



## THIẾT KẾ LAYOUT CHO TRANG WEB - 3

1. MỘT SỐ THUỘC TÍNH CSS CẦN THIẾT ĐƯỢC SỬ DỤNG
2. THIẾT KẾ CÁC PHẦN CHO WEBSITE



- TẠO MỘT KHUNG CHỨA CÓ ID LÀ CONTAINER VÀ DÀN TRANG HẾT DIỆN TÍCH TRÌNH DUYỆT.

```
<div id="container">
```

```
</div>
```

- THUỘC TÍNH CSS TƯƠNG ƯNG

```
body{margin:0px;}
```

```
#container{width:100%;}
```



- BÊN TRONG KHỐI CONTAINER, LẦN LƯỢT CHIA 5 KHỐI DIV: BANNER, SLIDE, CONTENT, BOTTOM VÀ COPYRIGHT.

```
<div id="container">
```

```
  <div id="banner"></div>
```

```
  <div id="slide"></div>
```

```
  <div id="content"></div>
```

```
  <div id="bottom"></div>
```

```
  <div id="copyright"></div>
```

```
</div>
```



- KHỐI BANNER  
CAO 150PX
- KHỐI SLIDE  
CAO 260PX
- KHỐI CONTENT  
CAO 600PX
- KHỐI BOTTOM  
CAO 368PX
- KHỐI COPYRIGHT  
CAO 156PX



- BÊN TRONG MỖI KHỐI BANNER, SLIDE, CONTENT VÀ BOTTOM, TA ĐẶT MỘT KHỐI CÓ ĐỘ RỘNG 944PX ĐỨNG CHÍNH GIỮA.

Ví dụ cho khối banner:

```
<div id="banner">
```

```
<div></div>
```

```
</div>
```

# KẾT QUẢ CHIA LAYOUT





- KHỐI BANNER  
CAO 150PX
- KHỐI SLIDE  
CAO 260PX
- KHỐI CONTENT  
CAO 600PX
- KHỐI BOTTOM  
CAO 368PX
- KHỐI COPYRIGHT  
CAO 156PX



# MỘT SỐ THUỘC TÍNH CSS CẦN THIẾT



# Thuộc tính margin

margin: 5px; margin:4px 5px; margin:1px 2px 3px 4px;



```
1 <html xmlns="http://www.w3.org/1999/xhtml">
2 <head>
3 </head>
4 <body>
5     <div class="box1"></div>
6     <div class="box1"></div>
7
8     <div class="box2"></div>
9     <div class="box2"></div>
10    <style>
11        body{margin:0px;}
12        .box1{
13            width:100px;
14            height:100px;
15            background-color:red;
16            margin:20px;
17        }
18        .box2{
19            width:100px;
20            height:100px;
21            background-color:blue;
22            margin:20px;
23            float:left;
24        }
25    </style>
26 </body>
27 </html>
```

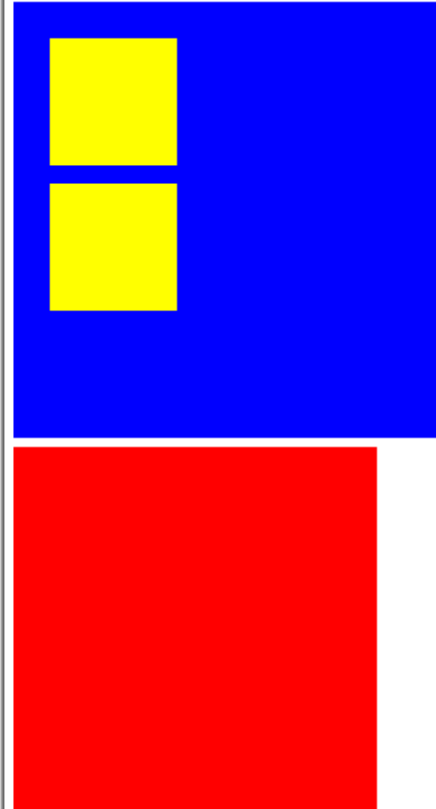


# Thuộc tính padding

padding: 5px; padding :4px 5px; padding :1px 2px 3px 4px;



```
1 <html xmlns="http://www.w3.org/1999/xhtml">
2 <head>
3 </head>
4 <body>
5   <div id="box1">
6     <div class="inbox"></div>
7     <div class="inbox"></div>
8   </div>
9
10  <div id="box2"></div>
11  <style>
12    body{margin:0px;}
13    #box1, #box2{
14      width:200px;
15      height:200px;
16      background-color:red;
17      margin:5px;
18    }
19    #box1{
20      padding:20px;
21      background-color:blue !important;
22    }
23    .inbox{
24      width:70px;
25      height:70px;
26      margin-bottom:10px;
27      background-color:yellow;
28    }
29  </style>
30 </body>
31 </html>
```



**Chú ý**: Thuộc tính padding sẽ làm tăng kích thước của khối.

# Thuộc tính position

Cách tính vị trí của khối: static (\*), relative, absolute, fixed.



