

Định dạng bằng CSS

NGUYỄN THỊ THÙY LIÊN

KHOA CNTT – ĐH PHENIKAA

LIEN.NGUYENTHITHUY@PHENIKAA-UNI.EDU.VN

CSS (Cascading Style Sheet):



Mẫu quy định cách thức thể hiện các thẻ HTML

Dùng để trình bày các phần tử trong HTML trên trình duyệt.

Giúp tiết kiệm được rất nhiều thời gian và công sức cho việc thiết kế web.

Có thể định nghĩa nhiều style vào một thẻ HTML (Cascading), giúp tải trang nhanh hơn

CSS có thể được đặt trong văn bản HTML hay ở một file khác

Khai báo và sử dụng style

Các loại style

Có 4 loại style:

Inline Style (Style được qui định trong 1 thẻ HTML cụ thể)

Internal Style (Style được qui định trong phần <head> của 1 trang HTML)

External Style (style được qui định trong file .CSS ngoài)

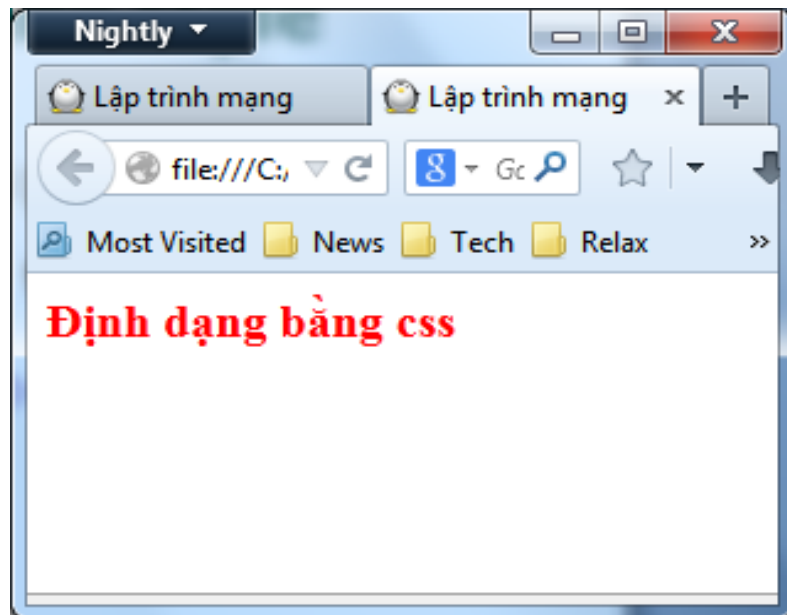
Browser Default (thiết lập mặc định của trình duyệt)

Thứ tự ưu tiên: Mức ưu tiên giảm dần từ trái sang phải

Cách chèn CSS – inline style

Đặt css trong thẻ mở với cú pháp

`style="tt1:gt1; tt2:gt2;..."`



`<h1 style="color: red; font-size: 20px"> Định dạng bằng css </h1>`

- Độ ưu tiên cao nhất
- Giới hạn áp dụng cho thẻ được chèn
- Nặng cho file HTML
- Không có tính linh động
- Lộn xộn giữa nội dung và trình bày

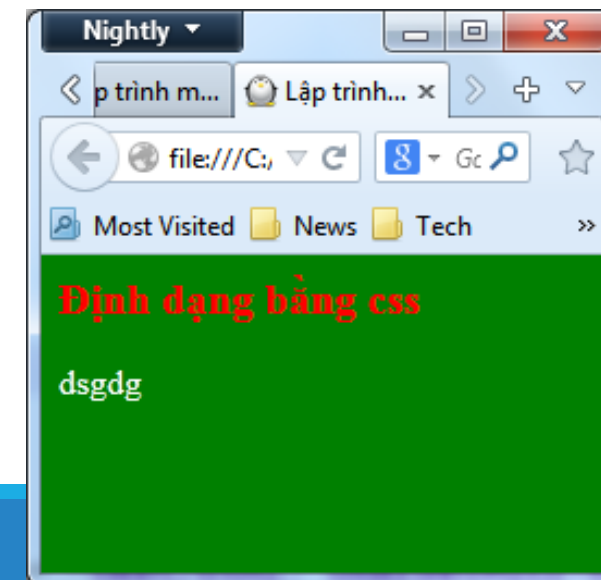
Cách chèn CSS – Internal style

Đặt trong **<head>...</head>**

Với **Internal style**:

- Dùng được cho nhiều thẻ
- Giới hạn áp dụng trong trang

```
2 <head>
3   <title>Lập trình mạng</title>
4   <style type="text/css">
5       body{
6           background: green;
7       }
8       p{
9           color:#ffffff;
10      }
11  </style>
```



Cách chèn CSS – External style

Định nghĩa style trong file riêng (thường có đuôi .CSS)

Nhúng file CSS đã định nghĩa vào trang web:

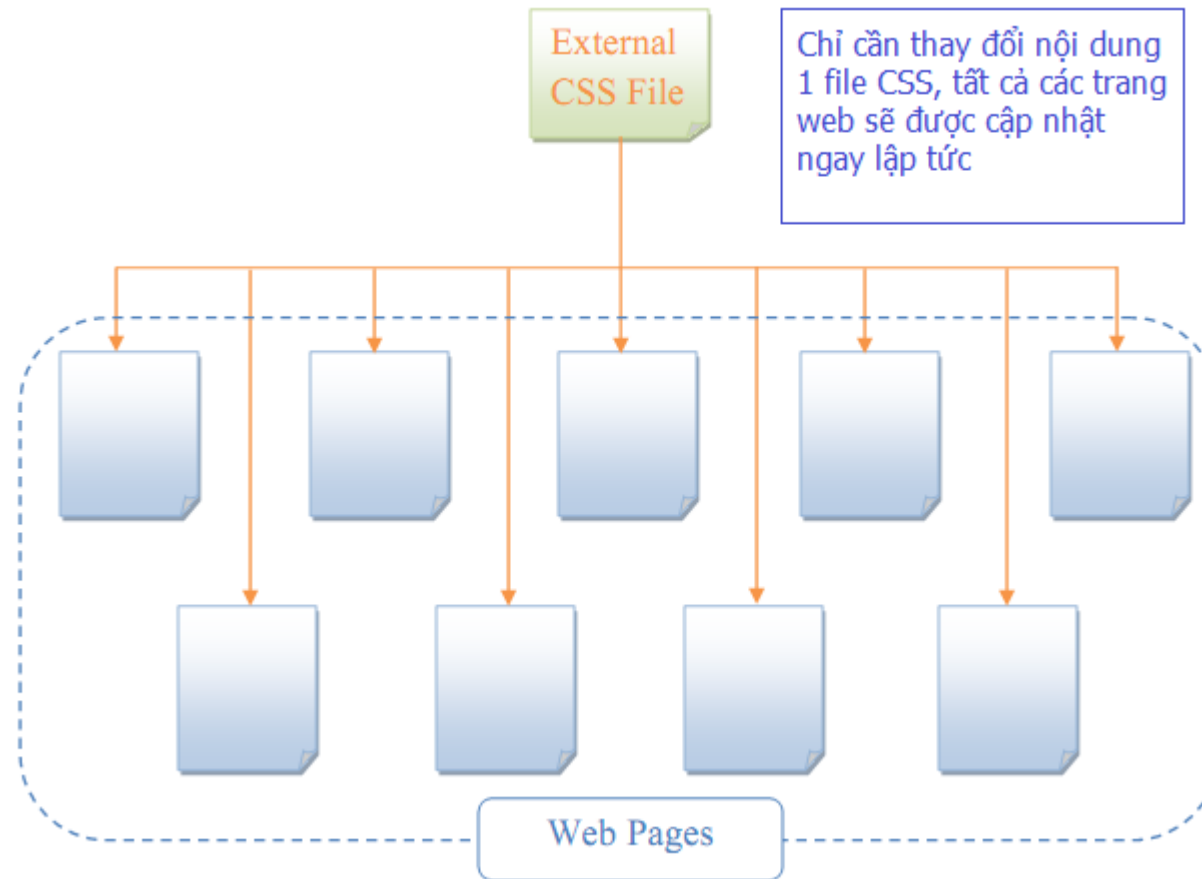
```
<link href="địa chỉ file"  
      rel="stylesheet" type="text/css">
```

```
<head>  
  <title>Lập trình mạng</title>  
  <meta http-equiv="content-type" content = "text/html; charset=utf-8">  
  <meta name="title" content = "Lập trình mạng">  
  <link href="images/icon.ico" rel = "shortcut icon" type="image/x-icon">  
  <link rel="stylesheet" href="css/system.css" type="text/css">  
  <script type="text/javascript" src="js/common.js"></script>  
</head>
```

Cách chèn CSS – External style

Hữu ích cho việc đồng bộ hay bảo trì website lớn

- Có tính kế thừa
- Áp dụng toàn bộ các trang trong website
- Thiết kế, chỉnh sửa dễ dàng
- Áp dụng được nhiều file CSS



Độ ưu tiên

Browser default < External < Internal < Inline style

Trong cùng 1 file giá trị khai báo sau ưu tiên hơn giá trị khai báo trước

Selector sử dụng id có mức độ ưu tiên cao hơn class, thẻ

Selector được viết càng rõ mức độ ưu tiên càng cao.

Sử dụng **!important** để tăng mức độ ưu tiên

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

Chú ý khi viết style

Style phân biệt chữ hoa, chữ thường

Để ghi chú trong style sử dụng:

`/ *`

`Đoạn ghi chú`

`* /`

Cú pháp CSS

Cú pháp cơ bản của CSS gồm 3 phần

- **Selector** (phần tử được hiển thị)
- **Property** (thuộc tính)
- **Value** (giá trị)

Selector { property : value }

Ví dụ:

```
body {color: black}  
p {font-family : "Tahoma"}
```

```
selector {  
    Property1: Value1;  
    Property2: Value2;  
}
```

Selectors

Tên thẻ: các khai báo CSS được áp dụng cho tất cả các thẻ được chỉ định

```
p {  
  color: red;  
}
```

Khai báo đồng thời nhiều thẻ: các khai báo CSS được áp dụng đồng thời cho nhiều thẻ được chỉ định

```
h1, h2, h3, h4, h5, h6 {  
  font-family: arial;  
}
```

Selectors

Tên thẻ và thuộc tính: các khai báo CSS được áp dụng cho những thẻ được chỉ định và có thuộc tính thỏa mãn điều kiện nào đó.

```
Input[name="search"] {  
    width: 200px  
}
```

Seletor đại diện cho tất cả các đối tượng

```
* {  
    margin: 0px;  
    padding : 0px}
```

Selectors - class

Thêm thuộc tính class vào trong thẻ mở

`<ten_thẻ class="tên_class">`

Ví dụ:

`<p class="red-text">Đoạn này màu đỏ</p>`

`<p class="blue-text">Đoạn này màu xanh</p>`

`<h2 class="blue-text">Tiêu đề màu xanh</h3>`

Selectors - class

Gắn với thẻ cụ thể: `tên_thẻ.tên_class`

```
p.red-text{  
  color:red;  
}  
p.blue-text{  
  color:blue;  
}
```

Không gắn với thẻ cụ thể:

```
.loai3{  
  color:green;  
}
```

Selectors - ID

Giá trị của các ID là duy nhất trên trang

Thêm thuộc tính **id** trong thẻ mở

```
<tên_thẻ id="định_danh">
```

Ví dụ:

```
<p id="red-text">Đoạn này màu đỏ</p>
```

```
<h1 id="blue-text">Tiêu đề xanh</h1>
```


Selectors - ID

Gắn với thẻ cụ thể: **tên_thẻ#định_danh{...}**

Không gắn với thẻ cụ thể: **#định_danh{...}**

Ví dụ:

```
p#red-text {  
    color : red;  
}  
#blue-text {  
    color : blue;  
}
```

Selectors

Selector	Ví dụ	Mô tả
.class	.intro	Phần tử có class = intro
#id	#firstname	Phần tử có id="firstname"
*	*	Tất cả các phần tử
element	p	Tất cả các thẻ <p>
element, element	div, p	Tất cả các thẻ <div> và thẻ <p>
element element	div p	Thẻ <p> nằm trong thẻ <div>

Selectors

Selector	Ví dụ	Mô tả
[attribute]	[target]	Phần tử có thuộc tính target
[target=_blank]	[target=_blank]	Phần tử có thuộc tính target=_blank
[<i>attribute</i> ~= <i>value</i>]	[title~=flower]	Phần tử có thuộc tính title có chứa từ flower
Element > element	div > p	Thẻ <p> có cha trực tiếp là thẻ <div>
Element + element	div + p	Thẻ <p> ngay sau thẻ <div>
Element ~ element	div ~ p	Thẻ <p> sau thẻ <div>

Kế thừa thuộc tính

Các phần tử con kế thừa thuộc tính CSS của phần tử cha

Một số thuộc tính cho phép được kế thừa

- color
- font-family
- text-align
- font-size
- line-height
- list-style

Một số thuộc tính không được kế thừa

- width
- border
- padding
- float
- height
- margin
- position
- ...

