

Bài giảng

PHÁT TRIỂN ỨNG DỤNG WEB

Lê Đình Thanh

Khoa Công nghệ Thông tin

Trường Đại học Công nghệ, ĐHQGHN

E-mail: thanhld@vnu.edu.vn Mobile: 0987.257.504

Chương 3

CSS

Nội dung

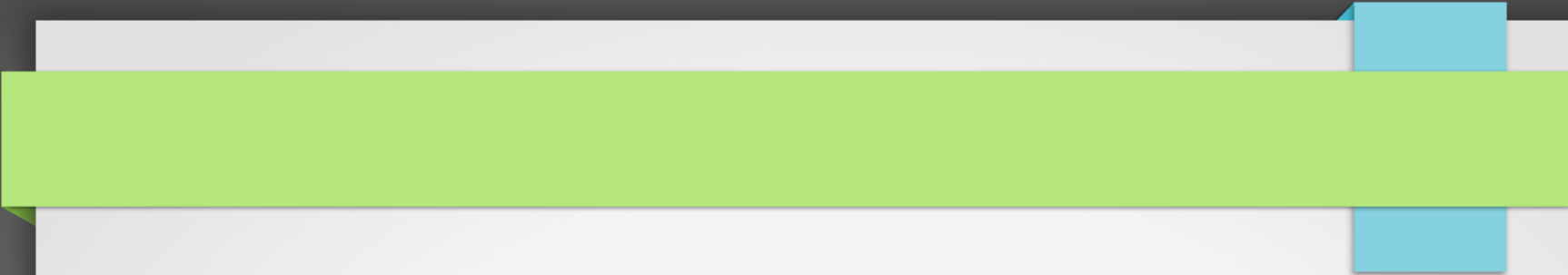
- HTML
- **CSS**
- JavaScript
- DOM

Nội dung web

- Một trang web bao gồm một tập các **đối tượng tài liệu** được
 - khai báo bằng HTML
 - lưu trữ theo cấu trúc DOM
 - định kiểu trình diễn bởi CSS
 - quản lý bởi JavaScript



Quan trọng



Bảng định dạng xếp tầng

CSS: Cascading Style Sheet



Lý do sử dụng CSS

- Nguyên lý **tách biệt nội dung và trình diễn**
 - Dùng HTML khai báo các đối tượng tài liệu (nội dung).
 - Dùng CSS để xác định cách trình diễn các đối tượng tài liệu.
- Nguyên lý **nhất quán** trong thiết kế giao diện
 - Nhiều đối tượng cần được định dạng theo một kiểu thống nhất. Ví dụ, tất cả các bảng có cùng một kiểu, tất cả các siêu liên kết có cùng một kiểu. Nhiều trang cần được dàn trang một cách thống nhất.
- Nguyên lý **W1Un, DRY**
 - Kiểu định dạng cần được định nghĩa một lần và sử dụng nhiều lần, nhiều nơi.

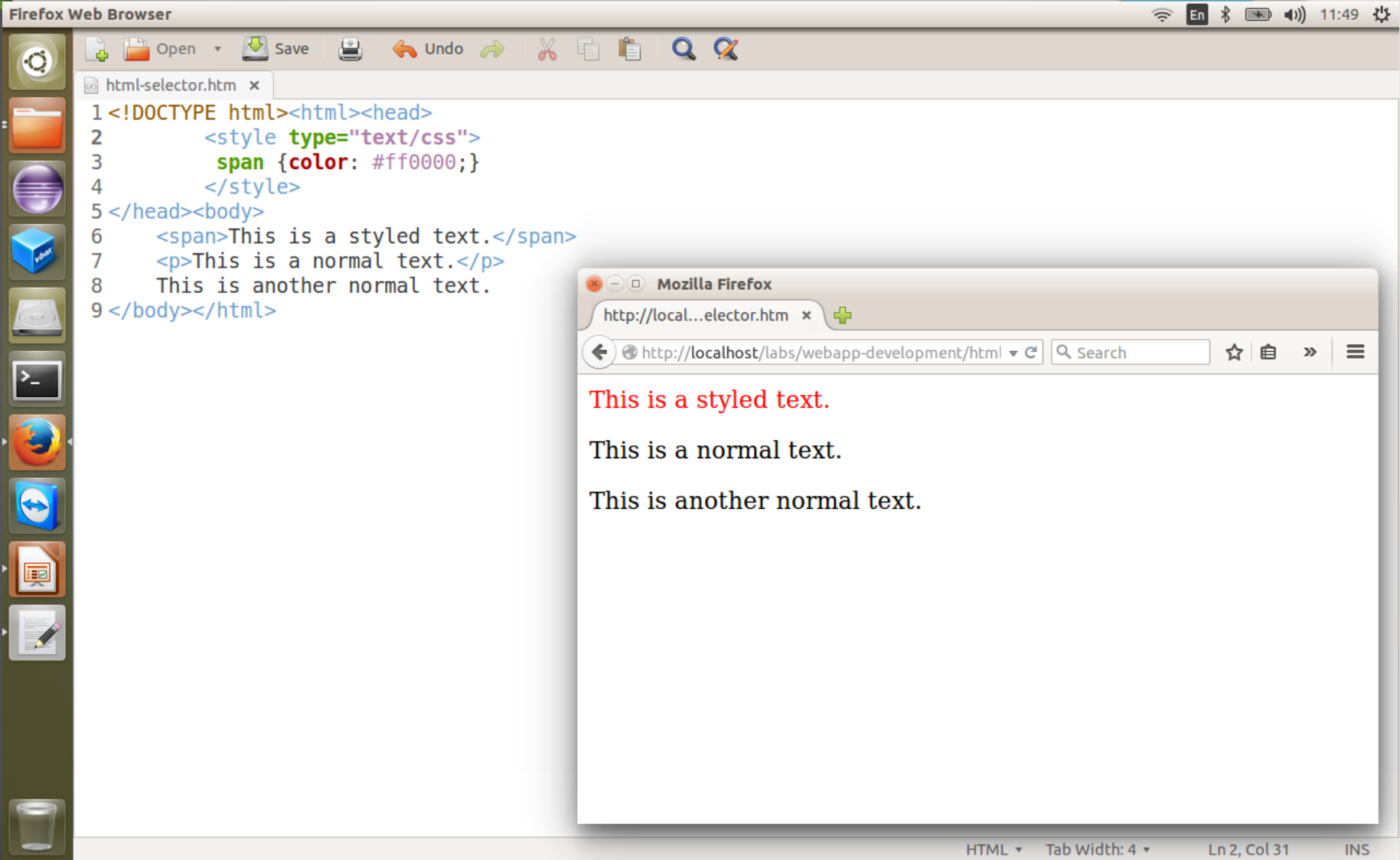
Style Rule

- Bộ chọn {thuộc-tính: giá-trị; thuộc-tính: giá-trị; ...}
 - Bộ chọn (selector): Lọc ra các đối tượng tài liệu khớp (match) với bộ chọn
 - Khai báo (declaration): thuộc-tính:giá-trị
 - Khối các khai báo (declaration block): {các khai báo}

Bộ chọn theo kiểu phần tử

- **HTML** {} – Chọn đối tượng html
- **BODY** {} – Chọn đối tượng body
- **TD** {} – Chọn các đối tượng td
- **TEXTAREA** {} – Chọn các đối tượng textarea
- **LABEL** {} – Chọn các đối tượng span
- **INPUT** {} – Chọn các đối tượng input
- **SELECT** {} – Chọn các đối tượng select
- **A** {} – Chọn các đối tượng liên kết <a>
- **UL** {} – Chọn các đối tượng danh sách không sắp xếp
- **OL** {} – Chọn các đối tượng danh sách sắp xếp
- **DL** {} – Chọn các đối tượng danh sách định nghĩa
- **LI** {} – Chọn các đối tượng mục trong danh sách
- **DL DT/DD** - Chọn các đối tượng mục trong danh sách định nghĩa

Bộ chọn theo kiểu phần tử



Bộ chọn theo thuộc tính

- Chọn các đối tượng có thuộc tính thỏa
 - `[thuoctinh]`
 - `[thuoctinh = giatri]`
 - `[thuoctinh *= giatri]` (chứa giá trị)
 - `[thuoctinh ~= tu]` (chứa từ)
 - `[thuoctinh |= tu]` (bắt đầu bằng từ)

Bộ chọn theo thuộc tính

Tryit Editor v3.5

Secure | https://www.w3schools.com/html/tryit.asp?filename=tryhtml_default

Ad closed by Google

Run »

Result Size: 392 x 341

```
<!DOCTYPE html><html><head><title>CSS</title>
<style type="text/css">
  [href] {color:green;}
  [href *= ".edu"] {font-weight:bold;}
  [lang |= "en"] {font-style:italic;}
</style>
</head><body>
  <h2>Một số liên kết</h2>
  <a href="http://vnu.edu.vn" lang="vi">Đại học Quốc gia Hà Nội</a><br>
  <a href="http://dantri.com.vn" lang="vi">Bảo điện tử Dân trí</a><br>
  <a href="https://google.com" lang="en-us">Google</a>
</body></html>
```

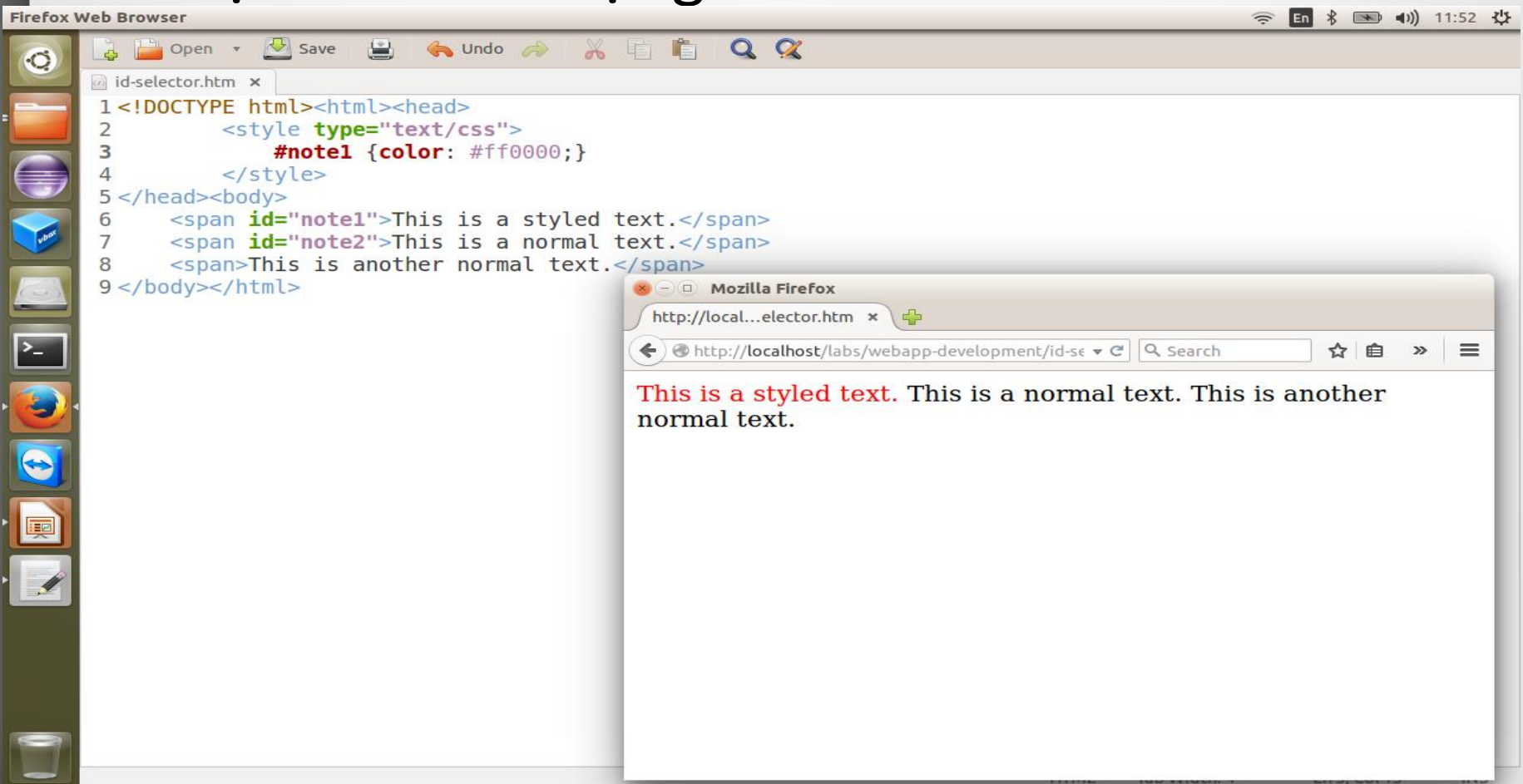
Một số liên kết

[Đại học Quốc gia Hà Nội](http://vnu.edu.vn)
[Bảo điện tử Dân trí](http://dantri.com.vn)
[Google](https://google.com)

EN 6:11 PM 7/31/2017

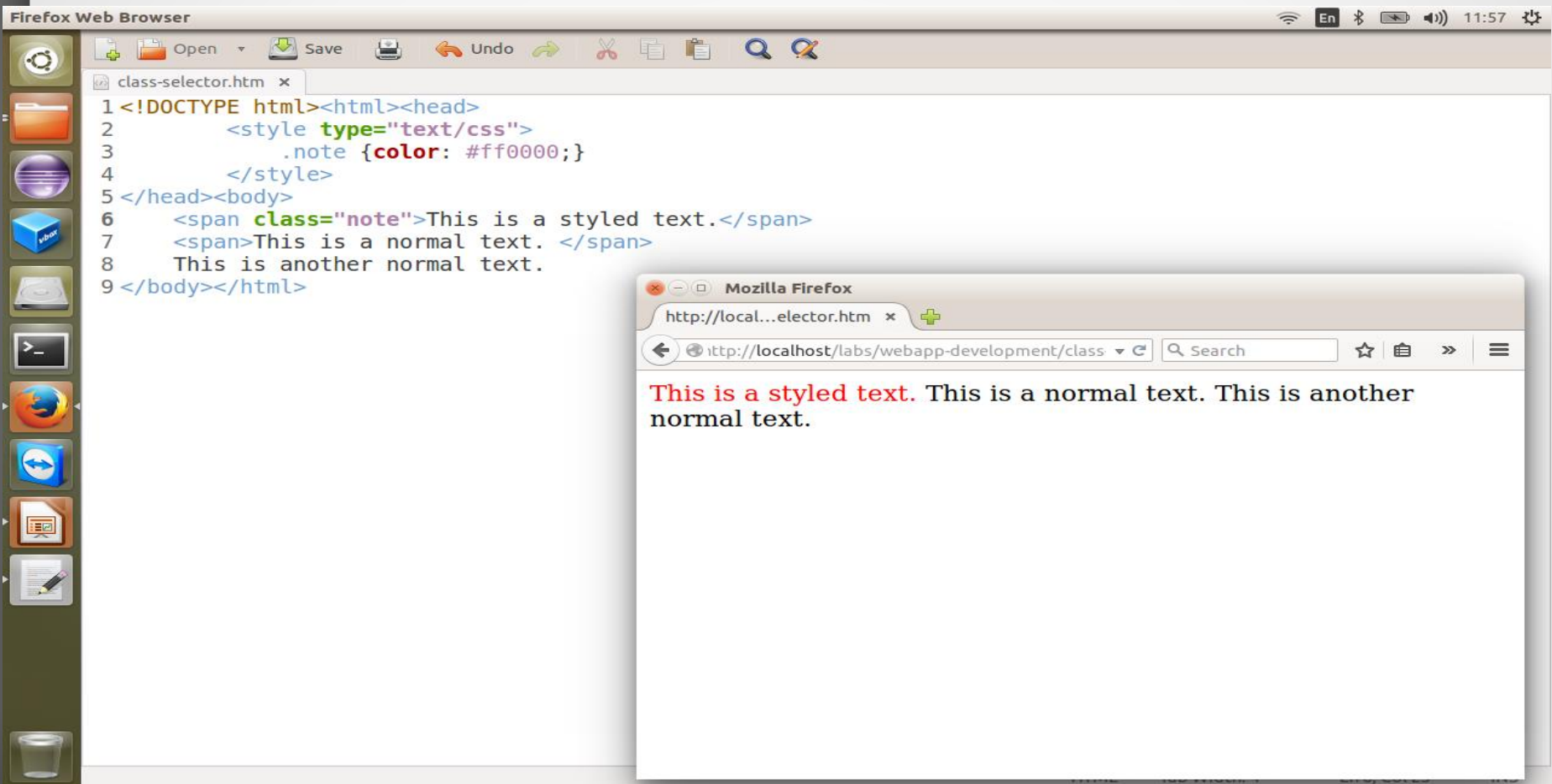
Bộ chọn theo định danh

- Chọn các đối tượng có *id="tenBochon"*



Bộ chọn theo lớp

- Chọn các đối tượng có *class="tenBochon"*



Bộ chọn nội tuyến

- Sử dụng thuộc tính **style** của các đối tượng HTML
- Nếu CSS không cần dùng cho nhiều đối tượng
- Ví dụ