- Giá trị static: là mặc định. Không tương thích với top và left (là hai thuộc tính xác định khoảng cách phía trên và khoảng cách từ mép trái).
- Giá trị relative: các khối sẽ tính tọa độ tương đối so với các khối ở ngay trước nó.
- Giá trị absolute: các khối sẽ tính tọa độ tuyệt đối theo khung chứa của nó (có thể là body, hoặc các thẻ div trực tiếp chứa nó).
- Giá trị fixed: các khối sẽ tính tọa độ tuyệt đối đối của khung nhìn trên trang (thường sẽ cho kết quả không thay đổi khi ta kéo thanh trượt).

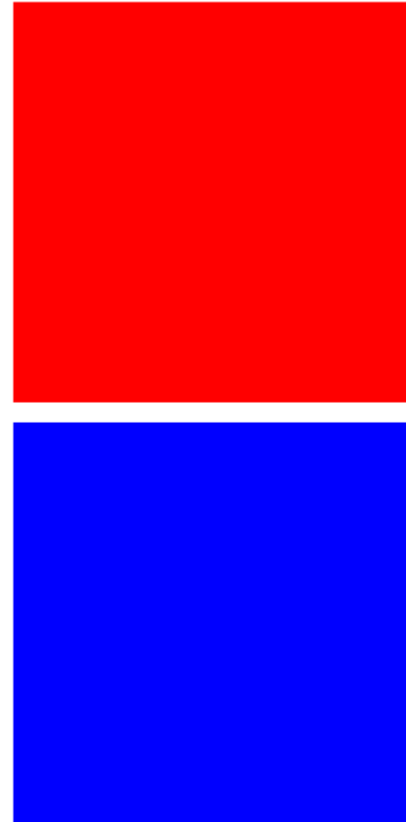
# Thuộc tính position

Cách tính vị trí của khối: static (\*), relative, absolute, fixed.



## Ví dụ về position:static;

```
1 <html xmlns="http://www.w3.org/1999/xhtml">
2 <head>
3 </head>
4 <body>
5   <div id="box1"></div>
6
7   <div id="box2"></div>
8   <style>
9     body{margin:0px;}
10    #box1, #box2{
11      width:200px;
12      height:200px;
13      background-color:red;
14      margin:10px;
15    }
16    #box2{
17      background-color:blue !important;
18    }
19    #box1{
20      position:static;
21      top:50px;
22      left:10px;
23    }
24  </style>
25 </body>
26 </html>
```



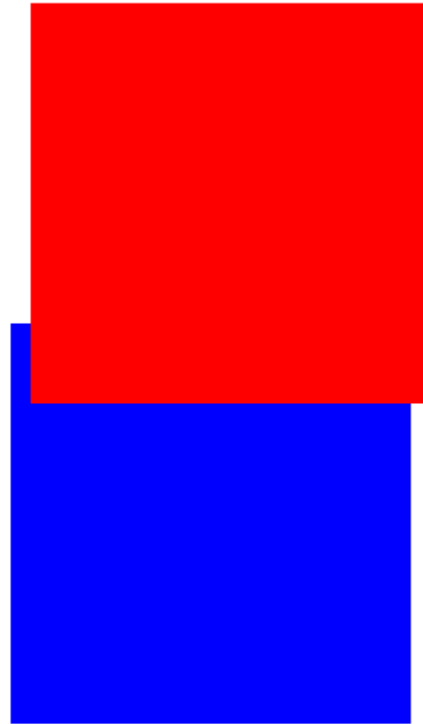
# Thuộc tính position

Cách tính vị trí của khối: static (\*), relative, absolute, fixed.



## Ví dụ về position:relative;

```
1 <html xmlns="http://www.w3.org/1999/xhtml">
2 <head>
3 </head>
4 <body>
5   <div id="box1"></div>
6
7   <div id="box2"></div>
8   <style>
9     body{margin:0px;}
10    #box1, #box2{
11      width:200px;
12      height:200px;
13      background-color:red;
14      margin:10px;
15    }
16    #box2{
17      background-color:blue !important;
18    }
19    #box1{
20      position:relative;
21      top:50px;
22      left:10px;
23    }
24  </style>
25 </body>
26 </html>
```



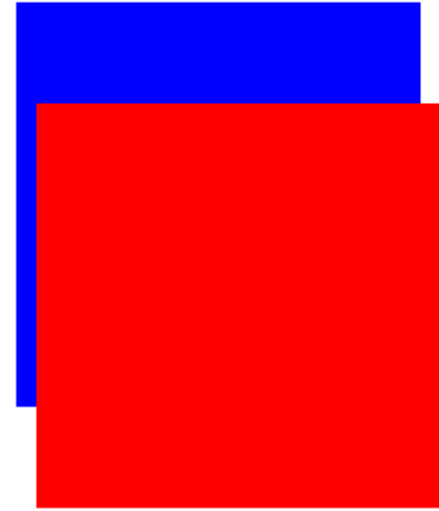
# Thuộc tính position

Cách tính vị trí của khối: static (\*), relative, absolute, fixed.