Đơn vị đo chiều dài trong CSS

Đơn vị	Mô tả	Đơn vị	Mô tả
%	Phần trăm	vh	1vh sẽ bằng 1% chiều cao của màn hình hiển thị
cm	centimeter	vw	1vw sẽ bằng 1% chiều rộng của màn hình hiển thị
in	inch(1 inch = 2.54 cm)	vmin	1% màn hình tối thiểu
mm	millimeter	vmax	1% màn hình tối đa
pc	pica (1pc = 12pt)	calc()	Đây là tính toán đơn vị, ví dụ: calc(100% - 20px)
px	Pixels (điểm ảnh trên màn hình máy tính)	em	1em ngang bằng với kích cỡ font hiện hành, 2em = 2 lần kích cỡ font hiện hành.
		rem	1rem bằng với kích cỡ font chữ của phần tử cha đầu tiên (html).
pt	point (1pt = 1/72inch)	ex	1ex bằng chiều cao của chữ x in thường của font hiện hành.

Đơn vị trong CSS

Đơn vị màu sắc

Đơn vị	Mô tả
Color-name	Tên màu tiếng Anh. Ví dụ: black, white, red, green, blue, cyan, magenta,...
RGB (r,g,b)	Màu RGB với 3 giá trị R, G, B có trị từ 0 – 255 kết hợp với nhau tạo ra vô số màu.
RGB (%r,%g,%b)	Màu RGB với 3 giá trị R, G, B có trị từ 0 – 100% kết hợp.
Hexadecimal RGB	Mã màu RGB dạng hệ thập lục. Ví dụ: #FFFFFF: trắng, #000000: đen, #FF00FF: đỏ tươi.

CSS Gradients

CSS3 Gradients giúp hiển thị màu dịch chuyển từ 2 hay nhiều màu khác nhau thành 1 dải màu.

2 loại

- Linear Gradients (goes down, up, left, right, diagonally)
- Radial gradients (center)
 - IE: 10 /
 - Chrome 26 /
 - firefox 16/
 - safari 6.1/
 - opera 12.1

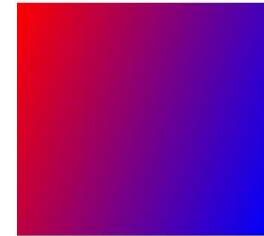


CSS Gradients

linear-gradient

background: **linear-gradient**(*direction, color-stop1, color-stop2, ...*);

- Direction: top/bottom/left/right hoặc sử dụng theo độ
- Ví dụ: background: linear-gradient(red, blue);
- background: linear-gradient(red, green, blue);
- background: linear-gradient(to right, rgba(255,0,0,0), rgba(255,0,0,1));

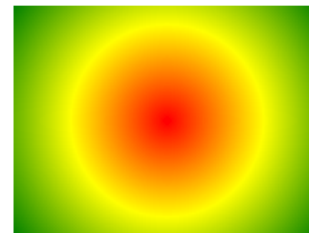
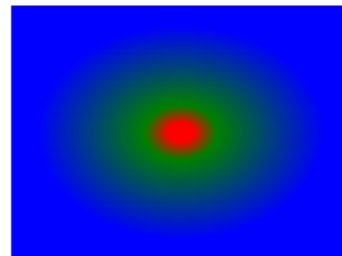
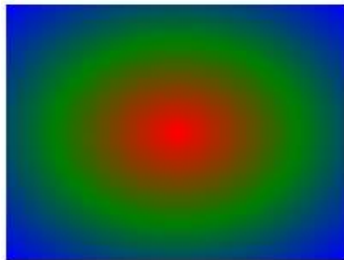


CSS Gradients

radial-gradient

background: **radial-gradient**(*shape size at position, start-color, ..., last-color*);

- background: radial-gradient(red, green, blue);
- background: radial-gradient(red 5%, green 15%, blue 60%);
- background: radial-gradient(circle, red, yellow, green);



Một số trường hợp cụ thể

CSS font

Font-family

- định nghĩa danh sách các font sẽ được dùng để hiển thị một thành phần trang web, nếu như trên máy tính truy cập chưa cài đặt font 1 thì font thứ 2 trong danh sách sẽ được sử dụng...

2 loại tên font:

- Family-name: tên cụ thể VD: arial, tahoma, verdana...
- Generic families: tên một họ gồm nhiều font: VD: serif, sans-serif,...

VD: `h1{font-family: arial, verdana, "Times New Roman"}`

CSS font

Sans-serif

Serif

Serif
(red serifs)

5 họ font:

- Serif
- Sans-serif
- Monospace
- Cursive
- Fantasy

Generic Font Family	Examples of Font Names
Serif	Times New Roman Georgia Garamond
Sans-serif	Arial Verdana Helvetica
Monospace	Courier New Lucida Console Monaco
Cursive	<i>Brush Script MT</i> <i>Lucida Handwriting</i>
Fantasy	Copperplate Papyrus

Google fonts

Tổng hợp các fonts chữ được cung cấp bởi Google:

<https://fonts.google.com/>

```
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Sofia">
```

```
body {  
    font-family: "Sofia", sans-serif;  
}
```

CSS Font

@font-face:

- Định nghĩa font
- @font-face {
 font-family: tên font
 src: bộ mã font
 font-style:
 font-weight}

```
VD: @font-face {  
    font-family: myFirstFont;  
    src: url(sansation_bold.woff);  
    font-weight: bold;  
}
```

CSS Font formats

TTF (TrueType Fonts): phổ biến (Mac OS, Windows)

OTF(OpenType Fonts): dựa trên TrueType, phổ biến trên máy tính

WOFF(WebOpen font format): thường sử dụng trên web, được khuyến nghị bởi W3C

SVG fonts/shapes:

EOT (Embedded Open Type Fonts): gói font được embed trên web.

CSS font

Font-style: kiểu chữ

- Normal: in thường
- Italic: in nghiêng
- Oblique: xiên

Font-variant: chọn chế độ bình thường và chế độ small-caps (chữ in hoa có kích cỡ nhỏ hơn in hoa chuẩn)

- Normal: chữ thường
- Small-caps: chữ in hoa nhỏ

VD: `h1 { font-variant: small-caps; }`

CSS font

Font-weight: độ đậm nhạt font chữ

- Normal
- Bold
- Bolder
- Lighter
- 100-900

Font-size: kích cỡ chữ

- Px, pt, xx-small, x-small, em, large, x-large, larger...

VD: `body{font-size: 12px;}`

`h1 { font-size: 2em }`

CSS font

Cú pháp rút gọn font:

```
font: <font-style>  
      | <font-variant>  
      | <font-weight>  
      | <font-size>  
      | <font-family>
```

VD: `p { font: italic bold 20px arial, tahoma; }`

CSS text - color

Color

- Định màu chữ
- VD: `p {color: #0000ff; } body{ color: grey;}`

Background-color

- Màu nền chữ
- VD: `p {background-color: #0000ff; }`

CSS Text - alignment


Text-align


- Căn chỉnh văn bản cho các thành phần trong trang web
 - Left
 - Right
 - Center
 - justify
- **Direction**
 - Hướng căn chữ (ltr, rtl)
 - Ex: `p { direction: rtl; unicode-bidi: bidi-override; }`


CSS Text - alignment


Vertical-align


- Căn chỉnh theo chiều dọc của một phần tử
 - baseline
 - Text-top
 - Text-bottom
 - Sub
 - Super
 - Top
 - Bottom
 - Middle
 - Độ lớn/ %

An  image with a default alignment.

An  image with a text-top alignment.

An  image with a text-bottom alignment.

An  image with a sub alignment.

An  image with a super alignment.

CSS Text - alignment

text-align-last

- Căn lề dòng cuối cùng của đoạn văn
- text-align-last: auto | left | right | center | justify | start| end

You can use the text-align-last property to align the last line of a text, if the text has the text-align property set to justify. This text is where you will see the result of the text-align-last property. You can use the text-align-last property to align the last line of a text, if the text has the text-align property set to justify. This text is where you will see the result of the text-align-last property.

CSS text – decoration - transformation

Text-decoration

- Xác định hiệu ứng hiển thị cho đoạn văn bản
 - Underline: gạch chân
 - Line-through: gạch ngang
 - Overline: gạch đầu
 - None: không hiệu ứng

```
a { text-decoration:none; }
```

Text-transform

- Xác định chế độ in hoa hay in thường
 - Uppercase: in hoa
 - Lowercase: in thường
 - Capitalize : in hoa ký tự đầu tiên mỗi từ
 - None: không hiệu ứng

```
H1{ text-transform: uppercase; }
```

CSS text - Spacing

Text-indent

- Tạo khoảng thụt đầu dòng cho dòng đầu tiên trong đoạn văn bản
- Đơn vị: px, % , pt....
- VD: `p { text-indent: 30px; }`

Line-height

- Độ cao của dòng
`p.small { line-height: 0.8; }`

CSS Text - spacing

Letter-spacing

- Định khoảng cách giữa các ký tự trong một đoạn văn bản
- VD: `p { letter-spacing: 5px }`

Word-spacing

- Khoảng cách giữa các chữ, từ
- Ex: `h1 { word-spacing: 10px; }`

white-space

- kiểm soát việc xuống dòng của văn bản.
- `nowrap` – giữ văn bản trên 1 dòng
- `normal` (default) – xuống dòng cho văn bản khi cần thiết

CSS Text Effect

text-shadow:

- Text-shadow: lệch ngang | lệch dọc | độ rộng | màu
- VD: text-shadow: 5px 5px 5px #FF0000;

Text shadow effect!

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

Text shadow effect!

CSS Text Effect

text-overflow:

text-overflow: clip | ellipsis

This is some long text that wil



This is some long text that ...

VD: text-overflow: ellipsis;

CSS Text Effect

word-wrap:

word-wrap: normal | break-word

This paragraph
contains a very long
word:
thisisaveryveryveryveryveryverylongword.
The long word will
break and wrap to
the next line.



This paragraph
contains a very long
word:
thisisaveryveryveryv
eryveryverylongword.
The long word will
break and wrap to
the next line.

CSS Text Effect

word-break

word-wrap: keep-all | break-all

This paragraph
contains some text.
The lines will break
at any character.



This paragraph contains
some text. The lines
will break at any
character.

CSS Multiple Columns

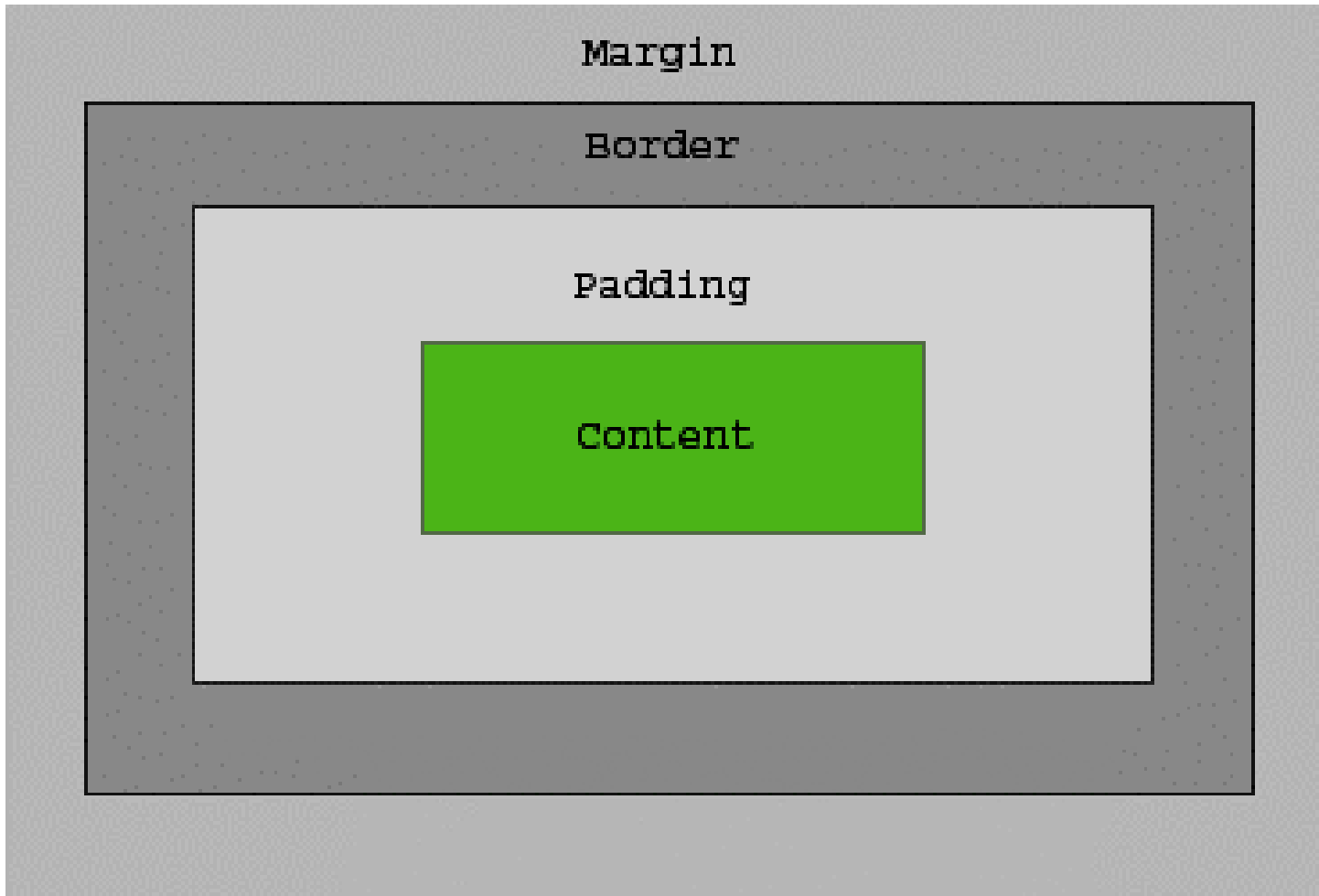
columns

- **column-count**: số cột
- **column-gap**: khoảng cách giữa các cột
- **column-rule**: đường kẻ phân tách giữa các cột
 - Column-rule - color | style | Width
- **column-width**: độ rộng các cột
- **column-span**: all (hoặc số cột có thể tràn qua)

Lorem ipsum dolor sit amet, consectetur 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. Duis autem vel eum

iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue dui dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue

nihil imperdiet doming id quod mazim placerat facer possim assum. Typi non habent claritatem insitam; est usus legentis in iis qui facit eorum claritatem. Investigationes demonstraverunt lectores legere me lius quod ii legunt saepius.



