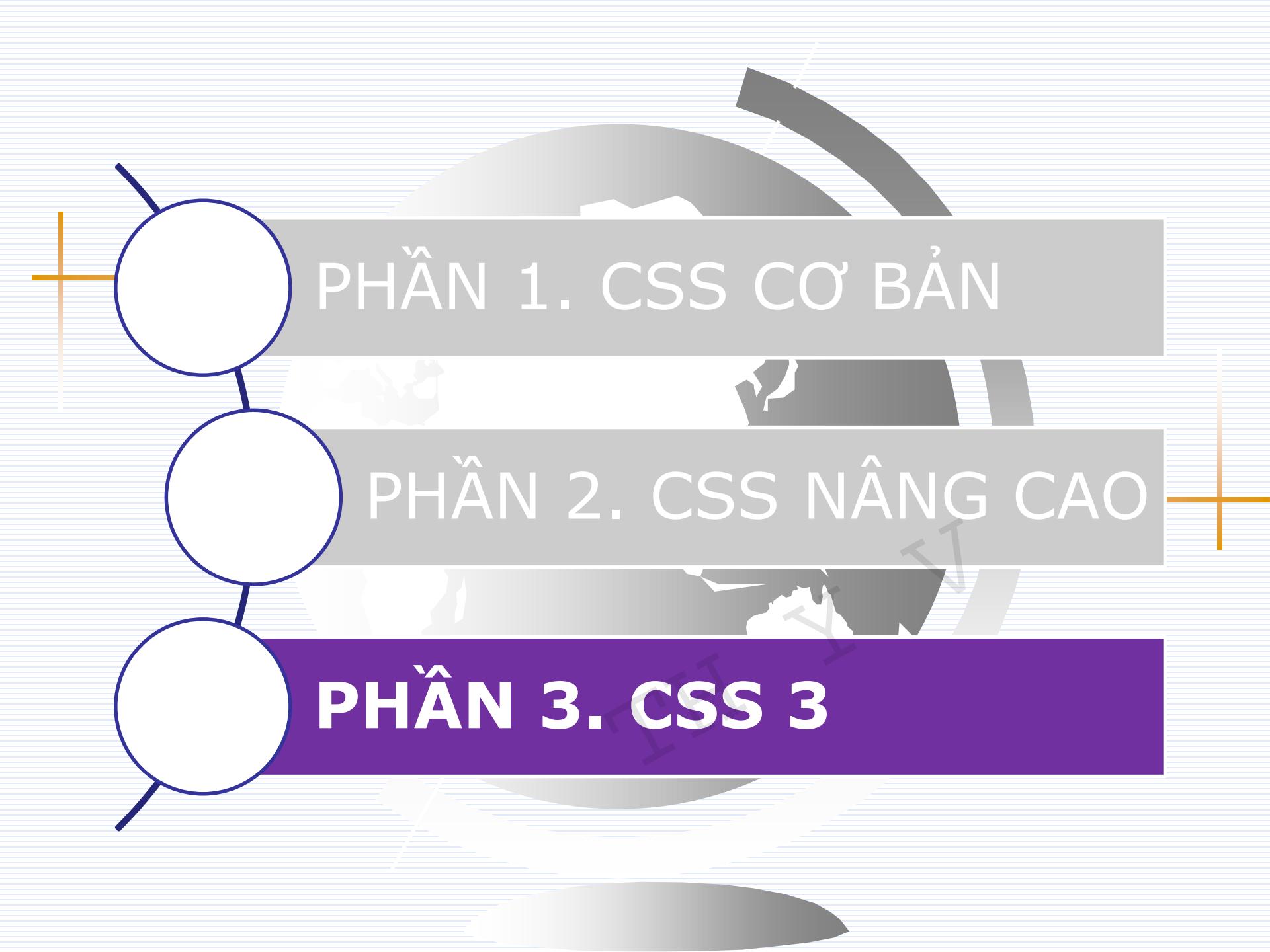


Demo image opacity

✓ ImageOpacity

```
6 <style type="text/css">
7 #first{
8     opacity:1.0;
9     filter:alpha(opacity=100);
10 }
11 #second{
12     opacity:0.6;
13     filter:alpha(opacity=60);
14 }
15 #third{
16     opacity:0.4;
17     filter:alpha(opacity=40);
18 }
19 #forth{
20     opacity:0.1;
21     filter:alpha(opacity=10);
22 }
23 </style></head>
```