## Ví dụ về position:absolute;

```
1 <html xmlns="http://www.w3.org/1999/xhtml">
2 <head>
3 </head>
4 <body>
5   <div id="box1"></div>
6
7   <div id="box2"></div>
8   <style>
9     body{margin:0px;}
10    #box1, #box2{
11      width:200px;
12      height:200px;
13      background-color:red;
14      margin:10px;
15    }
16    #box2{
17      background-color:blue !important;
18    }
19    #box1{
20      position:absolute;
21      top:50px;
22      left:10px;
23    }
24  </style>
25 </body>
26 </html>
```

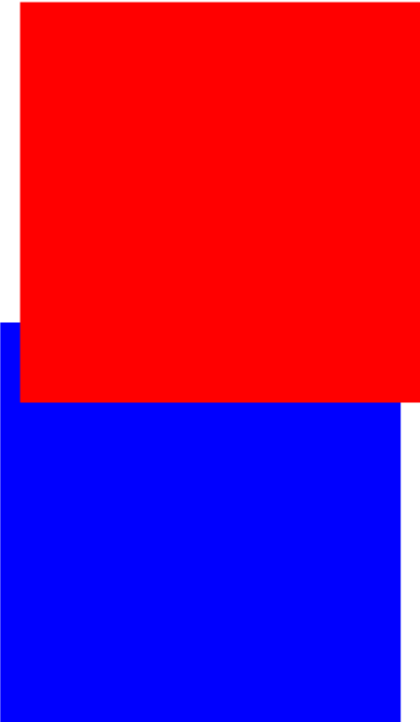


# Thuộc tính position

Cách tính vị trí của khối: `static (*)`, `relative`, `absolute`, `fixed`.

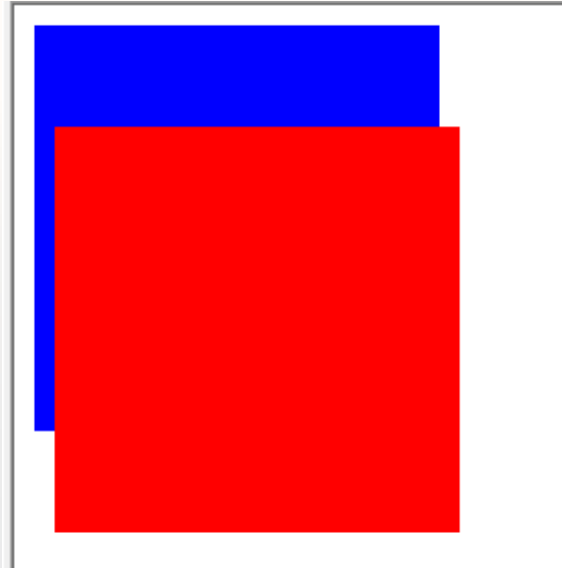


## position:relative;



```
body{margin:0px;}
#box1, #box2{
  width:200px;
  height:200px;
  background-color:red;
  margin:10px;
}
#box2{
  background-color:blue !important;
}
#box1{
  position:absolute;
  top:50px;
  left:10px;
}
```

## position:absolute;



Cả `relative` và `absolute` đều cho phép sử dụng thuộc tính `top` và `left`. NHƯNG:

- Với `relative`: khối thứ 2 (màu xanh blue) sẽ chịu ảnh hưởng từ khối 1 (màu đỏ).
- Với `absolute`: hai khối độc lập nhau và tính theo tọa độ của khối chứa nó (hiện tại là `body`).

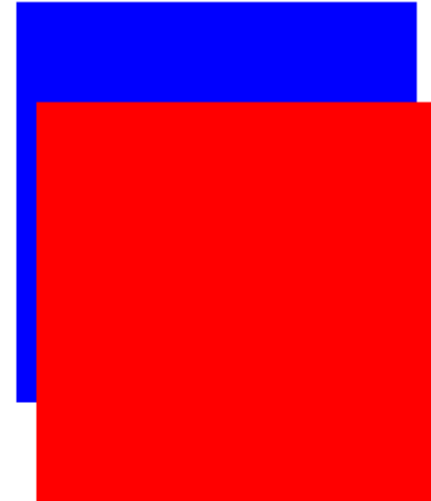
# Thuộc tính position

Cách tính vị trí của khối: static (\*), relative, absolute, fixed.



## Ví dụ về position:fixed;

```
1 <html xmlns="http://www.w3.org/1999/xhtml">
2 <head>
3 </head>
4 <body>
5     <div id="box1"></div>
6
7     <div id="box2"></div>
8     <style>
9         body{margin:0px;}
10        #box1, #box2{
11            width:200px;
12            height:200px;
13            background-color:red;
14            margin:10px;
15        }
16        #box2{
17            background-color:blue !important;
18        }
19        #box1{
20            position:fixed;
21            top:50px;
22            left:10px;
23        }
24    </style>
25 </body>
26 </html>
```



