

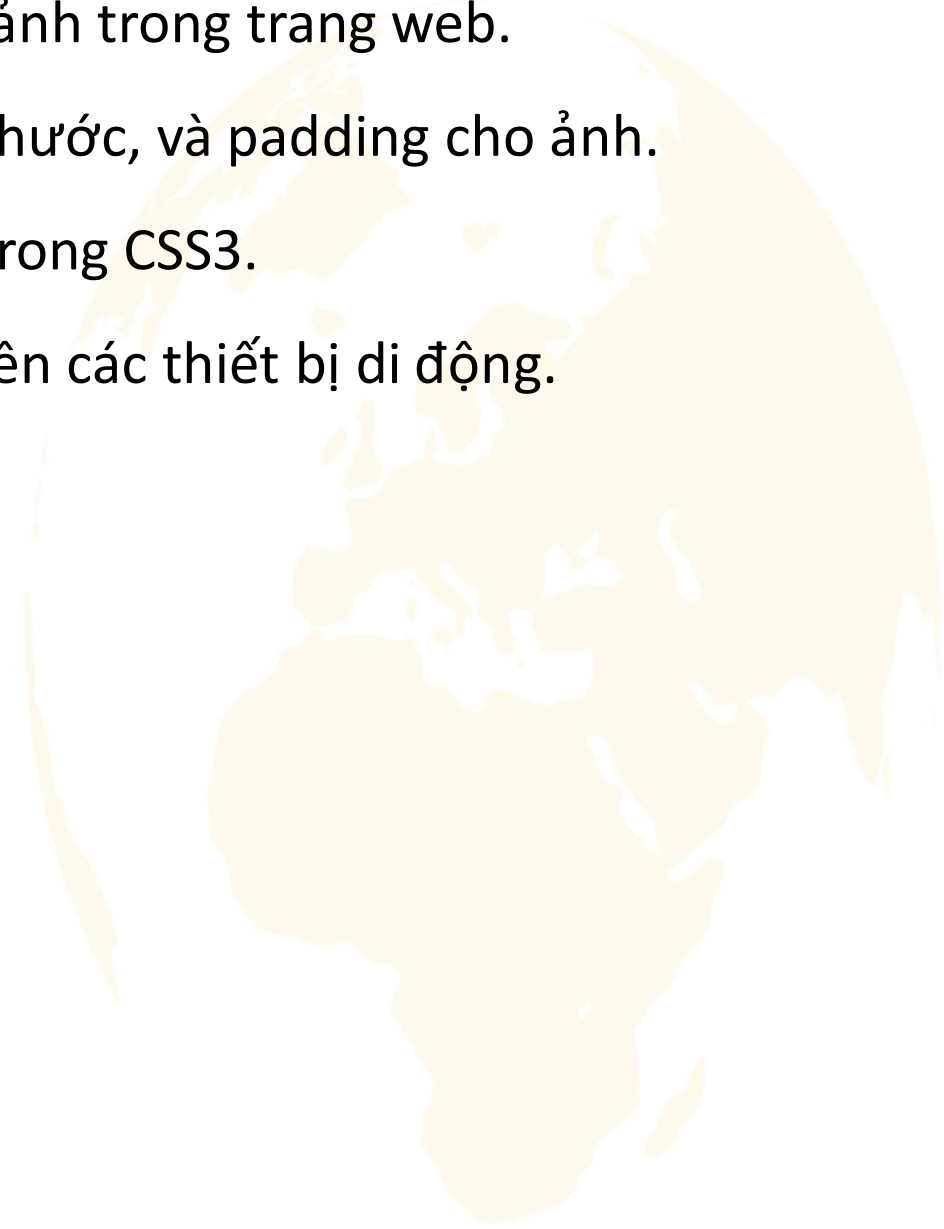
TRƯỜNG ĐÀO TẠO LẬP TRÌNH VIÊN VÀ QUẢN TRỊ MẠNG QUỐC TẾ BACHKHOA-APTECH

---

# BÀI 07: ĐỒ HỌA VÀ HÌNH ẢNH ĐỘNG TRONG CSS3

# MỤC TIÊU

- Định dạng hình ảnh trong trang web.
- Chèn ảnh, kích thước, và padding cho ảnh.
- Hình ảnh động trong CSS3.
- Sử dụng CSS3 trên các thiết bị di động.