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

Bộ chọn nội tuyến

Tryit Editor v3.5

Secure | https://www.w3schools.com/html/tryit.asp?filename=tryhtml_default

Not interested in this ad Ad knew too much Ad was inappropriate Ad covered content

Run » Result Size: 307 x 341

```
<!DOCTYPE html><html><head><title>CSS</title>
</head><body>
  <h2>Một số liên kết</h2>
  <a href="http://vnu.edu.vn" style="color:red;">Đại học Quốc gia Hà Nội</a><br>
  <a href="http://dantri.com.vn">Bảo điện tử Dân trí</a><br>
  <a href="https://google.com">Google</a>
</body></html>
```

Một số liên kết

[Đại học Quốc gia Hà Nội](http://vnu.edu.vn)
[Bảo điện tử Dân trí](http://dantri.com.vn)
[Google](https://google.com)

Windows taskbar: File Explorer, Firefox, Chrome, Edge, Word, 2:33 PM 8/3/2017

Bộ chọn tất cả

Tryit Editor v3.5

New tab

Secure | https://www.w3schools.com/html/tryit.asp?filename=tryhtml_default

Not interested in this ad

Ad knew too much

Ad was inappropriate

Ad covered content

Run »

Result Size: 307 x 341

```
<!DOCTYPE html><html><head><title>CSS</title>
<style type="text/css">
  * {color:red;}
</style>
</head><body>
<h2>Một số liên kết</h2>
<a href="http://vnu.edu.vn" style="color:red;">Đại học Quốc gia Hà Nội</a><br>
<a href="http://dantri.com.vn">Bảo điện tử Dân trí</a><br>
<a href="https://google.com">Google</a>
</body></html>
```

Một số liên kết

[Đại học Quốc gia Hà Nội](http://vnu.edu.vn)
[Bảo điện tử Dân trí](http://dantri.com.vn)
[Google](https://google.com)

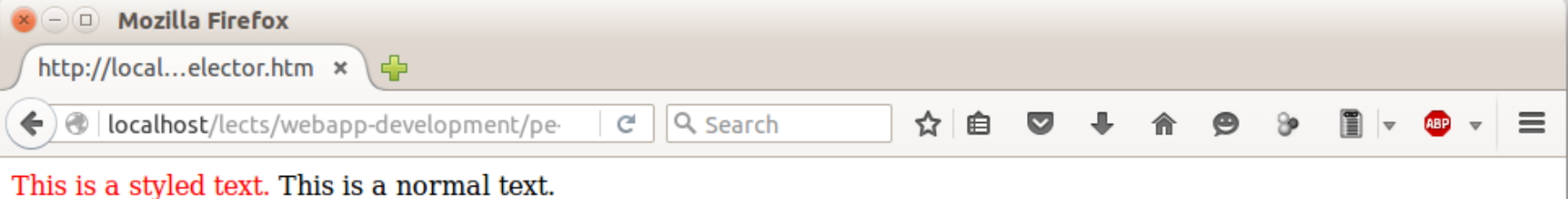
EN 2:36 PM 8/3/2017

Kết hợp bộ chọn

- $p > e$: Chọn đối tượng được chọn bởi e có cha được chọn bởi p
- $a e$: Chọn đối tượng được chọn bởi e ở bên trong đối tượng được chọn bởi a
- $prev + e$: Chọn đối tượng được chọn bởi e có đối tượng liền trước được chọn bởi $prev$
- $prev \sim e$: Chọn đối tượng được chọn bởi e nằm sau đối tượng được chọn bởi $prev$
- ...

Kết hợp $p > e$

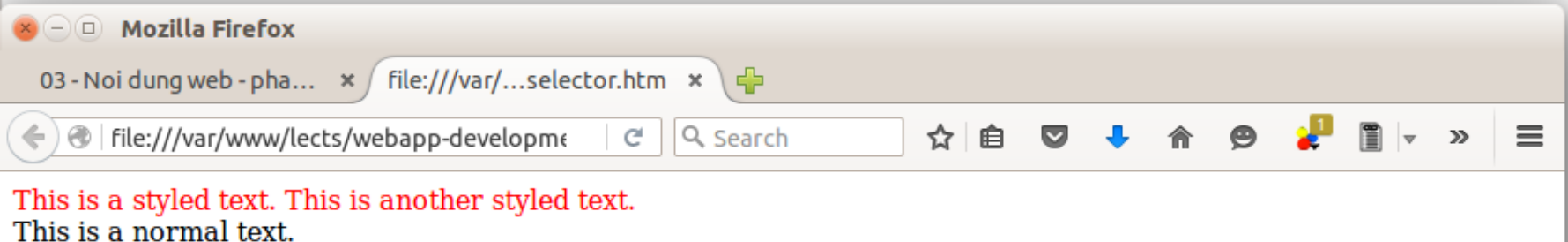
```
pe-selector.htm x
1 <!DOCTYPE html><html><head>
2     <style type="text/css">
3         div > span {color: #ff0000;}
4     </style>
5 </head><body>
6     <div>
7         <span>This is a styled text.</span>
8         <label><span>This is a normal text.</span></label>
9     </div>
10 </body></html>
```



Kết hợp a e

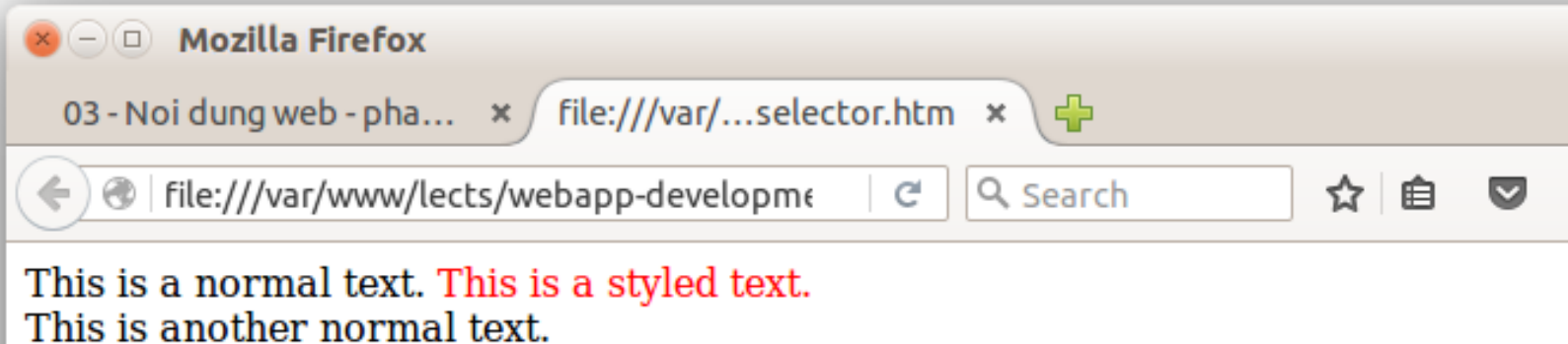
ae-selector.htm x

```
1 <!DOCTYPE html><html><head>
2     <style type="text/css">
3         div span {color: #ff0000;}
4     </style>
5 </head><body>
6     <div>
7         <span>This is a styled text.</span>
8         <label><span>This is another styled text.</span></label>
9     </div>
10    <span>This is a normal text.</span>
11</body></html>
```



Kết hợp prev + e

```
pe-selector.htm x
1 <!DOCTYPE html><html><head>
2     <style type="text/css">
3         span+span {color: #ff0000;}
4     </style>
5 </head><body>
6     <div>
7         <span>This is a normal text.</span>
8         <span>This is a styled text.</span>
9     </div>
10    <span>This is another normal text.</span>
11 </body></html>
```



Kết hợp **prev ~ e**

Tryit Editor v3.5

Secure | https://www.w3schools.com/html/tryit.asp?filename=tryhtml_default

Not interested in this ad | Ad knew too much | Ad was inappropriate | Ad covered content

Run »

Result Size: 307 x 341

```
<!DOCTYPE html><html><head><title>CSS</title>
<style type="text/css">
  h2 ~ a {color:red;}
</style>
</head><body>
<h2>Một số liên kết</h2>
<a href="http://vnu.edu.vn" style="color:red;">Đại học Quốc gia Hà Nội</a><br>
<span>
  <a href="http://dantri.com.vn">Bảo điện tử Dân trí</a>
</span><br>
<a href="https://google.com">Google</a>
</body></html>
```

Một số liên kết

[Đại học Quốc gia Hà Nội](http://vnu.edu.vn)
[Bảo điện tử Dân trí](http://dantri.com.vn)
[Google](https://google.com)

Lớp giả

Chọn các phần tử theo trạng thái (state) của chúng

:pseudo-class {}

Ví dụ

:hover :focus :enabled :disabled :read-only ...

:link :visited :active ...

:root :empty ...

:only-child :first-child :nth-child() :nth-of-type() ...

:is() :where() :has() ...

Tham khảo

<https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes>

Lớp giả

CSS Demo: :nth-child

RESET

HTML

CSS

OUTPUT

```
1 p {  
2   font-weight: bold;  
3 }  
4  
5 li:nth-child(-n+3) {  
6   border: 2px solid orange;  
7   margin-bottom: 1px;  
8 }  
9  
10 li:nth-child(even) {  
11   background-color: lightyellow;  
12 }  
13
```

NBA players with most championships:

- Bill Russell
- Sam Jones
- Tom Heinsohn
- K. C. Jones
- Satch Sanders
- John Havlicek
- Jim Loscutoff
- Frank Ramsey
- Robert Horry

Phần tử giả

Chọn phần tử được giả vờ thêm vào
`selector::pseudo-element {}`

Ví dụ

`::first-letter` – Ký tự đầu tiên

`::first-line` – Dòng đầu tiên

`::before` – Trước con cái

`::after` – Sau con út

`::marker` – Hộp đánh dấu trước `<li>`

Tham khảo

<https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-elements>

Phần tử giả

HTML Demo: ::first-letter

RESET

HTML

CSS

```
1 p::first-letter {  
2     font-size: 1.5rem;  
3     font-weight: bold;  
4     color: brown;  
5 }  
6
```

OUTPUT

Scientists exploring the depths of Monterey Bay unexpectedly encountered a rare and unique species of dragonfish. This species is the rarest of its species.

When Robison and a team of researchers discovered this fish, they were aboard a week-long expedition.

Bộ chọn hỗn hợp (compound)

- Chọn phần tử được đồng thời chọn bởi nhiều bộ chọn khác nhau (như kết hợp logic AND các bộ chọn)

`selector1Selector2Selector3`

Ví dụ

`h1.special`

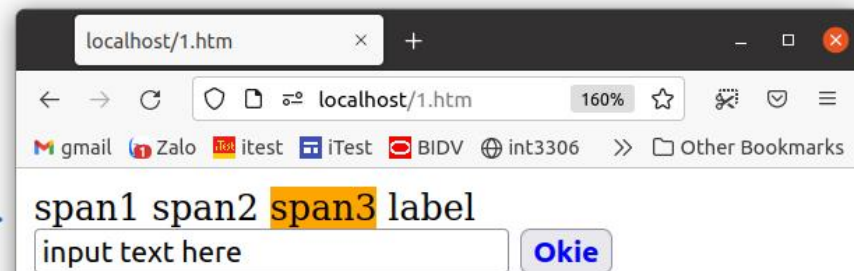
Chọn phần tử `<h1>` thuộc lớp `special` (class chứa `special`)

`.box.warning:first-child`

Chọn phần tử thuộc đồng thời hai lớp `box` và lớp `warning` (class chứa cả `box` và `warning`) và là con cả của một nút khác.