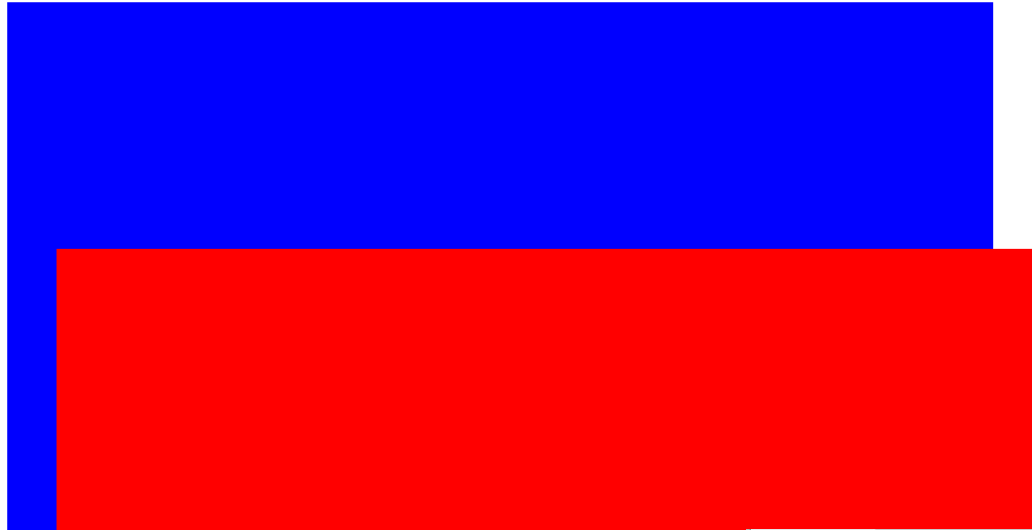


# Thuộc tính position

Cách tính vị trí của khối: static (\*), relative, absolute, fixed.

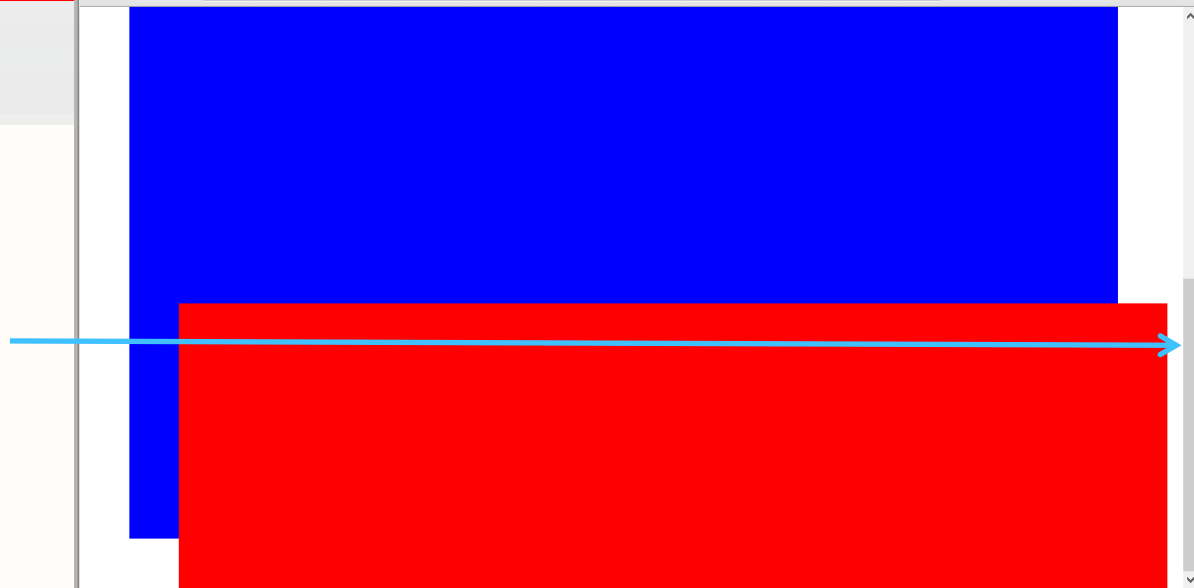


Ví dụ về `position:fixed;`



Chưa kéo thanh cuộn

Đã kéo thanh cuộn xuống

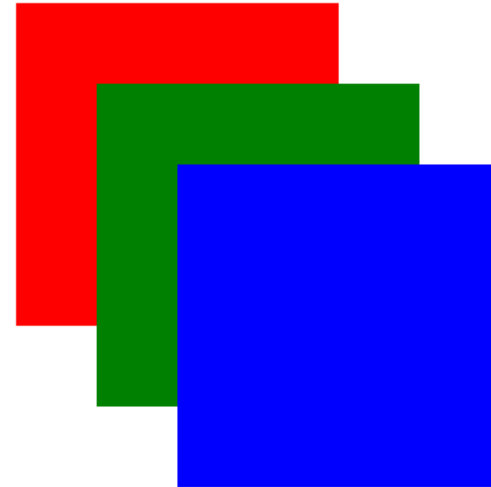


# Thuộc tính z-index

Chiều thứ 3 (chiều sâu) trong không gian, được tính bằng chỉ số.  
Z-index càng lớn, thì nó càng ở phía trên.



```
1 <html xmlns="http://www.w3.org/1999/xhtml">
2 <head>
3 </head>
4 <body>
5   <div id="box1"></div><!--z-index là 0-->
6   <div id="box2"></div><!--z-index là 1-->
7   <div id="box3"></div><!--z-index là 2-->
8   <style>
9     body{margin:0px;}
10    #box1, #box2, #box3{
11      width:200px;
12      height:200px;
13      margin:10px;
14      position:absolute;
15    }
16    #box1{
17      top:50px;
18      left:50px;
19      background-color:red;
20    }
21    #box2{
22      top:100px;
23      left:100px;
24      background-color:green;
25    }
26    #box3{
27      background-color:blue;
28      top:150px;
29      left:150px;
30    }
31  </style>
32 </body>
33 </html>
```