The Box Model

Solid

Dotted

Dashed

Double

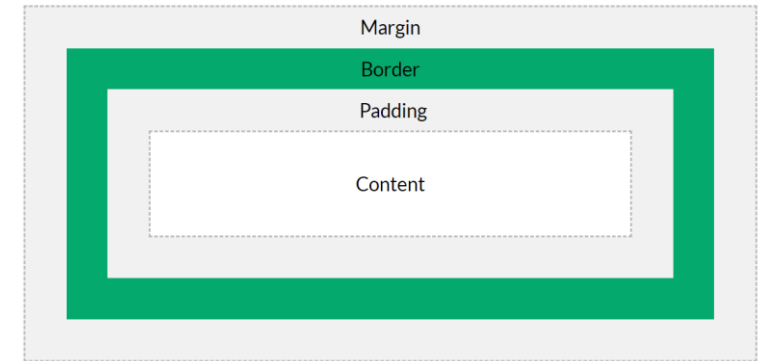
Groove

Ridge

Inset

Outset

CSS border



Border-width

- Độ rộng đường viền
 - Px, thin, medium, thick

Border-color

- Màu đường viền
 - VD: `p { border-color: red; }`

Border-style

- Kiểu đường viền

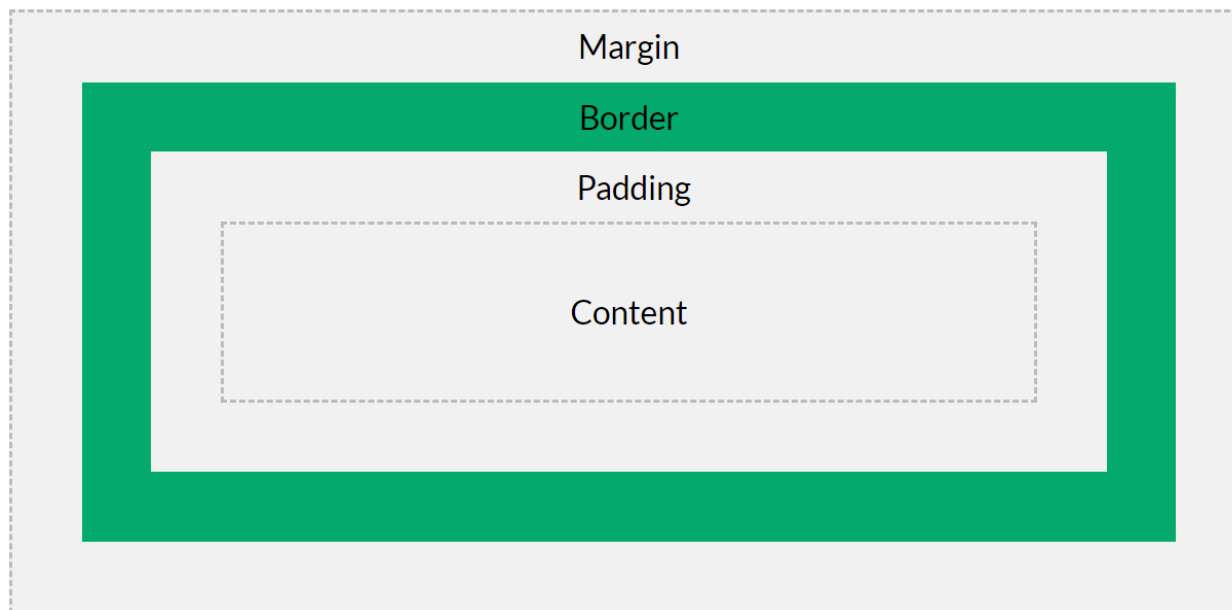
Dotted	Solid
Dashed	Double
Groove	Ridge
Inset	Outset

CSS border

Có thể sử dụng riêng các thuộc tính của css border cho từng cạnh đường viền

- **border-top**
- **border-left**
- **border-right**
- **border-bottom**

VD: `p { border-top-color: blue; }`
`h2 {border-bottom-width: 2px;}`



CSS border

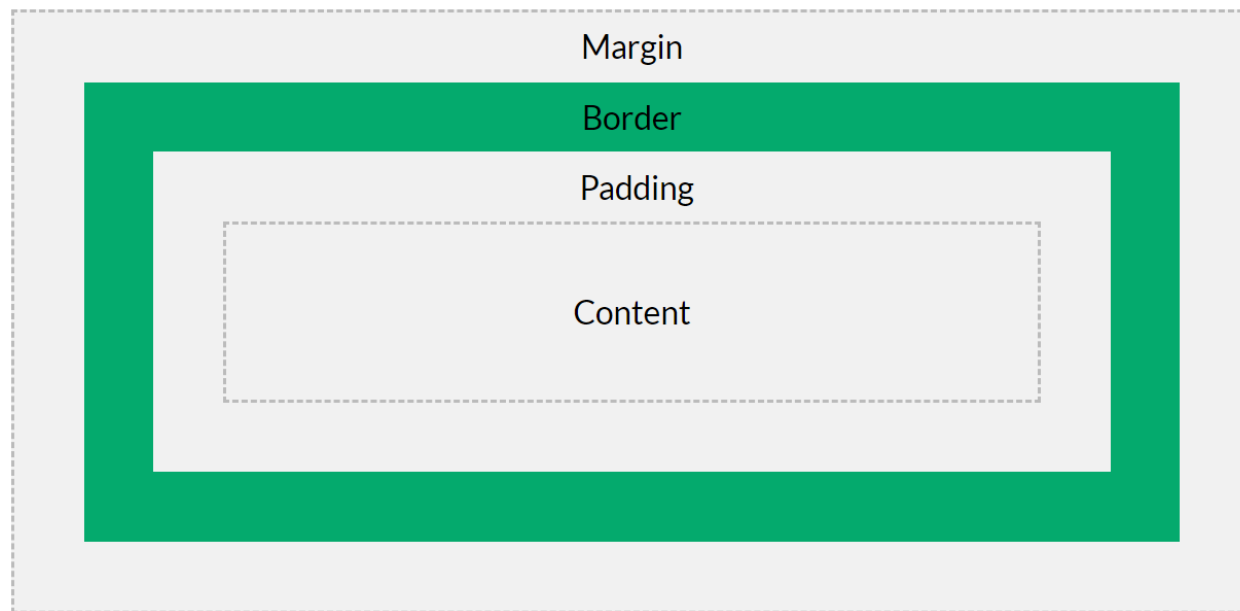
Cú pháp rút gọn

Border : <border-width>
 | <border-style>
 | <border-color>

VD:

```
Body { border: 1px solid #ccc; }
```

```
H1 { border-bottom: 2px dotted #ddd; }
```

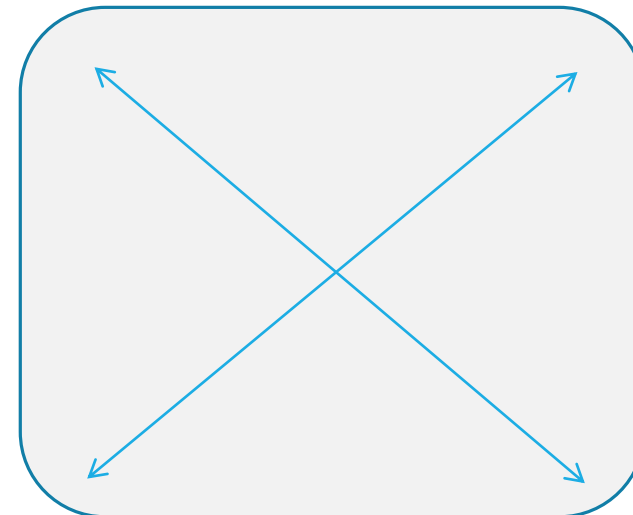


CSS Border - radius

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

border-radius

- Bo tròn góc đường viền
- VD: border-radius: 25px 5px 10px 15px
- Border-top-left-radius.
- Border-top-right-radius
- Border-bottom-left-radius
- Border-bottom-right-radius



CSS Border - image



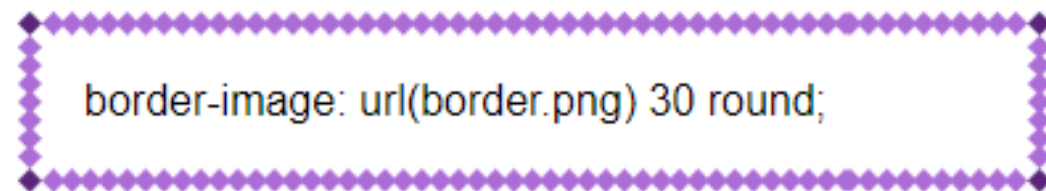
B1: tạo border trong suốt:

```
border: 60px solid transparent;
```



B2: Tạo “đường viền hình ảnh”:

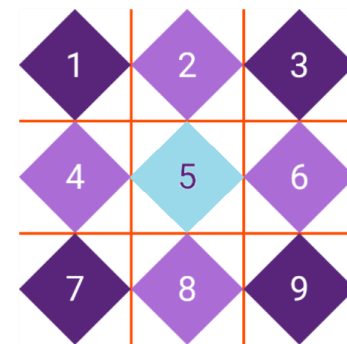
border-image-source: địa chỉ ảnh nền



border-image-width: độ dày đường viền

border-image-outset: mức độ border hiển thị ra ngoài

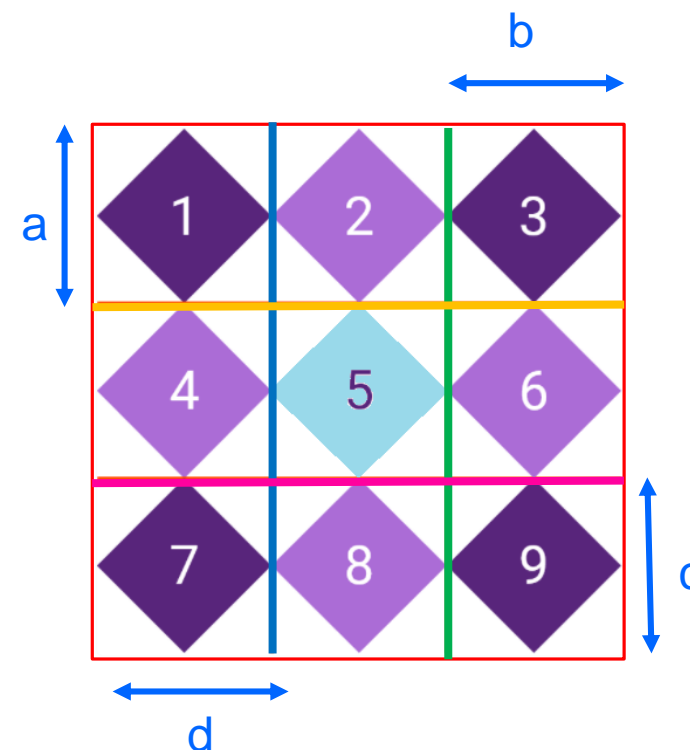
border-image-repeat: chế độ lặp (stretch/repeat/round)



CSS Border - image

border-image-slide: a b c d

- cách cắt ảnh nền làm đường viền
- Giá trị: 4 số:
 - a: khoảng cách/tỉ lệ cạnh top và lát cắt màu cam
 - b: khoảng cách/tỉ lệ cạnh top và lát cắt màu xanh lá
 - c: khoảng cách/tỉ lệ cạnh top và lát cắt màu hồng
 - d: khoảng cách/tỉ lệ cạnh top và lát cắt màu lam
- Vd: border-image-slide: 33% 33% 33% 33%

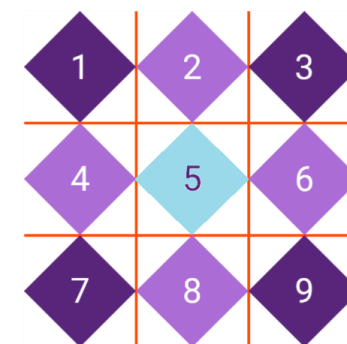
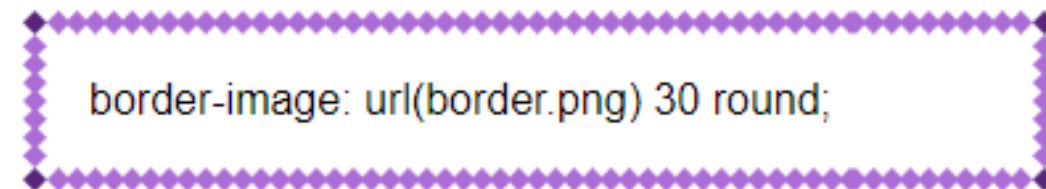