# ĐỊNH DẠNG HÌNH ẢNH 1-4

- Có rất nhiều định dạng đồ họa có sẵn; thường được sử dụng nhất là chuyên (JPEG), (GIF), (PNG).
- Sự khác biệt giữa các định dạng đồ họa phụ thuộc vào các đặc điểm sau:

## ➤ Color Depth (độ sâu của màu)

- Nó được xác định bởi số lượng màu sắc riêng biệt tùy theo phần cứng hoặc phần mềm.
- Độ sâu màu được xác định bởi số bit cho mỗi pixel (bpp) và nó cũng được gọi là độ sâu bit.
- Độ sâu màu cao hơn cho thấy phạm vi cao hơn của màu sắc được sử dụng.

## ➤ Compression/file size (tỉ lệ nén file)

- Nén lưu trữ những hình ảnh gốc trong một số ít các byte sử dụng một thuật toán.
- Hình ảnh này có thể được mở rộng trở lại kích thước ban đầu bằng cách sử dụng một thuật toán giải nén.

# ĐỊNH DẠNG HÌNH ẢNH 2-4

- Hai loại hình ảnh sử dụng thuật toán nén file như sau:

## Lossless compression(nén ít mất dữ liệu nhất)

- Kích thước tập tin được giảm xuống, nhưng giữ một bản sao của hình ảnh không nén ban đầu.
- Tránh tích lũy giai đoạn nén lại khi chỉnh sửa hình ảnh.

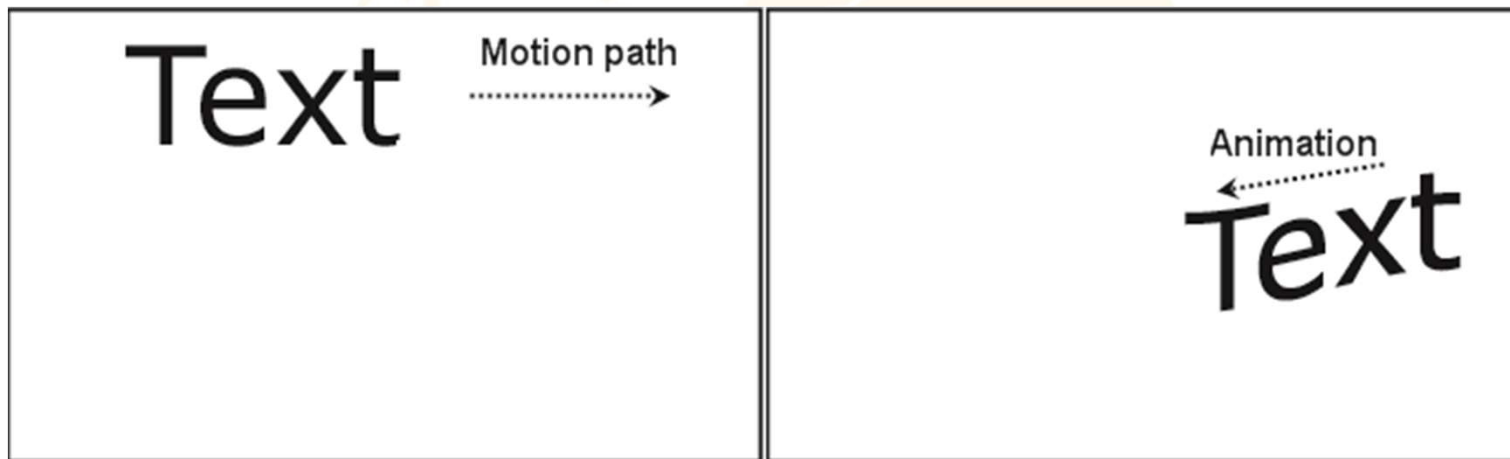
## Lossy compression (nén mất nhiều dữ liệu)

- Hình ảnh dường như là một bản sao của hình ảnh gốc nhưng trong thực tế nó không phải là một bản sao.
- Nén lossy đạt được kích thước file nhỏ hơn khi so sánh với nén lossless.
- Các thuật toán nén tổn hao cho phép biến đổi bao gồm chất lượng hình ảnh với kích thước tập tin.

# ĐỊNH DẠNG HÌNH ẢNH 3-4

## ➤ Animation (hoạt ảnh)

- Một số định dạng đồ họa bao gồm một loạt các bức ảnh được chạy sau khi ảnh khác đưa ra tạo hình ảnh động.
- Hình dưới đây cho thấy một đồ họa hoạt hình.