Bộ chọn hỗn hợp (compound)

```
1 <!DOCTYPE html><html><head>
2   <style>
3     span.warning {
4       background-color:orange;
5     }
6     input[type='button'] {
7       font-weight:bold;
8       color:blue;
9     }
10  </style>
11 </head><body>
12   <span>span1</span>
13   <span class="success">span2</span>
14   <span class="warning">span3</span>
15   <label class="warning">label</label>
16   <input value="input text here">
17   <input type="button" value="Okie">
18 </body></html>
```



Viết gộp các luật

- Nhiều Rule có chung một phần nội dung (một số khai báo giống nhau)

```
S1, S2, ..., Sn {  
    attribute1: value1;  
    attribute2: value2;  
    ...  
}  
S1 {attributeN:valueN; ...}  
S2 {attributeM:valueM; ...}
```

Chú thích

`/*Chú thích trong CSS*/`

Bảng định dạng (Style Sheet) trong

The screenshot displays a web development environment. The top window is a Notepad++ editor editing a file named 'internal-css.htm'. The code is as follows:

```
1 <html>
2   <head>
3     <style type="text/css">
4       span {background-color: yellow;} /*Bộ chọn theo kiểu phân tử HTML*/
5       .style1 {color:sienna;} /*Bộ chọn theo lớp*/
6       #style2 {color: red; font-weight:bold;} /*Bộ chọn theo định danh*/
7     </style>
8   </head>
9   <body>
10    <span class="style1"> Sử dụng CSS với bộ chọn theo lớp </span> <br>
11    <span id="style2">Sử dụng CSS với bộ chọn theo định danh </span> <br>
12    <span style="color: green;">Sử dụng CSS với bộ chọn không tên (nội tuyến)</sp
13    Không áp dụng CSS
14  </body>
15 </html>
```

The bottom window is a web browser displaying the rendered output of the file. The browser's address bar shows the file path: 'file:///C:/xampp/htdocs/labs/webapp-development/internal-css.htm'. The rendered content is:

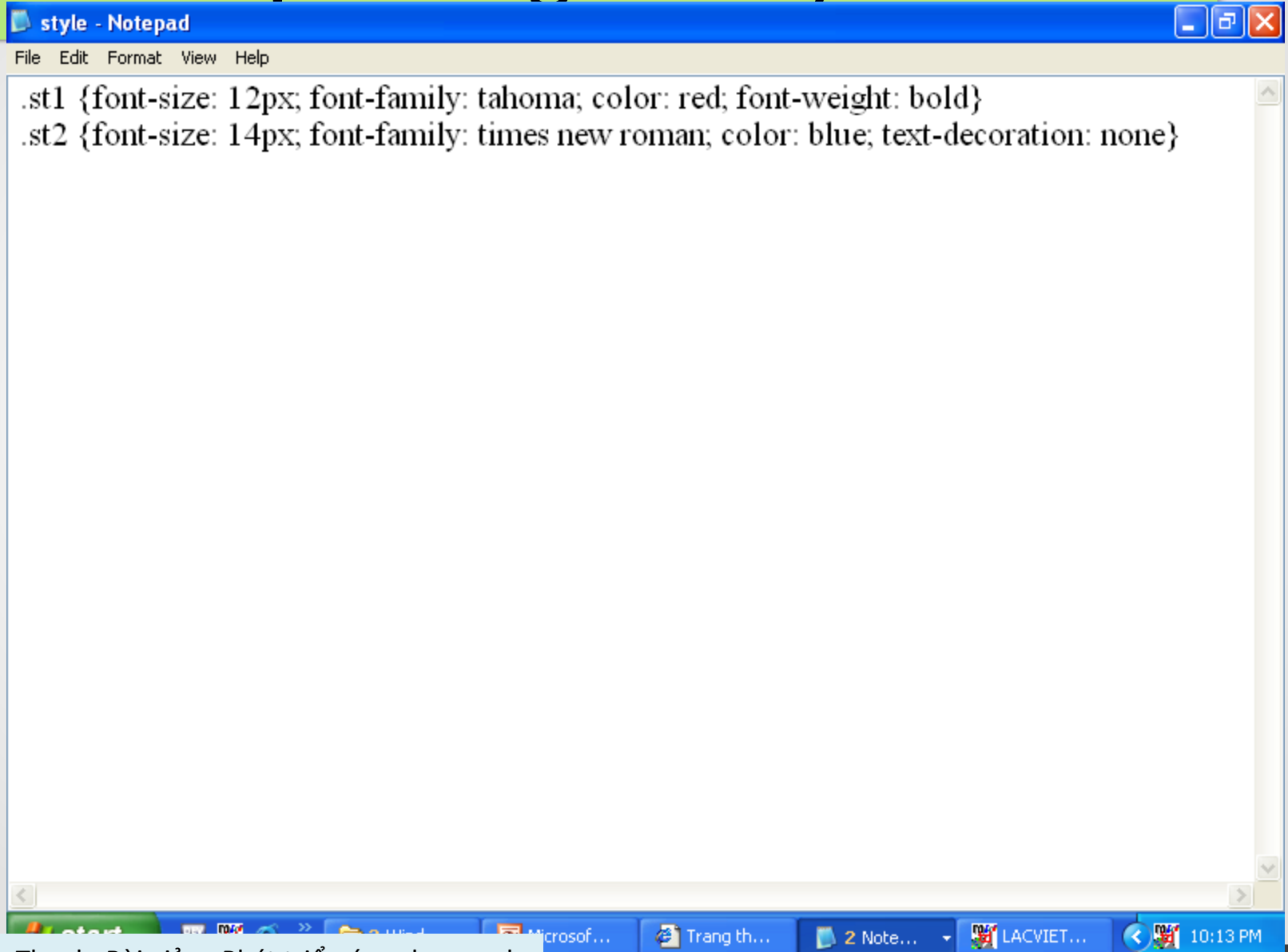
Sử dụng CSS với bộ chọn theo lớp
Sử dụng CSS với bộ chọn theo định danh
Sử dụng CSS với bộ chọn không tên (nội tuyến)
Không áp dụng CSS

The browser window also shows several tabs and a search bar at the top. The taskbar at the bottom of the screen shows various application icons and the system clock indicating 10:44 PM.

Style Sheet ngoài

- Các rules được định nghĩa và lưu trong tệp có tên mở rộng **.css**.
- Bao hàm tệp .css trong tệp HTML.

Tạo SS ngoài: style.css

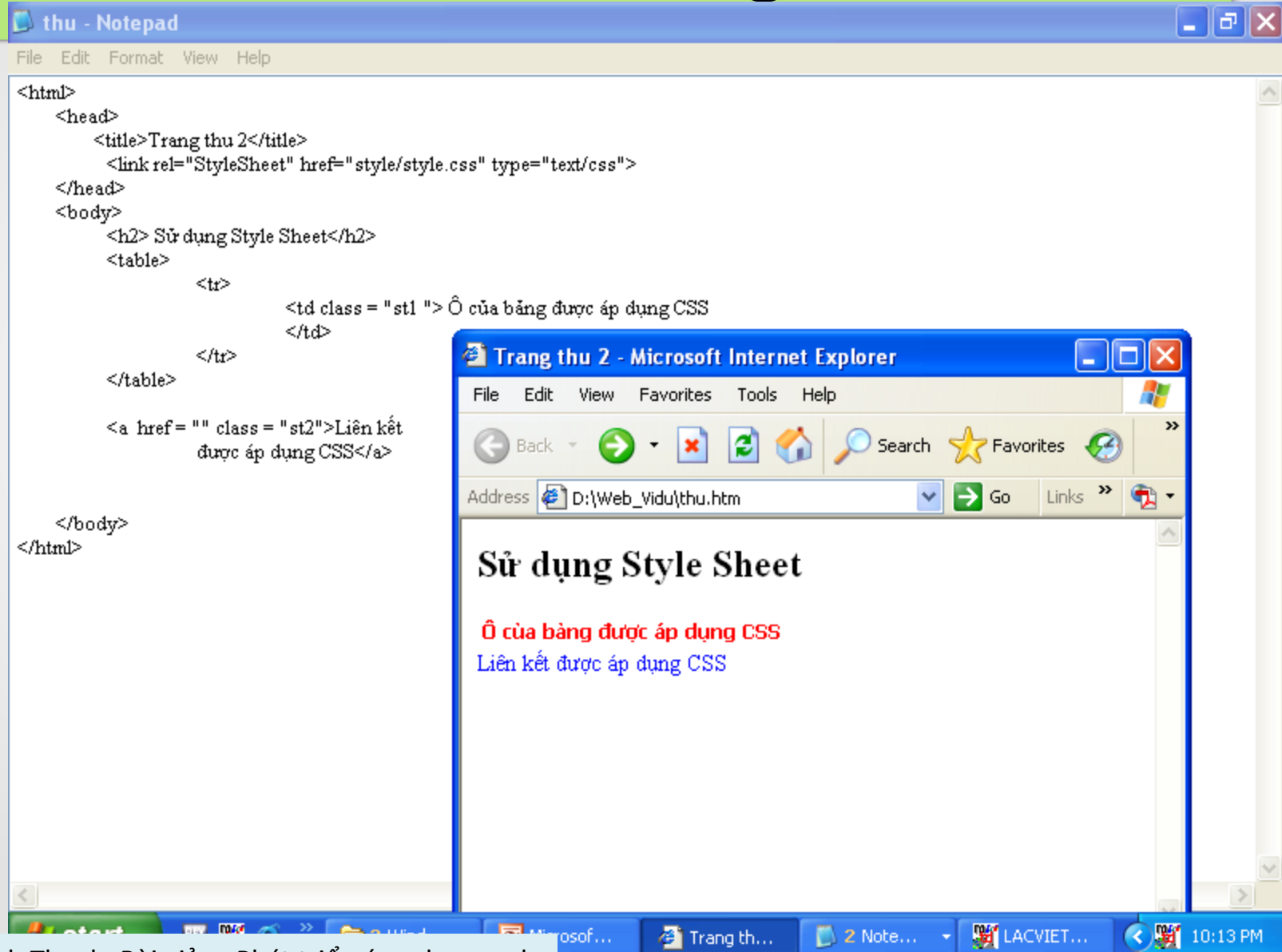
A screenshot of a Windows Notepad application window. The title bar is blue and reads "style - Notepad". Below the title bar is a menu bar with "File", "Edit", "Format", "View", and "Help". The main text area contains two lines of CSS code:

```
.st1 {font-size: 12px; font-family: tahoma; color: red; font-weight: bold}  
.st2 {font-size: 14px; font-family: times new roman; color: blue; text-decoration: none}
```

The window has standard Windows window controls (minimize, maximize, close) in the top right corner. At the bottom of the image, a portion of the Windows taskbar is visible, showing several open applications including "Trang th...", "2 Note...", and "LACVIET...". The system clock in the bottom right corner shows "10:13 PM".

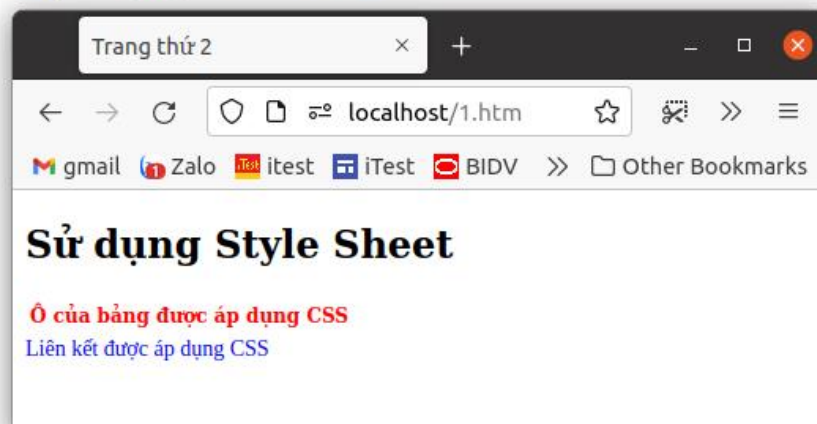
```
style - Notepad  
File Edit Format View Help  
.st1 {font-size: 12px; font-family: tahoma; color: red; font-weight: bold}  
.st2 {font-size: 14px; font-family: times new roman; color: blue; text-decoration: none}
```

Link SS ngoài



import SS ngoài

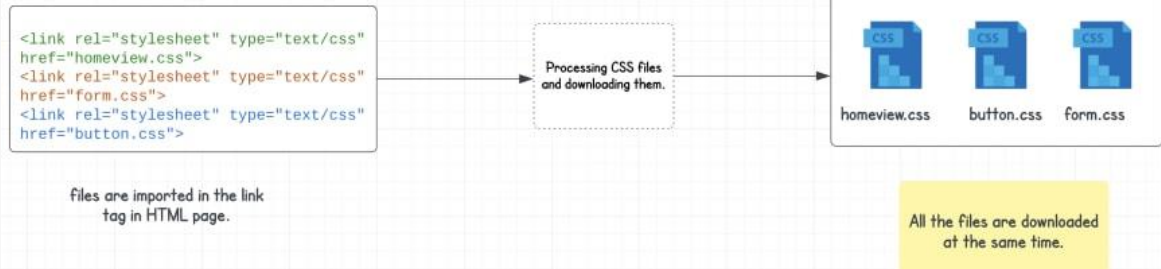
```
1 <!DOCTYPE html><html><head>
2   <title>Trang thứ 2</title>
3   <meta charset="utf-8">
4   <style type="text/css">
5     @import url("style.css");
6   </style>
7 </head><body>
8   <h2>Sử dụng Style Sheet</h2>
9   <table>
10    <tr>
11      <td class="st1">Ô của bảng được áp dụng CSS</td>
12    </tr>
13  </table>
14  <a href="" class="st2">Liên kết
15    được áp dụng CSS</a>
16</body></html>
```



<link> vs @import

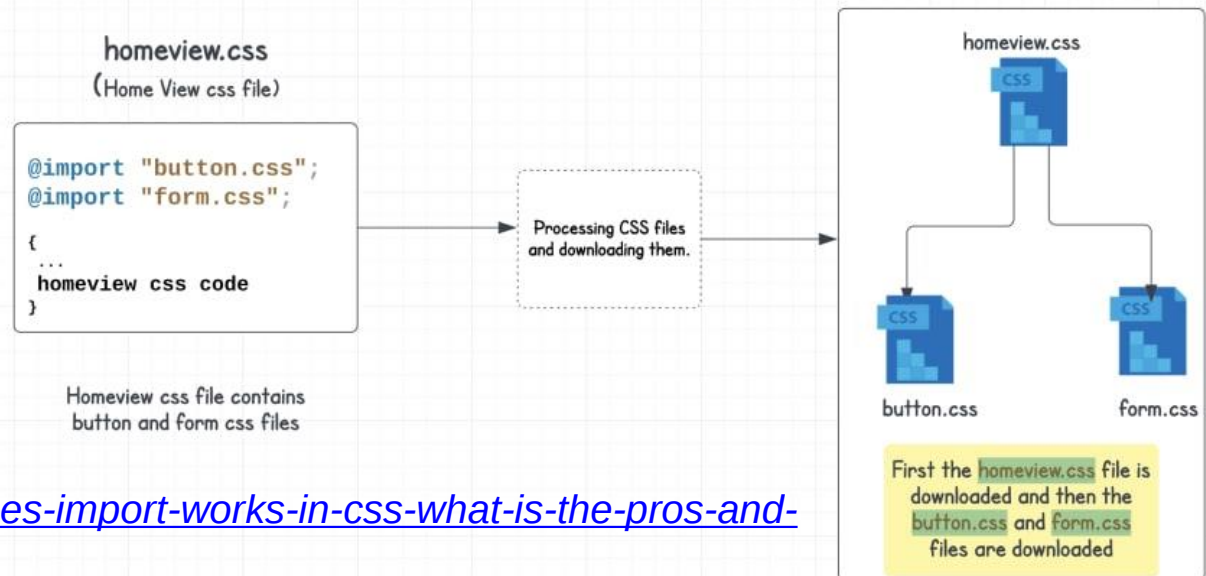
<link>: Nhập các luật từ một SS vào trang/tài liệu.

How downloading of scripts happens while using link element?



@import: Nhập các luật từ SS này vào SS khác.

How downloading of scripts happens while using imports?