Lưu ý: từ khóa **fill** được thêm vào cuối cùng nếu muốn hiển thị vùng trung tâm (số 5)

CSS Border - image

Cú pháp rút gọn:

border-image: *source slice width outset repeat;*

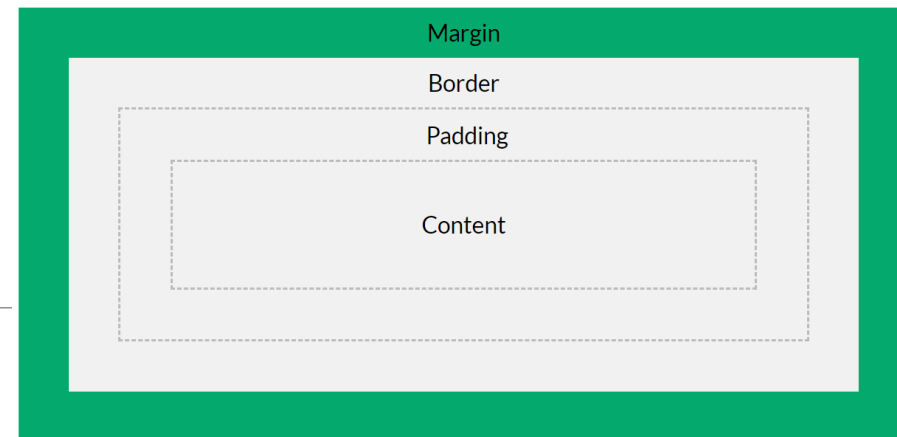
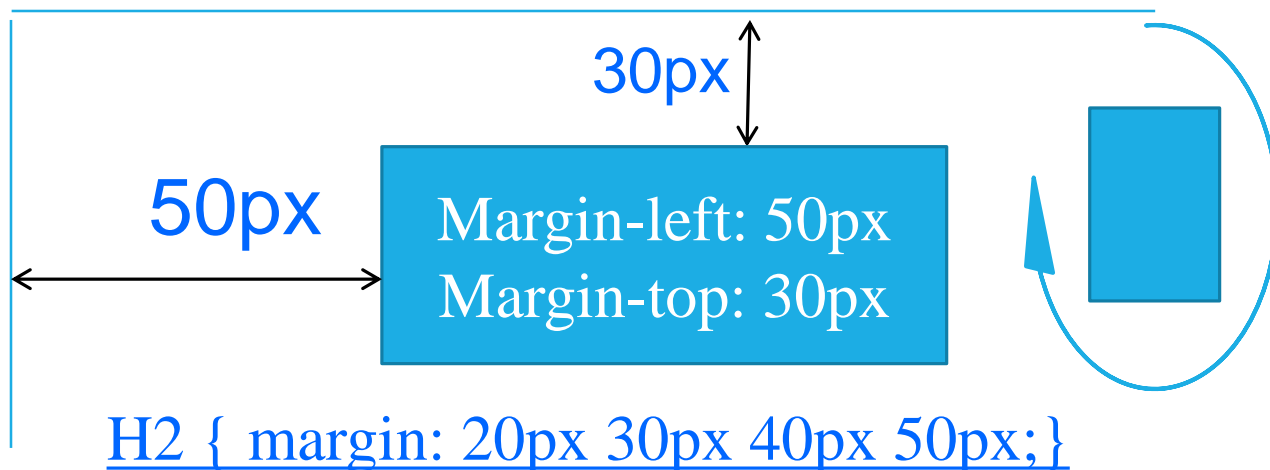


CSS Margin

Margin

- Xác định độ rộng của khoảng cách từ thành phần web này tới thành phần web khác hay với viền trang

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

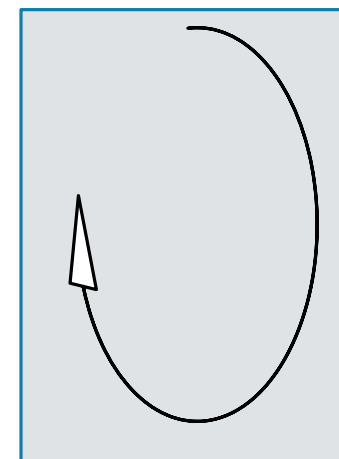
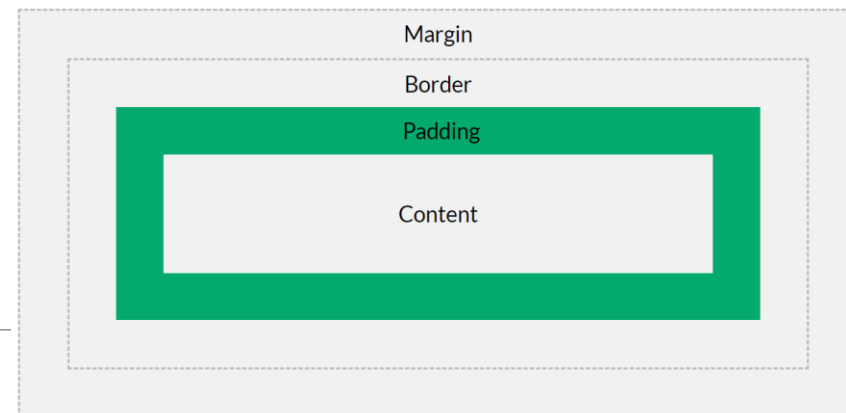
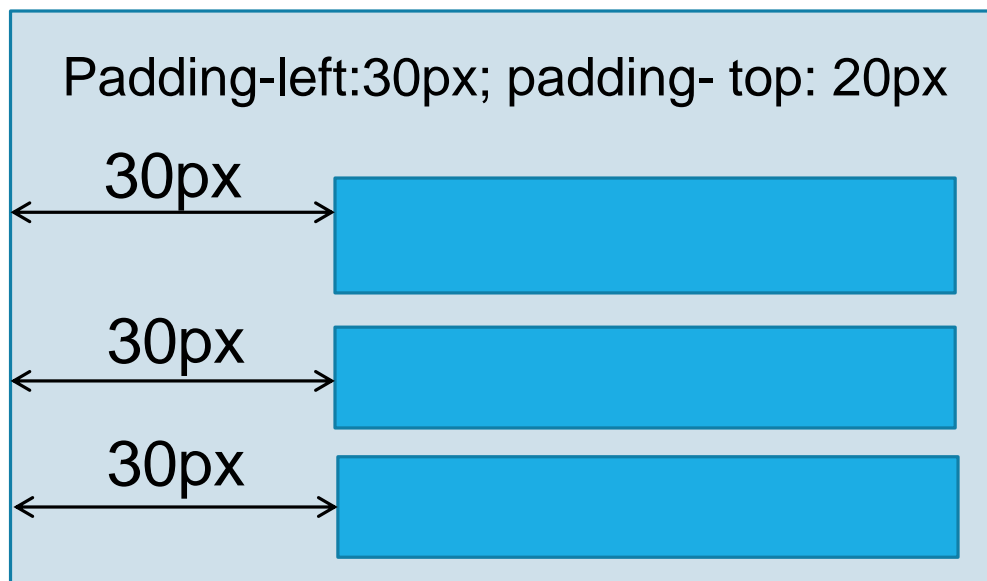


CSS padding

Padding

- Quy định khoảng cách giữa phần nội dung và viền

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



H2 {padding: 20px 30px 40px 50px;}

CSS height & width

Width

- Chiều rộng của một thành phần web
- VD: `p {width: 600px; } img { width: 50%;}`

Max-width

- Chiều rộng tối đa

Min-width

- Chiều rộng tối thiểu



CSS height & width

Height

- Chiều cao của 1 thành phần web
- VD: `h1 {height: 30px}`

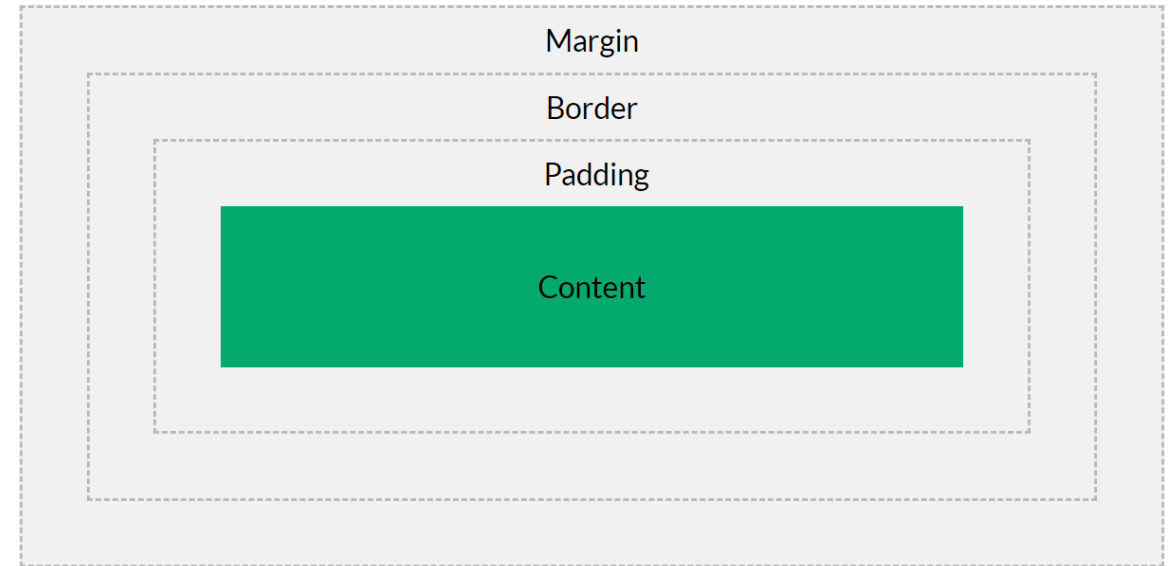
Max-height

- Chiều cao tối đa

Min-height

- Chiều cao tối thiểu

Chú ý: thông thường chiều cao một thành phần web do văn bản trong trang web quyết định

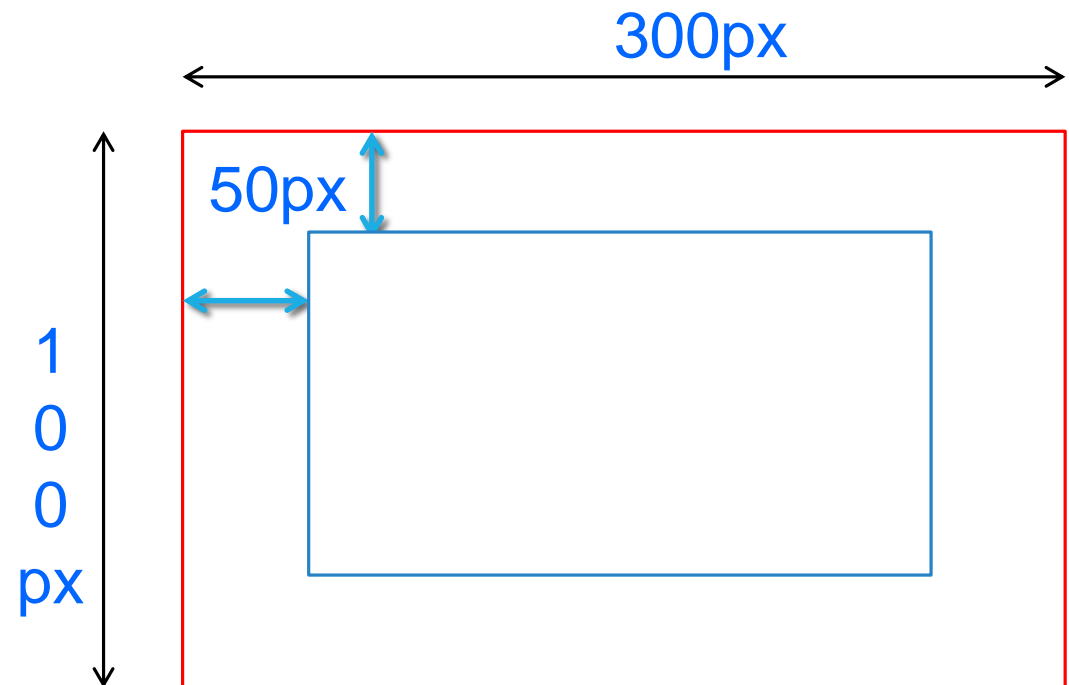


CSS Box-sizing

box-sizing: xác định cách tính kích thước của phần tử web

- Content-box
- Border-box

```
.div2 {  
  width: 300px;  
  height: 100px;  
  padding: 50px;  
  border: 1px solid red;  
  box-sizing: border-box;  
}
```