PHẦN 1. CSS CƠ BẢN

PHẦN 2. CSS NÂNG CAO

PHẦN 3. CSS 3

Giới thiệu CSS 3

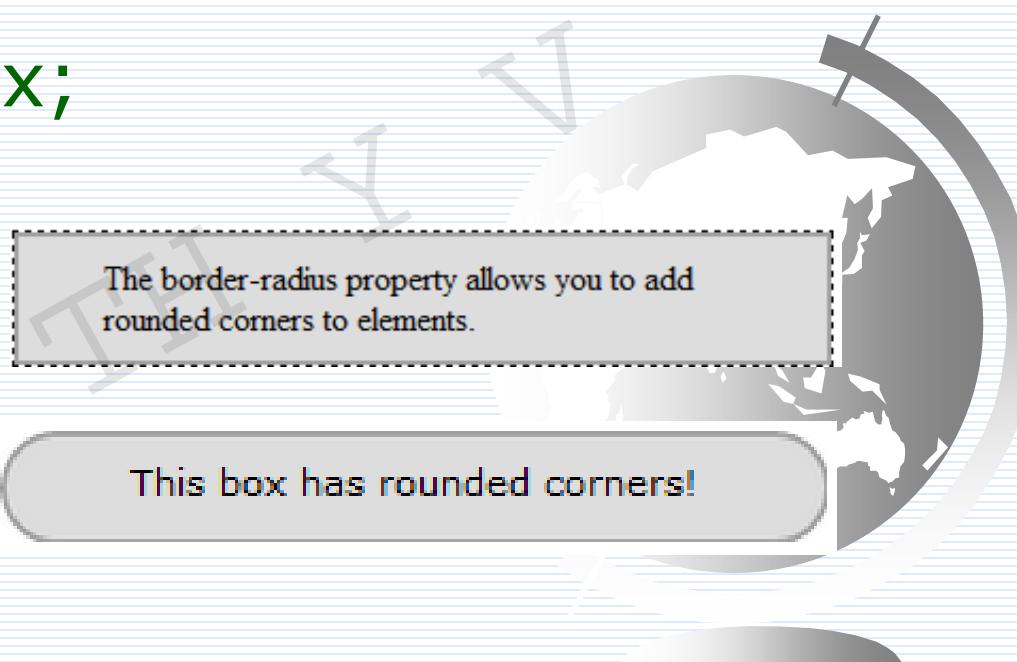
- ✓ CSS3 is split up into "modules". The old specification has been split into smaller pieces, and new ones are also added.
- ✓ Some of the most important CSS3 modules are:
 - Selectors
 - Box Model
 - Backgrounds and Borders
 - Text Effects
 - 2D/3D Transformations
 - Animations
 - Multiple Column Layout
 - User Interface



CSS3 Rounded Corners

- ✓ Thuộc tính border-radius dùng để bo đường viền
- ✓ Ví dụ

```
div  
{  
border:2px solid;  
border-radius:25px;  
}
```



The border-radius property allows you to add rounded corners to elements.

This box has rounded corners!