<https://dev.to/jasmin/how-does-import-works-in-css-what-is-the-pros-and-cons-136c>

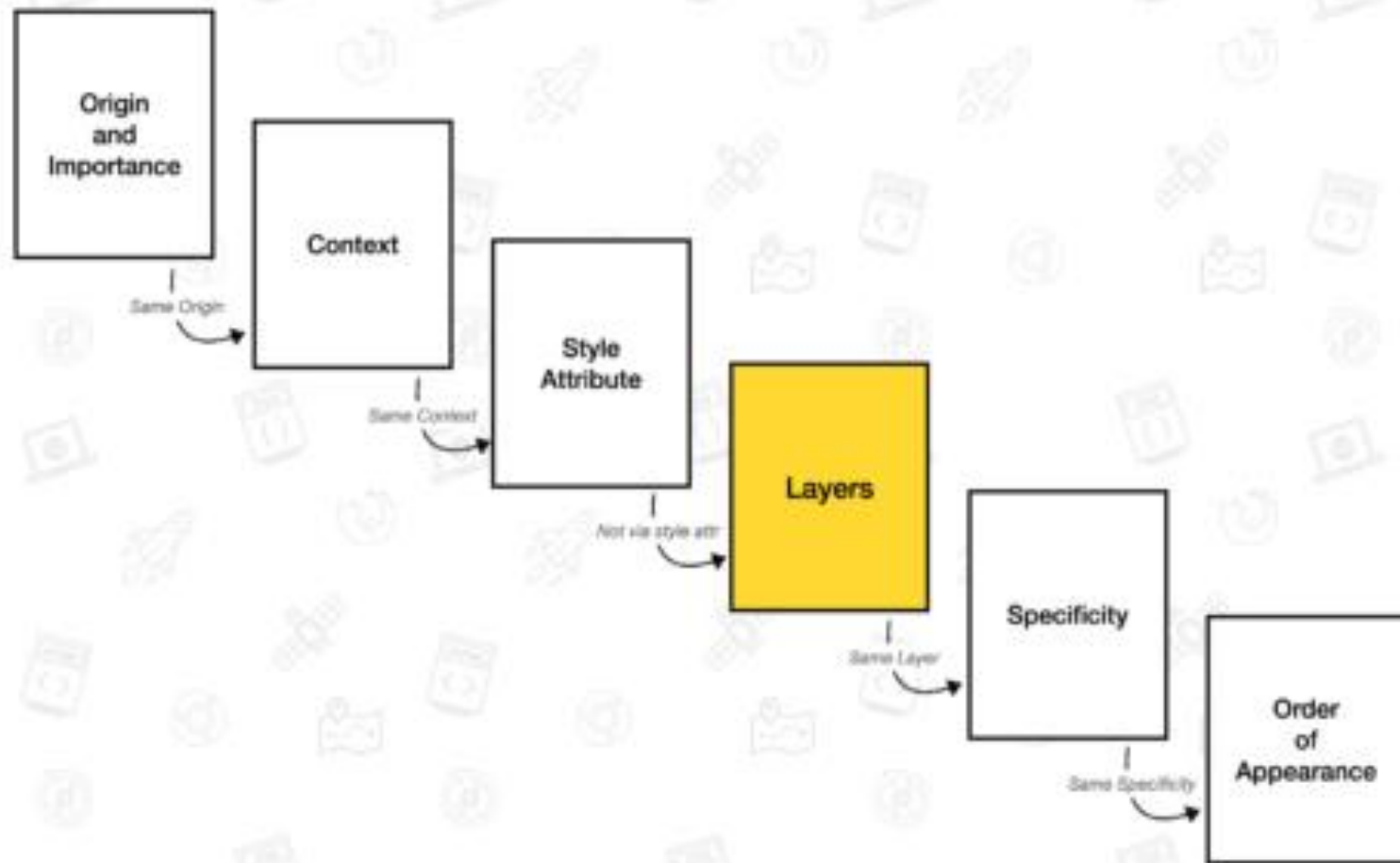
Cascading

- Mục đích
 - Giải quyết xung đột (**competing declarations**)
 - Nhiều khai báo cùng áp dụng cho một thuộc tính của một đối tượng nhưng giá trị của các khai báo khác nhau => sử dụng giá trị nào cho thuộc tính?

Cascading

- Input
 - Danh sách không có thứ tự các khai báo áp dụng cho mỗi thuộc tính của mỗi đối tượng (**declared values**).
- Output
 - Danh sách có thứ tự các giá trị (**cascaded value**) của các khai báo áp dụng cho mỗi thuộc tính của mỗi đối tượng.
- Algorithm
 - Sắp xếp các khai báo theo mức độ ưu tiên của chúng (**declaration's precedence**)

Cascading



Nguồn: <https://www.bram.us/2021/09/15/the-future-of-css-cascade-layers-css-at-layer/>

Origin

- Các style sheet do **lập trình** viên xác lập được gọi là **Author Origin**.
- Trình duyệt cũng có các style sheet **sẵn có**, được gọi là **User-Agent Origin**.
- **Người dùng**, thông qua các extension của trình duyệt, cũng có thể thiết lập các style sheet, được gọi là **User Origin**.

Khai báo important

- Khai báo được thêm **!important** ở cuối
 - Nên hạn chế sử dụng

```
[hidden] { display: none !important; }
```

Cascading

Origin and Importance

The [origin](#) of a declaration is based on where it comes from and its [importance](#) is whether or not it is declared with '[!important](#)' (see [below](#)). The precedence of the various origins is, in descending order:

1. Transition declarations [\[css-transitions-1\]](#)
2. [Important user agent](#) declarations
3. [Important user](#) declarations
4. [Important author](#) declarations
5. Animation declarations [\[css-animations-1\]](#)
6. [Normal author](#) declarations
7. [Normal user](#) declarations
8. [Normal user agent](#) declarations

Declarations from [origins](#) earlier in this list win over declarations from later origins.

Cascading

Context

A document language can provide for blending declarations sourced from different ***encapsulation contexts***, such as the nested tree contexts of shadow trees in the [DOM].

When comparing two declarations that are sourced from different encapsulation contexts, then for normal rules the declaration from the outer context wins, and for important rules the declaration from the inner context wins. For this purpose, [DOM] tree contexts are considered to be nested in shadow-including tree order.

Cascading

Element-Attached Styles

Separately for normal and important declarations, declarations that are attached directly to an element (such as the contents of a style attribute) rather than indirectly mapped by means of a style rule selector take precedence over declarations the same importance that are mapped via style rule.

Cascading

Layers

Declarations within each [origin](#) and [context](#) can be explicitly assigned to a [cascade layer](#). For the purpose of this step, any declaration not assigned to an explicit layer is added to an implicit final layer.

Cascade layers (like declarations) are ordered by order of appearance. When comparing declarations that belong to different layers, then for [normal](#) rules the declaration whose [cascade layer](#) is last wins, and for [important](#) rules the declaration whose cascade layer is first wins.

Cascading

Specificity

The [Selectors module \[SELECT\]](#) describes how to compute the specificity of a selector. Each declaration has the same specificity as the style rule it appears in. The declaration with the highest specificity wins.

Bộ chọn

Theo định danh

Theo thuộc tính, theo lớp, lớp giả

Theo phần tử, phần tử giả

*

Kết hợp

Tính cụ thể

<1, 0, 0>

<0, 1, 0>

<0, 0, 1>

<0, 0, 0>

Vector tổng

Ví dụ

ul#nav

<1, 0, 1>

ul#nav li.active a

<1, 1, 3>

Try it: <https://polypane.app/css-specificity-calculator/>

Old: <https://css-tricks.com/specifcs-on-css-specificity/>

Cascading

Tryit Editor v3.5

New tab

Secure | https://www.w3schools.com/html/tryit.asp?filename=tryhtml_default

Not interested in this ad

Ad knew too much

Ad was inappropriate

Ad covered content

Run »

Result Size: 307 x 341

```
<!DOCTYPE html><html><head><title>CSS</title>
<style type="text/css">
  label.note {color:red;}
  label {color:green;}
</style>
</head><body>
  <label class="note">Nhãn</label>
</body></html>
```

Nhãn

EN

3:54 PM
8/3/2017

Cascading

Order of Appearance

The last declaration in document order wins. For this purpose:

- Style sheets are ordered as in [final CSS style sheets](#).
- Declarations from [‘imported style sheets’](#) are ordered as if their style sheets were substituted in place of the [‘@import’](#) rule.
- Declarations from style sheets independently linked by the originating document are treated as if they were concatenated in linking order, as determined by the host document language.
- Declarations from style attributes are ordered according to the document order of the element the style attribute appears on, and are all placed after any style sheets.

Cascading

Tryit Editor v3.5

Secure | http://www.w3schools.com/html/tryit.asp?filename=tryhtml_default

Not interested in this ad | Ad knew too much | Ad was inappropriate | Ad covered content

Run »

Result Size: 307 x 341

```
<!DOCTYPE html><html><head><title>CSS</title>
<style type="text/css">
  label {color:red;}
</style>
<link rel="stylesheet" href="precedence.css"> <!-- label {color:yellow;} -->
</head><body>
  <label class="note">Nhãn</label>
</body></html>
```

Nhãn

Defaulting

When the [cascade](#) does not result in a value, the [specified value](#) must be found some other way. [Inherited properties](#) draw their defaults from their parent element through [inheritance](#); all other properties take their [initial value](#). Authors can explicitly request inheritance or initialization via the [‘inherit’](#) and [‘initial’](#) keywords.

Kế thừa

- Thuộc tính CSS có thể kế thừa (inherited) hoặc không
 - **inherited**: color, font-style, ...
 - **non-inherited**: width, border, ...

Kế thừa

```
<!DOCTYPE html><html><head>
<title>Page Title</title>
<style>
  div {
    color: red;
    border: solid 1px green;
  }
</style>
</head><body>
  <div>
    This is a div
    <p>This is a paragraph contained in a div.</p>
  </div>
  <p>This is a paragraph outside of any div.</p>
</body></html>
```

This is a div

This is a paragraph contained in a div.

This is a paragraph outside of any div.

Thay đổi hành vi mặc định

- Có thể thay đổi hành vi mặc định một cách rõ ràng (**Explicit Defaulting**) cho thuộc tính CSS bằng cách đặt giá trị:
 - **inherit**: kế thừa giá trị từ nút cha
 - **initial**: lấy giá trị mặc định của thuộc tính
 - **unset**: như inherit hoặc initial tùy thuộc vào thuộc tính naturally inherited hay không
 - **revert**: lấy giá trị từ nguồn trước
 - **revert-layer**: lấy giá trị từ CSS layer trước

Thay đổi hành vi mặc định

```
<!DOCTYPE html><html><head>
<title>Page Title</title>
<style>
  div {
    color: red;
    border: solid 1px green;
  }
  p {
    color: initial;
    border: inherit;
  }
</style>
</head><body>
  <div>
    This is a div
    <p>This is a paragraph contained in a div.</p>
  </div>
  <p>This is a paragraph outside of any div.</p>
</body></html>
```

This is a div

This is a paragraph contained in a div.

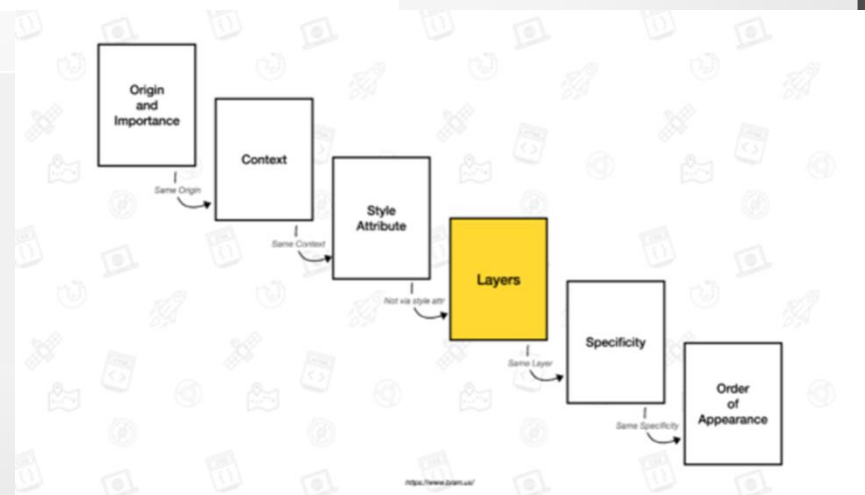
This is a paragraph outside of any div.

@layer

	🖥️					📱					
	Chrome 	Edge 	Firefox 	Opera 	Safari 	Chrome Android 	Firefox for Android 	Opera Android 	Safari on iOS 	Samsung Internet 	WebView Android 
@layer	✓ 99	✓ 99	✓ 97	✓ 85	✓ 15.4	✓ 99	✓ 97	✓ 68	✓ 15.4	✓ 18.0	✓ 99

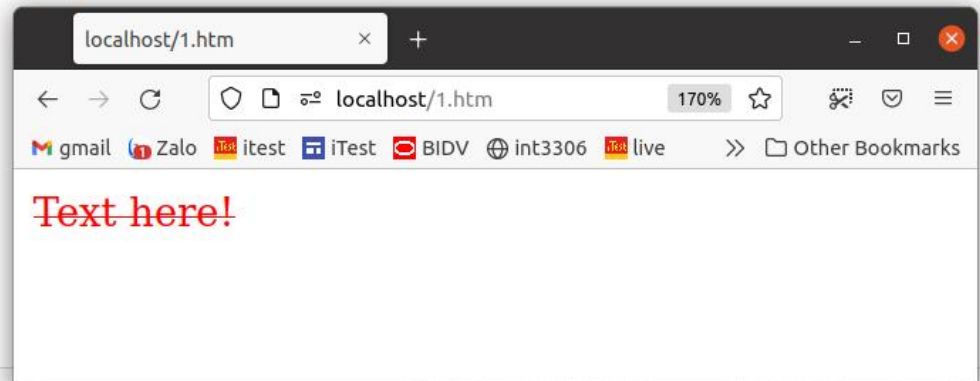
```

@layer layer-name {rules}
@layer layer-name;
@layer layer-name, layer-name, layer-name;
@layer {rules}
    
```