CSS Outline

Outline-width

- Độ rộng đường viền ngoài
 - Px, thin, medium, thick

Outline-offset

- Khoảng cách border và outline

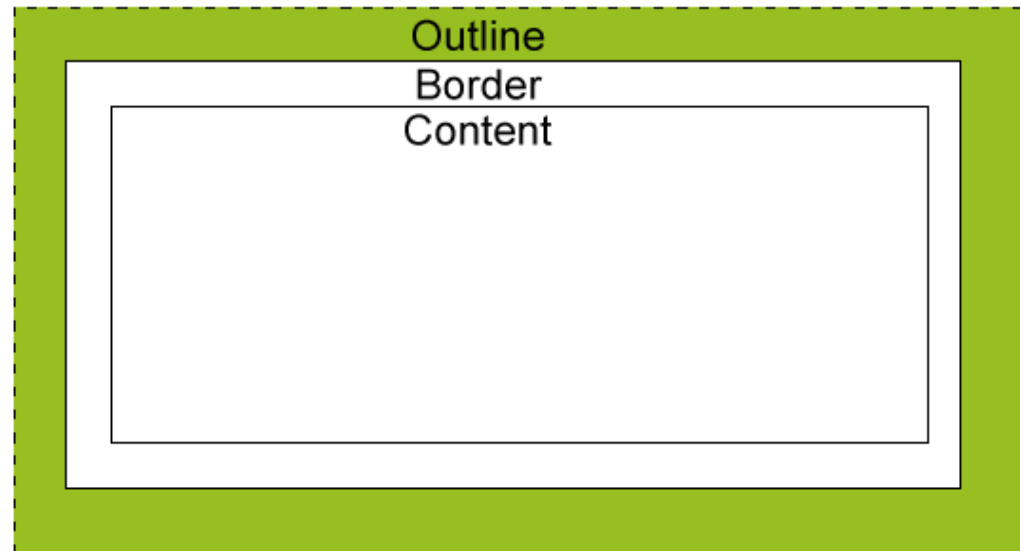
Outline-color

- Màu đường viền ngoài

Outline-style

- Kiểu đường viền ngoài

Dotted	Solid
Dashed	Double
Groove	Ridge
Inset	Outset



Outline : <outline-color>
| <border-style>
| <border-width>

CSS Background

background-color

- Xác định màu nền cho 1 thành phần trên trang web
- VD: `h1 { background-color : #ff0000; }`

background-image

- Chèn ảnh nền cho 1 thành phần trên trang web
- VD: `body { background-image: url("bg.jpg "), url("bg.jpg ")}`

background-repeat:

- Lặp lại ảnh nền
 - Repeat-x: lặp theo phương ngang
 - Repeat-y: lặp theo phương dọc
 - Repeat: lặp theo cả 2 phương (mặc định)
 - No-repeat: không lặp ảnh

CSS Background

background-attachment: Khóa ảnh nền

- Xác định tính cố định của ảnh nền so với nội dung trang web
 - **Scroll:** ảnh nền cuộn cùng nội dung trang web (mặc định)
 - **Fixed:** cố định ảnh nền so với nội dung trang web (ảnh nền đứng yên khi cuộn trang web)

background-position: Định vị ảnh nền

- Đặt ảnh nền ở vị trí xác định (mặc định là góc trên bên trái)
- Sử dụng 2 giá trị: vị trí theo chiều ngang vị trí theo chiều dọc (left, center, right, top, center, bottom, số và %)
- VD: background-position: 5cm 2cm, right bottom;

CSS Background

background-size

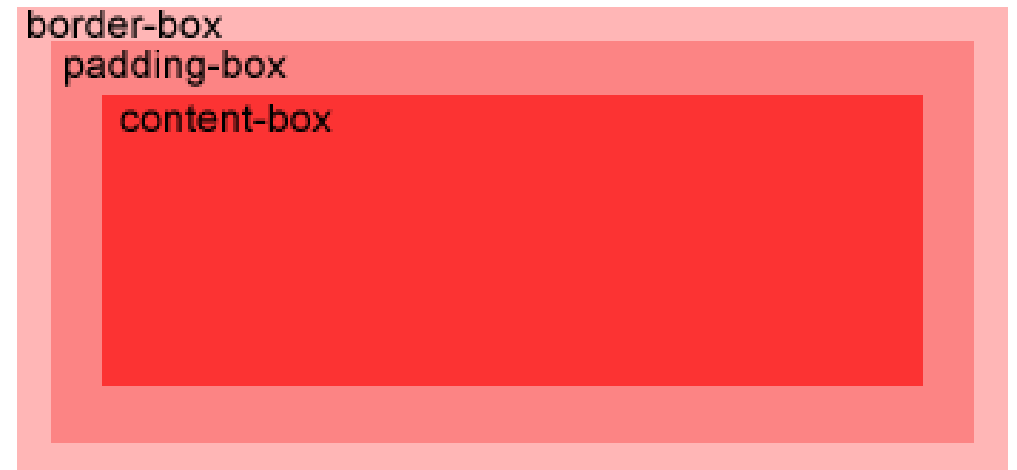
- Kích cỡ của background
 - Background-size: 100px 80px;
 - Contain: kích cỡ lớn nhất có thể nằm trong nền
 - Cover: phủ kín nền

background-origin

- Phạm vi xuất hiện của background là hình ảnh
 - border-box/padding-box/content-box)

background-clip

- Phạm vi xuất hiện của background là màu sắc
 - border-box/padding-box/content-box)



CSS Background

Cú pháp rút gọn css background: background: <background - color>
| <background-image>
| <background-position>
| <background-size>
| <background-repeat>
| <background-origin>
| <background-clip>
|<background-attachment>

background: lightblue url("tulips.jpg") no-repeat fixed right bottom;

CSS Box-shadow

box-shadow

- Đổ bóng khối

- Box-shadow:

lệch ngang | lệch dọc | độ mờ | độ rộng | màu | [inset]

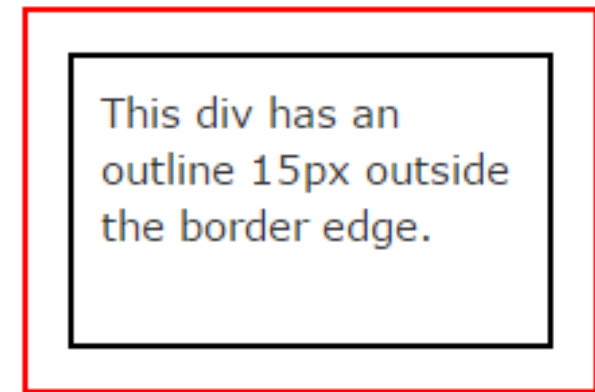
- VD: `box-shadow: 10px 10px 5px #888888;`



CSS User Interface

resize: cho phép người dùng co giãn kích thước phần tử (kết hợp overflow: auto)

- Horizontal: cho phép co giãn theo chiều ngang
- Vertical: cho phép co giãn theo chiều dọc
- Both: co giãn 2 chiều



```
div {  
    border: 1px solid black;  
    outline: 1px solid red;  
    outline-offset: 15px;  
}
```

CSS list

List-style-type: Kiểu bullet

- circle
- disc
- Square
 - 1. Decimal (1, 2, 3), decimal-leading-zero (01, 02, 03)
 - i. lower-roman (i, ii, iii)
 - I. upper-roman (I, II, III, IV)
 - a. lower-alpha, lower-latin (a, b, c),
 - A. upper-alpha, upper-latin (A, B, C)
- none

CSS list

List-style-image

- Thiết lập ảnh làm bullet
- Ex: `ul {list-style-image: url('sqpurple.gif');}`

List-style-position

- Vị trí xuất hiện của bullet
- Inside, outside

Inside:

•	Coffee
•	Tea
•	Coca-cola

Outside:

•	Coffee
•	Tea
•	Coca-cola

CSS list

Cú pháp rút gọn:

List-style : <list-style-type>
 | <list-style-position>
 | <list-style-image>

Ví dụ:

```
ul {  
  list-style: square inside url("sqpurple.gif");  
}
```

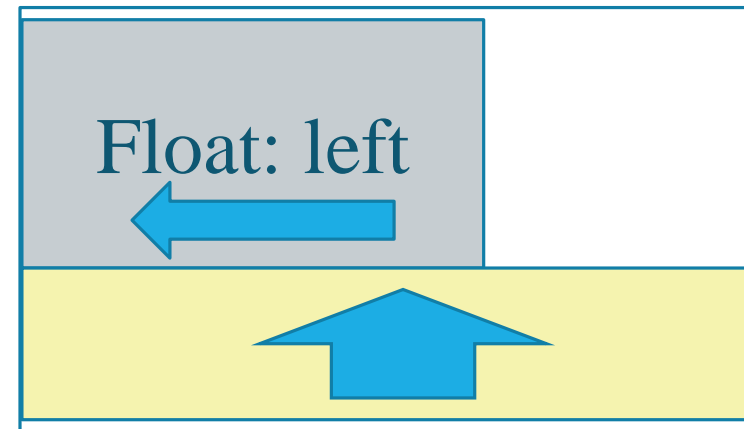
CSS float & clear

Float: cố định một thành phần trang web về bên trái hay bên phải không gian bao quanh nó.

Rất quan trọng khi dàn trang

Giá trị

- **Left:** đặt phần tử phía bên trái
- **Right:** đặt phần tử phía bên phải
- **None:** bình thường



CSS float & clear

Clear: (luôn đi cùng với float) Được gán vào các phần tử phía sau phần tử đã được float để quyết định định dạng của phần tử này.

Giá trị:

- Left : không tràn khi phần tử trước nó sử dụng float bên trái
- Right: không tràn khi phần tử trước nó sử dụng float bên phải
- Both: không tràn

Clearing floats

- Thêm phần tử <div> được định dạng với clear
- `:after { content: ""; display: block; clear: both; height: 0; }`

CSS Position

Position: Định vị một thành phần trên trang web

Giá trị:

- **Static:** vị trí mặc định của phần tử theo đúng vị trí được sắp xếp trong code
- **Absolute:** phần tử được định vị với vị trí tương đối so với phần tử chứa nó được thiết lập thuộc tính position với giá trị là relative, absolute hoặc fixed nằm gần với nó nhất.
- **Fixed:** phần tử được định vị với vị trí tương đối so với cửa sổ trình duyệt
- **Relative:** phần tử được định vị tương đối so với vị trí thông thường của mình.
- **Sticky:** Phần tử được định vị chuyển đổi giữa relative và fixed phụ thuộc vào vị trí của thanh cuộn

CSS layers

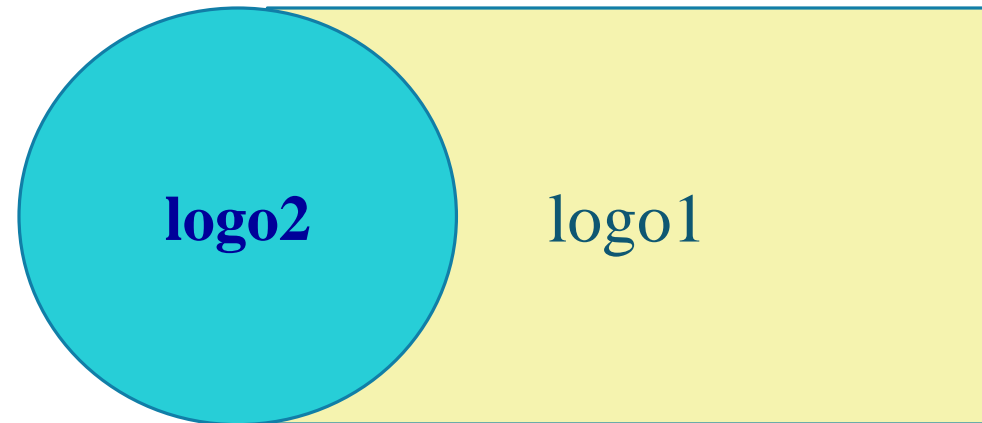
CSS hoạt động trên cả 3 chiều: cao (height), rộng (width), sâu(z-index)

Sử dụng thuộc tính **z-index** để thực hiện tạo các layer khác nhau.

Các layer có chỉ số cao hơn sẽ nằm bên trên layer có chỉ số thấp hơn.