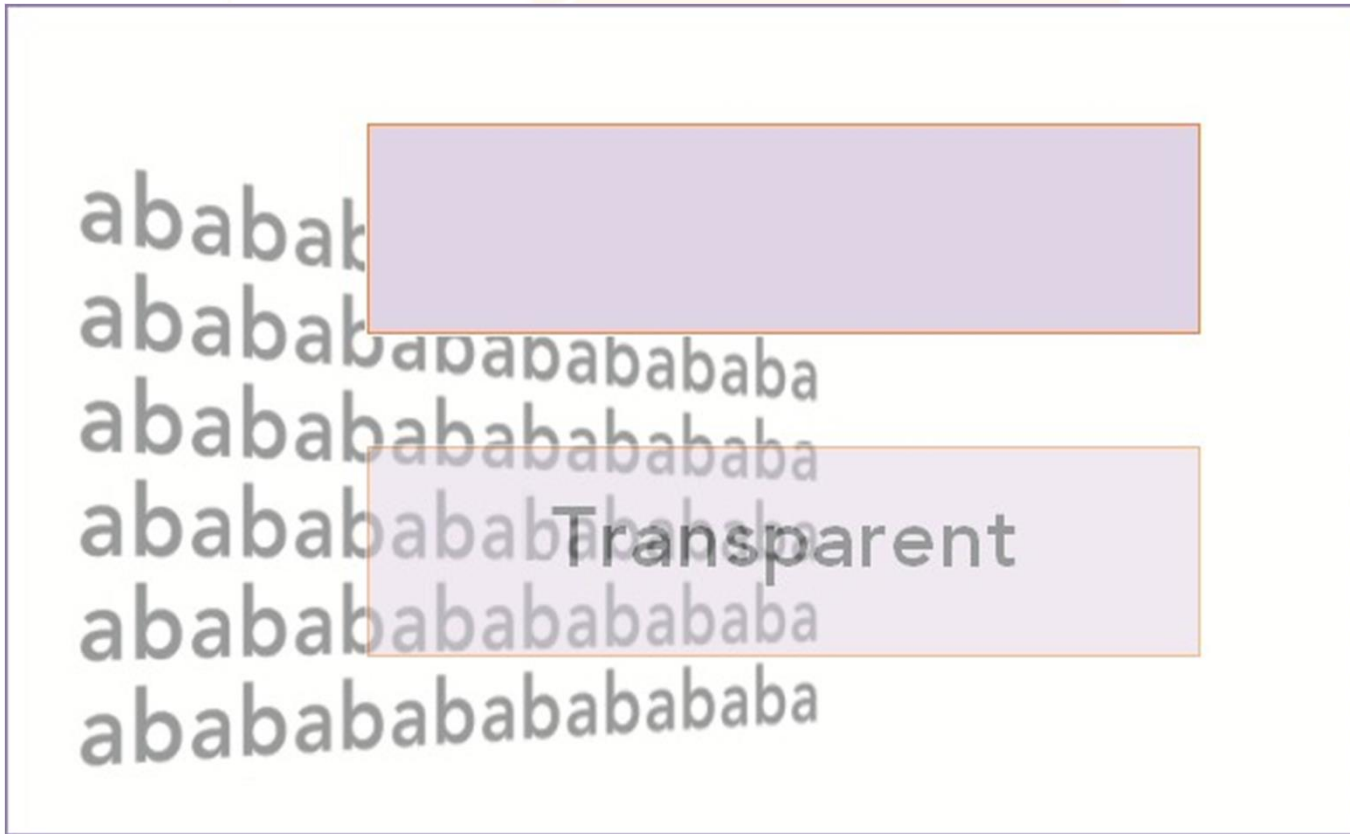


## ➤ Transparency (trong suốt)

- Hiện thị một hình ảnh trên một trang web xuất hiện trực tiếp với màu nền của trang là rất phổ biến trên trang web.
- Màu nền của trang web hiển thị thông qua các phần trong suốt của hình ảnh.

## ĐỊNH DẠNG HÌNH ẢNH 4-4

- Trong một hình ảnh trong suốt, một và chỉ một màu sắc có thể được ẩn.
- Nếu màu sắc được lựa chọn để làm trong suốt như là nền tảng của hình ảnh đưa vào, sau đó một hình ảnh hình dạng không đều xuất hiện phảng phất trên trang.
- Hình dưới đây cho thấy một hình ảnh trong suốt.



# ĐỊNH DẠNG HÌNH ẢNH CHO WEB 1-2

- Đối với các trang web, sử dụng JPEG và PNG đồ họa được khuyến khích vì nó cung cấp khả năng tương thích tối đa với tất cả các thiết bị truy cập vào trang web.
- Đối với hình ảnh, sử dụng các định dạng đồ họa JPEG và cho màn hình-ảnh và bản vẽ sử dụng các định dạng đồ họa PNG được khuyến khích.