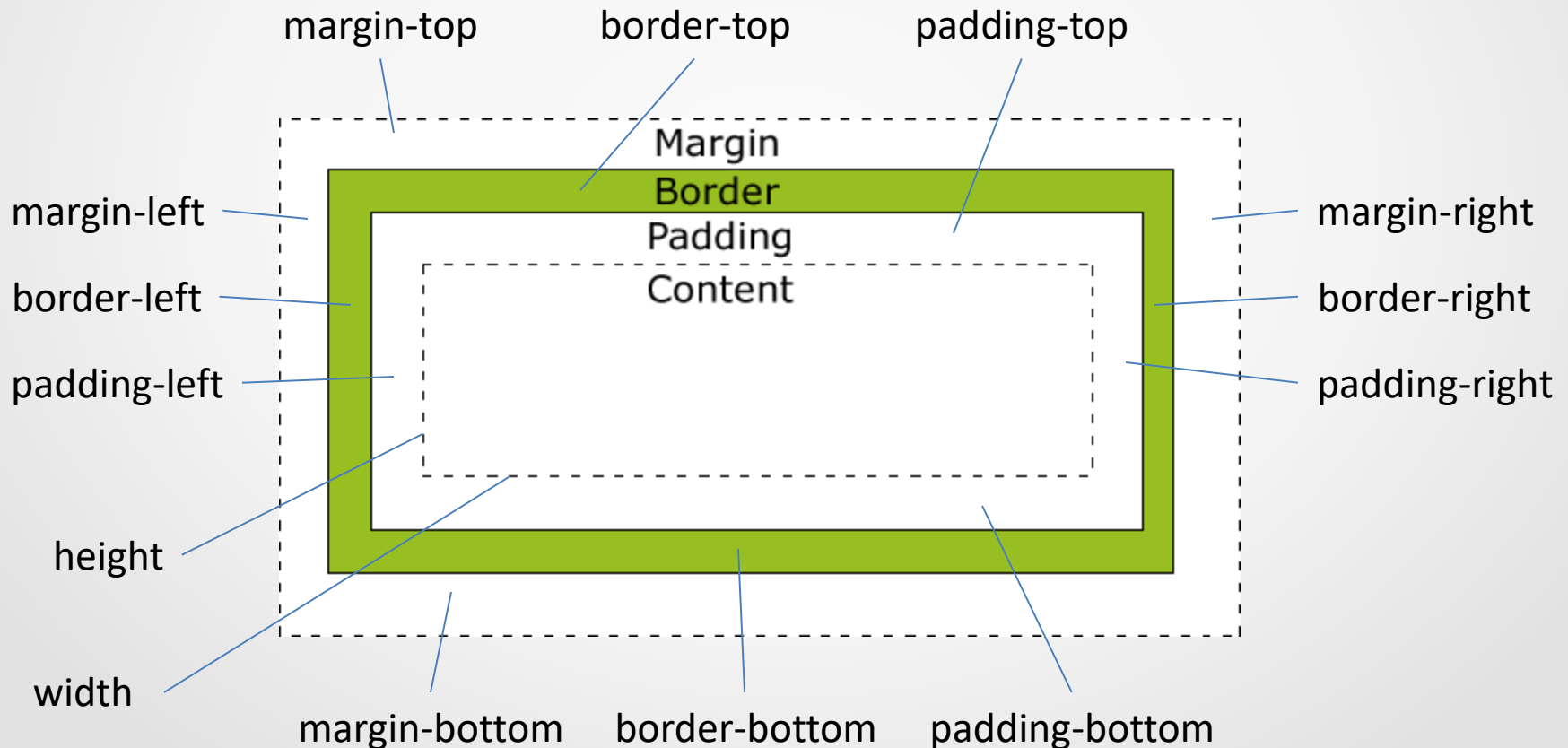
@layer

```
1 <!DOCTYPE html><html><head>
2   <meta charset="utf-8">
3   <style type="text/css">
4     .st0 {text-decoration:line-through;}
5     @layer a, b;
6     @layer b {
7       .st1 {color:red;text-decoration:underline;}
8     }
9     @layer a {
10      .st2 {color:blue;text-decoration:overline;}
11    }
12  </style>
13 </head><body>
14   <span class="st2 st1 st0">Text here!</span>
15 </body></html>
```



Hiển thị đối tượng tài liệu – Box model

- Mỗi đối tượng tài liệu được hiển thị trong một vùng hình chữ nhật (box model)



Box model - Vùng hiển thị nội dung

- Chiều cao và chiều rộng vùng hiển thị nội dung của đối tượng tài liệu thường được tính toán tự động để toàn bộ nội dung được nhìn thấy
- nhưng có thể ấn định bằng các thuộc tính

- `width, height`

- `max-width, max-height, min-width, min-height`

với giá trị được xác định bằng độ dài (pixel, pt, ...) hoặc phần trăm so với đối tượng chứa

Ví dụ `width:50%; height:10px;`

Box model - Vùng đệm

- Vùng đệm là vùng trống nằm giữa vùng hiển thị nội dung và đường viền, được xác định bởi các thuộc tính

`padding`, `padding-top`, `padding-bottom`,
`padding-right`, `padding-left`

với giá trị tính theo chiều dài hoặc phần trăm.

- Viết tắt:

`padding: top right bottom left`

`padding: top right(left) bottom`

`padding: top(bottom) right(left)`

`padding: top(right, bottom, left)`

Box model - Viền

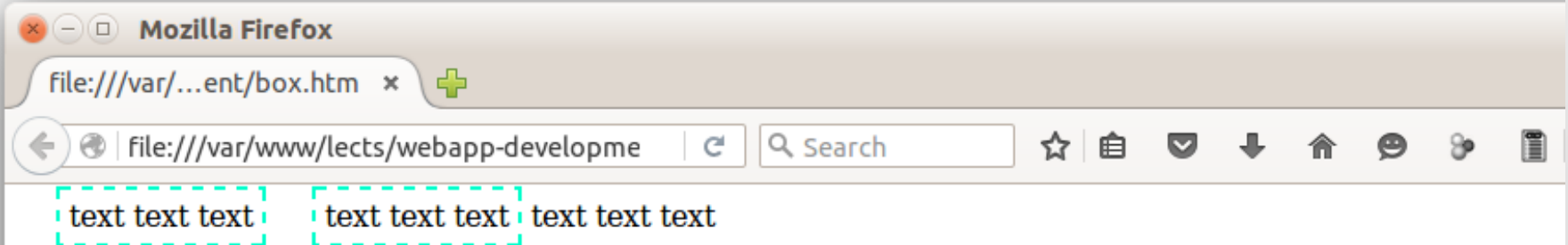
- border-[*side*-]style:
none/solid/dashed/dotted/double/groove/ridge/inset/outset/hidden
- border- [*side*-]width: thin/medium/thick/#px
- border- [*side*-]color: color-value
- border[*-side*]: style width color

Box model - Lề

- Lề là khoảng trống bao bên ngoài
- `margin[-side]: auto/inherit/#%/#px`

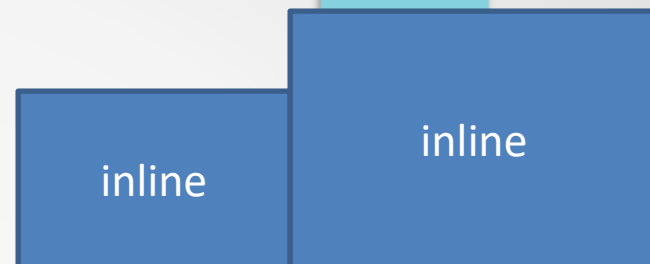
Box model

```
box.htm x
1 <!DOCTYPE html><html><head>
2     <style type="text/css">
3         label {
4             border:dashed 2px #00ffcc;
5             padding:5px;
6             margin-left:20px;
7         }
8     </style>
9 </head><body>
10    <label>text text text</label>
11    <label>text text text</label>
12    text text text
13 </body></html>
```



Hiển thị theo dòng và theo khối

- Kiểu hiển thị
 - `display: none/inline/block;`
- Khả năng nhìn thấy
 - `visibility: hidden;`
- Độ trong suốt
 - `opacity: 0.4;`
- collapsing-margins
 - trong một vài tình huống, lề trên và lề dưới của các hộp liền kề được kết hợp thành một lề



Các đối tượng có thể đứng cạnh nhau

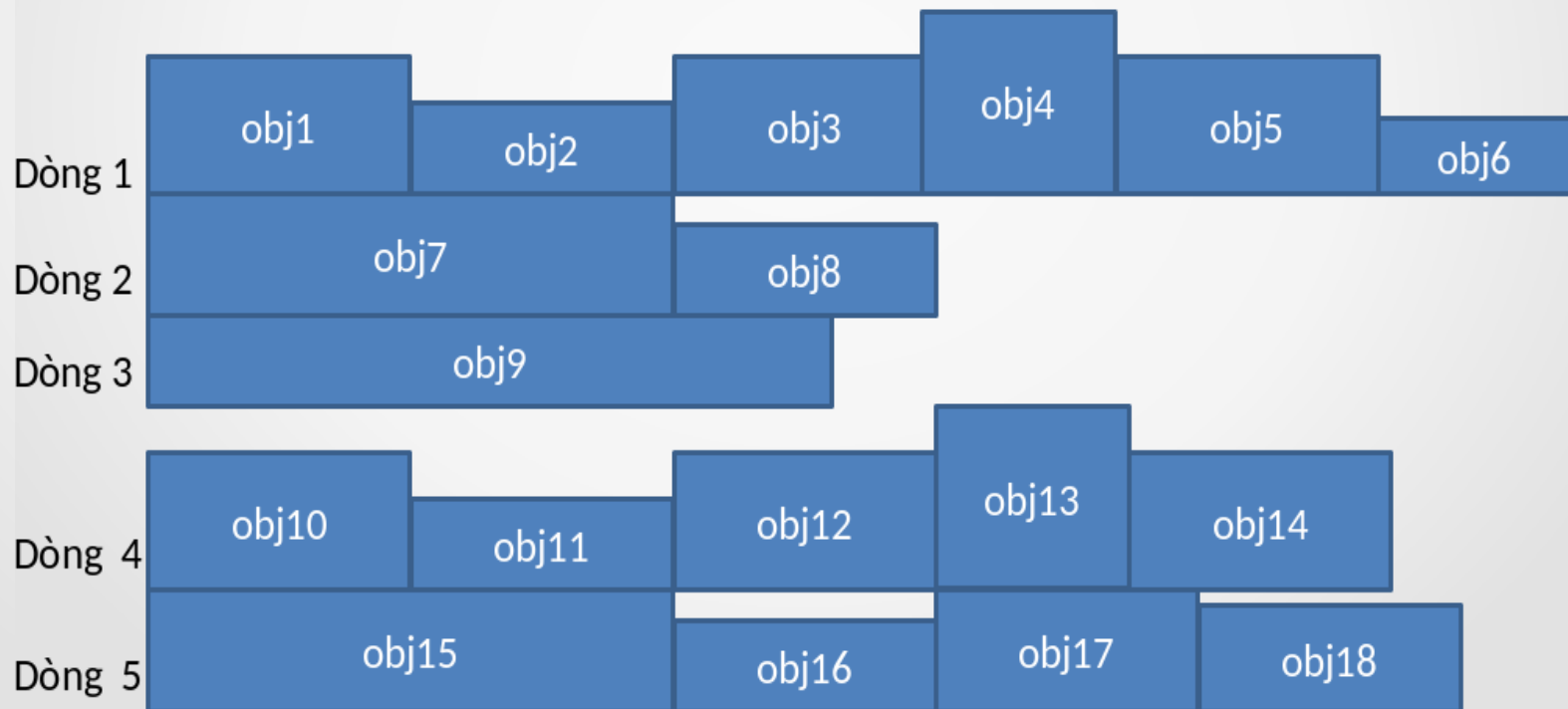


Đối tượng không thể đứng cạnh đối tượng khác

Vị trí hiển thị đối tượng

- **position:static**

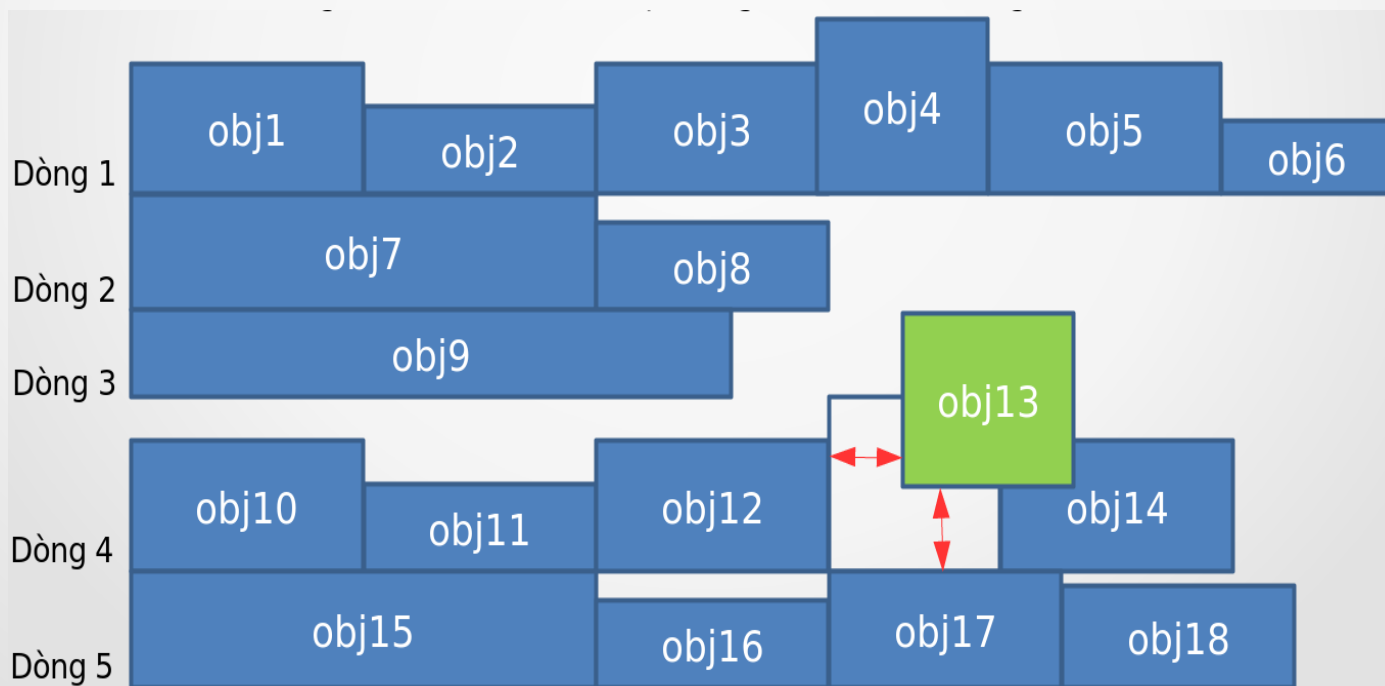
- Các đối tượng liên tiếp nhau được hiển thị theo **luồng bình thường (normal flow)**, tức là cạnh nhau từ trái sang phải rồi từ trên xuống dưới (đối tượng có kiểu hiển thị block không thể hiển thị bên cạnh (liền trái, liền phải) các đối tượng khác)



Vị trí hiển thị đối tượng

- **position:relative**

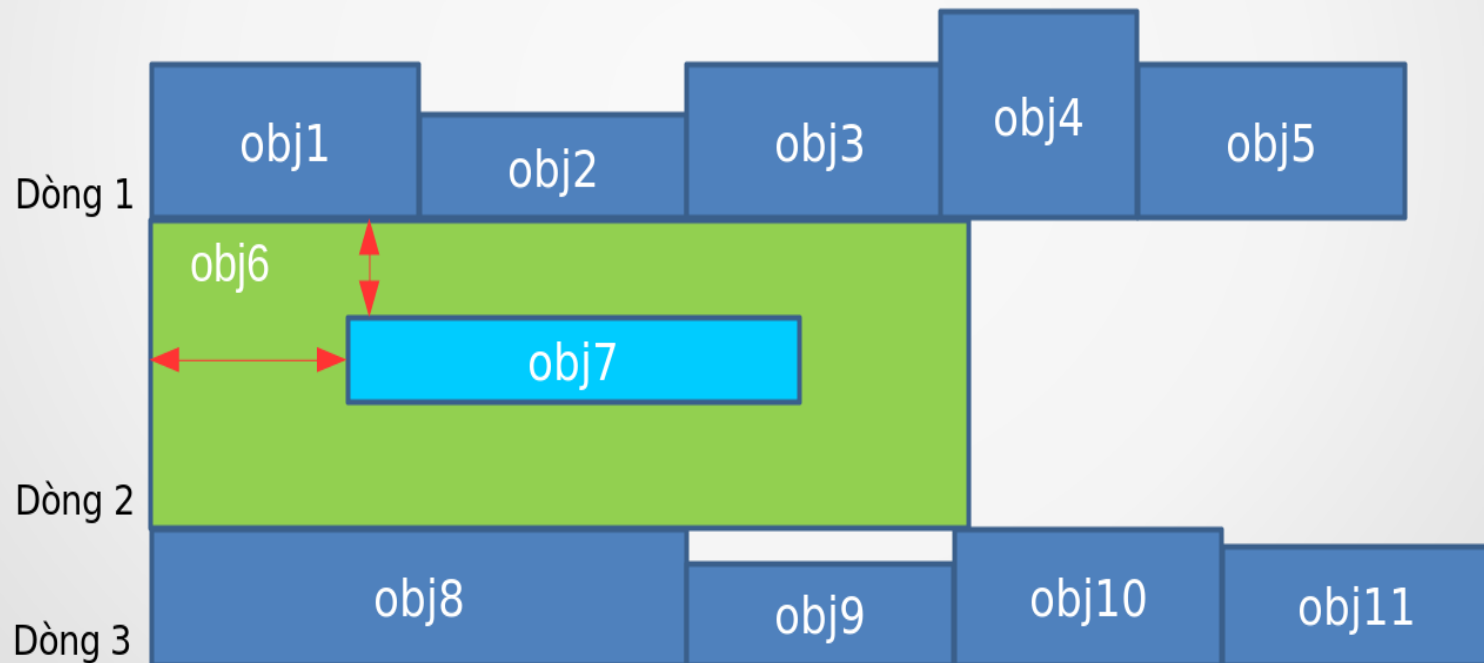
- Vị trí hiển thị đối tượng có thể được di rời trái (left), phải (right), lên trên (top), xuống dưới (bottom) so với vị trí tĩnh/trong luồng của nó.
- Vùng hiển thị trong luồng vẫn được giữ chỗ
- Đối tượng di rời có thể hiển thị chồng lên các đối tượng khác