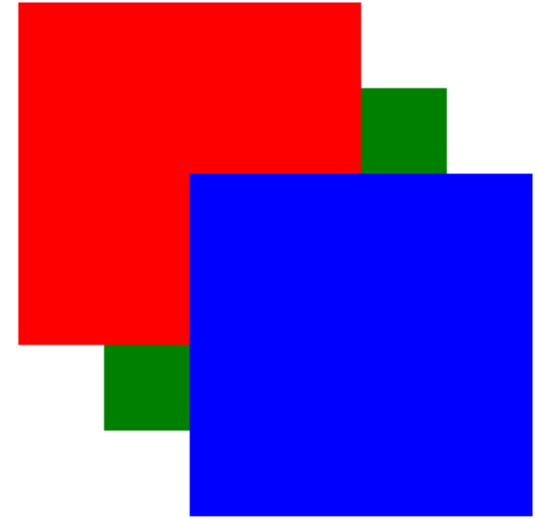
**Chú ý:** Theo mặc định, khối tạo ra sau sẽ nằm ở lớp trên so với khối tạo ra trước (thứ tự tạo ra trong html).

# Thuộc tính z-index



Chiều thứ 3 (chiều sâu) trong không gian, được tính bằng chỉ số. Z-index càng lớn, thì nó càng ở phía trên.

```
1 <html xmlns="http://www.w3.org/1999/xhtml">
2 <head>
3 </head>
4 <body>
5   <div id="box1"></div><!--z-index là 0-->
6   <div id="box2"></div><!--z-index là 1-->
7   <div id="box3"></div><!--z-index là 2-->
8   <style>
9     body{margin:0px;}
10    #box1, #box2, #box3{
11      width:200px;
12      height:200px;
13      margin:10px;
14      position:absolute;
15    }
16    #box1{
17      top:50px;
18      left:50px;
19      background-color:red;
20    }
21    #box2{
22      top:100px;
23      left:100px;
24      background-color:green;
25      z-index:-1;
26    }
27    #box3{
28      background-color:blue;
29      top:150px;
30      left:150px;
31    }
32  </style>
33 </body>
34 </html>
```



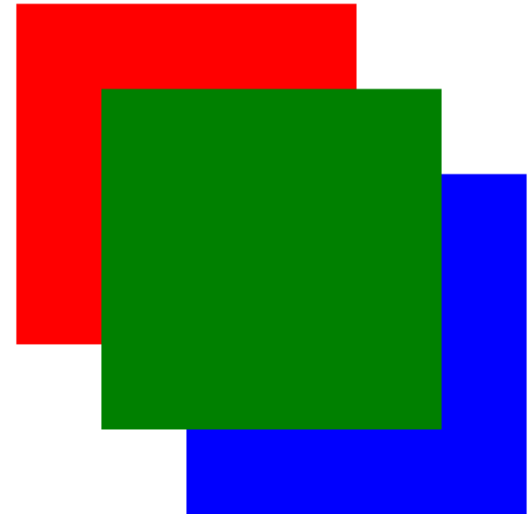
**Thay đổi chỉ số z-index của khối box2 là -1**

# Thuộc tính z-index

Chiều thứ 3 (chiều sâu) trong không gian, được tính bằng chỉ số. Z-index càng lớn, thì nó càng ở phía trên.



```
1 <html xmlns="http://www.w3.org/1999/xhtml">
2 <head>
3 </head>
4 <body>
5   <div id="box1"></div><!--z-index là 0-->
6   <div id="box2"></div><!--z-index là 1-->
7   <div id="box3"></div><!--z-index là 2-->
8   <style>
9     body{margin:0px;}
10    #box1, #box2, #box3{
11      width:200px;
12      height:200px;
13      margin:10px;
14      position:absolute;
15    }
16    #box1{
17      top:50px;
18      left:50px;
19      background-color:red;
20    }
21    #box2{
22      top:100px;
23      left:100px;
24      background-color:green;
25      z-index:3;
26    }
27    #box3{
28      background-color:blue;
29      top:150px;
30      left:150px;
31    }
32  </style>
33 </body>
34 </html>
```



**Thay đổi chỉ số z-index của khối box2 là 3**

# Thuộc tính z-index

Chiều thứ 3 (chiều sâu) trong không gian, được tính bằng chỉ số.



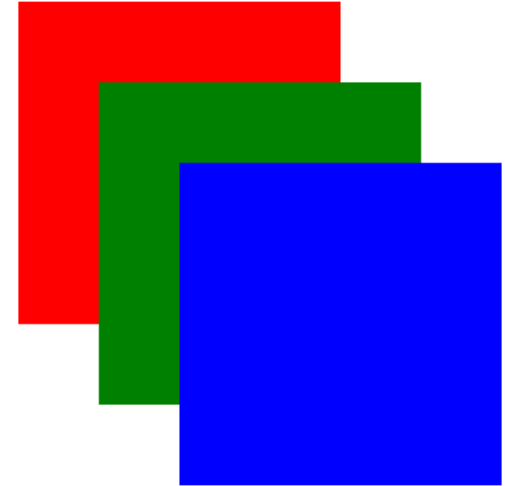
ĐIỀU GÌ XẢY RA NẾU CÓ NHIỀU KHỐI CÓ  
CÙNG Z-INDEX

