



# *Cascading Style Sheets* *- CSS*



# Nội dung

---

- What is CSS?
- Styling with Cascading Stylesheets (CSS)
- Selectors and style definitions
- Linking HTML and CSS
- Fonts, Backgrounds, Borders
- The Box Model
- Alignment, Z-Index, Margin, Padding
- Positioning and Floating Elements
- Visibility, Display, Overflow

# Cascading Style Sheets

---

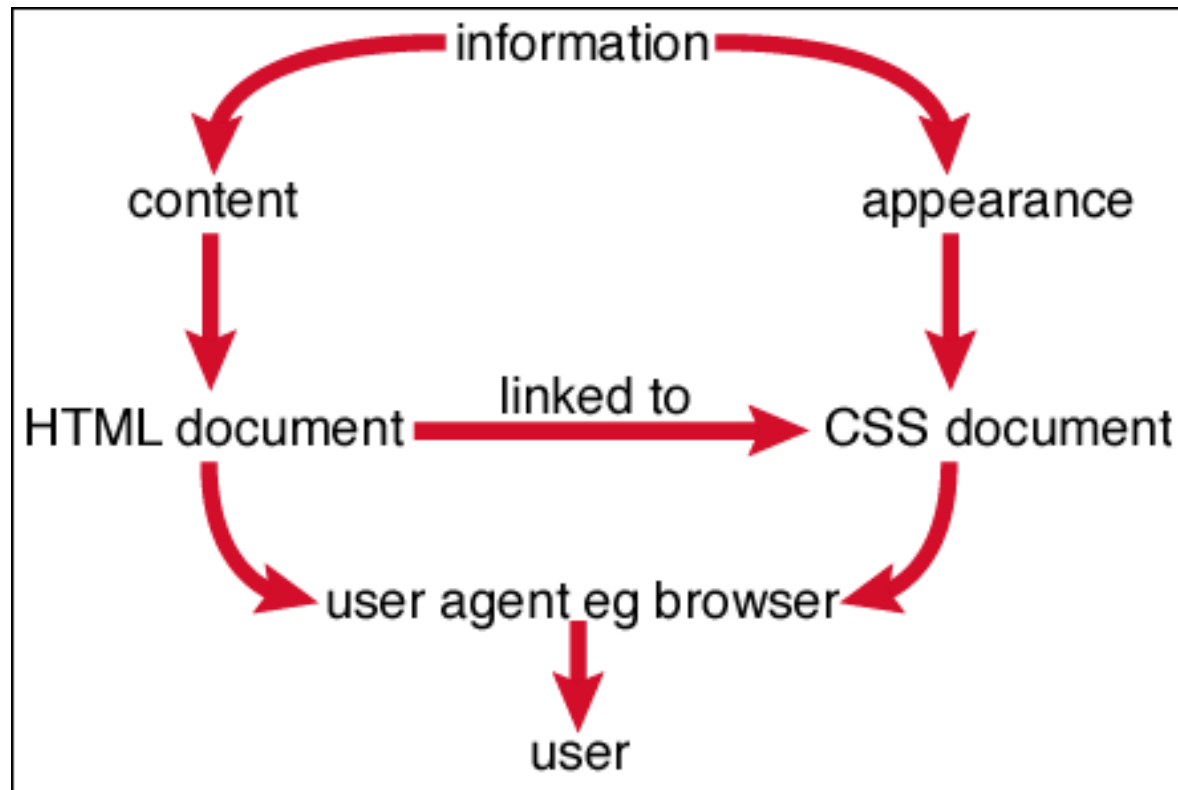
## ▪ **Vai trò của Cascading Style Sheets**

- ❖ **Tách** phần trình bày tài liệu khỏi nội dung tài liệu
- ❖ **Cung cấp** các tính năng khác để định dạng giao diện
  - Có thể xác định font, size, background color, background image, margins, etc.
- ❖ **Chia sẻ** các định dạng cho toàn bộ Web site
  - Giảm thời gian phát triển và bảo trì
- ❖ **Chỉ định một định nghĩa lớp** cho một kiểu, xác định hiệu quả các phần tử HTML mới
- ❖ **Linh hoạt**: các quy tắc được áp dụng theo cách phân cấp (quy tắc ưu tiên)

# How do Style Sheets work?

---

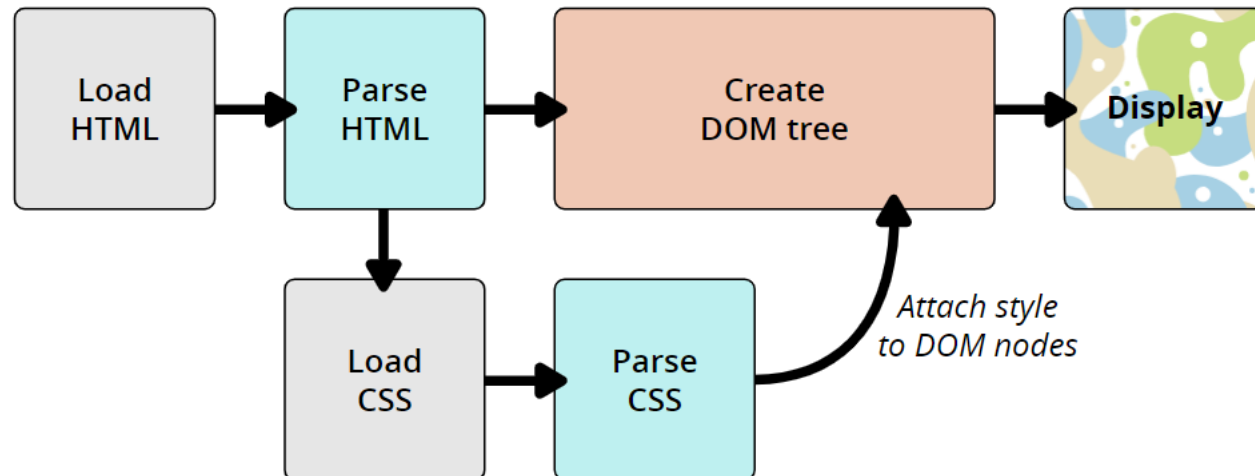
- Sử dụng **style sheets** tách biệt nội dung (content) và phần trình bày (appearance)



# How do Style Sheets work?

---

- **Cascading Style Sheet** là một chuỗi các lệnh (statements), mỗi lệnh thực hiện 2 việc
  - ❖ Xác định các phần tử trong tài liệu HTML mà nó ảnh hưởng đến
  - ❖ Cho trình duyệt biết cách hiển thị các phần tử HTML
- Khi trình duyệt hiển thị trang web thì phải kết hợp nội dung của tài liệu với thông tin Style Sheet



# Cascading Style Sheets

---

- **Cascading Style Sheets - CSS** là một tập tin văn bản đơn giản có tên với phần mở rộng **.css**
- **CSS Level 1 (1996)**
  - ❖ Áp dụng các kiểu đơn giản cho các phần tử HTML
- **CSS Level 2 (1998)**
  - ❖ Kết hợp và mở rộng CSS-1
  - ❖ Hỗ trợ media-specific style sheets (trình duyệt trực quan, thiết bị âm thanh, máy in, v.v.)

# Cascading Style Sheets

---

- **CSS, Level 3** (Đang trong quá trình phát triển)

- ❖ Kết hợp và mở rộng CSS-2
- ❖ Tập trung vào mô-đun hóa đặc tả CSS
- ❖ Bộ chọn mới, đường viền và hình nền lạ mắt, văn bản dọc, tương tác người dùng, giọng nói ....

- **Lưu ý:**

- ❖ *Một trình duyệt có thể không hỗ trợ tất cả các tính năng trong CSS level 1 và level 2*

# What kinds of style does CSS support?

---

## ■ CSS 1

- ❖ **Font properties:** such as typeface and emphasis
- ❖ **Color of text:** backgrounds, and other elements
- ❖ **Text attributes:** spacing between words, lines
- ❖ **Alignment** of text, images, tables, etc.
- ❖ **Margin**, border, padding, and positioning of most elements
- ❖ Dimension

## ■ CSS 2

- ❖ Relative and fixed positioning of most elements
- ❖ Bidirectional texts
- ❖ New font properties



# What kinds of style does CSS support?

---

- **CSS 3:** là một tiêu chuẩn mới nhất của các phiên bản css
  - Media Queries
  - Namespaces
  - Selectors Level 3
  - Color

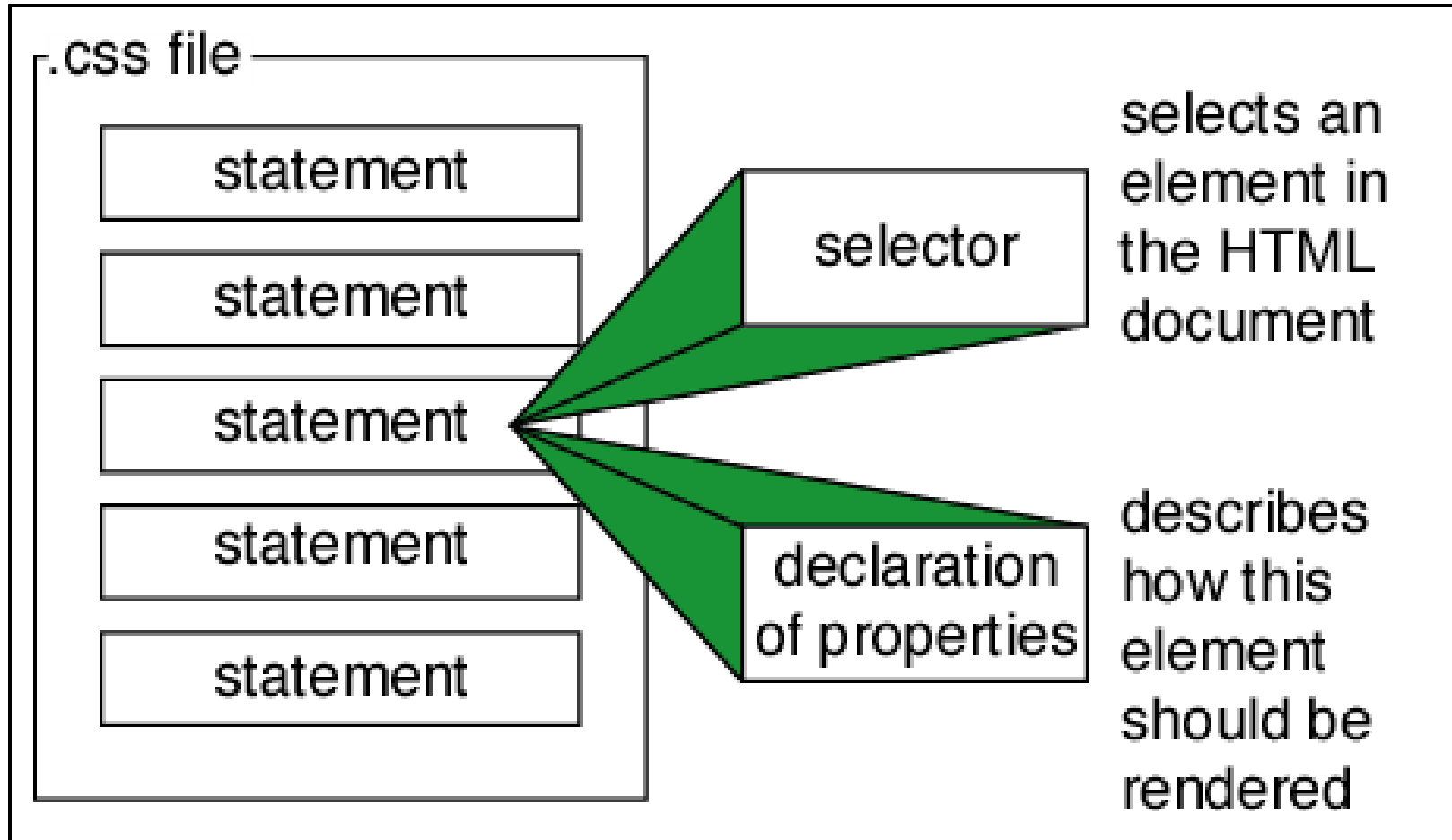
# Các thành phần của một style sheet

---

- **Câu lệnh (statements):** là một chuỗi hướng dẫn trong CSS
- **Phần tử (elements):** là nội dung bên trong các thẻ HTML.
- **Bộ chọn (selectors):** Phần của **câu lệnh (statement)** xác định các phần tử của trang web
- **Phần khai báo (declaration):** Phần của một câu lệnh cho trình duyệt biết cách các phần tử đã chọn hiển thị trên trình duyệt. **Một khai báo** chứa các Style sẽ được áp dụng cho phần tử đã chọn.

# Các thành phần của một style sheet

---



# Các thành phần của một style sheet

---

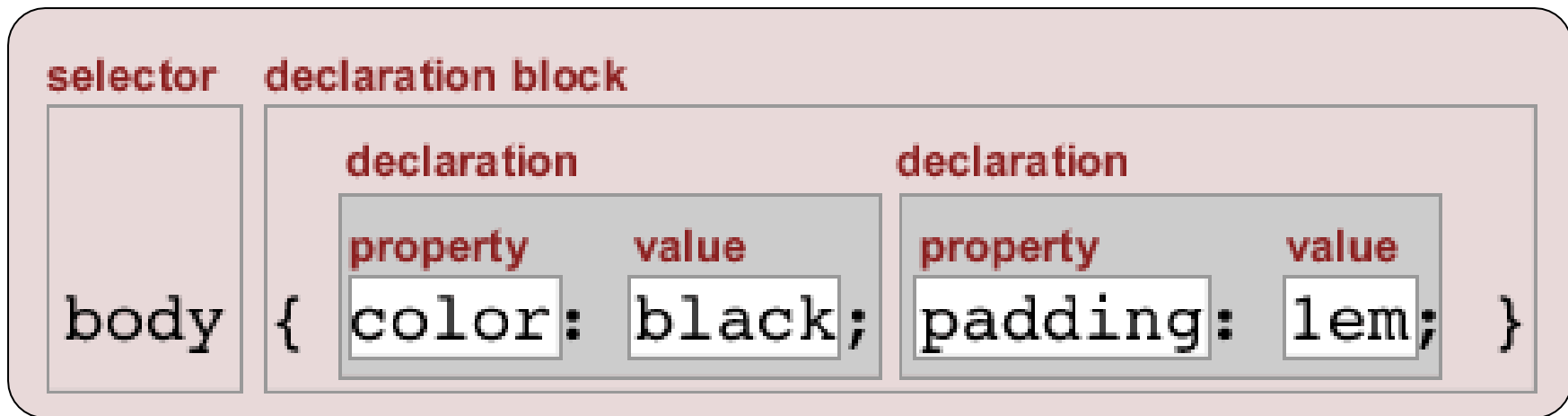
## ▪ Ví dụ: **statement**

```
body {  
    font-family: Verdana, "Minion Web", Helvetica, sans-serif;  
    font-size: 1em;  
    text-align: justify;  
}
```

- ❖ **body**: selector
- ❖ Câu lệnh (**statement**) ảnh hưởng đến các **phần tử** **<body>** của những trang có liên kết đến style sheet này
- ❖ Câu lệnh (**statement**): có một khai báo (**declaration**) gồm 3 thuộc tính: **font-family, font-size, text-align**

# Cú pháp của CSS

- **Một câu lệnh (Statement)** có một hoặc nhiều bộ chọn (selectors) và các khai báo (declaration)
  - ❖ **Bộ chọn (Selector):** chỉ định phần tử HTML
  - ❖ **Khai báo (Declaration):** là một hoặc nhiều thuộc tính được phân tách bằng dấu chấm phẩy “;”.
  - ❖ **Thuộc tính (Property):** gồm tên và giá trị được phân tách bằng dấu hai chấm “:”.