Vị trí hiển thị đối tượng

- **position:absolute**

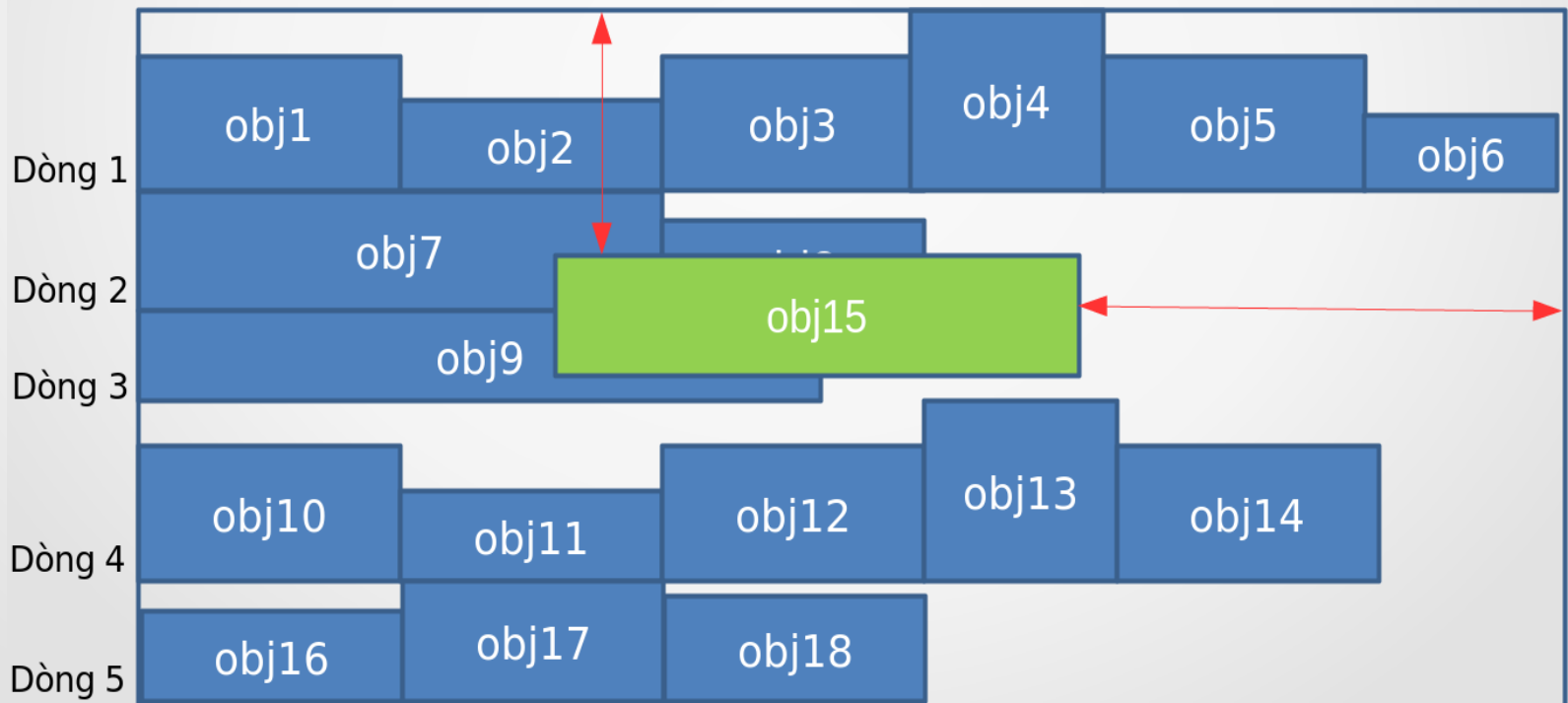
- Vị trí hiển thị đối tượng được xác định tương đối so với vị trí tổ tiên gần nhất không theo luồng tự nhiên.



Vị trí hiển thị đối tượng

- **position:fixed**

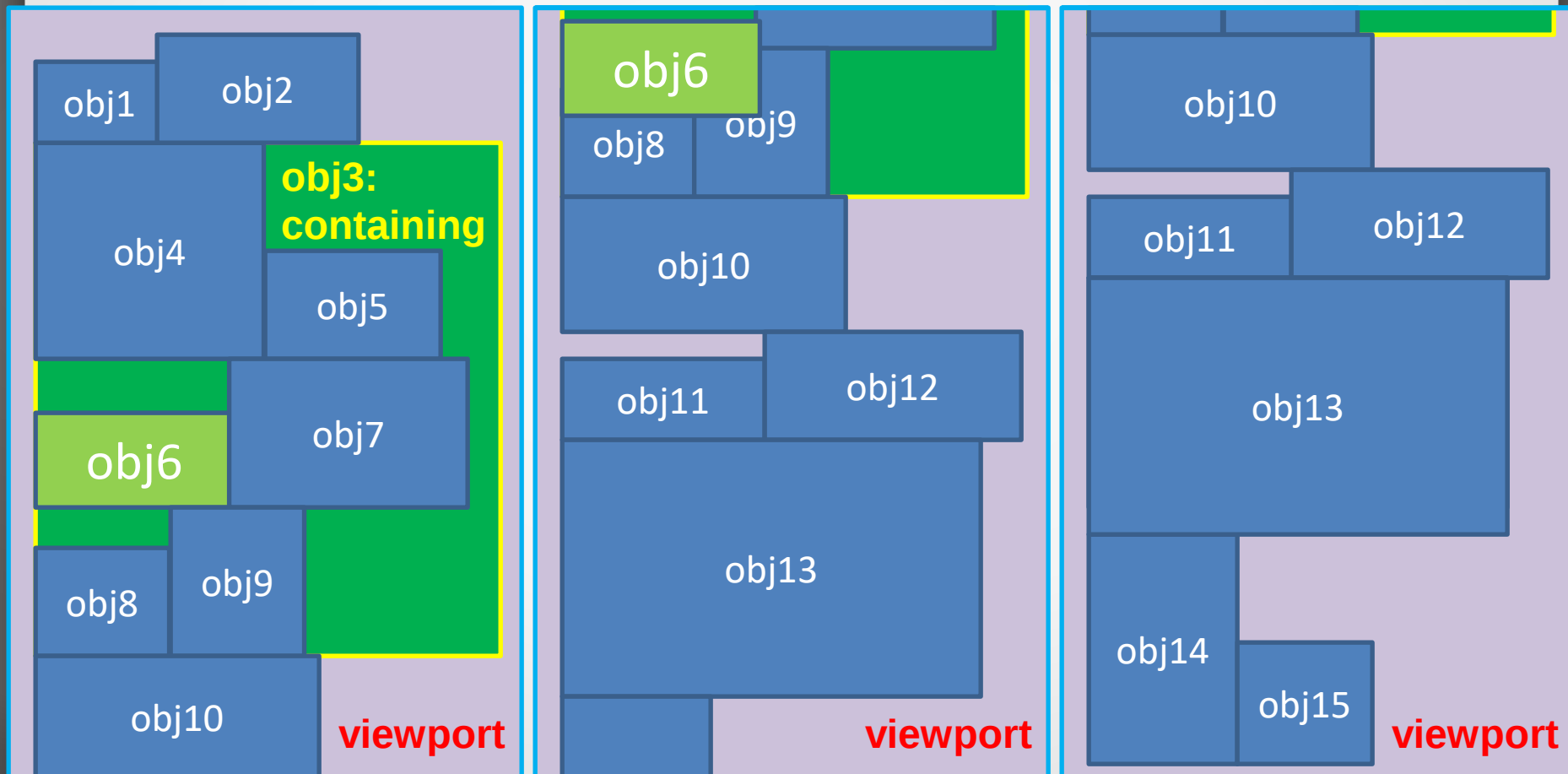
- Vị trí hiển thị đối tượng được xác định tương đối so với vị trí cửa sổ, không thay đổi khi cửa sổ được cuộn
- Đối tượng hiển thị cố định có thể hiển thị chồng lên các đối tượng khác



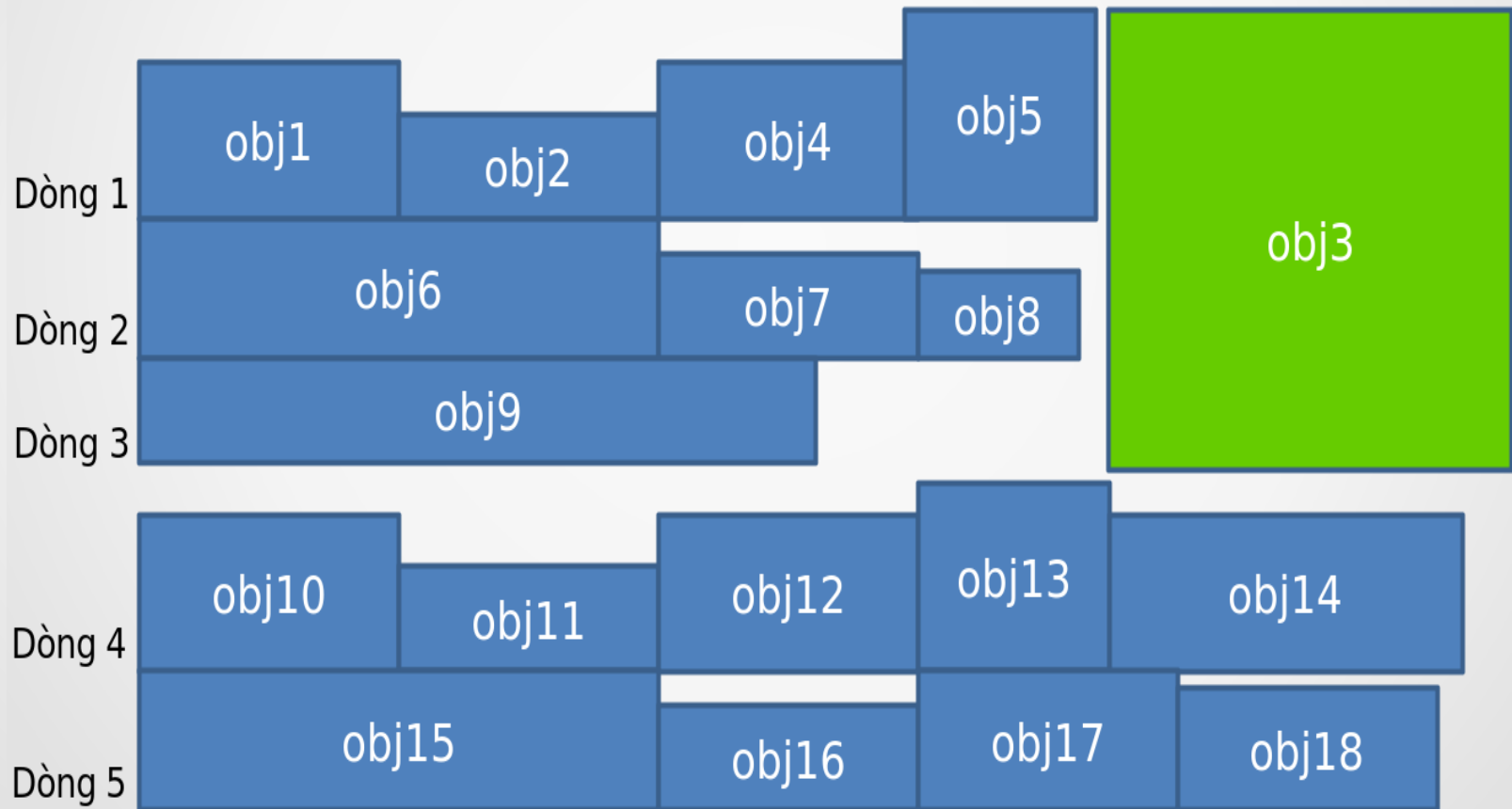
Vị trí hiển thị đối tượng

- position:sticky**

Như **position:fixed** khi hộp theo luồng tự nhiên nằm ngoài viewport NHƯNG giao (union) của viewport với hộp của đối tượng chứa còn đủ để hiển thị đối tượng. Các trường hợp khác, như **position:relative** với độ dời 0