CSS3 border-image

- ✓ Thuộc tính border-image dùng ảnh để làm đường viền cho đối tượng.
- ✓ Ví dụ :

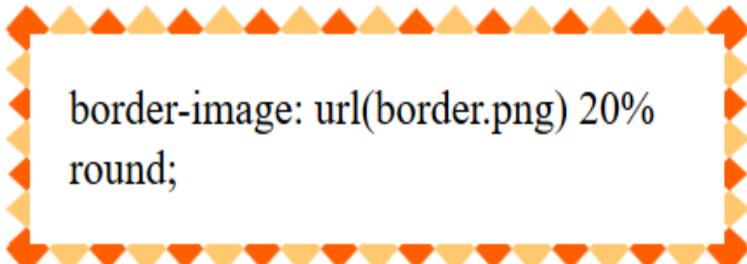
border-image: url(border.png) 30 round;



border.png

border-image: url(border.png) 50
round;

border-image: url(border.png) 50 round



border-image: url(border.png) 20%
round;

border-image: url(border.png) 20 round



CSS3 Multiple Backgrounds

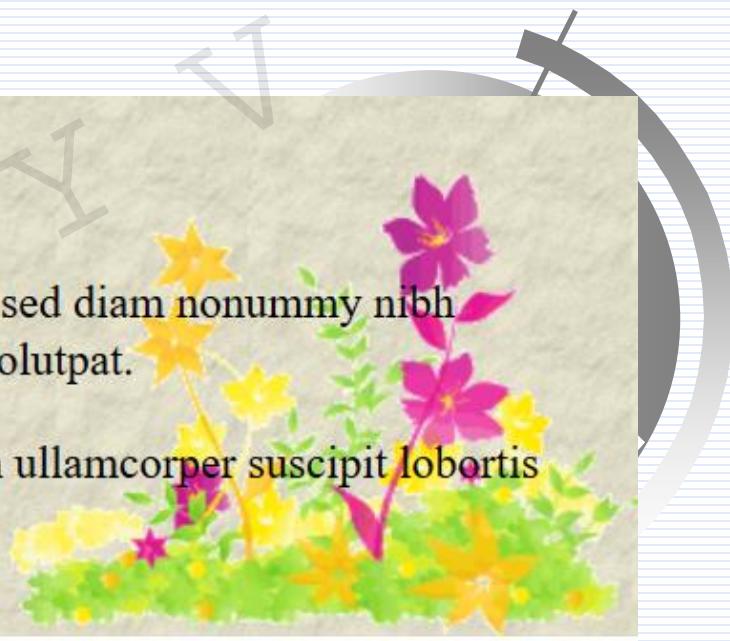
- ✓ Cho phép thêm nhiều hình nền vào 1 đối tượng.
 - ✓ Ví dụ :

```
#example1 {  
background-image: url(img_flwr.gif), url(paper.gif);  
background-position: right bottom, left top;  
background-repeat: no-repeat, repeat;  
}
```

Lorem Ipsum Dolor

LOREM IPSUM dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.



CSS3 Multiple Backgrounds (tt)

Một số thuộc tính mới:

✓ **Background-Size:** Kích thước ảnh nền

✓ **background-origin**

- border-box : bắt đầu từ viền góc trái trên
- padding-box: bắt đầu từ góc trái trên trừ padding ra
- content-box: gói gọn trong vùng content

```
#example1 {  
    border: 10px solid black;  
    padding: 35px;  
    background: url(img_flwr.gif);  
    background-repeat: no-repeat;  
    background-origin: content-box;  
}
```

✓ [Demo: CSS3/background Origin](#)



CSS3 Gradients

✓ Cú pháp:

[option]-gradient(*direction*, *color1*, *color*, ...);

✓ Ví dụ:

background: linear-gradient(red, yellow);

✓ [option]

- **Linear:** tô theo đường thẳng
- **Repeating-linear:** tô theo đường thẳng và lặp lại
- **Radial:** tô theo dạng đường tròn
- **Repeating-radial:** tô theo đường tròn và lặp lại

✓ Direction:

- **To right, to left, to bottom right, ...**
- **-360 độ đến 360 độ: -90deg, 180deg, ...**