# Thuộc tính z-index



Chiều thứ 3 (chiều sâu) trong không gian, được tính bằng chỉ số.

```
1 <html xmlns="http://www.w3.org/1999/xhtml">
2 <head>
3 </head>
4 <body>
5   <div id="box1"></div><!--z-index là 0-->
6   <div id="box2"></div><!--z-index là 1-->
7   <div id="box3"></div><!--z-index là 2-->
8   <style>
9     body{margin:0px;}
10    #box1, #box2, #box3{
11      width:200px;
12      height:200px;
13      margin:10px;
14      position:absolute;
15    }
16    #box1{
17      top:50px;
18      left:50px;
19      background-color:red;
20      z-index:0;
21    }
22    #box2{
23      top:100px;
24      left:100px;
25      background-color:green;
26      z-index:0;
27    }
28    #box3{
29      background-color:blue;
30      top:150px;
31      left:150px;
32      z-index:0;
33    }
34  </style>
35 </body>
36 </html>
```



**Nếu cùng chỉ số z-index, thì các khối sẽ được tính theo z-index mặc định: theo thứ tự tạo ra trong html.**

# Thuộc tính background



Dùng để đổ nền cho đối tượng: màu, ảnh, màu gradient...

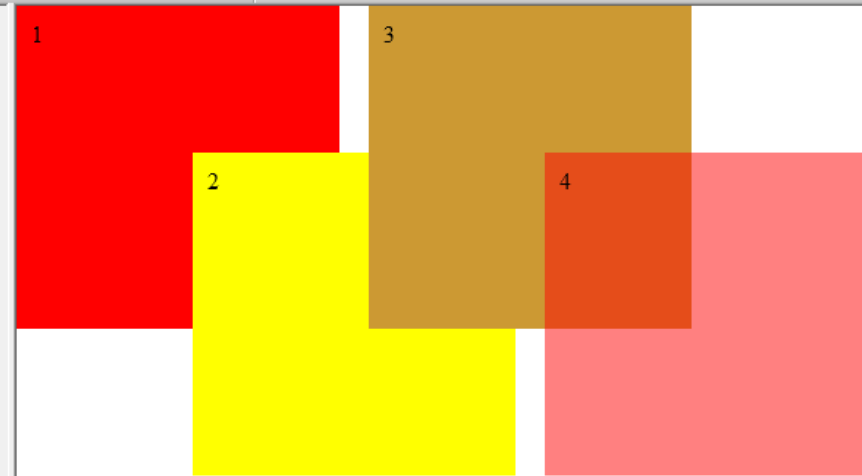
- Thuộc tính background có thể dùng để đổ nền cho một đối tượng. Phần nền có thể được tô bằng màu, bằng ảnh hoặc bằng màu gradient (chỉ cho phép với css3).
- Ta có thể dùng background-color để đổ màu nền (thay vì dùng background).
- Ta có thể dùng background-image để chèn ảnh làm nền (thay vì dùng background).

# Thuộc tính background-color

Dùng để đổ màu nền cho đối tượng: RGB, RGBA, tên màu, mã màu.



```
1 <html xmlns="http://www.w3.org/1999/xhtml">
2 <head>
3 </head>
4 <body>
5   <div id="box1">1</div>
6   <div id="box2">2</div>
7   <div id="box3">3</div>
8   <div id="box4">4</div>
9   <style>
10    body{margin:0px;}
11    #box1, #box2, #box3, #box4{
12      width:200px;
13      height:200px;
14      margin-right:-100px;
15      float:left;
16      padding:10px;
17    }
18    #box1{
19      background-color:red;
20    }
21    #box2{
22      background-color:rgb(255, 255, 0);
23      margin-top:100px;
24    }
25    #box3{
26      background-color:#C93;
27    }
28    #box4{
29      background-color:rgba(255, 0, 0, .5);
30      margin-top:100px;
31    }
32  </style>
33 </body>
34 </html>
```



RGB(0-255, 0-255, 0-255)

RGBA(0-255, 0-255, 0-255, 0-1)

Tên màu: red, green, blue,  
violet, darkgreen, lightgreen,  
darkblue, lightblue, brown,  
white, black...

Mã màu: theo mã hexa



# Thuộc tính background-image

Dùng để đổ nền cho đối tượng bằng ảnh.



```
1 <html xmlns="http://www.w3.org/1999/xhtml">
2 <head>
3 </head>
4 <body>
5   <div id="box1">1</div>
6   <style>
7     body{margin:0px;}
8     #box1{
9       width:200px;
10      height:200px;
11      margin-right:-100px;
12      float:left;
13      padding:10px;
14      background-image:url(media/homepage-USNWR2013.png) ;
15    }
16  </style>
17 </body>
18 </html>
```



Sử dụng giá trị url(địa chỉ đến file ảnh). Nên sử dụng địa chỉ tương đối. Mặc định, ảnh sẽ tự động lặp.