Vị trí hiển thị đối tượng: Đặt trôi



Vị trí hiển thị đối tượng: Đặt trôi

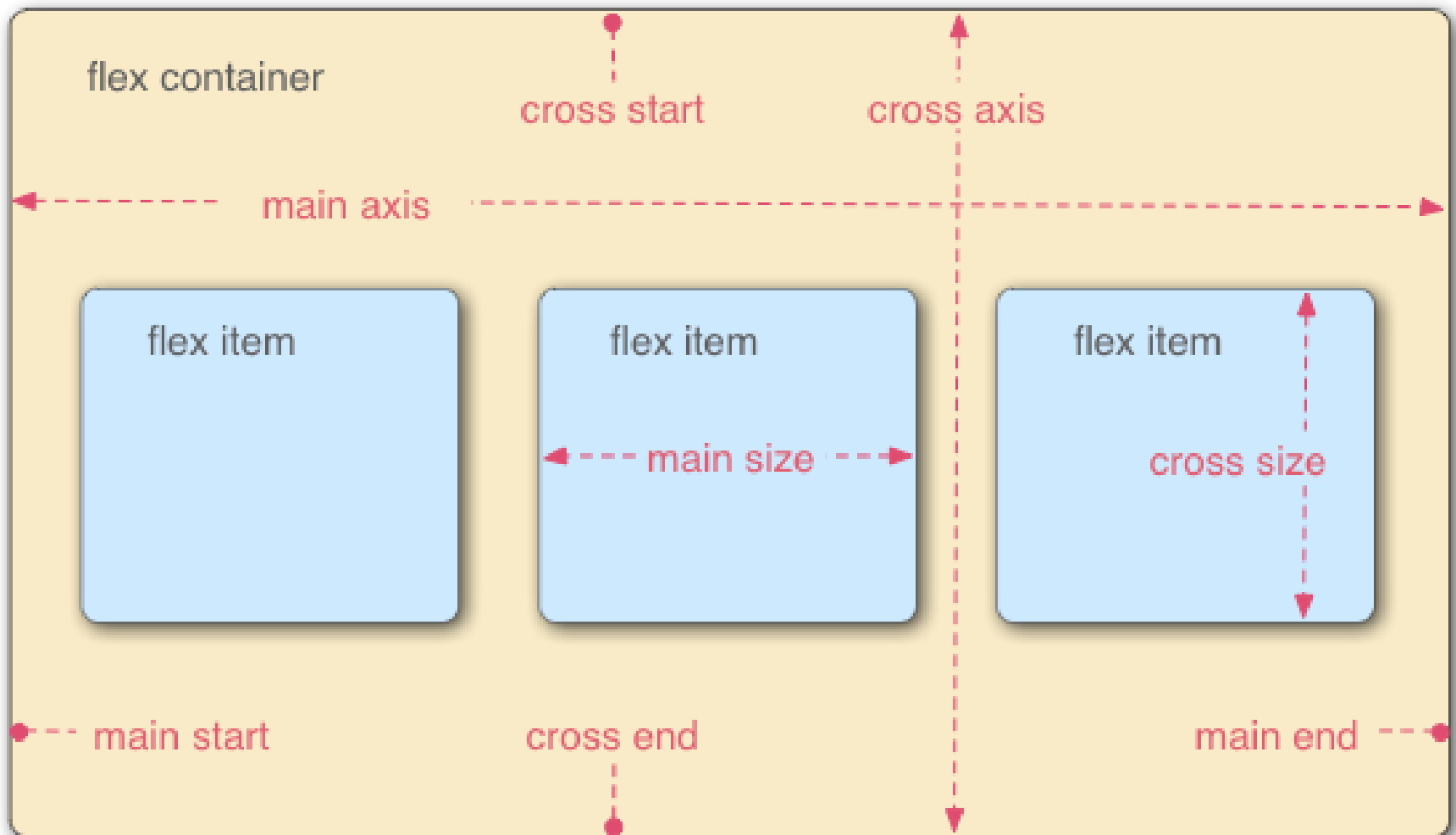
- `float: left/right;`
- Đối tượng được đặt trôi sẽ trôi về một phía (trái/phải) xa nhất có thể
- Các đối tượng sau đối tượng đặt trôi chảy theo đối tượng được đặt trôi.
- Sử dụng `clear: left/right/both;` để xóa trôi.

Flexbox

- Dàn trang **một chiều** (one-dimensional layout): ngang hoặc dọc
 - ✓ Các hộp linh động
 - giãn ra để phủ hết không gian nếu rộng
 - co lại nếu không gian hẹp

Tham khảo: https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox

Flexbox



Đối tượng chứa (Flex container)

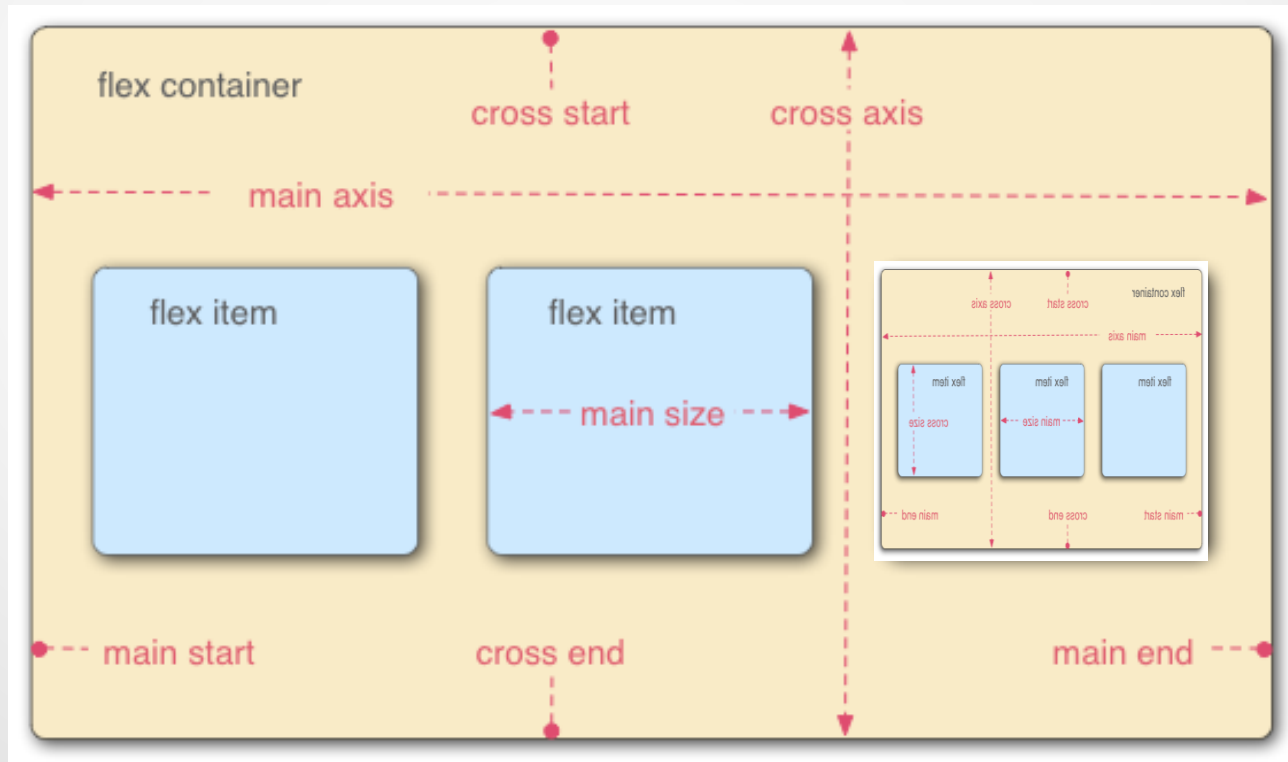
- `display: flex | inline-flex;` // Bắt buộc
- `flex-direction: row | row-reverse | column | column-reverse ;` // Các phần tử con được hiển thị theo hàng từ trái qua phải (mặc định) | theo hàng từ phải qua trái | theo cột từ trên xuống dưới | theo cột từ dưới lên trên
- `flex-wrap: wrap;` // cho phép sang dòng/cột mới nếu không đủ không gian
- `justify-content: space-around | ...;` // căn lề các phần tử theo trục chính
- `align-items: center | ...;` // căn lề các phần tử theo trục chéo/phụ

Các đối tượng con (Flex item)

- Kích thước theo trục chính
 - `flex: 200px;` // ít nhất bao nhiêu pixel
 - `flex: 3;` // chiếm mấy phần
- Thứ tự
 - `order: 2;`

Flexbox lồng nhau

- Các flexbox có thể lồng nhau
 - ✓ Flex item đồng thời là flex container



Ví dụ

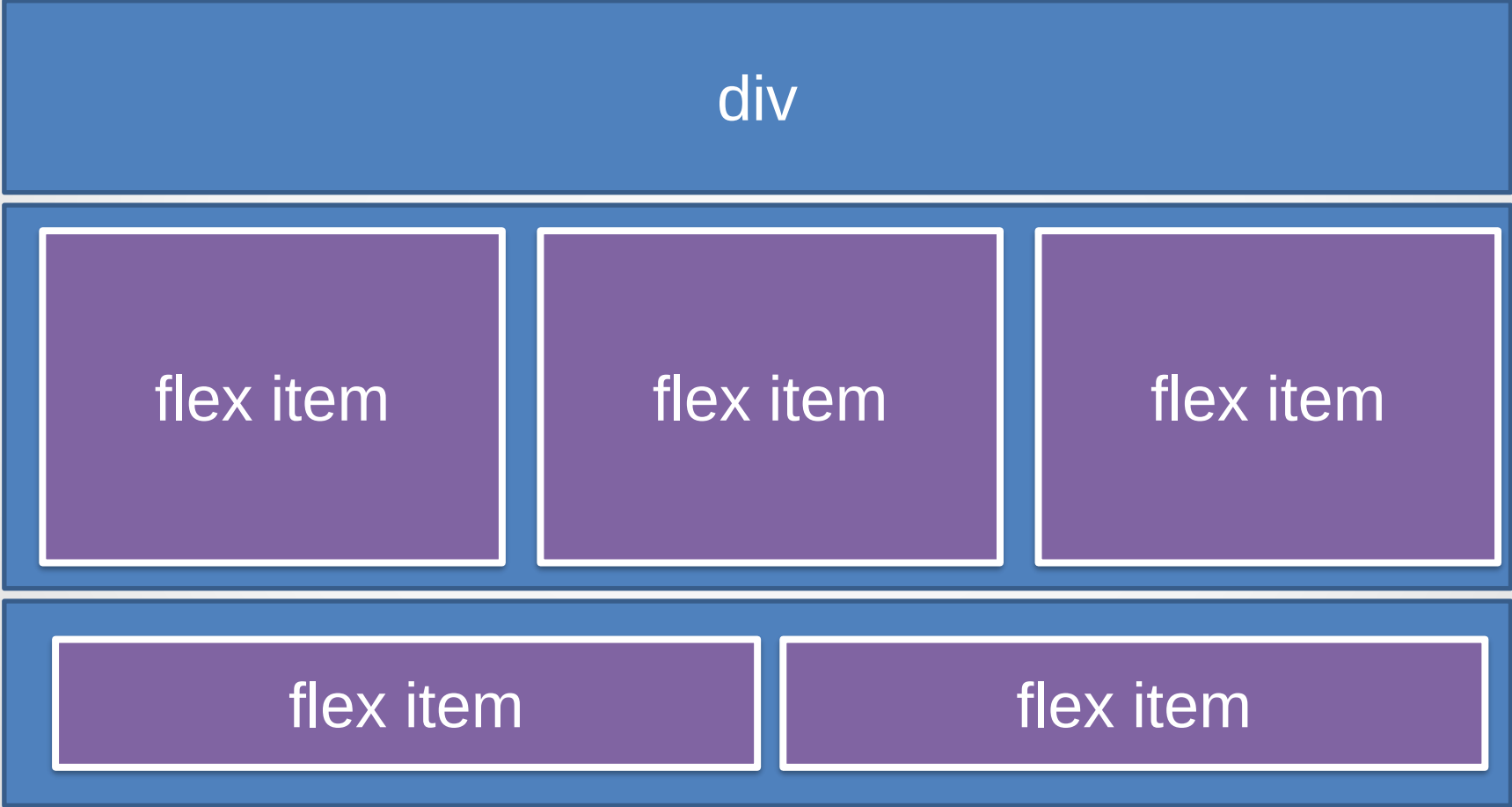
- Truy cập:
<https://itest.com.vn/lects/webappdev/layout/flexbox0.htm>
- Thay đổi chiều rộng cửa sổ và quan sát các bài viết được dàn trang theo flex như thế nào, đặc biệt chú ý các bài viết cuối cùng

Thực hành tốt nhất

- Layout từ trên xuống dưới
 - `<div>` không cần thêm CSS
 - hiển thị block theo mặc định
- Layout từ trái qua phải
 - flexbox để tạo hàng ngang

Thực hành tốt nhất

div

A diagram illustrating a flex container layout. A large blue rectangle labeled 'div' contains two rows of purple rectangles labeled 'flex item'. The first row contains three 'flex item' boxes of equal width, and the second row contains two 'flex item' boxes of equal width. The boxes are arranged in a grid-like fashion, with the first row having three items and the second row having two items.

flex item

flex item

flex item

flex item

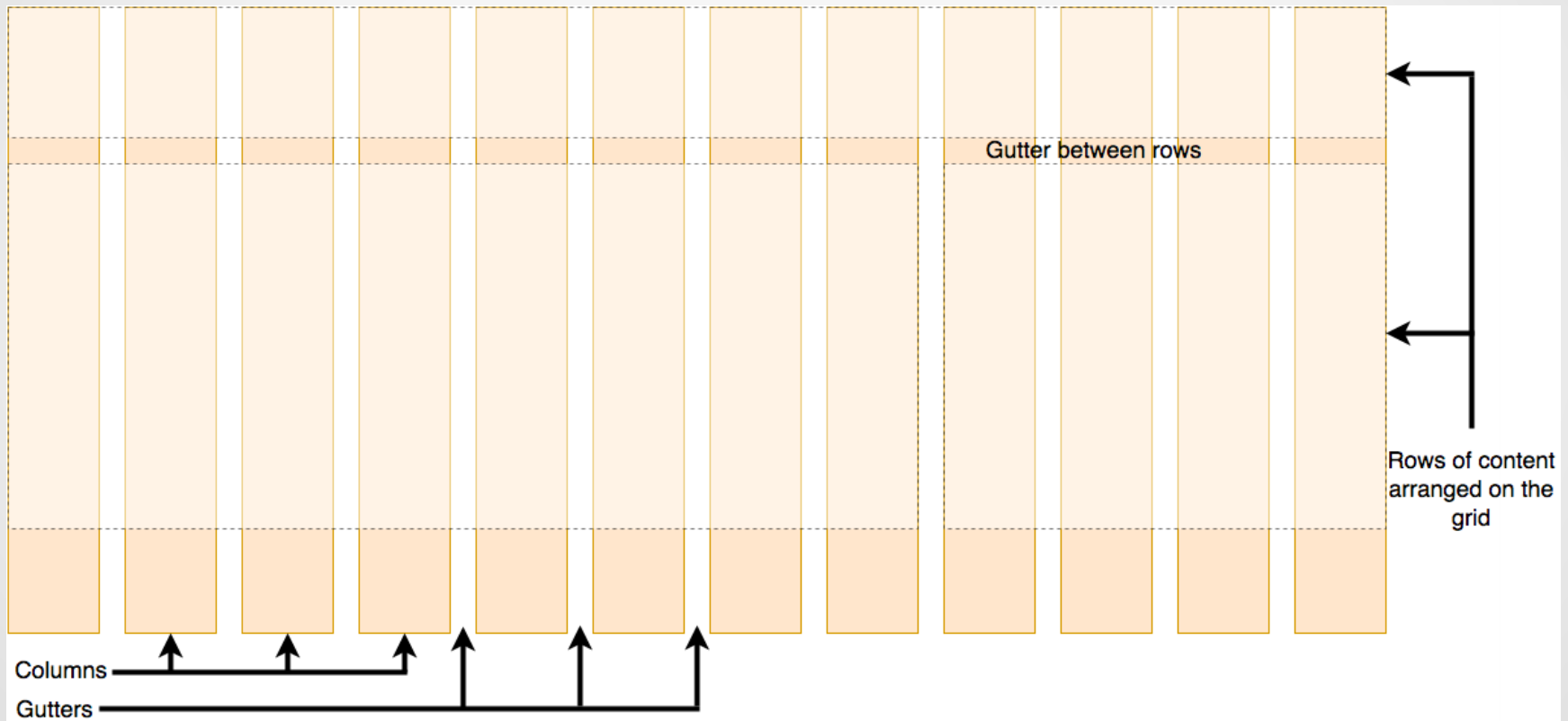
flex item

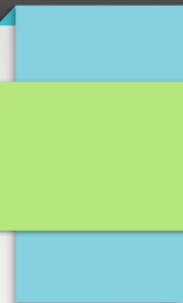
Grids

- Dàn trang hai chiều theo dòng và cột
- Tham khảo:

https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Grids

Grids





Grids

- Container

- Định nghĩa lưới

- Số hàng, cột tùy ý
 - Kích thước từng hàng, cột tùy ý

- Item

- Xác định vùng hiển thị trên lưới

- Hình chữ nhật trải trên các hàng liên tiếp và các cột liên tiếp

- ❖ Hàng bắt đầu
 - ❖ Hàng kết thúc
 - ❖ Cột bắt đầu
 - ❖ Cột kết thúc

Grid container

- `display: grid;` // Bắt buộc
- Định nghĩa các cột/hàng một cách rõ ràng.
 - `grid-template-columns: 1fr 4fr 1fr;`
 - `grid-template-columns: repeat(5, 1fr 2fr);`
- Các hàng/cột sẽ được định nghĩa ngầm khi đặt các item lên grid
 - `grid-gap: 20px;` // Kích thước gutter, khoảng cách giữa các ô
 - `gap: 20px;`