CSS3 Gradients

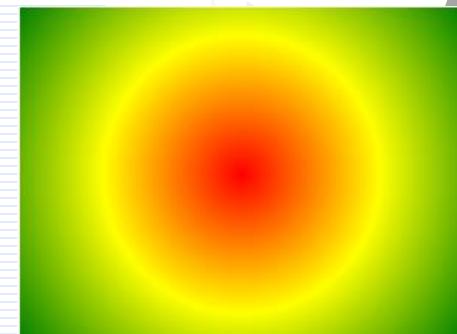
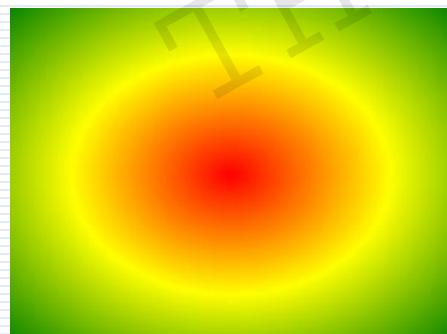
- ✓ `background: linear-gradient(to right, red, yellow);`



- ✓ `background: repeating-linear-gradient(red, yellow 10%, green 20%);`



- ✓ `background: radial-gradient(eclipse[circle], red, yellow, green);`



- ✓ [Demo](#)

CSS3 Shadow

- ✓ Text-shadow dùng để tạo hiệu ứng cho chữ
- ✓ Box-shadow: tạo hiệu ứng cho đối tượng
- ✓ Cú pháp:
`text-shadow: horizontal, vertical, blur, color;`
- ✓ `h1 { text-shadow: 5px 5px 5px red; }`

Text-shadow effect!

Text-shadow effect!

- ✓ Multiple shadow

`text-shadow: 0 0 3px #FF0000, 0 0 5px #0000FF;`

Text shadow effect!



CSS3 Shadow

- ✓ Box-shadow: tạo hiệu ứng cho đối tượng
- ✓ Cú pháp:

box-shadow: horizontal, vertical, blur, color;

- ✓ ví dụ:

box-shadow: 5px 5px 5px grey;

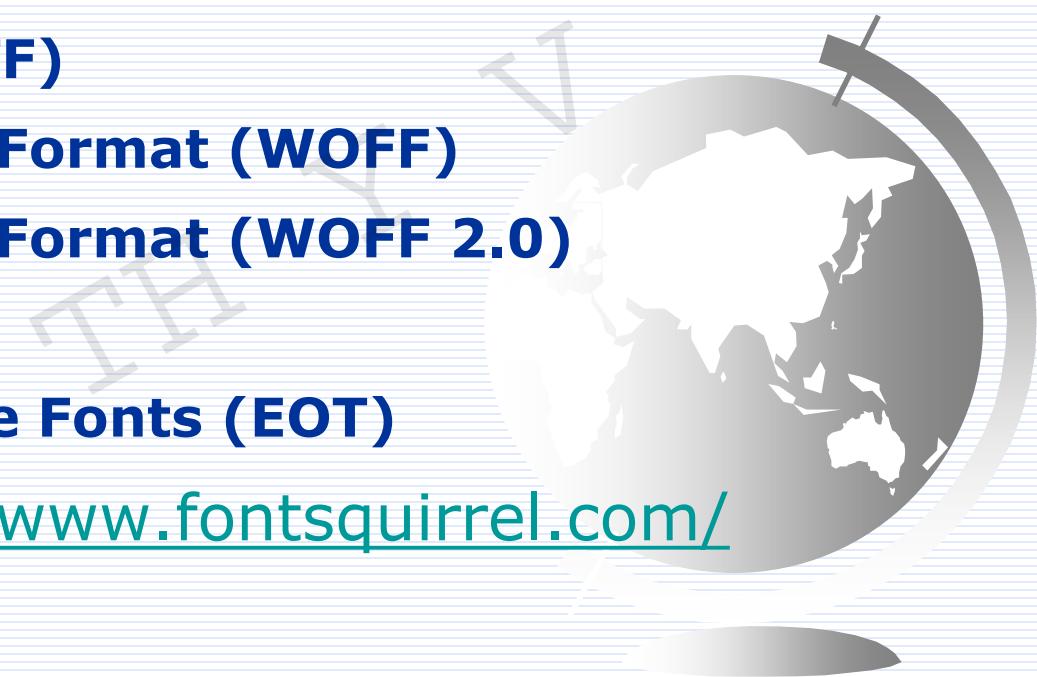
This is a yellow <div> element
with a blurred, grey box-shadow