VD: `#logo1 { z-index: 1;}`

`#logo2 {z-index: 2; }`



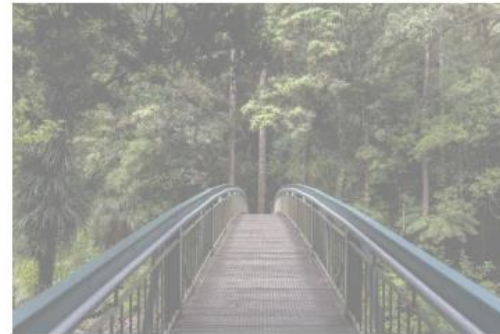
CSS Opacity

Opacity:

- 0.0 – 1.0
- Opacity: 0.5



opacity 0.2



opacity 0.5



opacity 1
(default)

<=IE8

- Filter:alpha(opactiy = x) , x = 0 - 100

Visibility

visibility

- Thuộc tính quy định phần tử có được hiển thị hay không
- **hidden**: phần tử không được hiển thị nhưng vẫn chiếm vị trí trên trang (giống như **opacity:0**)
- **visible**: hiển thị thông thường (mặc định)

CSS display

display:

- Inline: hiển thị trong dòng
 - `` `<a>`
- Block: hiển thị theo khối
 - `<h1>` `<p>` `` `<div>`
- Inline-block: hiển thị dòng, nhận các thuộc tính khối (width,height, padding)
- None: không hiển thị (kích thước của phần tử không được sử dụng để tính toán khi hiển thị)

Inline1 inline2 inline3



CSS Overflow

overflow

- Visible – default: hiển thị nguyên dạng
- Hidden: ẩn phần nội dung tràn ra ngoài
- Scroll: hiển thị thanh cuộn
- Auto: hiển thị thanh cuộn nếu nội dung bị tràn

Overflow-x

Overflow-y

Chỉ làm việc với phần tử block và có height xác định

Other CSS Properties

cursor: hình ảnh trỏ chuột xuất hiện khi trỏ chuột qua một phần tử

- crosshair
 - Pointer
 - Move
 - row-resize
 - Wait
- help
 - progress
 - col-resize
 - text
 - copy

CSS Flexbox

- Tạo cấu trúc bố cục linh hoạt mà không cần sử dụng float và position



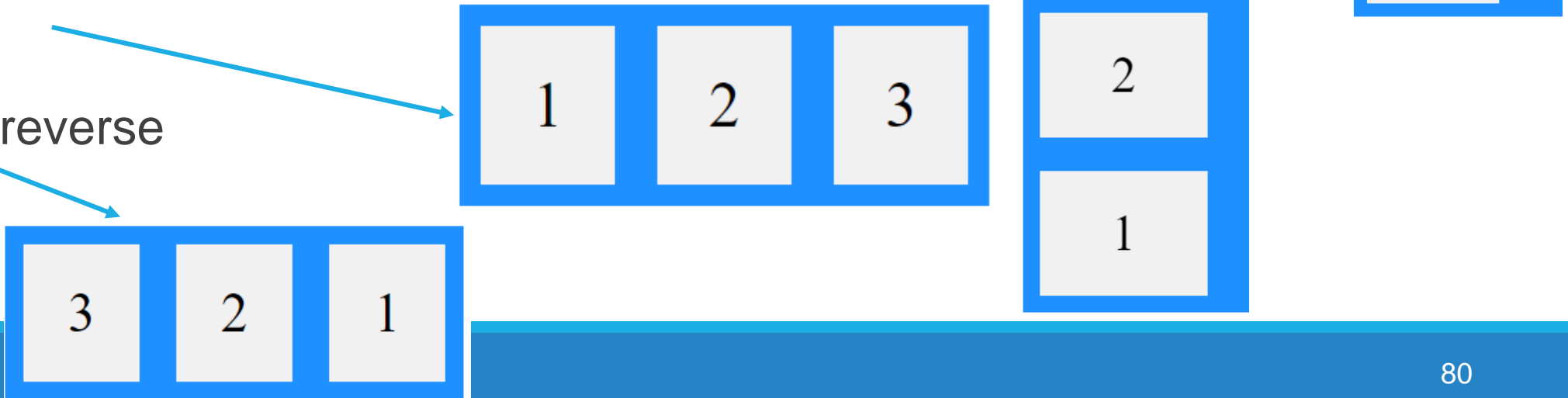
```
<div class="flex-container">  
  <div>1</div>  
  <div>2</div>  
  <div>3</div>  
</div>
```

CSS Flexbox container

display: flex

flex-direction: xác định hướng sắp xếp các phần tử con

- column
- column-reverse
- row
- row-reverse



CSS Flexbox container

flex-wrap: cho phép phần tử con xuống dòng hay không

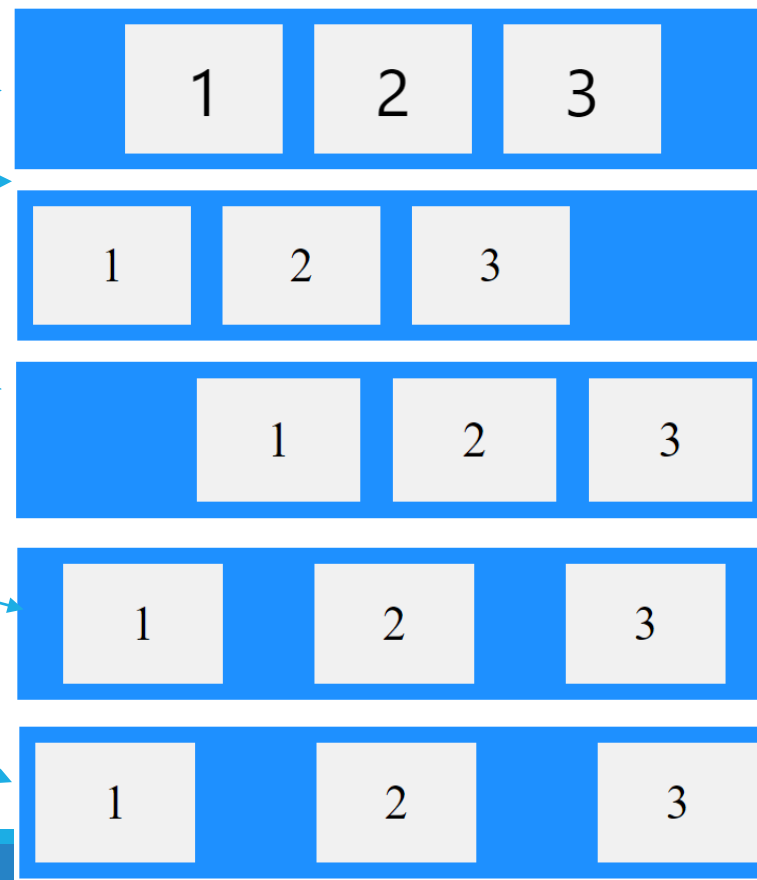
- wrap: xuống dòng theo thứ tự từ các phần tử cuối cùng
- Nowrap: không xuống dòng
- wrap-reverse: xuống dòng theo thứ tự từ các phần tử đầu tiên

flex-flow: viết tắt của 2 thuộc tính flex-direction và flex-wrap

CSS Flexbox container

justify-content: Căn các phần tử con theo chiều ngang

- center
- flex-start
- flex-end
- space-around
- space-between

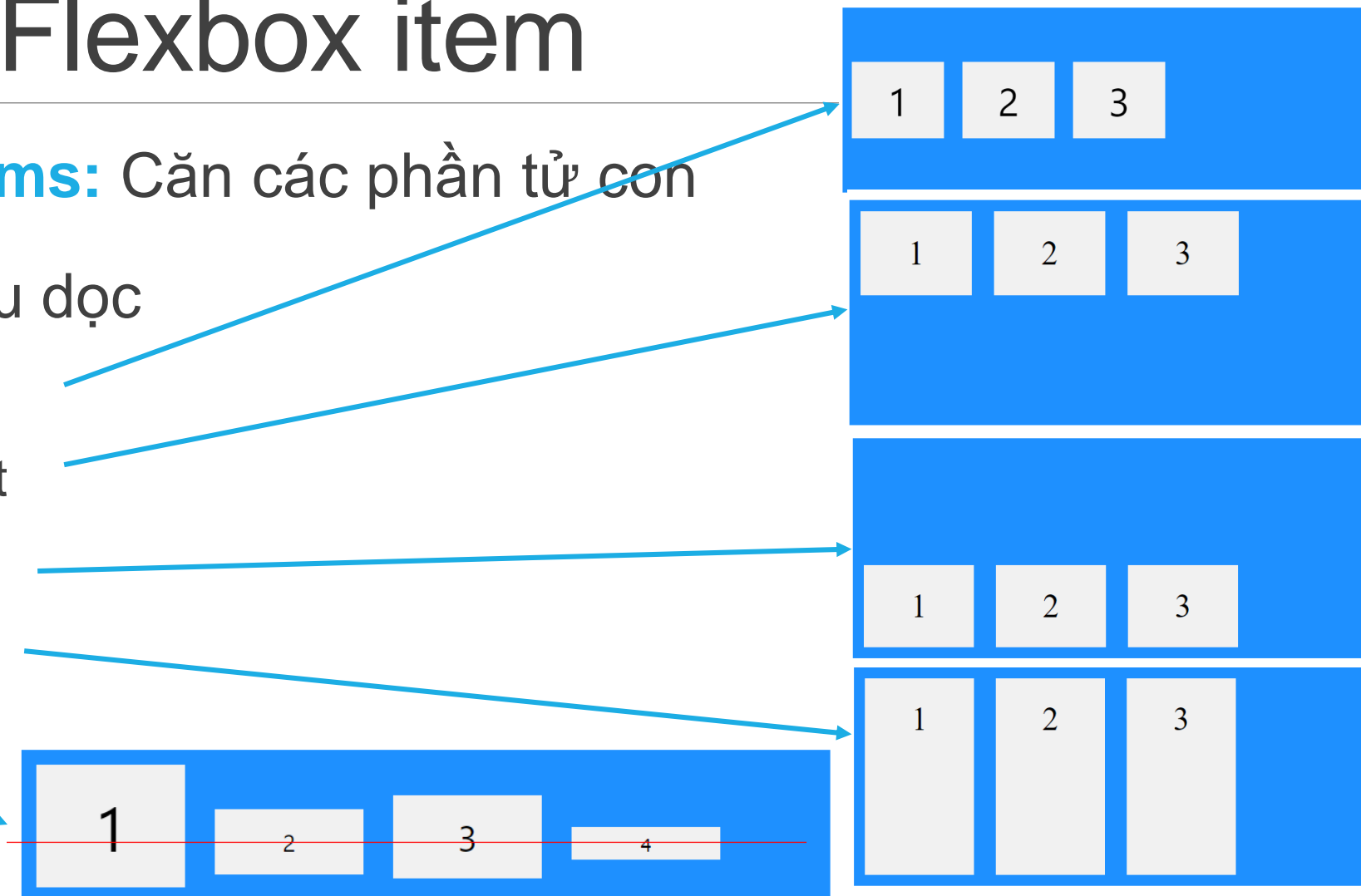


CSS Flexbox item

align-items: Căn các phần tử con

theo chiều dọc

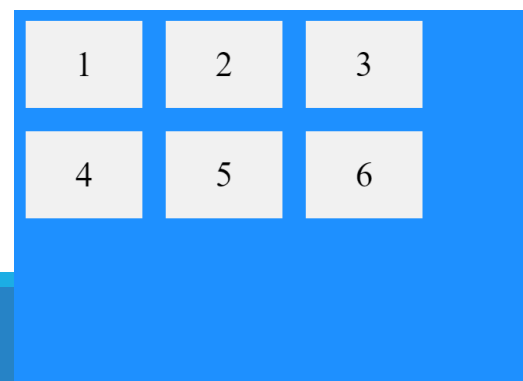
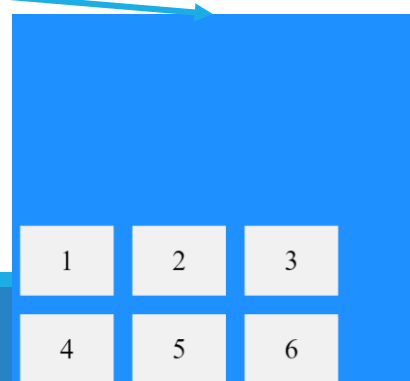
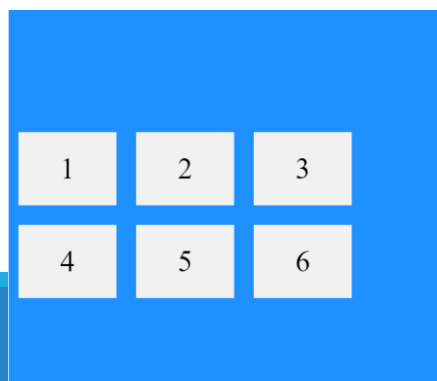
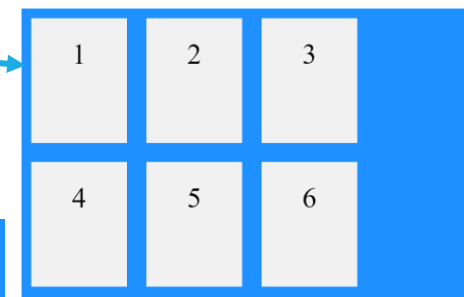
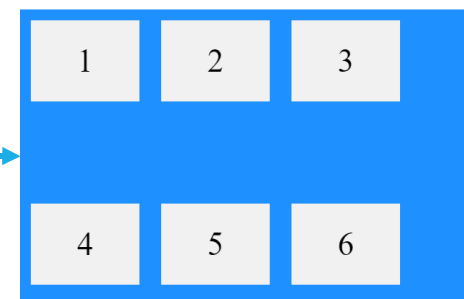
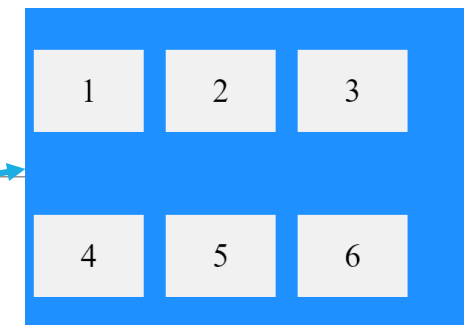
- center
- flex-start
- flex-end
- stretch
- baseline



CSS Flexbox item

align-content: căn chỉnh dòng flex

- space-around
- space-between
- Stretch
- center
- flex-start
- flex-end



CSS Flexbox item

order: sắp xếp thứ tự các phần tử

```
<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>
```



flex-grow: thiết lập tỉ lệ các phần tử

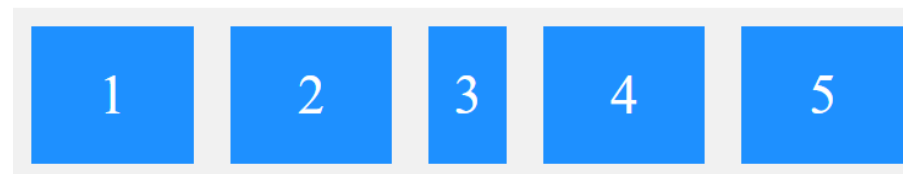
```
<div class="flex-container">  
  <div style="flex-grow: 1">1</div>  
  <div style="flex-grow: 1">2</div>  
  <div style="flex-grow: 3">3</div>  
</div>
```