# Thuộc tính background-image

Dùng để đổ nền cho đối tượng bằng ảnh.



```
1 <html xmlns="http://www.w3.org/1999/xhtml">
2 <head>
3 </head>
4 <body>
5     <div id="box1">1</div>
6     <style>
7         body{margin:0px;}
8         #box1{
9             width:200px;
10            height:200px;
11            margin-right:-100px;
12            float:left;
13            padding:10px;
14            background-image:url(media/homepage-USNWR2013.png);
15            background-repeat:repeat-x;
16        }
17    </style>
18 </body>
19 </html>
```



Để điều khiển sự lặp lại của ảnh, ta có thể sử dụng thuộc tính background-repeat. Thuộc tính này có thể nhận các giá trị:

- no-repeat: không lặp.
- repeat-x: lặp theo chiều x.
- repeat-y: lặp theo chiều y.
- repeat (mặc định): lặp theo hai chiều.



HIỂN THỊ CÁC PHẦN CỦA ẢNH (KHÔNG HIỂN THỊ TOÀN BỘ) BẰNG CÁCH SỬ DỤNG CSS SẼ GIÚP CHO NGƯỜI THIẾT KẾ GIẢM GÁNH NẶNG TRONG VIỆC QUẢN LÝ NHIỀU TẬP TIN MEDIA.

1

1

# Thuộc tính background-position

Dùng để điều chỉnh vị trí của ảnh nền được tạo bởi background-image.



```
1 <html xmlns="http://www.w3.org/1999/xhtml">
2 <head></head>
3 <body>
4   ảnh gốc
5   <div id="box1"></div>
6   hiển thị phần ảnh đầu tiên
7   <div id="box2"></div>
8   hiển thị phần ảnh thứ 2
9   <div id="box3"></div>
10  hiển thị phần ảnh thứ 3
11  <div id="box4"></div>
12  <style>
13    body{margin:0px;}
14    #box1, #box2, #box3, #box4{
15      width:21px;
16      height:21px;
17      background-image:url(media/anythingSlider_thumbNav_a_background1.png);
18      background-repeat:no-repeat;
19    }
20    #box1{
21      width:63px !important;
22    }
23    #box2{
24      background-position:0px 0px;
25    }
26    #box3{
27      background-position:-21px 0px;
28    }
29    #box4{
30      background-position:-42px 0px;
31    }
32  </style>
33 </body>
34 </html>
```

ảnh gốc



hiển thị phần ảnh đầu tiên



hiển thị phần ảnh thứ 2



hiển thị phần ảnh thứ 3



Trong đó, background-position có hai giá trị đi kèm: giá trị 1 (là vị trí dịch lùi theo chiều x), giá trị 2 (chiều y). Hoặc có thể định vị: left, top, right, bottom.



# MỘT SỐ HIỆU CHỈNH VỀ FONT CHỮ

# Một số hiệu chỉnh font chữ



- Thuộc tính color: màu font chữ.
- Thuộc tính font-size: kích thước font chữ. Đo bằng pt hoặc px.
- Thuộc tính font-family: tên font chữ.

# Một số hiệu chỉnh font chữ

Các thuộc tính: font-family(tên font); font-size(kích thước font); color(màu font).



```
1 <html xmlns="http://www.w3.org/1999/xhtml">
2 <head></head>
3 <body>
4   <div id="box">
5     <div class="b1">
6       Nội dung từ div box.b1
7     </div>
8     <div class="b1">
9       Nội dung từ div box.b1
10    </div>
11    <div class="b2">
12      Nội dung từ div box.b2
13    </div>
14  </div>
15  <style>
16    body{margin:0px;}
17    #box div{
18      font-family:Tahoma, Geneva, sans-serif;
19    }
20    #box .b1{
21      color:green;
22      font-size:12pt;
23    }
24    #box .b2{
25      font-style:italic;
26    }
27  </style>
28 </body>
29 </html>
```

Nội dung từ div box.b1  
Nội dung từ div box.b1  
*Nội dung từ div box.b2*



KHI CỐ ĐỊNH KÍCH THƯỚC CỦA MỘT KHỐI, LÀM THẾ NÀO ĐỂ QUY ĐỊNH PHẦN NỘI DUNG KHÔNG ĐƯỢC PHÉP TRÀN RA KHỎI KHỐI?

**overflow**

Quê hương là chùm khế ngọt  
Cho con trèo hái mỗi ngày

....

Quê hương là con diều biếc  
Tuổi thơ con thả trên đồng

Quê hương là chùm khế ngọt  
Cho con trèo hái mỗi ngày

....

Quê hương là con diều biếc



# Thuộc tính overflow

Cho phép nội dung chỉ hiển thị trong một vùng chỉ định.



```
1 <html xmlns="http://www.w3.org/1999/xhtml">
2 <head></head>
3 <body>
4   <div class="mine" id="b1">
5     Quê hương là chùm khế ngọt<br/>
6     Cho con trèo hái mỗi ngày<br/>
7     Quê hương là đường đi học<br/>
8     Con về rợp bướm vàng bay<br/>
9   </div>
10  <div class="mine" id="b2">
11    Quê hương là chùm khế ngọt<br/>
12    Cho con trèo hái mỗi ngày<br/>
13    Quê hương là đường đi học<br/>
14    Con về rợp bướm vàng bay<br/>
15  </div>
16  <style>
17    body{margin:0px;}
18    .mine{
19      width:100px;
20      height:100px;
21      background-color:#0F9;
22      color:brown;
23      margin-bottom:100px;
24    }
25    #b1{
26      overflow:hidden;
27    }
28    #b2{
29      overflow:scroll;
30    }
31  </style>
32 </body>
33 </html>
```

Quê hương là  
chùm khế ngọt  
Cho con trèo  
hái mỗi ngày  
Quê hương là

Quê hương là  
chùm khế  
ngọt  
Cho con trèo  
hái mỗi ngày

# Thuộc tính overflow

Cho phép nội dung chỉ hiển thị trong một vùng chỉ định.