Demo: [CSS3/shadow.html](#)



CSS3 Web Font

- ✓ Cho phép lập trình viên sử dụng các font chữ không có sẵn trong máy tính
- ✓ Định nghĩa luật : **@font-face**
- ✓ Các định dạng font được hỗ trợ:
 - **TrueType Fonts (TTF)**
 - **OpenType Fonts (OTF)**
 - **The Web Open Font Format (WOFF)**
 - **The Web Open Font Format (WOFF 2.0)**
 - **SVG Fonts/Shapes**
 - **Embedded OpenType Fonts (EOT)**
- ✓ Link tải font: <https://www.fontsquirrel.com/>



CSS3 Web Font

✓ Ví dụ :

```
@font-face {  
    font-family: myFirstFont;  
    src: url(sansation_light.woff);  
}  
  
h1 {  
    font-family: myFirstFont;  
}
```

The @font-face Rule



CSS3 2D, 3D Transforms

- ❑ translate(): di chuyển
- ❑ rotate(): xoay
 - ❑ rotateX(): xoay 3D quay trục X
 - ❑ rotateY(): xoay 3D quay trục Y
 - ❑ rotateZ(): xoay 3D quay trục Z
- ❑ scale(): tăng giảm kích thước
- ❑ skewX(): kéo dãn theo phương ngang
- ❑ skewY(): kéo dãn theo phương đứng
- ❑ matrix(): dùng để gộp tất cả phương thức trên vào 1 lệnh.
- ❑ Demo: [CSS3/transform.html](#)



CSS3 Transitions: Speed Curve

✓ **transition-timing-function** mô tả tốc độ thực hiện chuyển đổi

- Ease: tốc độ chậm – nhanh – chậm (mặc định)
- Linear: tốc độ cố định
- ease-in : bắt đầu chậm, sau nhanh
- ease-out: bắt đầu bình thường, sau chậm
- ease-in-out : đầu và cuối chậm, giữa nhanh
- cubic-bezier(n,n,n,n): tự định tốc độ thực hiện

✓ Demo: [CSS3/transition.html](#)



CSS3 Animation

- ❑ Dùng để thay đổi trạng thái, di chuyển đối tượng
 - ❑ Bước 1: định nghĩa thay đổi bằng **@keyframes name**
 - ❑ Bước 2: gán thay đổi cho đối tượng (**animation-name**) và thời gian thực hiện (**animation-duration**)

- ❑ Ví dụ :

```
/* The animation code */
@keyframes example {
  from {background-color: red;}
  to {background-color: yellow;}
}
/* The element to apply the animation to */
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}
```



CSS3 Animation

❑ Một số thuộc tính khác:

- ❑ **animation-delay**: độ trễ
- ❑ **animation-iteration-count**: số lần lặp lại. Giá trị **Infinite** là lặp lại mãi mãi
- ❑ **animation-direction**: hướng di chuyển.
 - ❑ Mặc định là thuận chiều kim đồng hồ.
 - ❑ **reverse**: ngược chiều kim đồng hồ
 - ❑ **Alternate**: Trước là thuận, sau là ngược
- ❑ **animation-timing-function**: chỉ định tốc độ di chuyển (giống như transition): ease, linear, ease-in, ease-out, ...
- ❑ **animation-fill-mode**: Trạng thái sau khi chuyển đổi:
none|forwards|backwards|both|initial|inherit;
- ❑ **Short hand**: **animation**: example 5s linear 2s infinite alternate;
- ❑ **Demo**: <CSS3/Animation.html>

CSS3 Image

- ❑ Một số thuộc tính mới:

- ❑ **round**: bo tròn ảnh

- ❑ **filter**: lọc ảnh

- ❑ Blur(px): làm mờ ảnh.