CSS Flexbox item

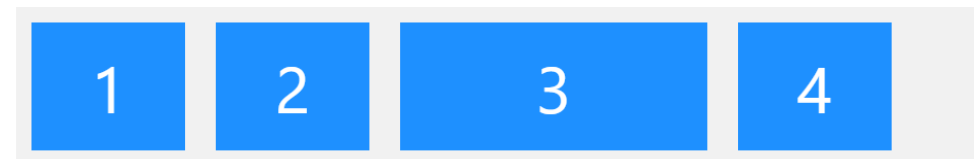
flex-shrink: thiết lập một phần tử co lại tương đối so với các phần tử khác khi co dẫn màn hình

```
<div class="flex-container">  
  <div>1</div>  
  <div>2</div>  
  <div style="flex-shrink:5">3</div>  
  <div>4</div>  
  <div>5</div>  
</div>
```



flex-basis: kích thước chiều rộng khởi tạo của phần tử

```
<div class="flex-container">  
  <div>1</div>  
  <div>2</div>  
  <div style="flex-basis: 200px">3</div>  
  <div>4</div>  
</div>
```



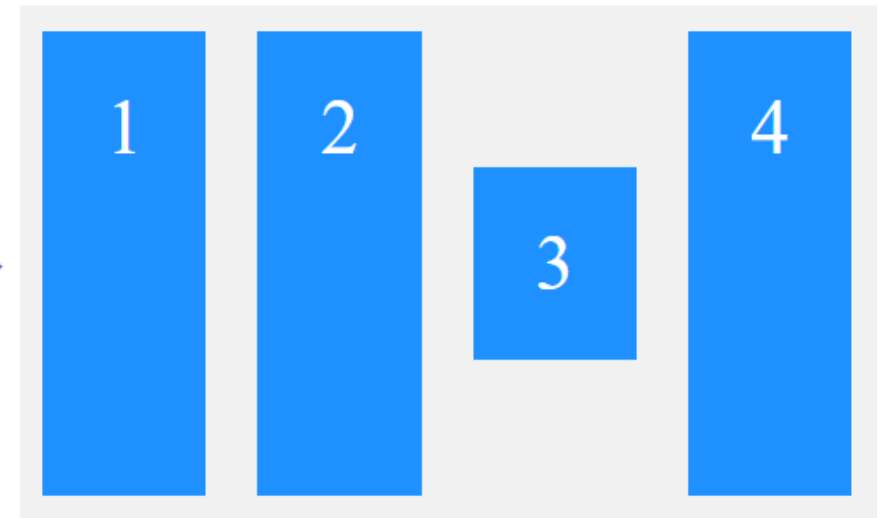
CSS Flexbox item

flex: gộp của các thuộc tính: flex-grow, flex-shrink, flex-basis

align-self: căn chỉnh một phần tử cụ thể

- center
- flex-start
- flex-end

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



CSS Pseudo-classes

Pseudo-classes cho phép xác định các hiệu ứng định dạng cho trạng thái đặc biệt của các phần tử

Cú pháp:

- **selector** : **pseudo-class** {
 property : value;
}

CSS Pseudo-classes

Selector	Ví dụ	Mô tả
:active	a:active	Thẻ a đang được chọn
:hover	a:hover	Link đang được di chuột qua
:link	a:link	Link chưa được chọn
:visited	a:visited	Link đã được thăm

CSS Pseudo-classes

Selector	Ví dụ	Mô tả
:checked	input:checked	Các phần tử <input> đang được chọn
:disabled	input:disabled	Các phần tử <input> ở trạng thái disabled
:enabled	input:enabled	Các phần tử <input> ở trạng thái enabled
:focus	input:focus	Các phần tử <input> ở trạng thái được trỏ chuột vào
:read-only	input:read-only	Các phần tử <input> ở trạng thái readonly
:valid :invalid	input:valid	Các phần tử <input> có dữ liệu nhập vào đúng/không đúng định dạng

CSS Pseudo-classes

Selector	Example	Example description
:empty	p:empty	Các phần tử <p> không có con
:first-child :last-child	p:first-child	Các phần tử thẻ <p> là con đầu tiên của phần tử cha
:first-of-type :last-of-type	p:first-of-type	Các phần tử là thẻ <p> đầu tiên của phần tử cha
:only-of-type	p:only-of-type	Các phần tử là thẻ <p> duy nhất của phần tử cha
:only-child	p:only-child	Các phần tử là con duy nhất của phần tử cha

CSS Pseudo-classes

Selector	Example	Example description
<code>:nth-child(n)</code>	<code>p:nth-child(2)</code>	Các phần tử thẻ <code><p></code> là con thứ 2 của phần tử cha
<code>:nth-last-child(n)</code>	<code>p:nth-last-child(2)</code>	Các phần tử thẻ <code><p></code> là con thứ 2 của phần tử cha tính từ cuối lên
<code>:nth-of-type(n)</code>	<code>p:nth-of-type(2)</code>	Các phần tử thẻ <code><p></code> là thẻ <code><p></code> thứ 2 của phần tử cha
<code>:nth-last-of-type(n)</code>	<code>p:nth-last-of-type(2)</code>	Các phần tử thẻ <code><p></code> là thẻ <code><p></code> thứ 2 của cha tính từ cuối lên

CSS pseudo-element

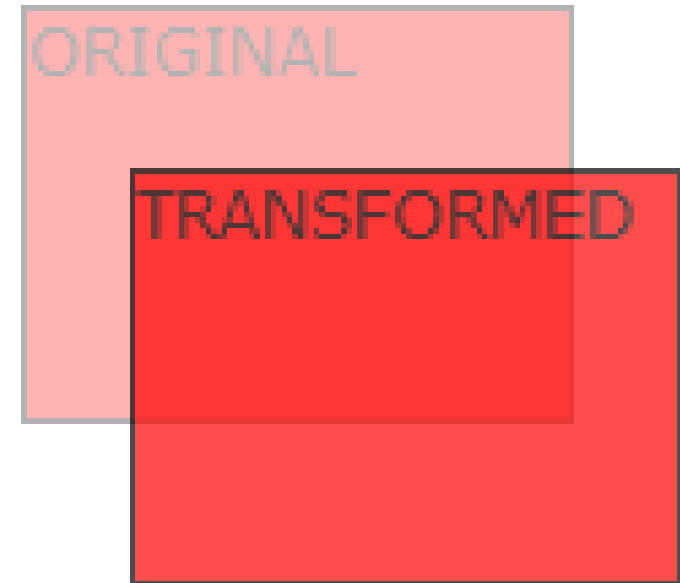
Selector	Example	Example description
::after	p::after	Thêm nội dung sau thẻ <p>
::before	p::before	Thêm nội dung trước thẻ <p>
::first-letter	p::first-letter	Định dạng Css tất cả các chữ cái đầu tiên trong thẻ <p>
::first-line	p::first-line	Định dạng Css tất cả các dòng đầu tiên trong thẻ <p>
::selection	p::selection	Định dạng Css phần dữ liệu được chọn bởi người dùng

CSS3 2D Transform

IE 10, Chrome 36, FF 16, S: 9 , O: 23

transform:

- translate (Xpx,Ypx): dịch chuyển
- translateX(Xpx)
- translateY(Ypx)



CSS3 2D Transform

IE 10, Chrome 36, FF 16, S: 9 , O: 23

transform:

- scale(X lần, Y lần): điều chỉnh kích thước
- scaleX(x lần)
- scaleY(y lần)

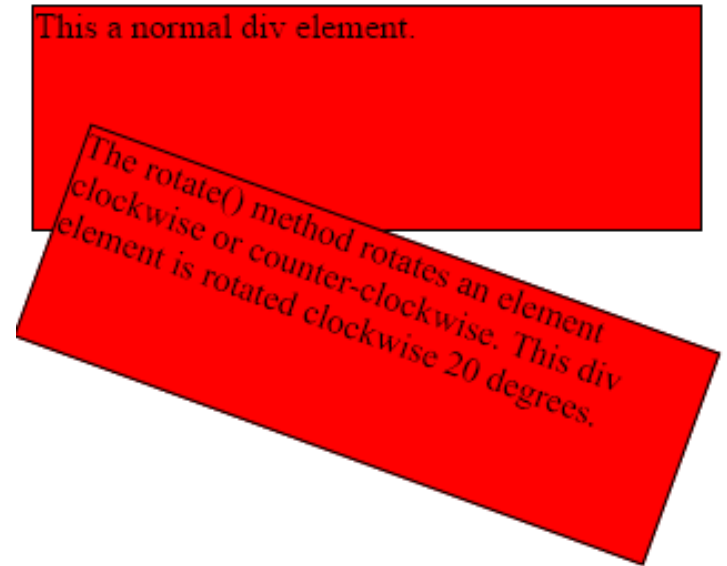


CSS3 2D Transform

IE 10, Chrome 36, FF 16, S: 3.2 , O: 23

transform

- rotate(Ndeg): xoay Ndeg theo chiều kim đồng hồ

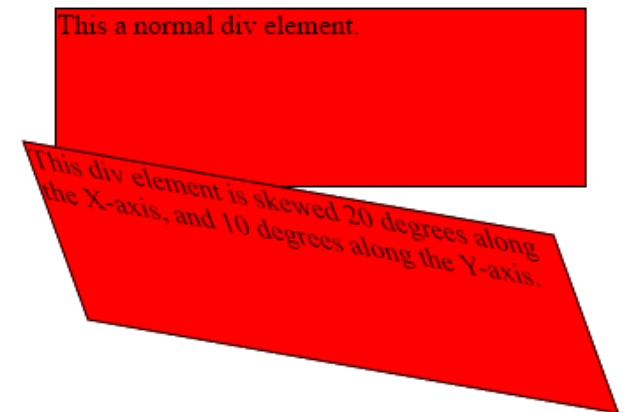
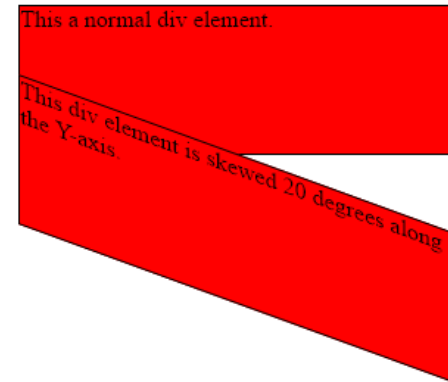
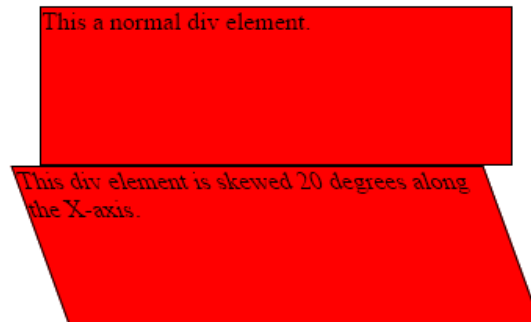


CSS3 2D Transform

IE 10, Chrome 36, FF 16, S: 3.2 , O: 23

transform

- skew(độ x, độ y): quay bao nhiêu độ theo trục X, Y
- skewX(Ndeg)
- skewY(Ndeg)



CSS3 2D Transform

IE 10, Chrome 36, FF 16, S: 3.2 , O: 23

matrix

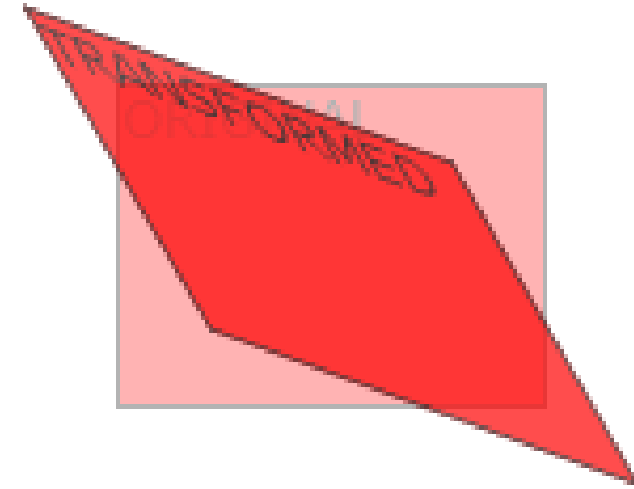
- `matrix(scaleX, skewY, skewX, scaleY, translateX, translateY)`
- `matrix(1, -0.3, 0, 1, 0, 0)`

CSS3 3D Transform

IE 10, Chrome 36, FF 16, S:4 , O: 23

transform

- `translate3d(x,y,z)`
 - `translateX(x)`, `translateY(y)`, `translateZ(z)`
- `scale3d(x,y,z)`
 - `scaleX(x)`, `scaleY(y)`, `scaleZ(z)`
- `rotate3(x,y,z ,angle)`
 - `rotateX(x)`, `rotateY(y)`, `rotateZ(z)`