# Loại CSS

---

## ▪ External Style Sheet

- ❖ Lưu code CSS trong một tập tin bên ngoài (**.css**)
- ❖ Áp dụng cho bất kỳ tài liệu nào liên kết đến tập tin **.css**

## ▪ Internal or Embedded Style Sheet

- ❖ Định nghĩa trong trang HTML
- ❖ Chỉ ảnh hưởng đến trang chứa style được định nghĩa

## ▪ Inline Styles

- ❖ Style được định nghĩa như một thuộc tính của phần tử HTML
- ❖ Chỉ áp dụng cho 1 phần tử trên trang

# External Style Sheet

---

- **External Style Sheet:** Sử dụng thẻ `<link>` trong phần `<head>` để liên kết đến tập tin **css**

```
<head>  
    <link rel="stylesheet" type="text/css" href="style1.css">  
</head>
```

## style1.css

```
h1 { text-align: center; font-family: Arial; }  
h2 { color: #440000;  
    text-align: center;  
    font-family: Arial Black, Arial, Helvetica;  
}
```

# Internal Style Sheet

---

- **Internal Style Sheet:** sử dụng thẻ `<style>` trong phần `<head>`

```
<head>
  <style>
    tagName {property: value;}
  </style>
</head>
```

- Ví dụ:

```
<head>
  <style type="text/css">
    hr { color: sienna; }
    p { margin-left: 20px; }
    body {background-image:url("images/back40.gif");}
  </style>
</head>
```



# Inline Styles

---

- **Inline Styles:** Sử dụng thuộc tính **style** trong thẻ HTML để chỉ định các thuộc tính áp dụng cho phần tử đó

```
<tagname style="property:value;">
```

- Ví dụ:

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

# Selectors

---

- **CSS Selectors**: xác định phần tử HTML sẽ được áp dụng quy tắc CSS. CSS selector được chia thành 5 nhóm
  - ❖ **Simple selectors**: chọn các phần tử dựa trên name, id, class
  - ❖ **Combinator selectors**: chọn các phần tử dựa trên mối quan hệ cụ thể giữa chúng
  - ❖ **Pseudo-class selectors**: chọn các phần tử dựa trên một trạng thái nhất định
  - ❖ **Pseudo-elements selectors**: chọn và tạo style cho một phần của phần tử
  - ❖ **Attribute selectors**: chọn các phần tử dựa trên một thuộc tính hoặc giá trị thuộc tính

# Selectors

---

- **CSS element Selector**: chọn các phần tử HTML dựa trên tên phần tử

Ví dụ:

```
p{text-align: center; color: red;}
```

- **CSS id Selector**: chọn các phần tử HTML dựa trên thuộc tính ID của phần tử

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

# Selectors

---

- **CSS class Selector**: chọn các phần tử HTML với một thuộc tính **class** cụ thể.

Ví dụ:

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

*Tất cả các phần tử HTML có **class** = "center" sẽ có màu đỏ và được canh giữa*

- **grouping selector**: chọn tất cả các phần tử HTML có cùng định nghĩa **Style**.

Ví dụ:

```
h1, h2, p {text-align: center; color: red;}
```

# Selectors

---

- **Pseudo-classes** để xác định một trạng thái đặc biệt của một phần tử.
  - ❖ **Style** cho một phần tử khi người dùng rê chuột qua nó (mouses over)
  - ❖ **Style** cho một liên kết khi truy cập (visited)
  - ❖ **Style** cho một phần tử khi gets focus

- **Cú pháp**

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

# Selectors

---

## ▪ Pseudo-classes:

❖ :hover, :visited, :active , :lang

Ví dụ:

```
/* unvisited link */
a:link {
    color: #FF0000;
}
/* visited link */
a:visited {
    color: #00FF00;
}
```

```
/* mouse over link */
a:hover {
    color: #FF00FF;
}
/* selected link */
a:active {
    color: #0000FF;
}
```

# Selectors

---

- **Pseudo-elements**: để tạo kiểu cho các phần cụ thể của một phần tử.

❖ **:first-line** , **:before**, **:after**

Ví dụ:

```
a:hover { color: red; }  
p:first-line { text-transform: uppercase; }  
.title:before { content: "»"; }  
.title:after { content: "«"; }
```

# Selectors

---

## ▪ Match relative to element placement

Ví dụ:

```
p a {text-decoration: underline}
```

❖ *Style được áp dụng cho tất cả các thẻ **<a>** bên trong **<p>***

## ▪ \* – universal selector (bộ chọn đa năng)

Vì dụ:

```
p * {color: black}
```

❖ *Style này được áp dụng với tất cả các con của phần tử **<p>***



# Selectors

---

- **+ selector**: áp dụng cho các phần tử "anh chị em kế tiếp"

Ví dụ: `img + .link {float:right}`

❖ *Style này áp dụng cho tất cả các “anh chị em” có **class=link** xuất hiện ngay sau thẻ **<img>***

- **> selector**: áp dụng cho các nút con trực tiếp

Ví dụ: `p > .error {font-size: 8px}`

❖ *Style này áp dụng cho tất cả các phần tử có **class = error**, con trực tiếp của thẻ **<p>***

# Selectors

---

- **[ ]**: định dạng các thuộc tính thẻ bằng **biểu thức chính quy**

Ví dụ:

```
img[alt~=logo] {border: none}
```

❖ Style này sẽ áp dụng cho tất cả các thẻ **<img>** có thuộc tính **alt** chứa từ **logo**

- **.class1.class2**: áp dụng cho 2 lớp (hoặc tất cả) tại cùng một thời điểm

# Giá trị trong các quy tắc của CSS

---

- **Color Values:** màu sắc được chỉ định bằng cách sử dụng giá trị RGB, HEX, HSL, RGBA và HSLA.
- **Hexadecimal Colors:** Màu thập lục phân được chỉ định bằng: #RRGGBB, trong đó: RR (red), GG (green) and BB (blue)

❖ Ví dụ

```
#p1 {background-color: #ff0000;} /* red */  
#p2 {background-color: #00ff00;} /* green */  
#p3 {background-color: #0000ff;} /* blue */
```

# Giá trị trong các quy tắc của CSS

---

- **RGB Colors:** được chỉ định bằng hàm **rgb ()**

- **Cú pháp:** `rgb(red, green, blue)`

❖ Mỗi tham số (**red, green, blue**) xác định cường độ của màu, có thể là:

- Một số nguyên từ **0 đến 255**
- Hoặc một giá trị phần trăm từ **0% đến 100%**.

- Ví dụ:
- ```
#p1 {background-color: rgb(255, 0, 0);} /* red */  
#p2 {background-color: rgb(0, 255, 0);} /* green */  
#p3 {background-color: rgb(0, 0, 255);} /* blue */
```

# Giá trị trong các quy tắc của CSS

---

- **RGBA Colors:** Giá trị màu **RGBA** là mở rộng của RGB, với **alpha** - chỉ định độ mờ của đối tượng.

❖ Cú pháp:

**rgba**(red, green, blue, alpha)

- ❖ Tham số **alpha** có giá trị trong khoảng từ 0,0 (hoàn toàn trong suốt) đến 1,0 (hoàn toàn không trong suốt).

```
#p1 {background-color:rgba(255, 0, 0, 0.3);}/*red with opacity*/  
#p2 {background-color:rgba(0, 255, 0, 0.3);}/*green with opacity*/  
#p3 {background-color: rgba(0, 0, 255, 0.3);}/*blue with opacity*/
```

Red

Green

Blue

# Giá trị trong các quy tắc của CSS

---

## ▪ **Numeric values**

- ❖ Pixels, ems, e.g. `12px` , `1.4em`
- ❖ Points, inches, centimeters, millimeters
- ❖ E.g. `10pt` , `1in`, `1cm`, `1mm`
- ❖ Percentages, e.g. `50%`
  - Percentage of what?...
- ❖ Zero can be used with no unit: `border: 0;`

# Default Browser Styles

---

## ▪ CSS mặc định của trình duyệt:

- ❖ Trình duyệt cung cấp một tập hợp các style mặc định (font, size, line spacing, paragraph...).
- ❖ Trình duyệt khác nhau có các style mặc định khác nhau.
- ❖ Được sử dụng khi không có thông tin CSS hoặc bất kỳ thông tin Style nào khác trong tài liệu
- ❖ Nếu không sử dụng Style mặc định thì Reset CSS bằng cách đưa giá trị của thuộc tính của các phần tử trên trang về 0.

Ví dụ:

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

```
body, h1, p, ul, li { margin: 0; padding: 0; }
```

# CSS Cascade (Precedence)

---

- **Cascade** xác định cách kết hợp các **giá trị** của thuộc tính trong style sheet từ các nguồn khác nhau.
- **Cascading Style Sheets**: thứ tự khai báo CSS phân tầng và sự ảnh hưởng đến định dạng của các phần tử trên trang web.
- Ví dụ: một trang web cùng liên kết đến 2 tập tin .css với nội dung:

```
//custom.css  
h1 { color: red; }  
//custom2.css  
h1 { color: blue; }
```

- *Trình duyệt sẽ áp dụng định dạng nào?*
- *Dựa vào **cơ chế "Cascading" trong CSS.***



# CSS Precedence Rules- Quy tắc ưu tiên CSS

---

- **Quy tắc ưu tiên CSS:** tập các quy tắc giúp trình duyệt xác định những **style** cần áp dụng cho các phần tử HTML
  - ❖ !important
  - ❖ Specificity
  - ❖ Source Order

# CSS Precedence Rules- Quy tắc ưu tiên CSS

---

- **Quy tắc !Important** được sử dụng để thêm mức độ quan trọng cho **property/value** hơn mức bình thường.

- **Cú pháp:**

```
Selector{  
    Property: value !important;  
}
```

- Quy tắc **!Important** có độ ưu tiên cao nhất

# CSS Precedence Rules- Quy tắc ưu tiên CSS

---

## ▪ Quy tắc !Important

Ví dụ:

```
<style>
  div {
    font-family: Arial;
    font-size: 16px !important;
  }
  .specialText {
    font-size: 18px;
  }
</style>
```

```
<div class="specialText"> This is special text. </div>
```

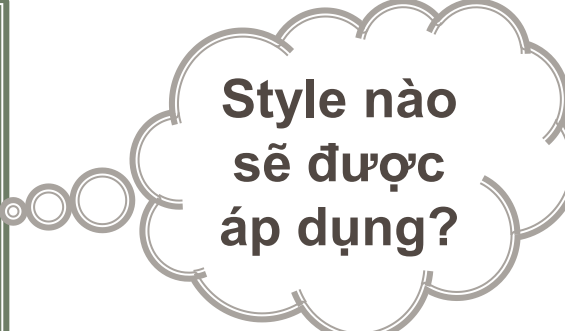
# CSS Precedence Rules- Quy tắc ưu tiên CSS

---

- **Specificity**: Tính cụ thể của quy tắc CSS phụ thuộc vào bộ chọn của nó. Bộ chọn CSS càng cụ thể thì **mức độ ưu tiên** của các khai báo thuộc tính CSS bên trong quy tắc CSS sở hữu bộ chọn càng cao.

- Ví dụ:

```
<style>
    #heading {color: orange; }
    .heading {color: red; }
    h1      {color: blue; }
</style>
```



Style nào  
sẽ được  
áp dụng?

```
<h1 class="heading" id="heading">Hello World!!</h1>
```

# CSS Precedence Rules- Quy tắc ưu tiên CSS

---

- **Specificity:** Các cấp độ "Specific" của CSS được xếp như sau:
  - ❖ **Inline styles:** thông qua thuộc tính **style** thì CSS có độ ưu tiên cao nhất.
  - ❖ **IDs – ID:** được sử dụng như một định danh duy nhất cho một phần tử trong trang, độ ưu tiên sau **Inline styles**
  - ❖ **Classes và pseudo-classes – Selector:** định danh theo nhóm. Mức độ "specific" cao hơn so với selector theo tên
  - ❖ **Elements và pseudo-elements:** Đây là loại selector kiểu "chung chung" nhất nên có độ ưu tiên thấp nhất.

# CSS Precedence Rules- Quy tắc ưu tiên CSS

---

- **Source Order:** nếu như các **CSS rule** có cùng **Importance** và **Specificity** thì trình duyệt sẽ xét đến thứ tự xuất hiện của nó. CSS rule nào được viết sau thì sẽ được ưu tiên áp dụng hơn so với CSS được viết trước.

- Ví dụ:

```
#heading { color: orange; color: grey; }  
.heading { color: red; }  
h1 { color: blue; }
```

```
//custom.css  
h1 { color: red; }  
//custom2.css  
h1 { color: blue; }
```

# CSS Precedence Rules- Quy tắc ưu tiên CSS

---

Lowest Priority	Highest Priority			
Highest Priority	Source	Browser	Reader	Author
	CSS Type	External	Internal	Inline
	Selector	Type	Class	ID

# Property Categories

---

- Text style – Fonts properties
- Text layout – Text alignments
- Foreground & Background
- Border
- Margin
- Padding
- Page layout
- Element type
- User interface



# Text Style Properties

---

## ▪ Font-specific

- ❖ font-weight
- ❖ font-family
- ❖ font-size
- ❖ font-style
- ❖ font-size-adjust
- ❖ font-stretch

## ▪ Text-specific

- ❖ text-decoration
- ❖ text-transform
- ❖ text-shadow

# Useful Font Properties

---

▪ **font-weight:** Relative weight - độ đậm của font chữ

❖ Giá trị: **normal** | **lighter** | **bold** | **bolder** | 100 | 200 | ... | 900 | inherit

```
H1 { font-weight : 200 }
```

```
H2 { font-weight : bolder }
```

Value	Name
100	Thin (Hairline)
200	Extra Light (Ultra Light)
300	Light
400	Normal
500	Medium

Value	Name
600	Semi Bold (Demi Bold)
700	Bold
800	Extra Bold (Ultra Bold)
900	Black (Heavy)

# Useful Font Properties

---

- **font-style:** thiết lập kiểu Font chữ trong font family

❖ Giá trị: **normal** | **italic** | **oblique**

Ví dụ:

```
P { font-style : normal; }
```

```
TH { font-style : italic; }
```

```
H1 { font-style: oblique; }
```

```
H2 { font-style: oblique 10deg; }
```

# Useful Font Properties

---

- **font-size** được chỉ định theo một trong những cách sau:
  - ❖ **Dùng từ khóa** có kích thước tuyệt đối (absolute-size) hoặc kích thước tương đối (relative-size)
    - **Absolute-size keywords:** dựa trên kích thước font chữ mặc định của người dùng, bao gồm các giá trị:

**xx-large | x-large | large | medium | small | x-small | xx-small**

- **Relative-size keywords:** font chữ sẽ lớn hơn hoặc nhỏ hơn so với kích thước font chữ của phần tử cha. Gồm các giá trị

**smaller | larger**

# Useful Font Properties

---

- **font-size** được chỉ định theo một trong những cách sau:
  - ❖ Dưới dạng **<length>** hoặc **<percentage>**: liên quan đến kích thước font chữ của phần tử trên trang.
    - **Giá trị <length>** : Đối với hầu hết các đơn vị font chữ tương đối, liên quan đến kích thước font chữ của phần tử cha. Gồm các loại đơn vị

Absolute length value: **pt, pc, in, cm, mm**

Relative length values: **em, ex, px, %**
    - **Giá trị <percentage>**: liên quan đến kích thước font chữ của phần tử cha

# Useful Font Properties

---

- **font-size:**

- **Cú pháp:**

```
/*<absolute-size> values */  
font-size: xx-small;  
font-size: x-small;  
font-size: small;  
font-size: medium;  
font-size: large;  
font-size: x-large;  
font-size: xx-large;  
font-size: xxx-large;
```

```
/*<relative-size> values */  
font-size: smaller;  
font-size: larger;  
/*<length> values */  
font-size: 12px;  
font-size: 0.8em;  
/*<percentage> values */  
font-size: 80%;  
/* Global values */  
font-size: inherit;  
font-size: initial;  
font-size: revert;  
font-size: unset;
```

# Useful Font Properties

---

## ■ font-size:

Ví dụ

Small H1

Larger H1

Kết quả

24 point H1

200% H1

HTML

```
<h1 class="small">Small H1</h1>
<h1 class="larger">Larger H1</h1>
<h1 class="point">24 point H1</h1>
<h1 class="percent">200% H1</h1>
```

CSS

```
.small
{font-size: xx-small;}
.larger
{font-size: larger;}
.point
{font-size: 24pt;}
.percent
{font-size: 200%;}
```

# Useful Font Properties

---

- **font-family:** chỉ định danh sách ưu tiên gồm một hoặc nhiều **font family names** hoặc **generic family names** cho phần tử đã chọn.
- **font-family** liệt kê một hoặc nhiều **font families**, được phân cách bằng dấu phẩy. Mỗi **font family** được chỉ định dưới dạng giá trị **<family-name>** hoặc **<generic-name>**.