- ❑ Brightness(%): độ sáng

- ❑ Contrast(%): độ tương phản

- ❑ grayscale(%): chuyển thành trắng đen

- ❑ ...

- ❑ Responsive: tự thay đổi kích thước khi màn hình thay đổi

- ❑ Kết hợp với Javascript tạo hiệu ứng bung ảnh

- ❑ Demo: [CSS3/image.html](#), [CSS3/image_model.html](#)

CSS3 Media Queriese

- Dùng để làm responsive website
 - @media screen and (min-width: 480px)

```
{      //code CSS  
}
```

: Chỉ định màn hình với kích thước 480px

- Demo: [CSS3/MediaQueries.html](#)



CSS3 Create Multiple Columns

- ✓ Thuộc tính column-count dùng để tạo nhiều cột trên văn bản
 - ✓ Ví dụ :

```
div
{
  -moz-column-count:3; /* Firefox */
  -webkit-column-count:3; /* Safari and Chrome */
  column-count:3;
}
```

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

Lorem ipsum dolor sit
 amet, consectetuer
 adipiscing elit, sed
 diam nonummy nibh
 euismod tincidunt ut

ad minim veniam, quis nostrud exercit ation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in

laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation

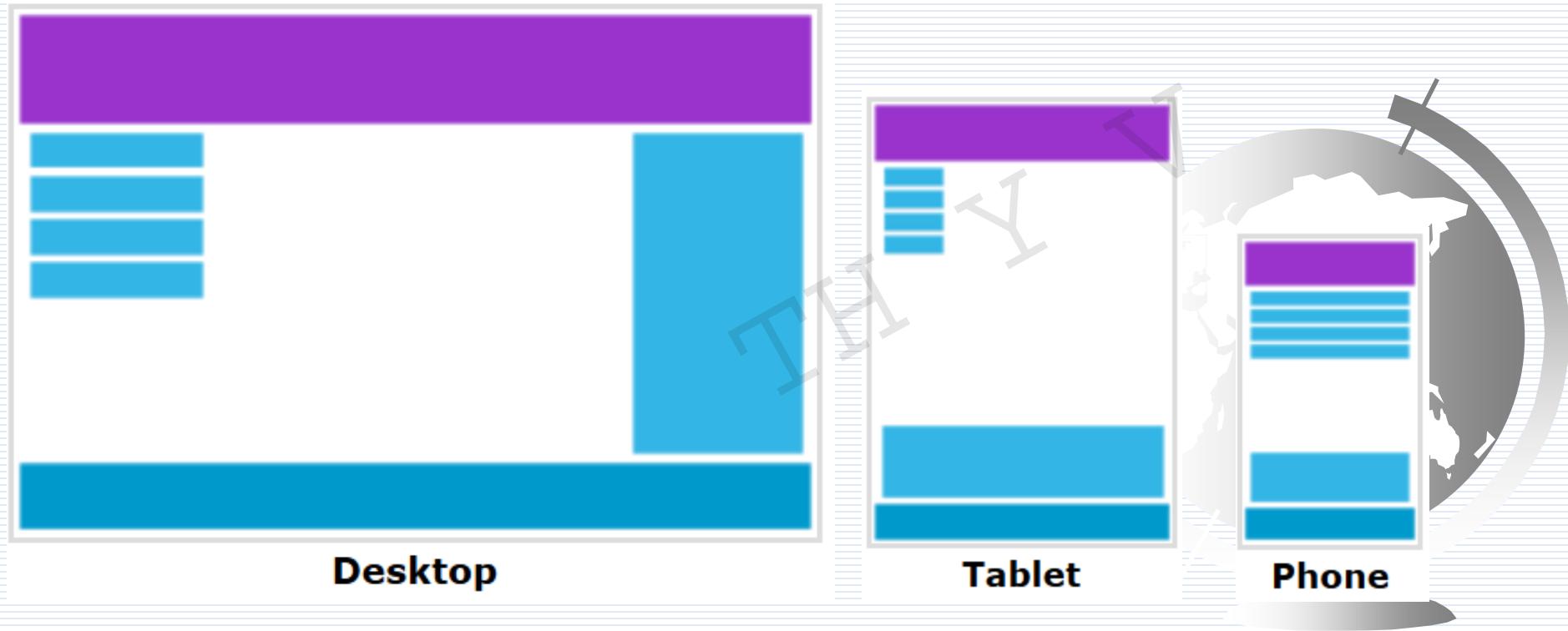
ullamcorper suscipit
lobortis nisl ut aliquip
ex ea commodo
consequat. Duis autem
vel eum iriure dolor in