Overflow có thể nhận các giá trị:

- **hidden**: nội dung vượt ra khỏi vùng kích thước hiển thị sẽ bị ẩn đi.
- **scroll**: sẽ bổ sung thêm thanh trượt (scrollbar) để cho phép người dùng kéo thanh trượt để xem tiếp nội dung.
- **visible**: mặc định (tương tự như khi không sử dụng thuộc tính overflow).
- Overflow có thể được điều khiển theo chiều x hoặc y, tương ứng overflow-x, overflow-y.



# SỰ KẾ THỪA THUỘC TÍNH TRONG CSS



- THUỘC TÍNH CSS SẼ CHỊU SỰ CHI PHỐI CỦA CÁC KHỐI CHỨA NÓ.
- NẾU MỘT KHỐI CHA ÁP DỤNG MỘT THUỘC TÍNH CSS NÀO ĐÓ, TRONG KHỐI CON KHÔNG CHỈ ĐỊNH LẠI THUỘC TÍNH CSS ĐÓ THÌ GIÁ TRỊ CỦA THUỘC TÍNH CSS NÀY SẼ ĐƯỢC KẾ THỪA TỪ KHỐI CHA.
- NẾU NGƯỢC LẠI, KHỐI CON CHỈ ĐỊNH LẠI THUỘC TÍNH CSS ĐÓ NHƯNG VỚI GIÁ TRỊ KHÁC VỚI GIÁ TRỊ ĐƯỢC CHỈ ĐỊNH Ở KHỐI CHA, THÌ THUỘC TÍNH CSS NÀY NHẬN GIÁ TRỊ TƯƠNG ƯNG VỚI GIÁ TRỊ ĐƯỢC CHỈ ĐỊNH Ở KHỐI CON.

# TÍNH KẾ THỪA TRONG CSS



```
1 <html xmlns="http://www.w3.org/1999/xhtml">
2 <head></head>
3 <body>
4   <div id="container">
5     <div id="sub1">
6       Nội dung trong sub1
7     </div>
8     <div id="sub2">
9       Nội dung trong sub2
10    </div>
11  </div>
12  <style>
13    body{margin:0px;}
14    #container{
15      color:green;
16      font-family:Tahoma, Geneva, sans-serif;
17    }
18    #sub1{
19      color:blue;
20    }
21    #sub2{
22      font-size:10pt;
23    }
24  </style>
25 </body>
26 </html>
```

Nội dung trong sub1  
Nội dung trong sub2





# PHẠM VI CỤC BỘ VÀ PHẠM VI TOÀN CỤC TRONG CSS

# PHẠM VI CỤC BỘ VÀ PHẠM VI TOÀN CỤC TRONG CSS



```
1 <html xmlns="http://www.w3.org/1999/xhtml">
2 <head></head>
3 <body>
4   <div id="space1">
5     <div class="sub">
6       Nội dung trong sub1
7     </div>
8   </div>
9   <div id="space2">
10    <div class="sub">
11      Nội dung trong sub2
12    </div>
13  </div>
14  <style>
15    body{margin:0px;}
16    .sub{
17      color:red;
18    }
19    #space1 .sub{
20      font-size:15px;
21      font-family:Tahoma, Geneva, sans-serif;
22    }
23    #space2 .sub{
24      font-size:20px;
25      font-family:"Courier New", Courier, monospace;
26    }
27  </style>
28 </body>
29 </html>
```

Nội dung trong sub1

Nội dung trong sub2



## HIỆU CHỈNH CSS CHO THẺ <A>

- **a:hover**: khi trỏ chuột di chuyển trên liên kết.
- **a:visited**: khi liên kết đã từng được xem qua.
- **a:link**: khi liên kết chưa từng được mở.



# HIỆU CHỈNH CSS CHO THẺ <A>



```
1 <html xmlns="http://www.w3.org/1999/xhtml">
2 <head></head>
3 <body>
4   <div id="space1">
5     <div class="sub">
6       <a href="#">Nội dung trong sub1</a>
7     </div>
8   </div>
9   <div id="space2">
10    <div class="sub">
11      <a href="#">Nội dung trong sub2</a>
12    </div>
13  </div>
14  <style>
15    body{margin:0px;}
16    #space1 .sub a:visited, #space1 .sub a:link{
17      font-size:15px;
18      font-family:Tahoma, Geneva, sans-serif;
19      color:red;
20    }
21    #space1 .sub a:hover{
22      color:green;
23    }
24    #space2 .sub a:visited, #space2 .sub a:link{
25      font-size:15px;
26      font-family:"Courier New", Courier, monospace;
27      color:green;
28    }
29    #space2 .sub a:hover{
30      color:yellow;
31    }
32  </style>
33 </body>
34 </html>
```

Nội dung trong sub1  
Nội dung trong sub2





# HIỆU CHỈNH CSS VỚI TRẠNG THÁI HOVER CHO THẺ DIV

Ví dụ:

```
#box:hover{...}
```

```
.box:hover{...}
```