CSS3 Transition

IE 10, Chrome 26, FF 16, S: 6.1 O:12

transition

- transition-duration: thời gian thực hiện (default: 0s)
- transition-property: thuộc tính áp dụng (default: all)
- transition-timing-function:
 - Ease: slow-fast-slow (default)
 - Linear: same speed
 - ease-in: slow-fast
 - ease-out: fast-slow
 - ease-in-out: slow-fast-fast-slow
- transition-delay: độ trễ (default: 0s)

CSS3 Transition

VD1:

```
div { width: 100px; height: 50px; background: red;
      color: white; font-weight: bold;
      transition: width 2s;
      transition-timing-function: linear;}
div:hover { width: 300px; }
```

•VD2:

```
div{
    transition: width 2s, height 3s, transform 2s;
}
```

CSS3 Animation

@keyframes

```
@keyframes animationname  
{ keyframes-selector{css-styles;}
```

+ keyframes-selector: 0-100% | from | to

```
VD: @keyframes mymove {  
  0%   {top: 0px;}  
  25%  {top: 200px;}  
  50%  {top: 100px;}  
  75%  {top: 200px;}  
  100% {top: 0px;}  
}  
animation: mymove 5s;
```

CSS3 Animation

animation : Chuyển đổi tượng từ trạng thái này sang trạng thái khác(@keyframes)

- animation-name: keyframename| none
- animation-duration: thời gian thực hiện (s, ms; default: 0s)
- animation-timing-function: ease|linear| ease-in | ease-out| ease-in-out
- animation-delay: độ trễ bắt đầu thực hiện animation
- animation-iteration-count: number|infinite: số lần lặp lại
- animation-direction: normal | alternate (chỉ áp dụng với số lần lặp >1)
- animation-play-state: running | paused
- animation-fill-mode: none | forwards | backwards | both

Media Queries

- **media:** Kiểm tra thông số và thực hiện các css rẽ nhánh

- width, height của viewport, device
- orientation (landscape or portrait)
- resolution

```
@media not| only mediatype and (media feature){  
    //css-code  
}
```

- Sử dụng các stylesheets khác nhau cho các media khác nhau bằng cách sử dụng link:

```
<link rel="stylesheet" media="mediatype and|not|only (media feature)"  
href="mystylesheet.css">
```


Media types

all: tất cả các thiết bị với mọi loại tính chất

print: máy in

screen: màn hình máy tính, tablet, smart-phone..

speech: screenreader có thể reads thành tiếng

Media Features

Value	Description
max-aspect-ratio	The maximum ratio between the width and the height of the display area
max-device-aspect-ratio	The maximum ratio between the width and the height of the device
max-device-height	The maximum height of the device, such as a computer screen
max-device-width	The maximum width of the device, such as a computer screen
max-height	The maximum height of the display area, such as a browser window
max-resolution	The maximum resolution of the device, using dpi or dpcm
max-width	The maximum width of the display area, such as a browser window

Media Features

Value	Description
min-aspect-ratio	The minimum ratio between the width and the height of the display area
min-device-aspect-ratio	The minimum ratio between the width and the height of the device
min-device-width	The minimum width of the device, such as a computer screen
min-device-height	The minimum height of the device, such as a computer screen
min-height	The minimu
min-resolution	The minimum resolution of the device, using dpi or dpcm
min-width	The minimum width of the display area, such as a browser window

Media Features

Value	Description
max-color /min-color	The maximum/minimum number of bits per color component for the output device
orientation	The orientation of the viewport (landscape or portrait mode)
resolution	The resolution of the output device, using dpi or dpcm
width	The viewport width
aspect-ratio	The ratio between the width and the height of the viewport
color	The number of bits per color component for the output device
grid	Whether the device is a grid or bitmap
height	The viewport height

CSS3

@media Rule

```
@media only screen and (max-width: 500px) {  
  .gridmenu {  
    width:100%;  
  }  
  
  .gridmain {  
    width:100%;  
  }  
  
  .gridright {  
    width:100%;  
  }  
}
```

CSS 3 responsive

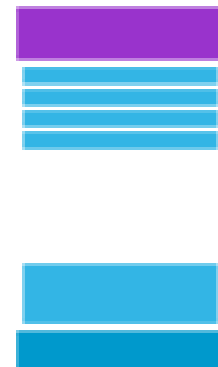
Desktop



tablet



phone



viewport

Không gian hiển thị website trên thiết bị người dùng: phone, table, desktop

Setting viewport

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

Quy tắc:

- Không sử dụng độ rộng cố định
- Không phụ thuộc vào một viewport cụ thể
- Sử dụng CSS media

Kích thước màn hình

max-width: 320px (điện thoại di động, hiển thị chiều dọc)

max-width: 480px (điện thoại di động, hiển thị chiều ngang)

max-width: 600px (máy tính bảng, hiển thị chiều dọc)

max-width: 800px (máy tính bảng, hiển thị chiều ngang)

max-width: 768px (máy tính bảng loại to, hiển thị chiều dọc)

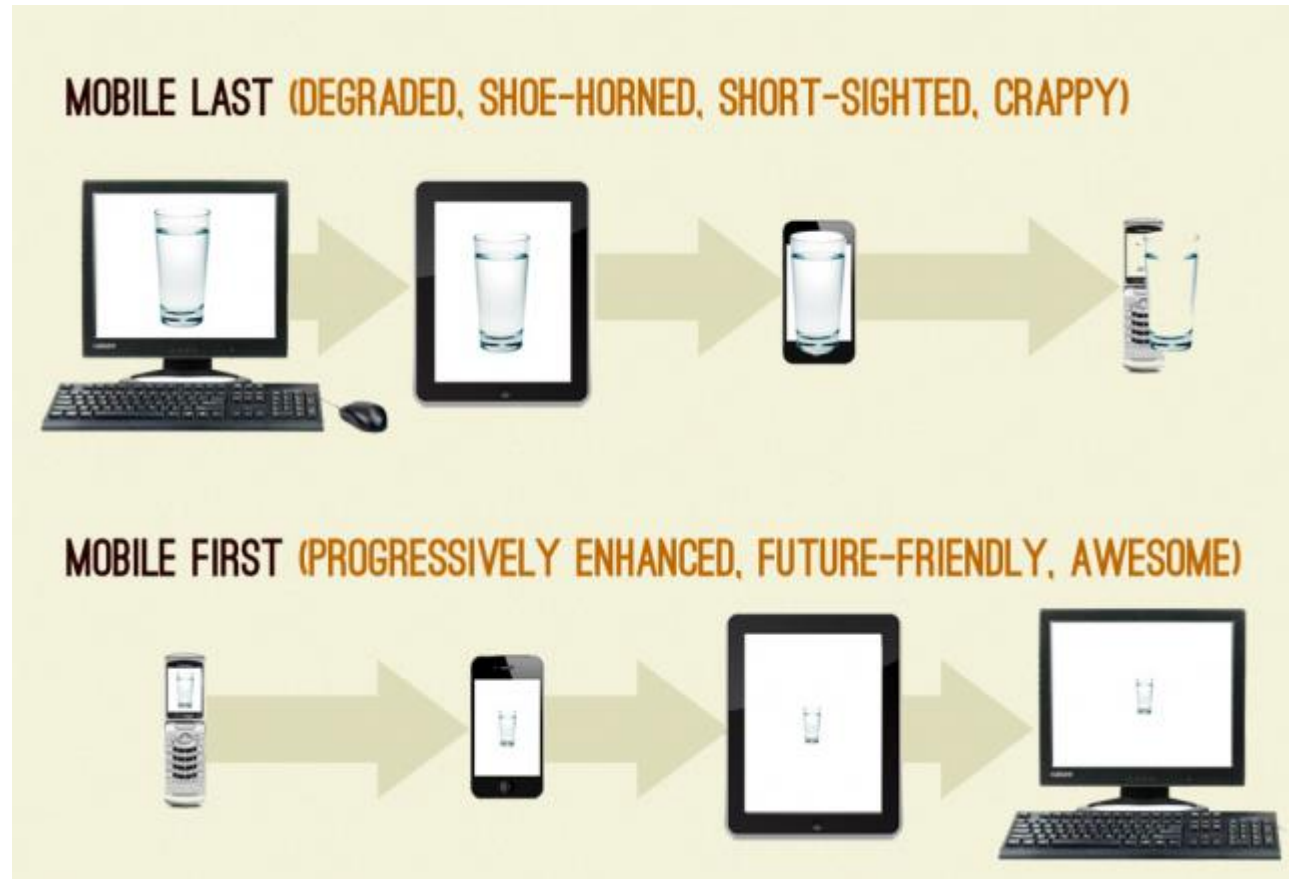
max-width: 1024px (máy tính bảng loại to, hiển thị chiều ngang)

min-width: 1025px (từ size này trở lên là danh cho desktop thông thường).

Cấu trúc CSS responsive

```
/* Các CSS này dành cho toàn bộ website và desktop */  
  
body {  
  
    background: #fff;  
  
    color: 333;  
  
}  
  
/* Dành cho điện thoại */  
  
@media all and (max-width: 480px) {  
  
  
  
}  
  
/* Dành cho máy tính bảng */  
  
@media all and (max-width: 1024px) {  
  
  
  
}
```

Mobile first



Cấu trúc responsive mobile last

```
@media screen and (max-width: 1024px) {  
  
}
```

```
@media screen and (max-width: 768px) {  
  
}
```

```
@media screen and (max-width: 480px) {  
  
}
```

```
@media screen and (max-width: 320px) {  
  
}
```

```
@media screen and (max-width: 240px) {  
  
}
```

Cấu trúc responsive mobile first

```
/* Dành cho điện thoại */
```

```
@media all and (min-width: 320px) {...}
```

```
/* Dành cho máy tính bảng chiều dọc */
```

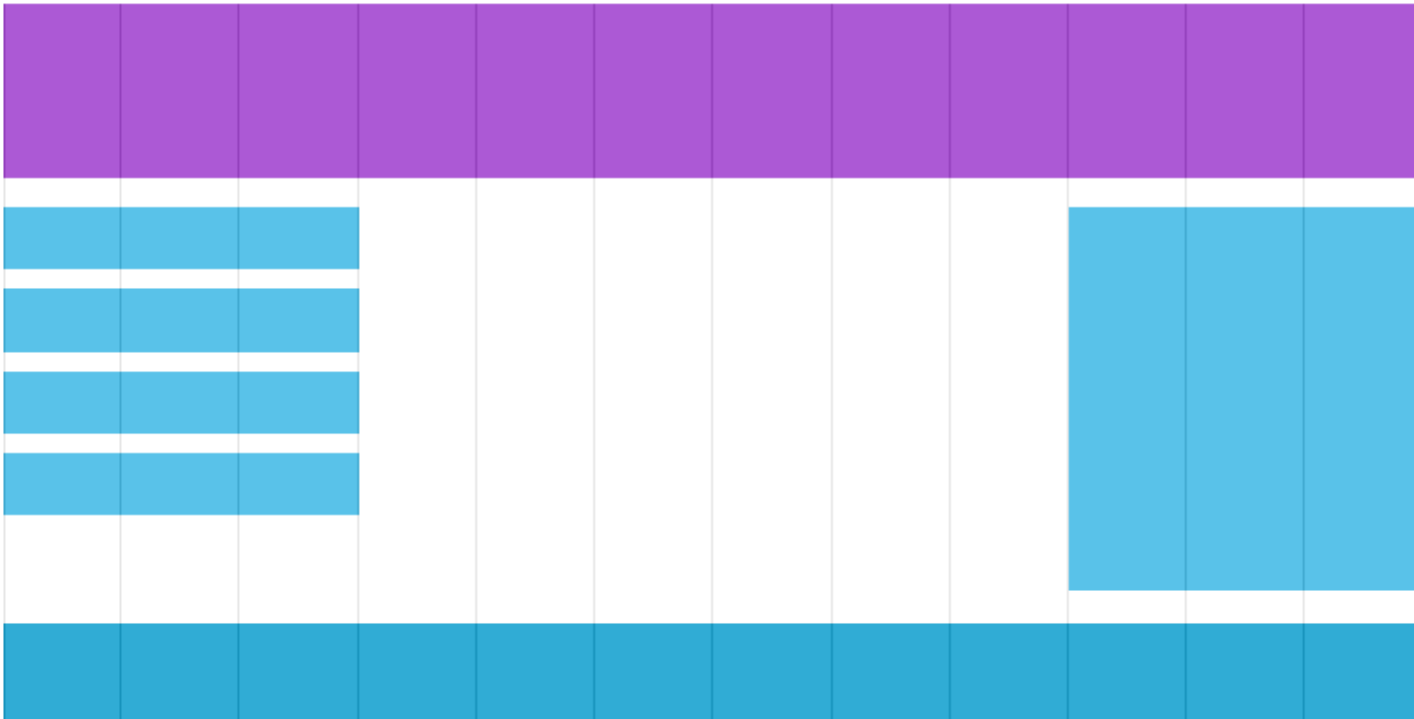
```
@media all and (min-width: 600px) {...}
```

```
/* Dành cho máy tính bảng chiều ngang */
```

```
@media all and (min-width: 1024px) {...}
```

```
/* Dành cho màn hình desktop */
```

```
@media all and (min-width: 1280px) {...}
```



Grid-view

Trang web được chia thành các cột (12 cột)

Tiện lợi trong việc thiết kế trang web, dàn layout

Responsive framework

W3.CSS: <http://www.w3schools.com/w3css/default.asp>

Bootstrap: <http://www.w3schools.com/bootstrap/default.asp>