# Useful Font Properties

---

- **<family-name>**: Tên của một **font family**. Một **font family** có chứa khoảng trắng phải được trích dẫn.

- Ví dụ:

"Times" và "Helvetica" là **font family**.

```
H1 { font-family: Arial } /* Arial is a font name */
```

```
H2 { font-family: serif }
```

*/\* serif là một từ khóa, gợi ý người dùng sử dụng font chữ thuộc "serif" font family \*/*

# Useful Font Properties

---

- **<generic-name>**: là một cơ chế dự phòng, để duy trì một số ý định của **style sheet author's intent** khi không có font chữ cụ thể khả dụng.

generic-name	Font
<b>serif</b>	Lucida Bright, Lucida Fax, Palatino, Palatino Linotype, Palladio, URW Palladio, serif.
<b>sans-serif</b>	Open Sans, Fira Sans, Lucida Sans, Lucida Sans Unicode, Trebuchet MS, Liberation Sans, Nimbus Sans L, sans-serif
<b>cursive</b>	Comic Sans MS
<b>fantasy</b>	Decorative fonts
<b>monospace</b>	Courier New (Font with fixed width)

# Text layout properties

---

## ▪ Cách bố cục văn bản trên một trang

- ❖ Letter-spacing
- ❖ Word-spacing
- ❖ Line-height
- ❖ Vertical-align
- ❖ Text-indent
- ❖ Text-align
- ❖ Direction

# Useful Text Properties

---

- **text-decoration:** thiết lập sự xuất hiện của các dòng trang trí trên văn bản, là cách viết tắt của
  - ❖ **text-decoration-line:** gạch dưới hoặc gạch ngang
  - ❖ **text-decoration-color:** màu sắc của trang trí.
  - ❖ **text-decoration-style:** kiểu của đường được sử dụng cho trang trí: **solid, wavy, dashed**
  - ❖ **text-decoration-thickness:** độ dày của đường được sử dụng để trang trí.

# Useful Text Properties

---

- **text-decoration:**

- **Cú pháp**

```
text-decoration: underline;  
text-decoration: overline red;  
text-decoration: none;  
/* Global values */  
text-decoration: inherit;  
text-decoration: initial;  
text-decoration: revert;  
text-decoration: unset;
```

# Useful Text Properties

---

## ▪ text-decoration:

Ví dụ:

```
<p class="wavy">Here's some text with wavy red underline!</p>  
<p class="both">This text has lines both above and below it.</p>
```

```
.wavy {  
  text-decoration-line: underline;  
  text-decoration-style: wavy;  
  text-decoration-color: red; }  
.both { text-decoration-line: underline overline; }
```

Kết quả

Here's some text with wavy red underline!

This text has lines both above and below it.

# Useful Text Properties

---

- **vertical-align**: Căn chỉnh theo chiều dọc hộp của **inline**, **inline-block**, **table-cell** box.

- **Cú pháp**:

```
/* Keyword values */
vertical-align: baseline;
vertical-align: sub;
vertical-align: super;
vertical-align: text-top;
vertical-align: text-bottom;
vertical-align: middle;
vertical-align: top;
vertical-align: bottom;
```

```
/* <length> values */
vertical-align: 10em;
vertical-align: 4px;
/*<percentage> values*/
vertical-align: 20%;
/* Global values */
vertical-align: inherit;
vertical-align: initial;
vertical-align: revert;
vertical-align: unset;
```

# Useful Text Properties

---

## ▪ Values for inline elements

❖ **Parent-relative values:** Các giá trị này căn chỉnh theo chiều dọc của phần tử so với phần tử chứa nó

<b>Baseline</b>	Căn theo đường cơ sở của phần tử gốc
<b>Sub</b>	Căn theo <b>subscript-baseline</b> của phần tử gốc.
<b>Super</b>	Căn theo <b>superscript-baseline</b> của phần tử gốc
<b>text-top</b>	Căn lề <b>trên</b> so với phần trên cùng của font của phần tử gốc.
<b>text-bottom</b>	Căn lề <b>dưới</b> so với phần dưới cùng của font của phần tử gốc
<b>Middle</b>	Căn giữa phần tử với đường cơ sở cộng với một nửa chiều cao x của phần tử gốc.



# Useful Text Properties

---

## ▪ Values for inline elements

❖ **Parent-relative values:** Các giá trị này căn chỉnh theo chiều dọc của phần tử so với phần tử chứa nó

<b>&lt;length&gt;</b>	Căn chỉnh so với độ dài đã cho phía trên đường cơ sở của phần tử gốc. Giá trị âm được phép.
<b>&lt;percentage&gt;</b>	Căn chỉnh theo tỷ lệ phần trăm đã cho trên đường cơ sở của phần tử gốc

# Useful Text Properties

---

## ▪ Values for inline elements

- ❖ **Line-relative values**: căn chỉnh phần tử theo chiều dọc so với toàn bộ dòng

<b>top</b>	Căn lề trên so với phần trên cùng của toàn bộ dòng.
<b>bottom</b>	Căn lề dưới so với phần dưới cùng của toàn bộ dòng.

- ❖ **Values for table cells**: căn chỉnh phần tử theo chiều dọc so với toàn bộ dòng

<b>top</b>	Căn lề trên của ô với đầu hàng.
<b>bottom</b>	Căn lề dưới của ô với cạnh dưới cùng của hàng.
<b>middle</b>	Căn giữa hộp đệm của ô trong hàng.

# Useful Text Properties

---

## ■ Ví dụ:

```
img.a { vertical-align: baseline;}  
img.b { vertical-align: text-top;}  
img.c { vertical-align: text-bottom;}
```

```
image with a default alignment.<br>  
image with a text-top alignment.<br>  
image with a text-bottom alignment.
```



image with a text-top alignment.

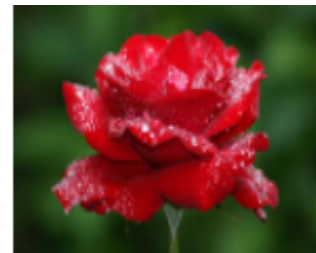


image with a text-bottom alignment.



image with a default alignment.

# Useful Text Properties

---

- **text-indent**: độ dài của khoảng trống (thụt lề) trước các dòng văn bản trong một khối.

❖ Specifies the indentation: **first line** of the paragraph

- **Cú pháp:**

```
/* <length> values */  
text-indent: 3mm;  
text-indent: 40px;  
/* Global values */  
text-indent: inherit;  
text-indent: initial;  
text-indent: revert;  
text-indent: unset;
```

```
/*<percentage> so với chiều rộng  
khối chứa*/  
text-indent: 15%;  
/* Keyword values */  
text-indent: 5em each-line;  
text-indent: 5em hanging;  
text-indent: 5em hanging each-line;
```

# Useful Text Properties

---

## ■ **text-indent**

❖ **<length>**: giá trị tuyệt đối. Các đơn vị có thể có:

**+/- pt, pc, in, cm, mm | +/- em, ex, px, %**

❖ **<percentage>**: Thụt lề theo tỉ lệ % của chiều rộng của khối chứa

❖ **Each-line**: Thụt dòng đầu tiên của vùng chứa

❖ **Hanging**: Tất cả các dòng ngoại trừ dòng đầu tiên sẽ được thụt lề

Ví dụ:

```
P { text-indent: -25px } /* Hanging indent */
```

# Useful Text Properties

---

## ▪ text-indent

Ví dụ:

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam  
nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat  
volutpat.</p>
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam  
nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat  
volutpat.</p>
```

```
p { text-indent: 5em; background: powderblue; }
```

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

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

# Useful Text Properties

---

- **line-height:** khoảng cách giữa hai đường cơ sở liên tiếp trong một đoạn văn
- **Cú pháp:**

```
/* Keyword value */  
line-height: normal;  
/* Giá trị không đơn vị: nhân với  
kích thước font chữ của phần tử */  
line-height: 3.5;  
/*<length> values */  
line-height: 3em;
```

```
/*<percentage>values*/  
line-height: 34%;  
/*Global values */  
line-height: inherit;  
line-height: initial;  
line-height: revert;  
line-height: unset;
```

# Useful Text Properties

---

## ▪ **line-height:**

- ❖ **Normal:** giá trị mặc định khoảng 1,2, tùy thuộc vào font-family của phần tử
- ❖ **Number:** số không có đơn vị này nhân với kích thước font chữ của phần tử.
- ❖ **Length:** tính toán chiều cao của line box
- ❖ **Percentage:** Giá trị được tính = <percentage> nhân với kích thước font chữ của phần tử



# Useful Text Properties

---

- **line-height:**

- ❖ normal | number | pt, pc, in, cm, mm | em, ex, px, %

- Ví dụ:

```
div{ line-height: 1.2; font-size: 10pt; }/*number/unitless*/  
div{ line-height: 1.2em; font-size: 10pt; }/*length*/  
div{ line-height: 120%; font-size: 10pt; }/*percentage*/  
div{ font: 10pt/1.2 Georgia,"Bitstream Charter",serif; }/*  
font shorthand */
```

# Background Properties

---

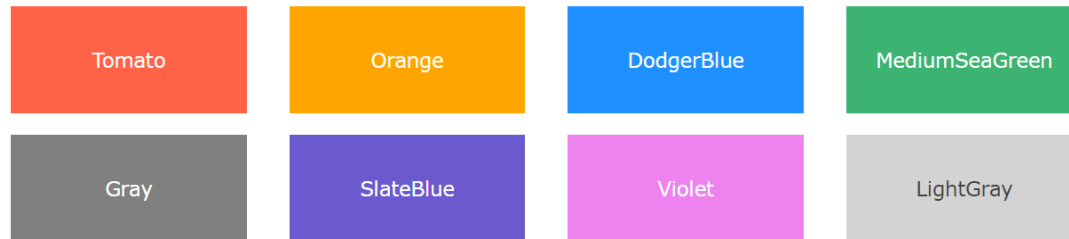
## ▪ Các thuộc tính thiết lập nền của một phần tử

- ❖ Background-color
- ❖ Background-image
- ❖ Background-attachment
- ❖ Background-repeat
- ❖ Background-position
- ❖ background

# Colors

---

- **Color Names:** được chỉ định bằng tên màu



- **Color Values:** màu sắc cũng có thể được chỉ định bằng cách sử dụng giá trị RGB, HEX, HSL, RGBA và HSLA.

❖ **Hexadecimal Colors:** Màu thập lục phân được chỉ định bằng: #RRGGBB, trong đó: RR (red), GG (green) and BB (blue)

❖ Ví dụ:

```
#p1 {background-color: #ff0000;} /* red */
#p2 {background-color: #00ff00;} /* green */
#p3 {background-color: #0000ff;} /* blue */
```

# Colors

---

## ■ Color Values:

❖ **RGB Colors:** được chỉ định bằng hàm **rgb ()**

❖ **Cú pháp:**

**rgb(red, green, blue)**

❖ Mỗi tham số (**red, green, blue**) xác định cường độ của màu, có thể là:

- Một số nguyên từ **0 đến 255**
- Hoặc một giá trị phần trăm từ **0% đến 100%**.

```
#p1 {background-color: rgb(255, 0, 0);} /* red */  
#p2 {background-color: rgb(0, 255, 0);} /* green */  
#p3 {background-color: rgb(0, 0, 255);} /* blue */
```

# Colors

---

## ■ Color Values:

❖ **RGBA Colors:** Giá trị màu RGBA là mở rộng của RGB, với kênh alpha - chỉ định độ mờ của đối tượng, được chỉ định bằng hàm `rgba ()`

❖ Cú pháp: `rgba(red, green, blue, alpha)`

- Tham số **alpha** có giá trị trong khoảng từ 0,0 (hoàn toàn trong suốt) đến 1,0 (hoàn toàn không trong suốt).

❖ Ví dụ

```
#p1 {background-color: rgba(255, 0, 0, 0.3);}
#p2 {background-color: rgba(0, 255, 0, 0.3);}
#p3 {background-color: rgba(0, 0, 255, 0.3);}
```

Red

Green

Blue

# Useful Color and Background Properties

---

- **background-color:** thiết lập màu nền cho một phần tử
- **Cú pháp:**

```
/* Keyword values */  
background-color: red;  
background-color: indigo;  
/* Hexadecimal value */  
background-color: #bbff00;  
/* Fully opaque */  
background-color: #bf0;
```

```
/* Fully opaque shorthand */  
background-color: #11ffee00;  
/* Fully transparent */  
background-color: #1fe0;  
/* Fully transparent shorthand */  
background-color: #11ffeeff;  
/* Fully opaque */  
background-color: #1fef;
```

# Useful Color and Background Properties

---

- **background-color:** thiết lập màu nền cho một phần tử
- **Cú pháp:**

```
/* RGB value */  
background-color: rgb(255, 255, 128); /* Fully opaque */  
background-color: rgba(117, 190, 218, 0.5);  
/* 50% transparent */ /* HSL value */  
background-color: hsl(50, 33%, 25%);  
/* Fully opaque */  
background-color: hsla(50, 33%, 25%, 0.75);  
/* Special keyword values */  
background-color: currentcolor;  
background-color: transparent;
```

# Image and Background Properties

---

- **background-image:** thiết lập một hoặc nhiều hình nền cho một phần tử.

- **Cú pháp:** `background-image: url(filename);`

- **Ví dụ:** `H2 {background-image:url(Bluedrop.gif);}`



# Image and Background Properties

---

- **background-repeat:**

- **Cú pháp:**

```
/* Keyword values */  
background-repeat: repeat-x;  
background-repeat: repeat-y;  
background-repeat: repeat;  
background-repeat: space;  
background-repeat: round;  
background-repeat: no-repeat;
```

```
/* Two-value syntax: horizontal |  
vertical */  
background-repeat: repeat space;  
background-repeat: repeat repeat;  
background-repeat: round space;  
background-repeat: no-repeat round;
```

# Image and Background Properties

---

## ■ background-repeat:

Ví dụ:

```
/* Background repeats */  
.one { background-repeat: no-repeat; }  
.two { background-repeat: repeat; }  
.three { background-repeat: repeat-x; }  
.four { background-repeat: repeat-y; }  
.five { background-repeat: space; }  
.six { background-repeat: round; }
```

# Image and Background Properties

---

## ■ background-repeat:

Ví dụ:

no-repeat



repeat-y



repeat



space



round



# Image and Background Properties

---

## ▪ background-repeat:

### ❖ <repeat-style>:

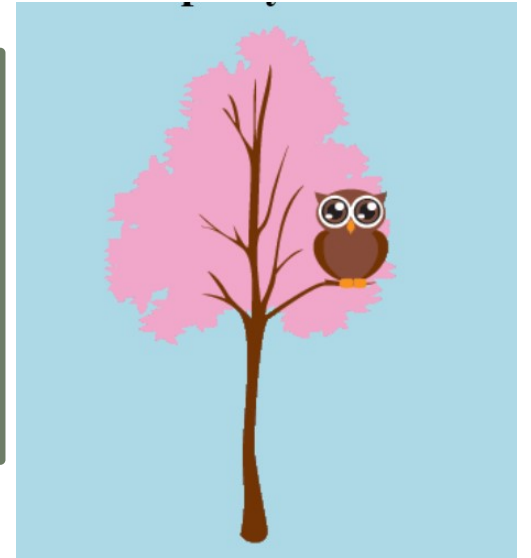
Single value	Two-value equivalent
repeat-x	repeat no-repeat
repeat-y	no-repeat repeat
repeat	repeat repeat
space	space space
round	round round
no-repeat	no-repeat no-repeat

# Image and Background Properties

---

- **Background:** Cho phép kết hợp các thuộc tính trong một phần tử duy nhất.
- Ví dụ:

```
body
{
    background: lightblue url("img_tree.gif")
               no-repeat fixed center;
}
```



# Image and Background Properties

---

- **Background-attachment:** Chỉ định xem hình nền là cố định hay được cuộn với tài liệu.

- **Cú pháp:**

```
/* Keyword values */
background-attachment: scroll;
background-attachment: fixed;
background-attachment: local;
/* Global values */
background-attachment: inherit;
background-attachment: initial;
background-attachment: revert;
background-attachment: unset;
```

# Image and Background Properties

---

## ▪ **Background-attachment:**

- ❖ **Fixed:** Hình nền sẽ không cuộn theo trang.
- ❖ **Local:** Hình nền sẽ cuộn với nội dung của phần tử
- ❖ **Scroll:** Hình nền sẽ cuộn theo trang. Đây là mặc định.
- ❖ **Initial:** thiết lập thuộc tính này thành giá trị mặc định của nó
- ❖ **Inherit:** Kế thừa thuộc tính từ phần tử cha.

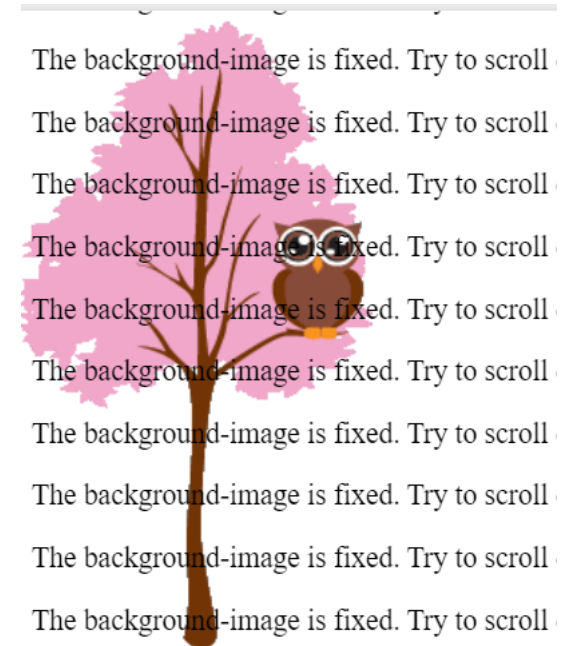
# Image and Background Properties

---

## ■ Background-attachment:

❖ Ví dụ:

```
body{  
  background-image: url("img_tree.gif");  
  background-repeat: no-repeat;  
  background-attachment: fixed;  
}
```





# Image and Background Properties

---

- **background-position**: thiết lập vị trí bắt đầu của hình nền.
- **Cú pháp**:

```
/*Keyword values*/  
background-position: top;  
background-position: bottom;  
background-position: left;  
background-position: right;  
background-position: center;  
/*<percentage> values */  
background-position: 25% 75%;
```

```
/* <length> values */  
background-position: 0 0;  
background-position: 1cm 2cm;  
background-position: 10ch 8em;  
/*Multiple images */  
background-position: 0 0, center;
```

# Image and Background Properties

---

- **background-position:**

- **Cú pháp:**

```
/* Edge offsets values */
background-position: bottom 10px right 20px;
background-position: right 3em bottom 10px;
background-position: bottom 10px right;
background-position: top right 10px;
/* Global values */
background-position: inherit;
background-position: initial;
background-position: revert;
background-position: unset;
```

# Image and Background Properties

---

## ▪ background-position:

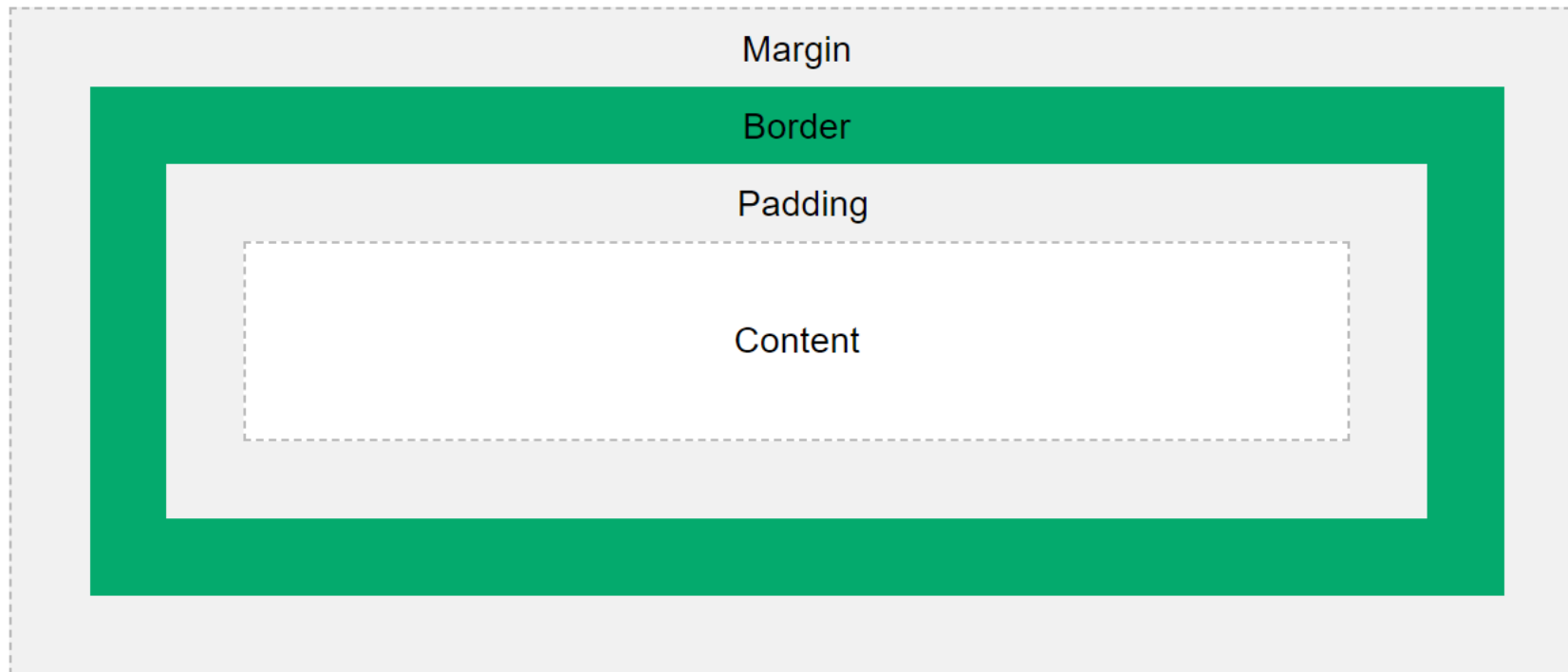
Ví dụ:

```
body {  
    background-image: url('w3css.gif');  
    background-repeat: no-repeat;  
    background-attachment: fixed;  
    background-position: 50% 50%;  
}
```

# Box Model

---

- **Box model:** Mọi phần tử có thể hiển thị như một hộp chứa với các thuộc tính margins, borders, padding, và actual content



# Box Model

---

- **Content area** là hình chữ nhật nhỏ nhất chứa dữ liệu được hiển thị tạo nên phần tử
- **Padding**: là khoảng trống giữa nội dung và các đường viền của phần tử. **Padding** lấy nền của phần tử
- **Border**: Đường viền bao quanh phần **padding** và **content**
- **Margin**: là khoảng cách giữa borders của phần tử và "containing box", Margin thường là trong suốt

# Box Model

---

- **Padding**: gồm 4 vùng **top**, **right**, **bottom**, **left**

padding-top  
padding-right  
padding-bottom  
padding-left

Ví dụ:

```
div {  
    border: 1px solid black;  
    background-color: lightblue;  
    padding-top: 50px;  
    padding-right: 30px;  
    padding-bottom: 50px;  
    padding-left: 80px;  
}
```

This div element has a top padding of 50px, a right padding of 30px, a bottom padding of 50px, and a left padding of 80px.

# Box Model

---

- **Margin:** gồm 4 vùng **top**, **right**, **bottom**, **left**

```
margin-top  
margin-right  
margin-bottom  
margin-left
```

Ví dụ

```
div {  
    border: 1px solid black;  
    margin-top: 100px;  
    margin-bottom: 100px;  
    margin-right: 150px;  
    margin-left: 80px;  
    background-color: lightblue;  
}
```

This div element has a top margin of 100px, a right margin of 150px, a bottom margin of 100px, and a left margin of 80px.

# Box Model

---

## ▪ **Border Properties**

### ❖ **width**

- `<length>` | thin | **medium** | thick

### ❖ **color**

- `<color value>` | transparent
- Giá trị mặc định là giá trị của thuộc tính "màu" của phần tử

### ❖ **style**

- **none** | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset



# Box Model

---

## ■ Border Properties

Ví dụ:

```
h1 { border: 5px solid red;}  
h2 { border: 4px dotted blue;}  
div{ border: double;}
```

**A heading with a solid red border**

**A heading with a dotted blue border**

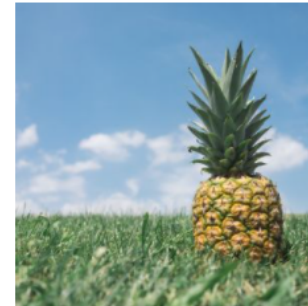
A div element with a double border.

# Float and clear

---

- **Float:** đặt một phần tử ở bên trái hoặc bên phải vùng chứa của nó, cho phép **văn bản** và các **phần tử inline** bao bọc xung quanh nó.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac...



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac...

# Float and clear

---

- **Float:**

- **Cú pháp**

```
/* Keyword values */  
float: left;  
float: right;  
float: none;  
float: inline-start;  
float: inline-end;
```

```
/* Global values */  
float: inherit;  
float: initial;  
float: revert;  
float: unset;
```

# Float and clear

---

## ▪ **Float:**

- ❖ **Left:** Phần tử phải nổi ở phía bên trái của khối chứa nó.
- ❖ **Right:** Phần tử phải nổi ở phía bên phải của khối chứa nó.
- ❖ **None:** Phần tử không được nổi.
- ❖ **inline-start:** Phần tử phải nổi ở phía bắt đầu của khối chứa nó.
- ❖ **inline-end:** Phần tử phải nổi ở phía cuối của khối chứa nó.

# Float and clear

---

- **Clear**: Khi sử dụng thuộc tính **float**, để phần tử tiếp theo bên dưới không bị trôi lên bên phải hoặc bên trái ta sử dụng thuộc tính **clear**.
- **Cú pháp**:

```
/* Keyword values */  
clear: none;  
clear: left;  
clear: right;  
clear: both;  
clear: inline-start;  
clear: inline-end;
```

```
/* Global values */  
clear: inherit;  
clear: initial;  
clear: revert;  
clear: unset;
```

# Float and clear

---

## ▪ Clear:

Ví dụ:

```
.div1 {  
  float: left;  
  padding: 10px;  
  border: 3px solid #73AD21;  
}  
.div2 {  
  padding: 10px;  
  border: 3px solid red;  
}
```

```
.div3 {  
  float: left;  
  padding: 10px;  
  border: 3px solid #73AD21;  
}  
.div4 {  
  padding: 10px;  
  border: 3px solid red;  
  clear: left;  
}
```

# Float and clear

---

## ■ Clear:

Ví dụ:

### Without clear

div1 div2 - Notice that div2 is after div1 in the HTML code. However, since div1 floats to the left, the text in div2 flows around div1.

### With clear

div3

div4 - Here, clear: left; moves div4 down below the floating div3. The value "left" clears elements floated to the left. You can also clear "right" and "both".

# CSS Positioning Schemes

---

- **Position CSS:** thiết lập vị trí của các phần tử trên trang
- **Cú pháp:**

```
position: static;  
position: relative;  
position: absolute;  
position: fixed;  
position: sticky;
```

```
/* Global values */  
position: inherit;  
position: initial;  
position: revert;  
position: unset;
```



# CSS Positioning Schemes

- **Static:** Các phần tử HTML được định vị tĩnh theo mặc định, các thuộc tính *top*, *bottom*, *left*, và *right* không được áp dụng.

- Ví dụ:



style.css

```
.parent {  
    border: 2px black dotted;  
    display: inline-block;  
}  
.box {  
    display: inline-block;  
    background: red;  
    width: 100px;  
    height: 100px;  
}  
#two {  
    background: green;  
}
```

index.html

```
<div class="parent">  
  <div class="box" id="one">One</div>  
  <div class="box" id="two">Two</div>  
  <div class="box" id="three">Three</div>  
  <div class="box" id="four">Four</div>  
</div>
```

# CSS Positioning Schemes

---

- **Relative**: phần tử được định vị so với vị trí bình thường của nó. Khi thiết lập top, right, bottom, left sẽ điều chỉnh phần tử khỏi vị trí bình thường của nó.
- Ví dụ:

```
#two {  
    top: 20px;  
    left: 20px;  
    background: green;  
    position: relative;  
}
```

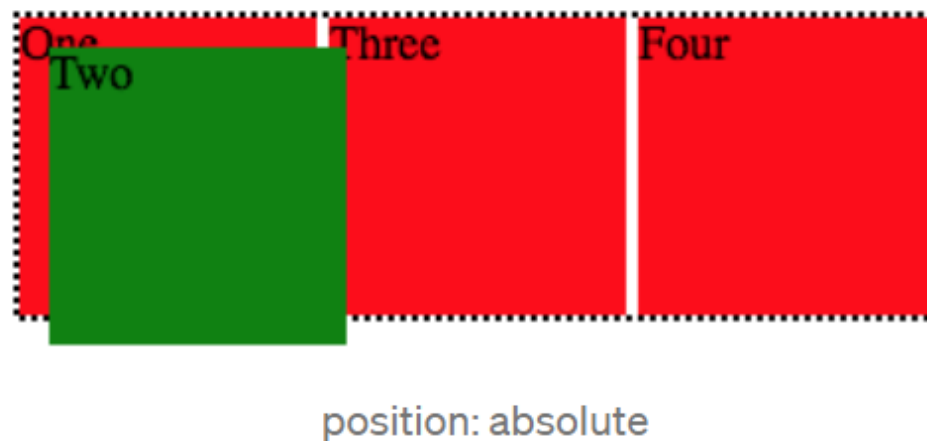


# CSS Positioning Schemes

---

- **absolute**: phần tử được định vị tương đối so với phần tử chứa nó gần
- Ví dụ:

```
#two {  
    top: 20px;  
    left: 20px;  
    background: green;  
    position: absolute;  
}
```



# CSS Positioning Schemes

---

- **Fixed**: được định vị so với chế độ xem, phần tử cố định khi cuộn trang. Các thuộc tính top, right, bottom, left được áp dụng.
- Ví dụ:

```
#two {  
    top: 20px;  
    left: 20px;  
    background: green;  
    position: fixed;  
}
```

# Useful Positioning Properties

---

## ▪ **top, left, bottom, right**

- ❖ Chỉ định các cạnh top/left/bottom/right của phần tử so với phần tử chứa nó
- ❖ `<length> | % | auto`

## ▪ **position**

- ❖ Mô tả cách vị trí được xác định cho phần tử chứa (parent window)
- ❖ `absolute | relative | static | fixed`

## ▪ **visibility**

- ❖ Xác định một lớp có thể hiển thị hoặc bị ẩn
- ❖ `visible | hidden`

## ▪ **z-index**

- ❖ Chỉ định phần tử nào sẽ xuất hiện bên trên / bên dưới các phần tử khác
- ❖ `<integer> | auto`
- ❖ The higher the numbers, the higher the level

# Opacity Property

---

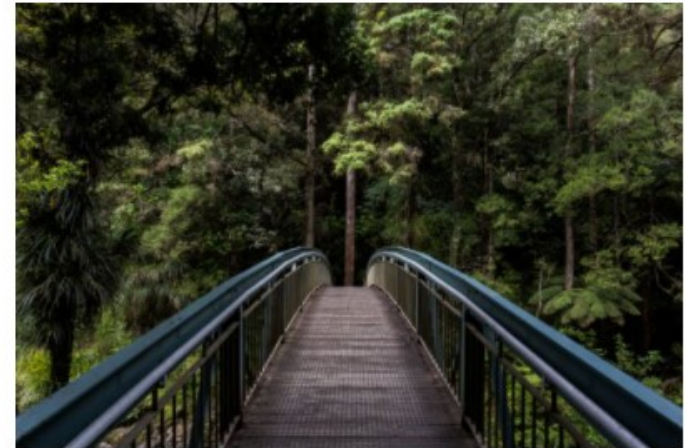
- **opacity:**



opacity 0.2



opacity 0.5



opacity 1  
(default)

# Opacity Property

---

- **opacity**: chỉ định mức độ trong suốt của một phần tử. Độ trong suốt có giá trị trong khoảng 0, 1: trong đó
  - ❖ 1 hoàn toàn không trong suốt
  - ❖ 0,5 là 50% nhìn xuyên
  - ❖ 0 là hoàn toàn trong suốt.
  - ❖ Với các trình duyệt **Mozilla** cũ, sử dụng **–moz-opacity**
  - ❖ Đối với **IE** sử dụng bộ lọc **alpha (opacity = value)**, value: từ 0 - 100;

# Visibility Property

---

- **Visibility:** khả năng hiển thị hoặc ẩn một phần tử mà không thay đổi bố cục của tài liệu
- Cú pháp

```
/* Keyword values */  
visibility: visible;  
visibility: hidden;  
visibility: collapse;
```

```
/* Global values */  
visibility: inherit;  
visibility: initial;  
visibility: revert;  
visibility: unset;
```



# Opacity Property

---

## ■ Visibility

- ❖ **Visible**: Giá trị mặc định. Phần tử có thể nhìn thấy được
- ❖ **Hidden**: phần tử không hiển thị, nhưng vẫn chiếm vị trí trên trang (similar to `opacity:0`)
- ❖ **Collapse**: Chỉ dành cho **table rows (<tr>), row groups (<tbody>), columns (<col>), column groups (<colgroup>)** Giá trị này loại bỏ một hàng hoặc cột, nhưng nó không ảnh hưởng đến bố cục bảng. Khoảng trống được chiếm bởi hàng hoặc cột sẽ có sẵn cho nội dung khác.

# Opacity Property

---

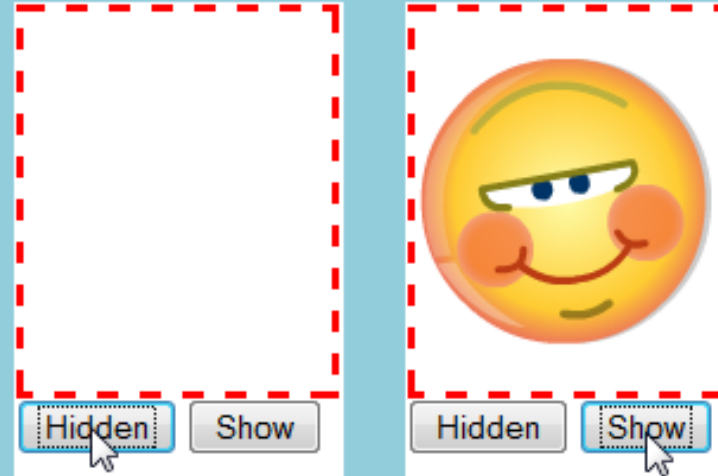
## ■ Visibility

❖ Ví dụ:

```
.frame
{
  border: dashed medium red;
  width: 130px;
}

html file
<body>
<div class="frame">
<p></p>
</div>
<input type="button"
  value="Hidden"
  onclick="myImg.style.visibility='hidden'">

<input type="button"
  value="Show"
  onclick="myImg.style.visibility='visible'">
</body>
```



# Display Property

---

- **Display:** kiểm soát sự hiển thị của phần tử và cách nó được hiển thị và nếu các dấu ngắt nên được đặt trước và sau phần tử
- **Cú pháp:**

```
/* legacy values */  
display: block;  
display: inline;  
display: inline-block;  
display: flex;  
display: inline-flex;  
display: grid;  
display: inline-grid;  
display: flow-root;
```

```
/* two-value syntax */  
display: block flow;  
display: inline flow;  
display: inline flow-root;  
display: block flex;  
display: inline flex;  
display: block grid;  
display: inline grid;  
display: block flow-root;
```

# Display Property

---

## ■ Display:

- ❖ **inline** Hiển thị một phần tử dạng inline (<span>). Mọi thuộc tính height and width không được áp dụng
- ❖ **Block**: Hiển thị một phần tử dạng block (như <p>, <div>). Bắt đầu trên một dòng mới và chiếm toàn bộ chiều rộng
- ❖ **inline-block**: Hiển thị một phần tử dạng vùng chứa khối cấp inline. Bản thân phần tử được định dạng là phần tử nội tuyến, nhưng áp dụng các giá trị **height** và **width**
- ❖ **Flex**: Hiển thị một phần tử dưới dạng vùng chứa linh hoạt cấp khối
- ❖ **inline-flex**: Hiển thị một phần tử dưới dạng một vùng chứa linh hoạt cấp inline

# Display Property

---

## ▪ Display

- ❖ **Grid**: Hiển thị một phần tử dưới dạng vùng chứa lưới cấp khối **inline-level flex container**
- ❖ **inline-grid**: Hiển thị một phần tử dưới dạng vùng chứa lưới cấp nội tuyến **inline-level grid container**
- ❖ **Table**: phần tử hoạt động giống như một phần tử `<table>`

# Display Property

---

## ■ Display

Ví dụ:

```
<article class="container">
  <span>First</span>
  <span>Second</span>
  <span>Third</span>
</article>
<article class="container">
  <span>First</span>
  <span>Second</span>
  <span>Third</span>
</article>
```

```
article span {
  background-color: black;
  color: white;
  margin: 1px;
}
article, span {
  padding: 10px;
  border-radius: 7px;
}
article, div {
  margin: 20px;
}
```

# Display Property

---

## ■ Display

Ví dụ:



`display: block;`



`display: inline;`



`display: inline-block;`

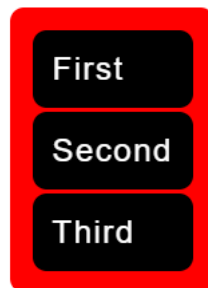
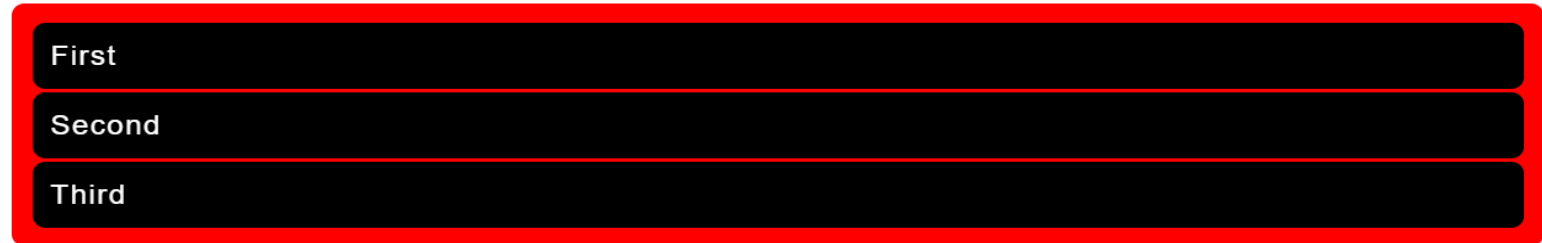
# Display Property

---

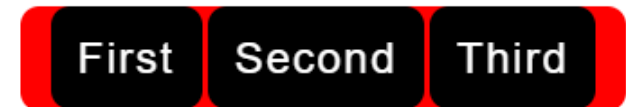
## ■ Display

Ví dụ:

`display: grid;`



`display: inline-grid;`



`display: table;`



# Overflow Property

---

- **overflow**: xác định hành vi của phần tử khi nội dung cần nhiều không gian hơn kích thước đã chỉ định bởi thuộc tính size hoặc các lý do khác
- **Cú pháp**:

```
/* Keyword values */  
overflow: visible;  
overflow: hidden;  
overflow: clip;  
overflow: scroll;  
overflow: auto;  
overflow: hidden visible;
```

# Overflow Property

---

## ▪ overflow:

Value	Description
<b>Visible</b> (thuộc tính mặc định)	Phần tràn không được cắt bớt, hiển thị bên ngoài hộp của phần tử.
<b>hidden</b>	Phần tràn được cắt bớt và phần còn lại của nội dung sẽ không hiển thị
<b>scroll</b>	Phần tràn được cắt bớt, nhưng một thanh cuộn được thêm vào để xem phần còn lại của nội dung
<b>auto</b>	Nếu phần tràn bị cắt bớt, một thanh cuộn sẽ được thêm vào để xem phần còn lại của nội dung

# Table Properties

---

- **Border**: xác định đường viền cho các bảng

Ví dụ: `table, th, td { border: 1px solid black; }`

- **border-collapse**: thiết lập các đường viền bảng thu gọn thành một đường viền duy nhất

Ví dụ: `table { border-collapse: collapse; }`  
`th, td { border: 1px solid black; }`

ID	Name	Age
1	John Carter	30
2	Harry Potter	11
3	Peter Parker	21

Separate Border (Default)

ID	Name	Age
1	John Carter	30
2	Harry Potter	11
3	Peter Parker	21

Collapse Border

# Table Properties

---

- **CSS padding**: thêm khoảng trống giữa nội dung ô bảng và đường viền ô.

Ví dụ:

```
th, td { padding: 15px; }
```

- **CSS border-spacing**: điều chỉnh khoảng cách giữa các đường viền của các ô

Ví dụ:

```
table { border-spacing: 10px; }
```

# Table Properties

---

- **zebra-striped tables**: sử dụng bộ chọn **nth-child()** và thêm màu nền cho tất cả các hàng chẵn (hoặc lẻ)

Ví dụ:

```
table {  
    border-collapse: collapse;  
    width: 100%;  
}  
th, td {  
    text-align: left;  
    padding: 8px;  
}  
tr:nth-child(even) {  
    background-color: #f2f2f2;  
}
```

# Table Properties

---

- **zebra-striped tables**

Ví dụ:

First Name	Last Name	Points
Peter	Griffin	\$100
Lois	Griffin	\$150
Joe	Swanson	\$300
Cleveland	Brown	\$250

# Table Properties

---

- **Hoverable Table:** bộ chọn **:hover** trên `<tr>` để **highlight** các hàng trong bảng khi di chuột qua

- Ví dụ:

```
table {  
    border-collapse: collapse;  
    width: 100%;  
}  
th, td {  
    text-align: left;  
    padding: 8px;  
}  
tr:hover {background-color: yellow;}
```

# Table Properties

---

## ▪ Hoverable Table:

Ví dụ:

First Name	Last Name	Points
Peter	Griffin	\$100
Lois	Griffin	\$150
Joe	Swanson	\$300
Cleveland	Brown	\$250



# Table Properties

---

- **Table Width and Height:** thiết lập chiều rộng và chiều cao của bảng

Ví dụ: 

```
table { width: 100%; } th { height: 40px; }
```

- **Controlling the Table Layout:** sử dụng để bố trí các ô, hàng và cột trong bảng, thuộc tính này nhận một trong hai giá trị
  - ❖ **Auto:** độ rộng của bảng và các ô của nó được điều chỉnh để phù hợp với nội dung.
  - ❖ **Fixed:** bố cục theo chiều ngang của bảng không phụ thuộc vào nội dung của các ô.

# List Properties

---

- **list-style**: áp dụng trên phần tử danh sách HTML như CSS list-style-type, CSS list-style-image và CSS list-style-position
- **Cú pháp**

```
/* type */  
list-style: square;  
/* image */  
list-style: url('../img/shape.png');  
/* position */  
list-style: inside;
```

# List Properties

---

- **list-style:**

- **Cú pháp**

```
/* type | position */  
list-style: georgian inside;  
/* type | image | position */  
list-style: lower-roman url('../img/shape.png') outside;  
/* Keyword value */  
list-style: none;
```

# List Properties

---

## ▪ list-style:

Ví dụ:

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

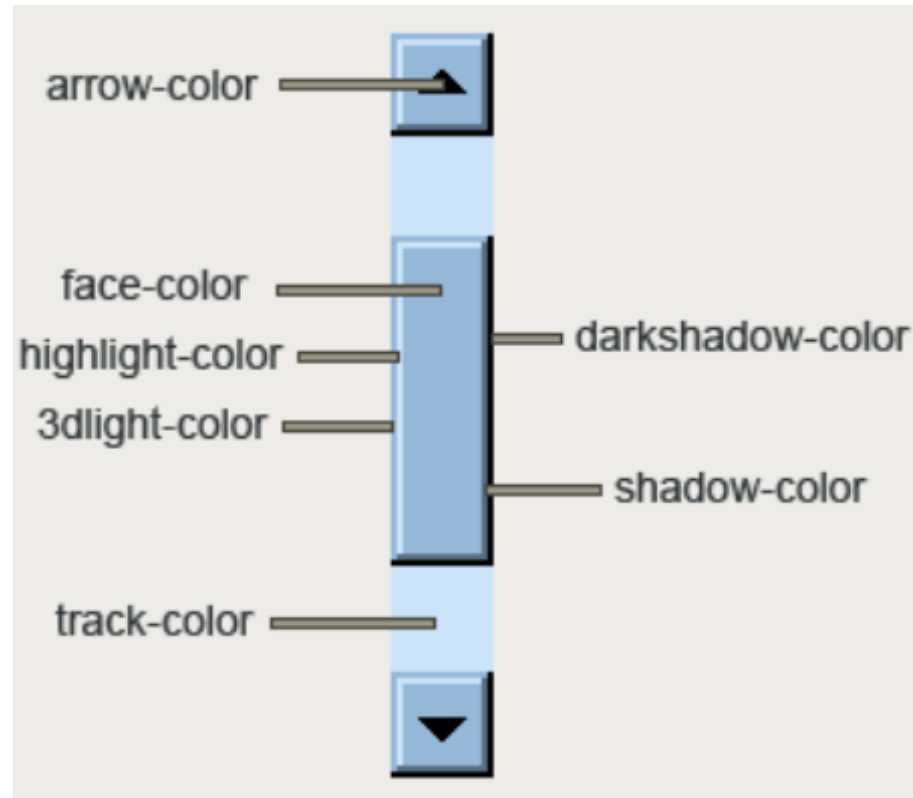
```
<ul>  
    <li>Coffee</li>  
    <li>Tea</li>  
    <li>Coca Cola</li>  
</ul>
```

- Coffee
- Tea
- Coca Cola

# Scrollbar Properties

---

- **Scroll bars**: cho phép cửa sổ trình duyệt hiển thị văn bản liên tục. **CSS Scrollbars** giúp tùy chỉnh sự xuất hiện của thanh cuộn.



# Scrollbar Properties

---

## ■ **scrollbar-color** :

- ❖ **scrollbar-track-color**: Sets the color for scroll bar track
- ❖ **scrollbar-face-color**: Sets the color for the scroll bar slider and the boxes that contain the scrollbar arrows
- ❖ **scrollbar-arrow-color**: Sets the scroll bar arrow color
- ❖ **scrollbar-3dlight-color**: Sets the scroll bar 3D light color
- ❖ **scrollbar-highlight-color**: Sets the scroll bar highlight color
- ❖ **scrollbar-shadow-color**: Sets the scroll bar shadow color
- ❖ **scrollbar-darkshadow-color**: Sets the scroll bar dark shadow color

# Scrollbar Properties

---

## ■ scrollbar-color:

Ví dụ:

```
html {  
    scrollbar-face-color: #FF0000;  
    scrollbar-shadow-color: #0000FF;  
    scrollbar-highlight-color: #00FF00;  
    scrollbar-3dlight-color: #FF00FF;  
    scrollbar-darkshadow-color: #00FFFF;  
    scrollbar-track-color: #FFFF00;  
    scrollbar-arrow-color: #000000;  
}
```

# Scrollbar Properties

---

- **scrollbar-width**: độ dày tối đa của thanh cuộn khi chúng được hiển thị.









Ví dụ:

```
scrollbar-width: none;  
scrollbar-width: thin;  
scrollbar-width: auto;
```



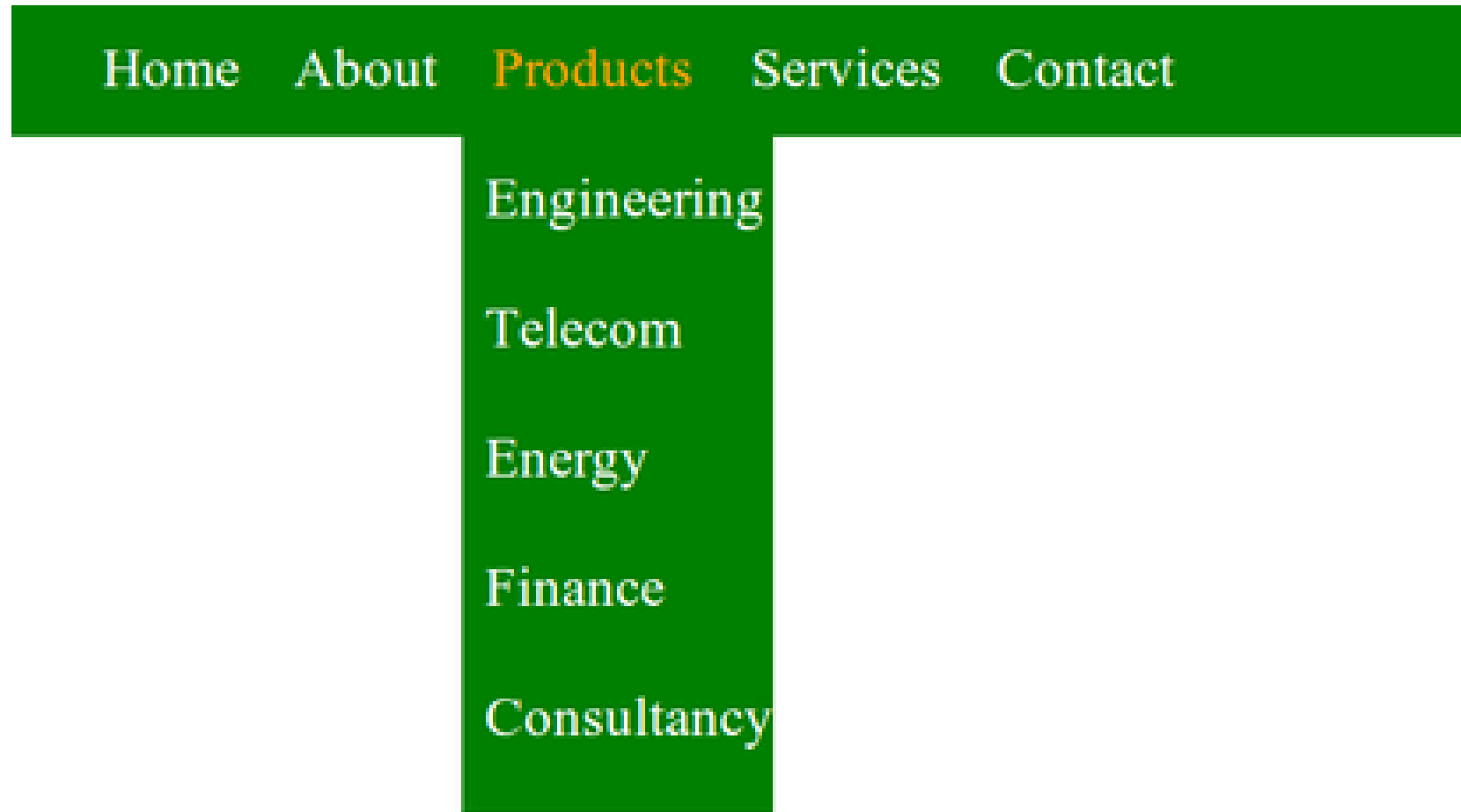
# Screen Properties

- **cursor Property**: chỉ định loại con trỏ sẽ được hiển thị khi con trỏ chuột được đặt trên nội dung

Icon	Value	Live example (move mouse over box)
	auto	<i>The User Agent determines the cursor to</i>
	default	style="cursor: default;"
	hand	style="cursor: hand;"
	pointer	style="cursor: pointer;"
	hand & pointer	style="cursor: pointer; cursor: hand;"
	crosshair	style="cursor: crosshair;"
	text	style="cursor: text;"
	wait	style="cursor: wait;"
	help	style="cursor: help;"

# Menu with CSS

---



# Menu with CSS

```
<style type="text/css">
nav {
    display: block;
    position: absolute;
    top: 0;
    width: 100%;
    background-color: green;
}
li{
    list-style-type: none;
    display: inline;
    margin-right: 20px;
    font-size: 25px
}
a:link {
    color: #fff;
    text-decoration: none;
}
a:hover {
    color: orange;
    text-decoration: none;
}
li > ul { display: none }
li:hover ul {
    display: block; position: absolute;
    left: 200px;
    background-color: green; margin: 0;
}
li:hover ul li a:link{
    display: block; margin-left: -30px;
}
</style>
```

```
<body>
<nav>
<ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">About</a></li>
    <li> <a href="#">Products</a>
        <ul>
            <li><a href="#">Engineering</a></li>
            <li><a href="#">Telecom</a></li>
            <li><a href="#">Energy</a></li>
            <li><a href="#">Finance</a></li>
            <li><a href="#">Consultancy</a></li>
        </ul>
    </li>
    <li><a href="#">Services</a></li>
    <li><a href="#">Contact</a></li>
</ul>
</nav>
</body>
```

# Borders

---

- **CSS Border**: thiết lập đường viền của một phần tử
- **Cú pháp**:

```
selector { border: border-width border-style border-color; }
```

```
selector  
{  
    border-color: Top Right Bottom Left;  
    border-style: Top Right Bottom Left;  
    border-width: Top Right Bottom Left;  
}
```

# Borders

---

- **border-style:**

- **Cú pháp**

```
/*top and bottom |left and right */  
border-style: dotted solid;  
/*top|left and right | bottom*/  
border-style: hidden double dashed;  
/* top | right | bottom | left */  
border-style: none solid dotted dashed;
```

```
/* Keyword values */  
border-style: none;  
border-style: hidden;  
border-style: dotted;  
border-style: dashed;  
border-style: solid;  
border-style: double;  
border-style: groove;  
border-style: ridge;  
border-style: inset;  
border-style: outset;
```

# Borders

---

## ▪ border-style:

Ví dụ:      HTML

```
<pre class="b1">none</pre>
<pre class="b2">hidden</pre>
<pre class="b3">dotted</pre>
<pre class="b4">dashed</pre>
<pre class="b5">solid</pre>
<pre class="b6">double</pre>
<pre class="b7">groove</pre>
<pre class="b8">ridge</pre>
<pre class="b9">inset</pre>
<pre class="b10">outset</pre>
```

CSS

```
pre{
    height: 80px;
    width: 120px;
    margin: 20px;
    padding: 20px;
    display: inline-block;
    background-color: palegreen;
    border-width: 5px;
    box-sizing: border-box;
}
```

# Borders

---

## ■ border-style:

Ví dụ:

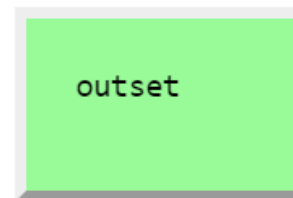
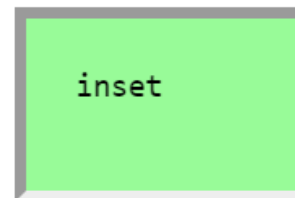
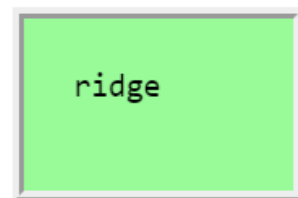
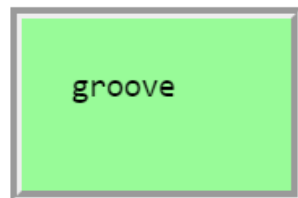
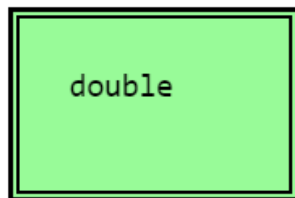
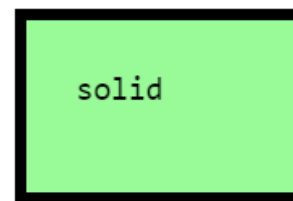
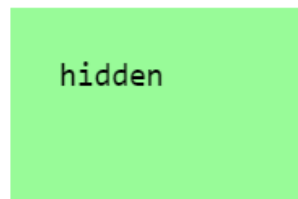
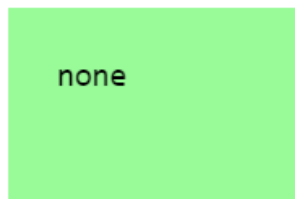
```
/*border-style example classes */  
.b1 { border-style: none; }  
.b2 { border-style: hidden; }  
.b3 { border-style: dotted; }  
.b4 { border-style: dashed; }  
.b5 { border-style: solid; }  
.b6 { border-style: double; }  
.b7 { border-style: groove; }  
.b8 { border-style: ridge; }  
.b9 { border-style: inset; }  
.b10 { border-style: outset; }
```

# Borders

---

## ■ border-style:

Ví dụ:





# Borders

---

- **border-color**: thiết lập màu đường viền của các phần tử
- Cú pháp:

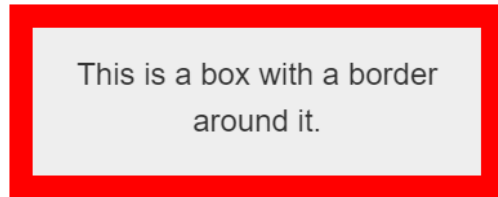
```
/* <color> values */  
border-color: red;  
/* top and bottom | left and right */  
border-color: red #f015ca;  
/* top | left and right | bottom */  
border-color: red rgb(240,30,50,.7) green;  
/* top | right | bottom | left */  
border-color: red yellow green blue;
```

# Borders

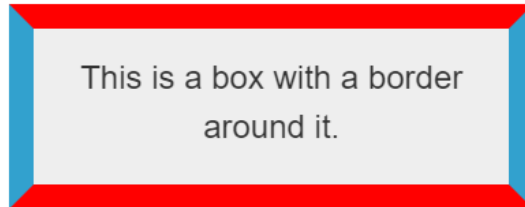
---

- **border-color:**

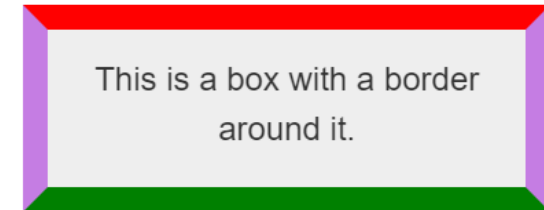
- Ví dụ:



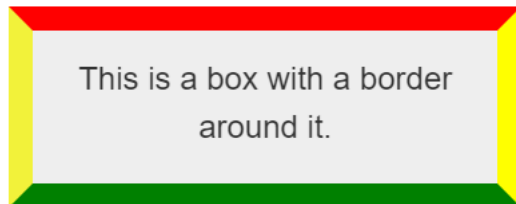
`border-color: red;`



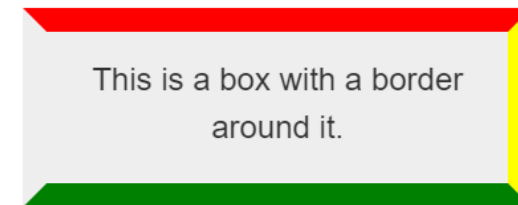
`border-color: red #32a1ce;`



`border-color: red rgba(170, 50, 220, .6) green;`



`border-color: red yellow green hsla(60, 90%, 50%, .8);`



`border-color: red yellow green transparent;`

# Rounded Borders

---

- **CSS border-radius:** làm tròn các góc của cạnh viền ngoài của phần tử
- **Cú pháp:**

```
/*Radius is set for all 4 sides */  
border-radius: 10px;  
/*top-left-and-bottom-right | top-right-and-bottom-left*/  
border-radius: 10px 5%;  
/*top-left | top-right-and-bottom-left | bottom-right */  
border-radius: 2px 4px 2px;  
/*top-left | top-right | bottom-right | bottom-left */  
border-radius: 1px 0 3px 4px;
```

# Rounded Borders

---

- **CSS border-radius:**

- **Cú pháp:**

```
/*(first radius values) / radius */  
border-radius: 10px / 20px;  
/*(first radius values) / top-left-and-bottom-right | top-  
right-and-bottom-left */  
border-radius: 10px 5% / 20px 30px;  
/*(first radius values) / top-left | top-right-and-bottom-  
left | bottom-right */  
border-radius: 10px 5px 2em / 20px 25px 30%;  
/*(first radius values) / top-left | top-right | bottom-  
right | bottom-left */  
border-radius: 10px 5% / 20px 25em 30px 35em;
```

# Rounded Borders

---

## ■ CSS border-radius:

Ví dụ:



`border-radius: 30px;`



`border-radius: 25% 10%;`



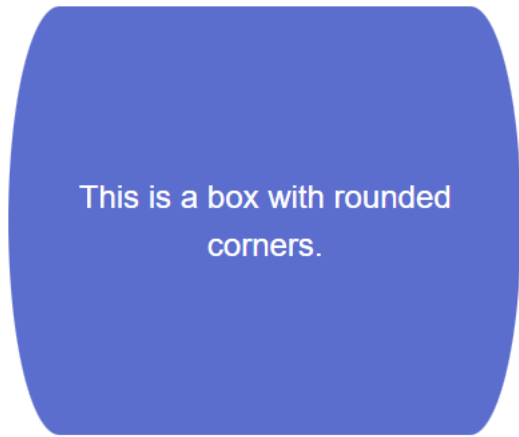
`border-radius: 10% 30% 50% 70%;`

# Rounded Borders

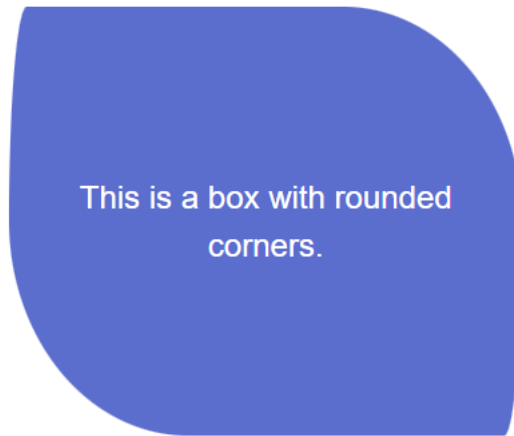
---

## ■ CSS border-radius:

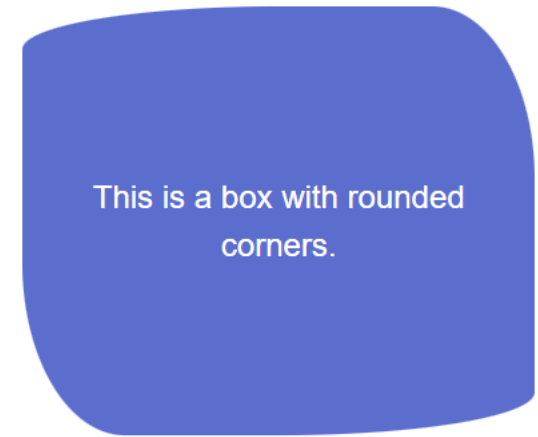
Ví dụ:



`border-radius: 10% / 50%;`



`border-radius: 10px 100px / 120px;`



`border-radius: 50% 20% / 10% 40%;`

# Rounded Borders

---





## ▪ **border-radius property:**




- ❖ Nếu thuộc tính gồm một, hai, ba hoặc bốn giá trị `<length>` hoặc `<percentage>` thì sử dụng để thiết lập một bán kính duy nhất cho các góc.
- ❖ Nếu giá trị của thuộc tính theo sau là `"/` và một, hai, ba hoặc bốn giá trị `<length>` hoặc `<percentage>` thì thiết lập bán kính bổ sung, do đó có thể tạo góc hình elip.

# Rounded Borders

---

## ▪ border-radius property:

values	Description
radius	
top-left-and-bottom-right	
top-right-and-bottom-left	
top-left-and-bottom-right	

values	Description
top-right	
bottom-right	
bottom-left	



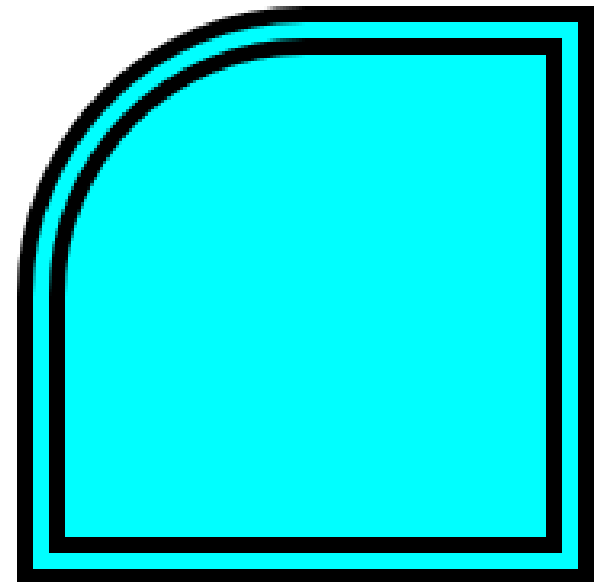
# Rounded Borders

---

- **border-radius property:**

- Ví dụ:

```
#vd
{
    border-top-left-radius: 50%;
    border: black 10px double;
    width: 100px;
    height: 100px;
    background-color: aqua;
}
```



# CSS Multiple Backgrounds

---

- **CSS background** có thể thêm nhiều hình nền cho một phần tử, thông qua thuộc tính background-image.
- Các hình nền khác nhau được phân tách bằng dấu phẩy và các hình được xếp chồng lên nhau, hình ảnh đầu tiên gần với người xem nhất. Các thuộc tính của hình nền:

- background-size
- background-origin
- background-clip

# CSS Multiple Backgrounds

---

## ▪ Ví dụ:

```
#example1 {  
    background-image: url(img_flwr.gif), url(paper.gif);  
    background-position: right bottom, left top;  
    background-repeat: no-repeat, repeat;  
    padding: 15px;  
}
```

```
<div id="example1">  
    <h1>Lorem Ipsum Dolor</h1>  
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam  
        nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat  
        volutpat.</p>  
    <p>Ut wisi enim ad minim veniam, quis nostrud exerci tation ull corper  
        suscipit lobortis nisl ut aliquip ex ea commodo consequat.</p>  
</div>
```

# CSS Multiple Backgrounds

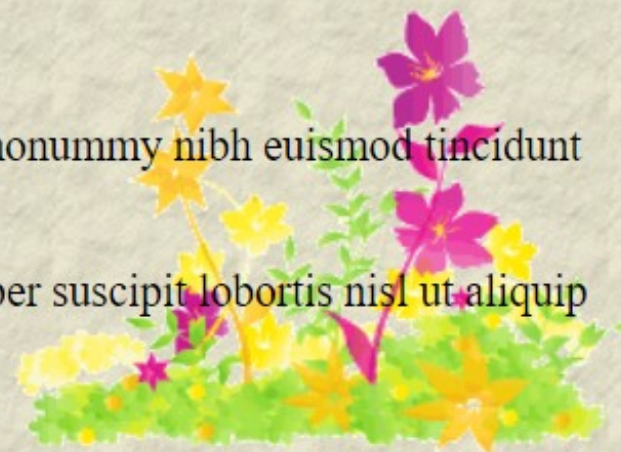
---

- Ví dụ:

## Lorem Ipsum Dolor

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 exercitation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.



# CSS Multiple Backgrounds

---

- **CSS Background Size**: chỉ định kích thước của hình nền. Kích thước có thể được chỉ định theo **lengths**, **percentages** hoặc bằng cách sử dụng một trong hai từ khóa: **contain** hoặc **cover**

- Ví dụ:

```
#div1 {  
    background: url(img_flower.jpg);  
    background-size: 100px 80px;  
    background-repeat: no-repeat;  
}
```

# CSS Multiple Backgrounds

---

- **background-size property** chấp nhận nhiều giá trị cho kích thước nền, danh sách các giá trị được phân tách bằng dấu phẩy, khi làm việc với nhiều nền.
- Ví dụ:

```
#example1 {  
    background: url(img_tree.gif) left top no-repeat,  
                url(img_flwr.gif) right bottom no-repeat,  
                url(paper.gif) left top repeat;  
    background-size: 50px, 130px, auto;  
}
```

# CSS Multiple Backgrounds

---

- **background-size property**

Ví dụ:



# Text effects

---

- **text-shadow**: thêm bóng cho văn bản, chấp nhận nhiều kiểu bóng, mỗi kiểu được phân cách bằng dấu phẩy.
- Ví dụ:

```
h1 {  
    text-shadow: 2px 2px;  
}
```

```
<h1>Text-shadow effect!</h1>
```

**Text-shadow effect!**



# Text effects

---

## ■ text-shadow

Ví dụ:

❖ Thêm màu cho bóng

```
h1 {  
    text-shadow: 2px 2px red;  
}
```

Text shadow effect!

Hiển thị văn bản màu trắng với bóng đen

```
h1 {  
    color: white;  
    text-shadow: 2px 2px 4px #000000;  
}
```

Text shadow effect!

# Text effects

---

- **text-transform:** chỉ định kiểu chữ của một phần tử. Nó có thể được sử dụng để làm cho văn bản xuất hiện ở dạng viết hoa toàn bộ hoặc viết thường hoặc viết hoa mỗi từ
- **Cú pháp:**

```
/* Keyword values */  
text-transform: none;  
text-transform: capitalize;  
text-transform: uppercase;  
text-transform: lowercase;  
text-transform: full-width;  
text-transform: full-size-kana;
```

# Text effects

---

## ▪ text-transform:

Ví dụ:

```
div.a {  
    text-transform: uppercase;  
}  
div.b {  
    text-transform: lowercase;  
}  
div.c {  
    text-transform: capitalize;  
}
```

# CSS box-shadow

---

- **box-shadow CSS**: thêm hiệu ứng đổ bóng xung quanh khung của phần tử, có thể đặt nhiều hiệu ứng, mỗi hiệu ứng được phân tách bằng dấu phẩy.

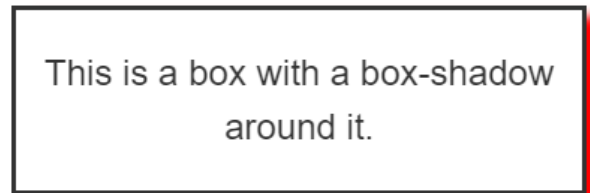


# CSS box-shadow

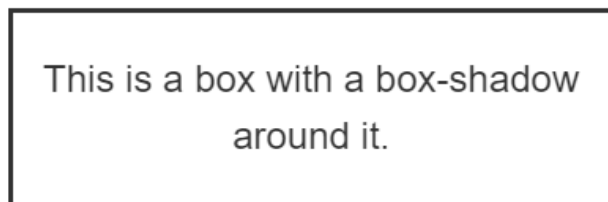
---

## ■ box-shadow CSS:

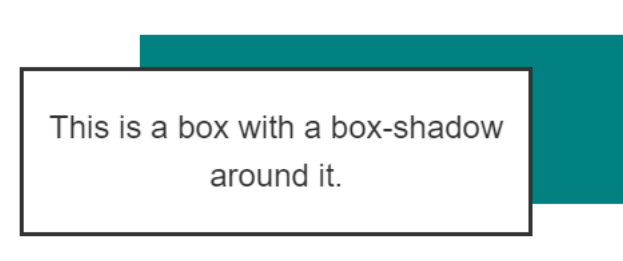
Ví dụ:



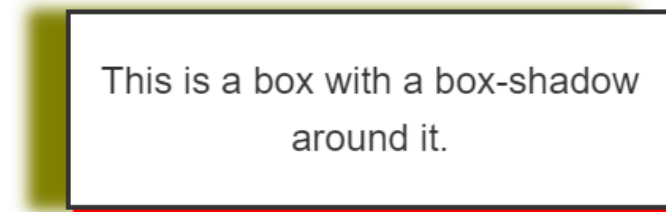
**box-shadow: 10px 5px 5px red;**



**box-shadow: 12px 12px 2px 1px rgba(0, 0, 255, .2);**



**box-shadow: 60px -16px teal;**



**box-shadow: 3px 3px red, -1em 0 .4em olive;**

# CSS box-shadow

---

## ■ box-shadow CSS:

Ví dụ:

```
#example1 {  
    border: 1px solid;  
    padding: 10px;  
    box-shadow: 5px 10px inset;  
}  
#example2 {  
    border: 1px solid;  
    padding: 10px;  
    box-shadow: 5px 10px 20px red inset;  
}
```

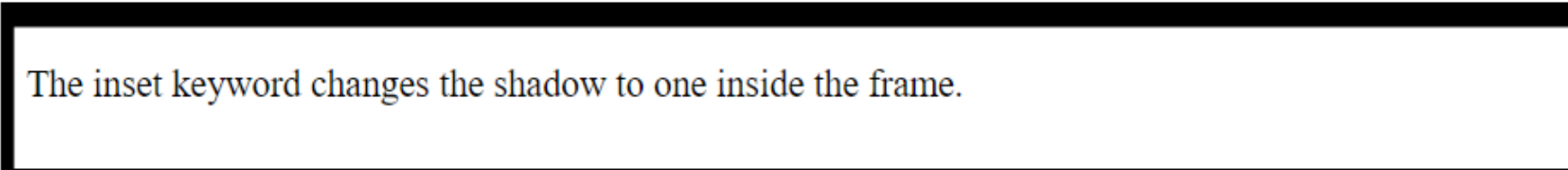
# CSS box-shadow

---

## ■ box-shadow CSS:


Ví dụ:

**box-shadow: 5px 10px inset:**



The inset keyword changes the shadow to one inside the frame.

**box-shadow: 5px 10px 20px red inset:**



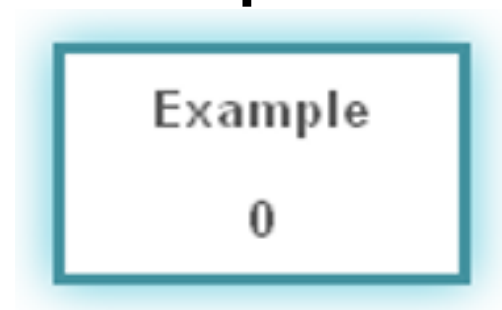
Inset, red and blur.

# CSS box-shadow

---

## ■ box-shadow CSS:

Ví dụ:



```
#example0{  
  box-shadow:0px 0px 20px 0px #000;  
  -moz-box-shadow:0px 0px 20px 0px #000;  
  -webkit-box-shadow:0px 0px 20px 0px #000;  
}
```



```
#example1{  
  box-shadow:0px 0px 20px 10px #000;  
  -moz-box-shadow:0px 0px 20px 10px #000;  
  -webkit-box-shadow:0px 0px 20px 10px #000;  
}
```



# CSS box-shadow

---

## ■ box-shadow CSS:

Ví dụ:



```
#example2{  
  box-shadow:20px 0px 20px 10px #000;  
  -moz-box-shadow:20px 0px 20px 10px #000;  
  -webkit-box-shadow:20px 0px 20px 10px #000;  
}
```



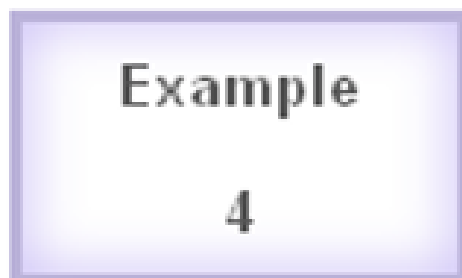
```
#example3{  
  box-shadow:20px 20px 20px 10px #000;  
  -moz-box-shadow:20px 20px 20px 10px #000;  
  -webkit-box-shadow:20px 20px 20px 10px #000;  
}
```

# CSS box-shadow

---

## ■ box-shadow CSS:

Ví dụ:



```
#example4{  
  box-shadow:inset 0px 0px 20px 0px #000;  
  -moz-box-shadow:inset 0px 0px 20px 0px #000;  
  -webkit-box-shadow:inset 0px 0px 20px 0px #000;  
}
```



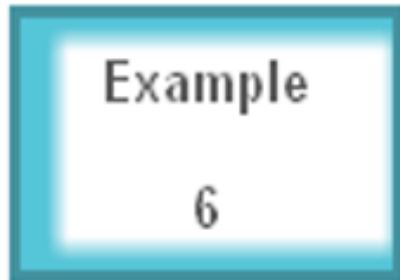
```
#example5{  
  box-shadow:0px 0px 10px 5px #000;  
  -moz-box-shadow:0px 0px 10px 5px #000;  
  -webkit-box-shadow:0px 0px 10px 5px #000;  
}
```

# CSS box-shadow

---

## ■ box-shadow CSS:

Ví dụ:



```
#example6{  
  box-shadow:inset 5px 0px 5px 5px #222;  
  -moz-box-shadow:inset 5px 0px 5px 5px #222;  
  -webkit-box-shadow:inset 5px 0px 5px 5px #222;  
}
```



```
#example7{  
  box-shadow:inset 5px 5px 5px 5px #222;  
  -moz-box-shadow:inset 5px 5px 5px 5px #222;  
  -webkit-box-shadow:inset 5px 5px 5px 5px #222;  
}
```

# CSS box-shadow

---

## ■ Multiple Shadows:

Ví dụ:

```
#example1 {  
  border: 1px solid;  
  padding: 10px;  
  box-shadow: 5px 5px blue, 10px 10px red, 15px 15px green;  
  margin: 20px;  
}  
#example2 {  
  border: 1px solid;  
  padding: 10px;  
  box-shadow: 5px 5px 8px blue, 10px 10px 8px red, 15px 15px 8px green;  
  margin: 20px;  
}
```

# CSS box-shadow

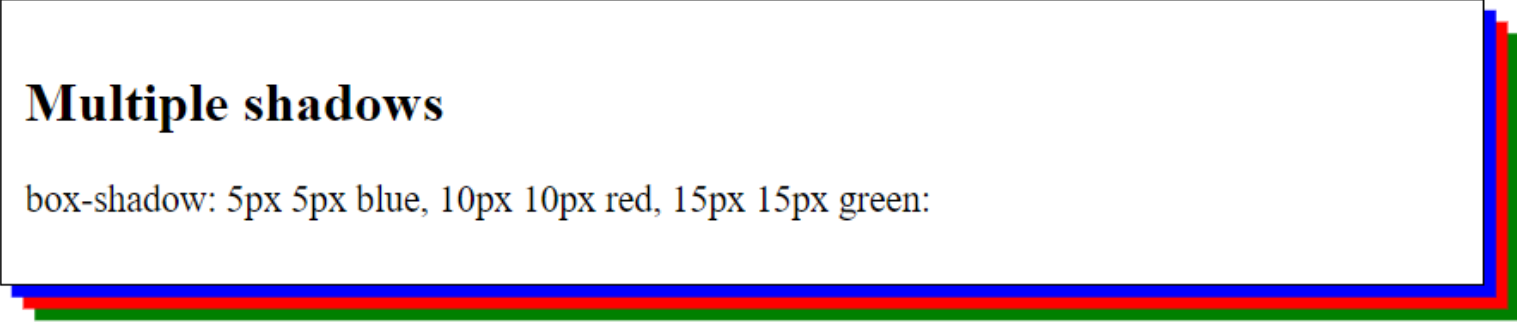
---

## ■ Multiple Shadows:

Ví dụ:

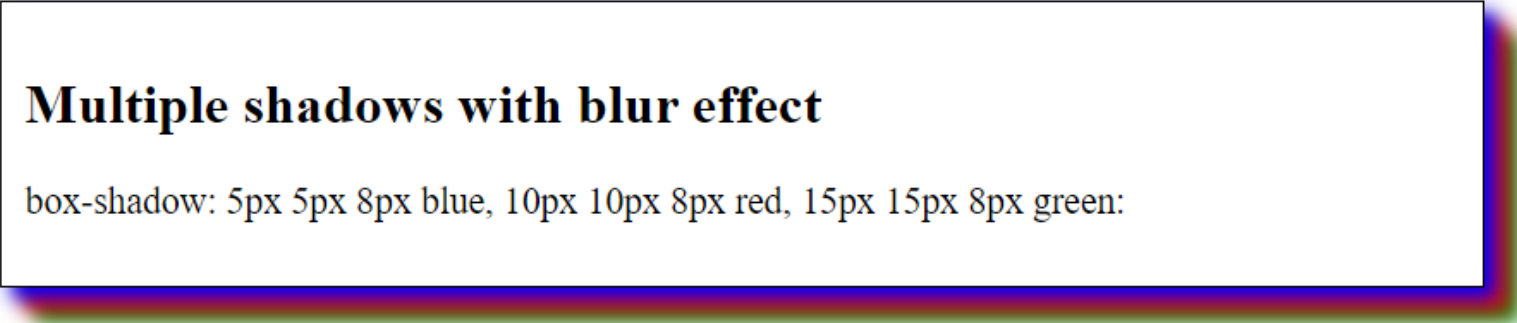
### **Multiple shadows**

`box-shadow: 5px 5px blue, 10px 10px red, 15px 15px green;`



### **Multiple shadows with blur effect**

`box-shadow: 5px 5px 8px blue, 10px 10px 8px red, 15px 15px 8px green;`



# Transforms & Transitions

---

- **CSS 2D Transforms**: di chuyển, xoay, chia tỷ lệ và làm nghiêng các phần tử theo 2 chiều
- **2D transformation methods**:

- `translate()`
- `rotate()`
- `scaleX()`
- `scaleY()`
- `scale()`

- `skewX()`
- `skewY()`
- `skew()`
- `matrix()`

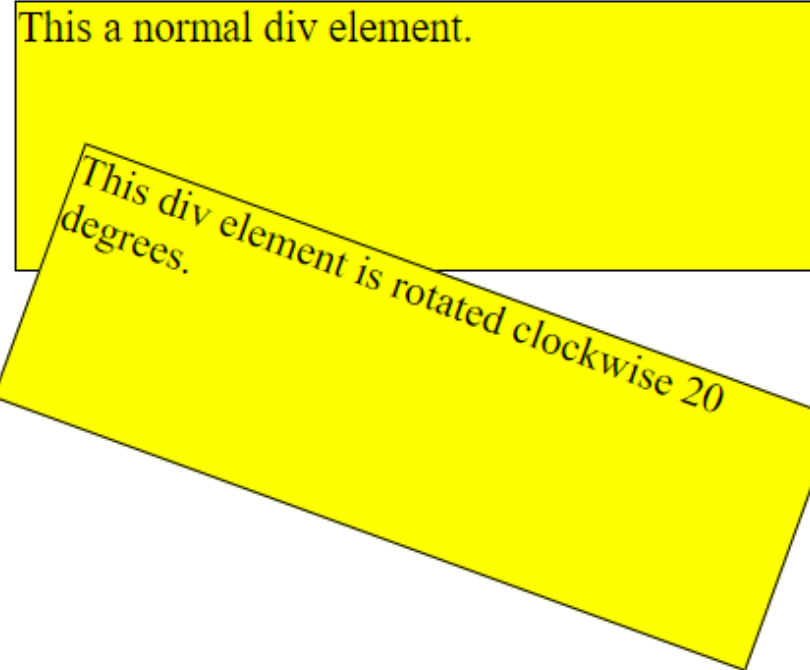
# Transforms & Transitions

---

## ■ CSS 2D Transforms:

Ví dụ:

```
div {  
  width: 300px;  
  height: 100px;  
  background-color: yellow;  
  border: 1px solid black;  
}  
div#myDiv {  
  transform: rotate(20deg);  
}
```



# Transforms & Transitions

---

- **CSS 3D Transforms**: di chuyển, xoay, và làm nghiêng các phần tử theo 3 chiều
- Các phương thức xoay 3 chiều
  - ❖ rotateX()
  - ❖ rotateY()
  - ❖ rotateZ()



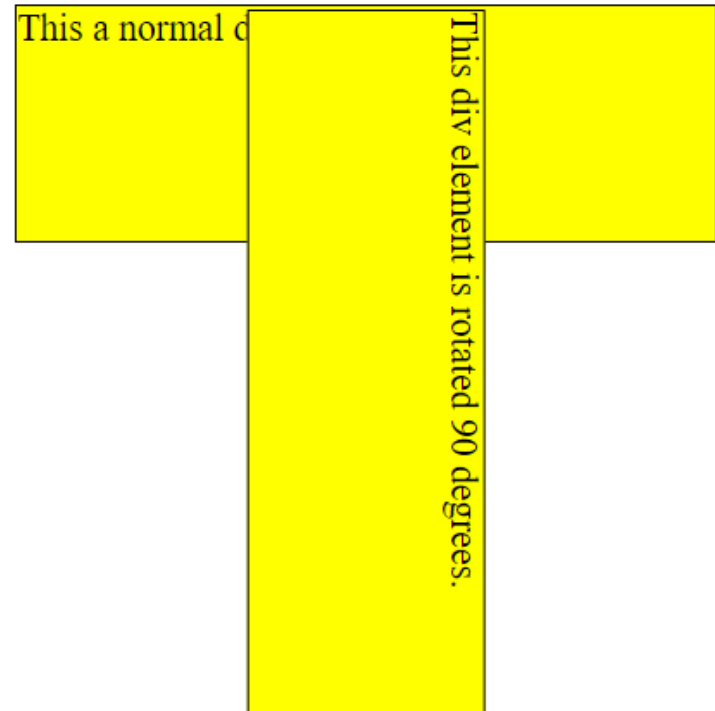
# Transforms & Transitions

---

## ■ CSS 3D Transforms

Ví dụ:

```
div {  
    width: 300px;  
    height: 100px;  
    background-color: yellow;  
    border: 1px solid black;  
}  
#myDiv {  
    transform: rotateZ(90deg);  
}
```



# Transforms & Transitions

---

- **CSS transitions**: sử dụng thuộc tính transition để thiết lập quá trình chuyển đổi của một đối tượng. Có thể kiểm soát các thành phần riêng lẻ của transition với các thuộc tính sau:
  - ❖ transition
  - ❖ transition-delay
  - ❖ transition-duration
  - ❖ transition-property
  - ❖ transition-timing-function

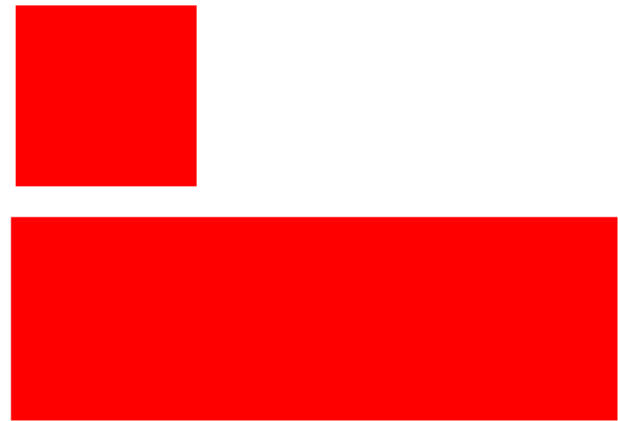
# Transforms & Transitions

---

## ■ CSS transitions:

Ví dụ:

```
<style>
  div {
    width: 100px;
    height: 100px;
    background: red;
    transition: width 2s;
  }
  div:hover {
    width: 300px;
  }
</style>
```



*Hiệu ứng: khi rê chuột vào đối tượng sẽ làm thay đổi độ rộng từ 100px thành 300px*

# Transforms & Transitions

---

## ■ CSS transitions

**Ví dụ 1:** thay đổi kích thước font chữ **4s** với độ trễ **2s** giữa thời gian người dùng rê chuột qua phần tử

```
#delay {  
    font-size: 14px  
    transition-property: font-size;  
    transition-duration: 4s  
    transition-delay: 2s;  
}  
#delay:hover { font-size: 36px; }
```

# Transforms & Transitions

---

## ■ CSS transitions

**Ví dụ 2:** Hộp sẽ thay đổi: chiều rộng, chiều cao, màu nền, khi rê chuột qua hộp.

```
.box {  
  border-style: solid;  
  border-width: 1px;  
  display: block;  
  width: 100px;  
  height: 100px;  
  background-color: #0000FF;  
  transition: width 2s, height 2s, background-color 2s, transform 2s;  
}
```

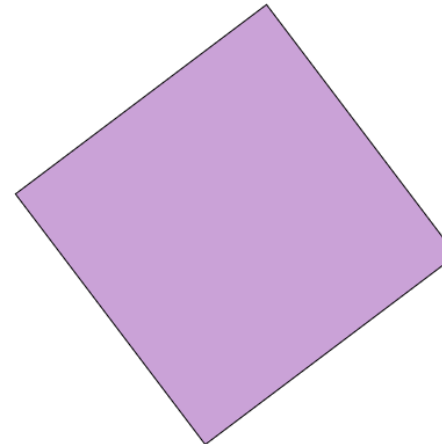
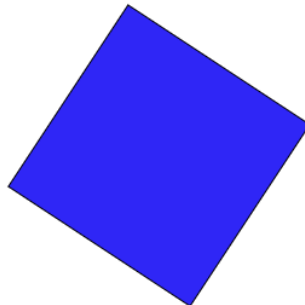
# Transforms & Transitions

---

## ■ CSS transitions

Ví dụ 2 (tt):

```
.box:hover {  
    background-color: #FFCCCC;  
    width: 200px;  
    height: 200px;  
    transform: rotate(180deg);  
}
```



# Transforms & Transitions

---

## ■ CSS transitions

**Ví dụ 3:** highlight các mục trong menu khi người dùng rê trỏ chuột qua chúng.

HTML

```
<nav>
  <a href="#">Home</a>
  <a href="#">About</a>
  <a href="#">Contact Us</a>
  <a href="#">Links</a>
</nav>
```

# Transforms & Transitions

---

## ■ CSS transitions

Ví dụ 3 (tt):

CSS

```
a {  
    flex: 1;  
    background-color: #333;  
    color: #fff;  
    border: 1px solid  
    padding: 0.5rem;  
    text-align: center;  
    text-decoration: none;  
    transition: all 0.5s ease-out; }
```

```
nav {  
    display: flex;  
    gap: 0.5rem; }  
a:hover, a:focus  
{  
    background-color: #fff;  
    color: #333;  
}
```



# Transforms & Transitions

---

- **CSS transitions**

Ví dụ 3 (tt):



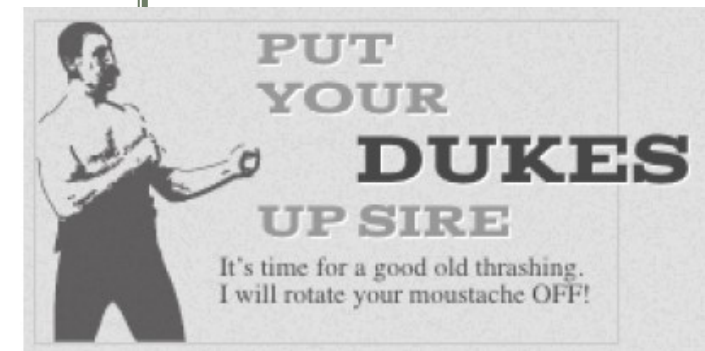
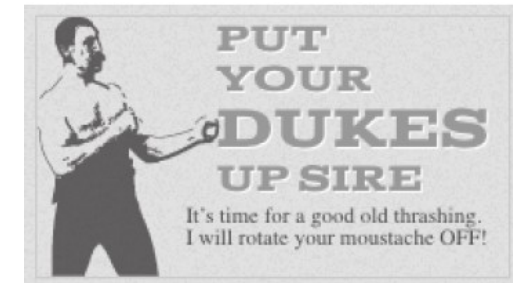
# Transforms & Transitions

---

## ■ CSS transitions

Ví dụ 4: chuyển từ "dukes" sang bên phải khi người dùng rê chuột qua nó,

```
#ourExample h1:hover span {  
    color: #484848;  
    -webkit-transform: translateX(40px);  
    -moz-transform: translateX(40px);  
    -ms-transform: translateX(40px);  
    -o-transform: translateX(40px);  
    transform: translateX(40px);  
}
```

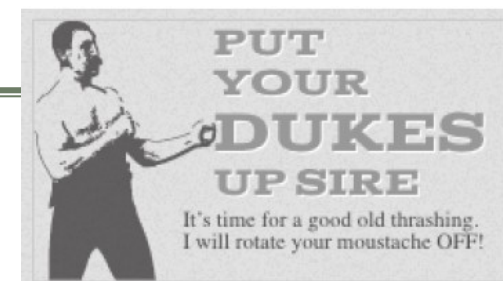


# Transforms & Transitions

- **Scaling**: Hàm **scale (x, y)** chia tỷ lệ một phần tử theo các yếu tố xác định theo chiều ngang và chiều dọc. Nếu chỉ có một giá trị được cung cấp, thì giá trị đó sử dụng cho cả x và y.

## Ví dụ 5

```
#ourExample h1: hover span {  
  color: #484848;  
  -webkit-transform: translateX(40px) scale(1.5);  
  -moz-transform: translateX(40px) scale(1.5);  
  -ms-transform: translateX(40px) scale(1.5);  
  -o-transform: translateX(40px) scale(1.5);  
  transform: translateX(40px) scale(1.5);  
}
```

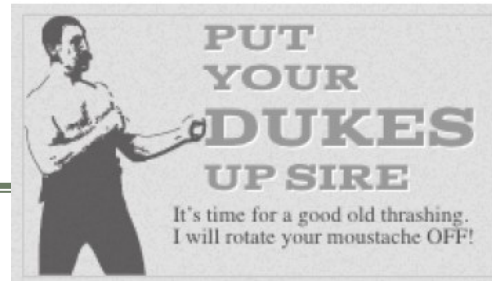


# Transforms & Transitions

---

- **Rotation:** Hàm rotate() xoay một phần tử xung quanh điểm gốc (đây là tâm của phần tử), theo một giá trị góc được chỉ định.

Ví dụ 6:



```
#ourExample h1:hover span {  
  color: #484848;  
  -webkit-transform: rotate(10deg) translateX(40px) scale(1.5);  
  -moz-transform: rotate(10deg) translateX(40px) scale(1.5);  
  -ms-transform: rotate(10deg) translateX(40px) scale(1.5);  
  -o-transform: rotate(10deg) translateX(40px) scale(1.5);  
  transform: rotate(10deg) translateX(40px) scale(1.5);  
}
```