Grid item

- Hình chữ nhật chiếm các ô trên lưới
 - Liên tiếp từ cột nào đến cột nào
 - `grid-column: c | start / end;`
 - `grid-column-start: c;`
 - `grid-column-end: c;`
 - Liên tiếp từ hàng nào đến hàng nào
 - `grid-row: c | start / end;`
 - `grid-row-start: c;`
 - `grid-row-end: c;`

Thực hành tốt nhất

- Ngay hàng thẳng lối
 - Dùng grid
 - Layout từ trên xuống dưới
 - ✓ Các hàng trên lưới
 - Layout từ trái qua phải
 - ✓ item trải nhiều ô liên tiếp trên cùng hàng
 - **Tốt hơn là dùng flexbox**
 - Các hàng độc lập với nhau => dễ quản lý hơn
- Phá cách
 - Dùng grid

Ngay hàng thẳng lối - Ví dụ

This is my lovely blog

Other things

Nam vulputate diam nec tempor bibendum. Donec luctus augue eget malesuada ultrices. Phasellus turpis est, posuere sit amet dapibus ut, facilisis sed est.

My article

Duis felis orci, pulvinar id metus ut, rutrum luctus orci. Cras porttitor imperdiet nunc, at ultricies tellus laoreet sit amet. Sed auctor cursus massa at porta. Integer ligula ipsum, tristique sit amet orci vel, viverra egestas ligula. Curabitur vehicula tellus neque, ac ornare ex malesuada et. In vitae convallis lacus. Aliquam erat volutpat. Suspendisse ac imperdiet turpis. Aenean finibus sollicitudin eros pharetra congue. Duis ornare egestas augue ut luctus. Proin blandit quam nec lacus varius commodo et a urna. Ut id ornare felis, eget fermentum sapien.

Nam vulputate diam nec tempor bibendum. Donec luctus augue eget malesuada ultrices. Phasellus turpis est, posuere sit amet dapibus ut, facilisis sed est. Nam id risus quis ante semper consectetur eget aliquam lorem. Vivamus tristique elit dolor, sed pretium metus suscipit vel. Mauris ultricies lectus sed lobortis finibus. Vivamus eu urna eget velit cursus viverra quis vestibulum sem. Aliquam tincidunt eget purus in interdum. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

Contact me@mysite.com

ABC Company

Ngay hàng thẳng lối - Ví dụ

- Lưới gồm 3 hàng, 4 cột
- header: Toàn bộ hàng 1
- aside: 1 ô bên trái hàng 2
- article: 3 ô bên phải hàng 2
- footer: hàng 3
 - nửa trái: 2 ô bên trái hàng 3
 - nửa phải: 2 ô bên phải hàng 3

Ngay hàng thẳng lối - Ví dụ - DOM

- .container
 - header
 - article
 - aside
 - footer
 - div

Ngay hàng thẳng lối - Ví dụ - CSS

```
.container {  
    display: grid;  
    grid-template-columns: repeat(4, 1fr);  
    grid-gap: 20px;  
}
```

Ngay hàng thẳng lối- Ví dụ - CSS

header	{ grid-column: 1 / 5; grid-row: 1;}
article	{ grid-column: 2 / 5; grid-row: 2; }
aside	{ grid-column: 1; grid-row: 2; }
footer	{ grid-column: 1 / 3; grid-row: 3; }
.container > div	{ grid-column: 3 / 5; grid-row: 3; }

Ngay hàng thẳng lối- Ví dụ - Hoàn thiện

Tham khảo

<https://itest.com.vn/lects/webappdev/layout/grid0.htm>

Thiết kế phá cách

- Các boxes có thể trải ra trên nhiều dòng và cột khác nhau, không ngay hàng thẳng lối.
 - Tạo grid
 - Đặt các items trên grid , mỗi item có thể trải trên nhiều hàng

Thiết kế phá cách - Ví dụ

This is my lovely blog

Main

Duis felis orci, pulvinar id metus ut, rutrum luctus orci. Cras porttitor imperdiet nunc, at ultricies tellus laoreet sit amet. Sed auctor cursus massa at porta. Integer ligula ipsum, tristique sit amet orci vel, viverra egestas ligula. Curabitur vehicula tellus neque, ac ornare ex malesuada et. In vitae convallis lacus. Aliquam erat volutpat. Suspendisse ac imperdiet turpis. Aenean finibus sollicitudin eros pharetra congue. Duis ornare egestas augue ut luctus. Proin blandit quam nec lacus varius commodo et a urna. Ut id ornare felis, eget fermentum sapien.

Second

Nam vulputate diam nec tempor bibendum. Donec luctus augue eget malesuada ultrices. Phasellus turpis est, posuere sit amet dapibus ut, facilisis sed est.

Eighth

Seventh

Sixth

Fifth

Fourth

Third

Contact me@mysite.com

Thiết kế phá cách - Ví dụ - DOM

- div.container
 - header
 - div.main
 - div.second
 - div.b1
 - div.b2
 - div.b3
 - div.b4
 - div.b5
 - dib.b6
 - footer

Thiết kế phá cách - Ví dụ - CSS

```
.container {  
    display: grid;  
    grid-template-columns: repeat(12, 1fr);  
    grid-gap: 20px;  
}
```

```
header { grid-column: 1 / 13; grid-row: 1; }  
footer { grid-column: 1 / 13; grid-row: 12; }
```

Thiết kế phá cách - Ví dụ - CSS

```
div.main {  
  grid-column: 1/7;  
  grid-row: 2/5;  
}
```

```
div.second {  
  grid-column: 7/11;  
  grid-row: 2/3;  
}
```

Thiết kế phá cách - Ví dụ - CSS

```
div.b1 { grid-column: 1 / 3;      grid-row: 7/12;  }  
div.b2 { grid-column: 3 / 5;      grid-row: 6/12;  }  
div.b3 { grid-column: 5 / 7;      grid-row: 5/12;  }  
div.b4 { grid-column: 7 / 9; grid-row: 4/12; }  
div.b5 { grid-column: 9 / 11; grid-row: 3/12;      }  
div.b6 { grid-column: 11 / 13; grid-row: 2/12;      }
```

Thiết kế phá cách - Ví dụ - Hoàn thiện

Tham khảo

<https://itest.com.vn/lects/webappdev/layout/grid1.htm>

Multiple-column

- Dàn trang **một chiều** (one-dimensional layout): ngang hoặc dọc

Tham khảo: https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox

Cao độ

- Đối tượng có z-index cao hơn sẽ che/nằm trước đối tượng có z-index nhỏ hơn khi các vùng hiển thị của chúng giao nhau.

Giá trị màu

- Tên: “red”, ...
- RGB: “rgb(255, 0, 0)”, ...
- Hexa: “#ff0000”

Định dạng nền

- background-color: #6495ed;
- background-image:url('paper.gif');
- background-repeat:repeat-x;
- background-attachment:fixed;
- background-position:right top;
- background:#ffffff url('img_tree.png') no-repeat right top;

Định dạng văn bản

- `color:#00ff00;`
- `text-align:center/left/right/justify;`
- `vertical-align:bottom/top/middle;`
- `text-decoration:none/underline/ line-through/ overline/blink;`
- `text-transform:uppercase/ lowercase/ capitalize;`
- `text-indent:50px;`

Định dạng font chữ

- font-family:"Times New Roman", Times, serif;
- font-size:40px;
- font-style:normal/italic/bold;

Định dạng danh sách

- `list-style-type: none/ circle/ square/ upper-roman/ lower-alpha/...;`
- `list-style-image: url('sqpurple.gif');`

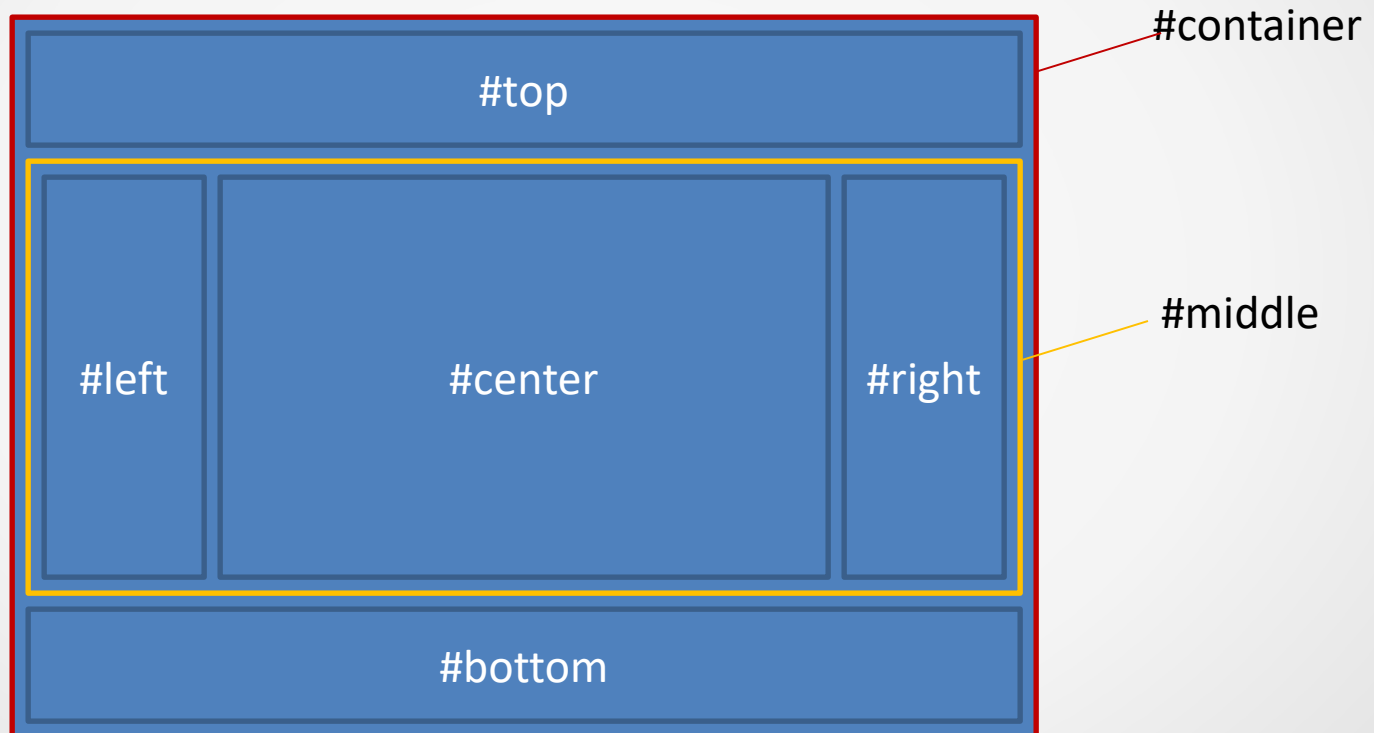
Định dạng bảng

- `border-collapse: collapse;`

Hiển thị một miếng ảnh

- `background:url(img_navsprites.gif) 0 0;`
- `width:46px;`
- `height:44px;`

Ví dụ: Dàn trang



Ví dụ: Dàn trang

HTML

```
<div id="container">
  <div id="top"></div>
  <div id="middle">
    <div id="left"></div>
    <div id="center"></div>
    <div id="right"></div>
    <br class = "spacer"/>
  </div>
  <div id="bottom"></div>
</div>
```

CSS

```
#left, #right, #center {
  display: inline;
  float:left;
}
.spacer {clear:both;}
```

Tạo viền tròn góc

- `border:2px solid;`
- `border-radius:25px;`
- `-moz-border-radius:25px; /* Old Firefox */`

Tạo bóng

- `box-shadow: 10px 10px 5px #888888;`

Sử dụng ảnh làm viền

- `border-image:url(border.png) 30 30 round;`
- `-moz-border-image:url(border.png) 30 30 round; /* Old Firefox */`
- `-webkit-border-image:url(border.png) 30 30 round; /* Safari and Chrome */`
- `-o-border-image:url(border.png) 30 30 round; /* Opera */`

Định dạng nền

- `background:url(img_flwr.gif);`
- `background-repeat:no-repeat;`
- `background-size:100% 100%;`
- `-webkit-background-origin:content-box; /*
Safari */`
- `background-origin:content-box;`

Hiệu ứng văn bản

- `text-shadow: 5px 5px 5px #FF0000;`
- `word-wrap: normal | break-word;`
- `word-break: normal | break-all | hyphenate;`
- `text-overflow: clip | ellipsis | string;`

Biến đổi 2D

- `transform: rotate(30deg);`
- `-ms-transform: rotate(30deg); /* IE 9 */`
- `-webkit-transform: rotate(30deg); /* Safari and Chrome */`
- `-o-transform: rotate(30deg); /* Opera */`
- `-moz-transform: rotate(30deg); /* Firefox */`
- `transform: translate(50px,100px);`
- `-ms-transform: translate(50px,100px); /* IE 9 */`
- `-webkit-transform: translate(50px,100px); /*`

Biến đổi 3D

- `transform: rotateX(120deg);`
- `-webkit-transform: rotateX(120deg); /* Safari and Chrome */`
- `-moz-transform: rotateX(120deg); /* Firefox */`
- `transform: rotateY(130deg);`
- `-webkit-transform: rotateY(130deg); /* Safari and Chrome */`
- `-moz-transform: rotateY(130deg); /* Firefox */`

Chuyển kiểu

- transition: property1 time1, property2 time2, property3 time3;
- -moz-transition: width 2s, height 2s, -moz-transform 2s;
- -webkit-transition: width 2s, height 2s, -webkit-transform 2s;
- -o-transition: width 2s, height 2s, -o-transform 2s;

Hoạt cảnh

- Định nghĩa các cảnh trong hoạt cảnh
 - @keyframes **kfname**
 - {
 - 0% {background: red;}
 - 25% {background: yellow;}
 - 50% {background: blue;}
 - 100% {background: green;}
 - }
- Sử dụng hoạt cảnh
 - animation-name: **kfname**;
 - animation-duration: 5s;

Tiếp theo
Thiết kế đáp ứng