RESPONSIVE WEB DESIGN



Giới thiệu

- ✓ Responsive web design sử dụng HTML và CSS để thiết kế website tương thích với mọi thiết bị như: Desktop, tablet, phones.



Khái niệm Viewport

- ✓ Viewport được định nghĩa là khung nhìn của người dùng, tùy vào từng thiết bị thì khung nhìn sẽ lớn hoặc nhỏ (hoặc ta cũng có thể gọi là màn hình thiết bị).
- ✓ Bằng việc thiết lập viewport, website sẽ co dãn kích thước vừa đúng với lại kích thước của thiết bị



Without the viewport meta tag



With the viewport meta tag

Khái niệm Viewport

✓ Khai báo:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Trong đó:

- **width=device-width**: Thiết lập độ rộng trang bằng với kích thước màn hình thiết bị
- **initial-scale=1.0**: Chỉ định việc phóng to/ thu nhỏ

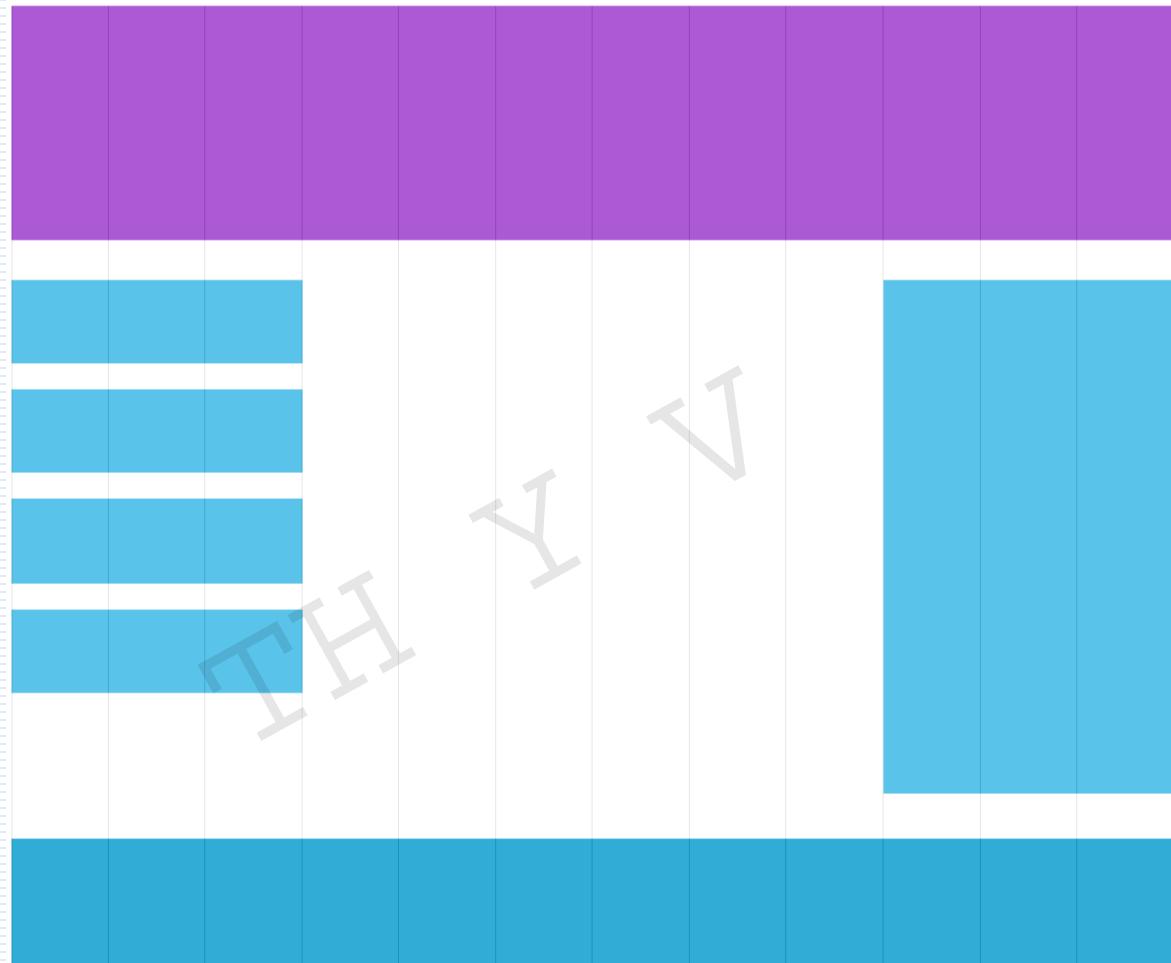
✓ Chú ý:

- ✓ Không được thiết lập kích thước 1 phần tử nào đó lớn hơn kích thước màn hình -> tạo thanh cuộn chiều ngang
- ✓ Không viết cho 1 kích thước cố định nào cả
- ✓ Dùng Css Media để tùy chỉnh website vừa với kích thước từng loại thiết bị

Gridview

- ✓ Để trang web linh động trong việc thay đổi kích thước, chúng ta sẽ chia web ra thành 12 cột, gọi là lưới (grid)

```
.col-1 {width: 8.33%;}  
.col-2 {width: 16.66%;}  
.col-3 {width: 25%;}  
.col-4 {width: 33.33%;}  
.col-5 {width: 41.66%;}  
.col-6 {width: 50%;}  
.col-7 {width: 58.33%;}  
.col-8 {width: 66.66%;}  
.col-9 {width: 75%;}  
.col-10 {width: 83.33%;}  
.col-11 {width: 91.66%;}  
.col-12 {width: 100%;}
```



Gridview

- ✓ Padding và border nằm trong chiều rộng đối tượng

```
{   box-sizing: border-box; }
```

Thiết lập cột trôi về trái

```
[class*="col-"] {  
  float: left;  
  padding: 15px;  
  border: 1px solid red;  
}
```

Dòng sau đó không trôi

```
.row::after {  
  content: "";  
  clear: both;  
  display: block;  
}
```



Kết hợp media queries

- ✓ Padding và border nằm trong chiều rộng đối tượng

```
{ box-sizing: border-box; }
```

Thiết lập cột trôi về trái

```
[class*="col-"] {  
  float: left;  
  padding: 15px;  
  border: 1px solid red;  
}
```

Dòng sau đó không trôi

```
.row::after {  
  content: "";  
  clear: both;  
  display: block;  
}
```

Thiết lập độ rộng 100% cho các cột khi màn hình nhỏ hơn 480px

```
@media only screen and (min-width: 480px) {
```

```
[class*="col-"] {  
  width: 100%;  
}
```

Demo: ResponsiveWeb.html

Chania

The Flight

The City

The Island

The Food

The City

Chania is the capital of the Chania region on the island of Crete. The city can be divided in two parts, the old town and the modern city.

What?

Chania is a city on the island of Crete.

Where?

Crete is a Greek island in the Mediterranean Sea.

How?

You can reach Chania airport from all over Europe.

Resize the browser window to see how the content respond to the resizing.