## ➤ JPEG

- Sử dụng nén lossy có nghĩa là chất lượng hình ảnh bị mất trong quá trình nén hình ảnh.
- Hầu hết các biên tập viên JPEG cho phép người sử dụng để xác định số lượng chi tiết mà người dùng sẵn sàng để mất.
- Nếu chất lượng được giảm xuống, sau đó sự mất mát có thể nhìn thấy; JPEG là khoảng một nửa kích thước của PNG.

## ➤ PNG

- Sử dụng nén lossless, có nghĩa là không có mất bất kỳ chi tiết hình ảnh.
- Được thiết kế để chuyển hình ảnh trên Internet và không dành cho in ảnh chất lượng chuyên nghiệp.
- Vì vậy, nó không hỗ trợ không gian màu RGB như CMYK.
- Hỗ trợ màu sắc cao và một phần minh bạch sử dụng các kênh alpha.

# ĐỊNH DẠNG HÌNH ẢNH CHO WEB 2-2

## ➤ GIF

- Sử dụng nén lossless có nghĩa là không có giảm chất lượng khi hình ảnh được nén.
- Hình ảnh không nén lưu trữ thông tin của nó trong một dạng tuyến tính.
- Mỗi dòng của các điểm ảnh được đọc từ trái sang phải.
- Một tập tin GIF lưu trữ xen kẽ các dòng của hình ảnh theo một thứ tự khác nhau.
- Đồ họa hoạt hình được lưu trữ ở định dạng GIF.

Khả năng tương thích và sự xuất hiện là những từ khóa trên trang web.

Các hình ảnh được chèn vào phải được nhìn thấy và không bị biến dạng khi xuất hiện trên bất kỳ thiết bị của người nhận.

Các nhà thiết kế web có thể tạo các giả định rằng trang web sẽ mở ra một máy tính mà sẽ có độ phân giải tối thiểu là 800x600 pixel.

Nếu một trang web dựa điện thoại di động cần phải được tạo ra sau đó các thông số kỹ thuật sẽ thay đổi.



# CHÈN HÌNH ẢNH 1-6

Các phần tử IMG là một phần tử rỗng, cho phép người dùng để chèn một hình ảnh.

Nó cho phép chèn các hình ảnh và biểu đồ.

Các định dạng đồ họa thường được sử dụng được hỗ trợ được cụ thể là, GIF, JPEG, bitmap (BMP), và PNG.

Thẻ <img> giành ra không gian cho hình ảnh và không chèn hình ảnh trong trang HTML.

Nó tạo ra một mối liên hệ giữa hình ảnh và các trang HTML.

## CHÈN HÌNH ẢNH 2-6

- Bảng sau liệt kê các thuộc tính thường được sử dụng trong những phần tử IMG.

Thuộc tính	Mô tả
src	Xác định đường dẫn hình ảnh
height	Chỉ ra chiều cao hình ảnh
width	Chỉ ra chiều rộng hình ảnh

- ví dụ

```
<body>  
  
</body>
```

# CHÈN HÌNH ẢNH 3-6

- Hình ảnh có thể lưu trữ trong thư mục con của thư mục chứa file HTML.
- Trong trường hợp này ta dùng như sau.

```
<body>  
  
</body>
```

- Căn lề

```
<body>  
  
</body>
```

# CHÈN HÌNH ẢNH 4-6

- Một số giá trị của float.

Giá trị	Mô tả
left	Phần tử căn trái.
right	Phần tử căn phải
none	Giá trị mặc định.
inherit	Tuân theo phần tử cha nó

HTML5 giới thiệu một thẻ `<figure>` mới hoạt động như một bộ chứa thẻ `<img>`.

Nó không phải là một thay thế cho `<img>` thẻ.

Thẻ `<figure>` quy định cụ thể nội dung khép kín, chẳng hạn như hình ảnh minh họa, sơ đồ, hình ảnh, danh sách mã.

Nội dung của phần tử `<figure>` là liên quan đến dòng chảy chính, vị trí của nó là độc lập với dòng chảy chính.

# CHÈN HÌNH ẢNH 5-6

- Ví dụ về thẻ <figure>.

```
<figure>
  
</figure>
```

- Ưu điểm chính của việc sử dụng <figure> thẻ là nó cho phép người dùng sử dụng thẻ <figcaption> cùng với nó.
- Thẻ <figcaption> cho phép người sử dụng thêm một chú thích cho hình ảnh.
- Chú thích luôn luôn xuất hiện cùng với các hình ảnh ngay cả khi hình ảnh nổi trong cách bố trí trang web.
- Ví dụ về việc sử dụng <figcaption> tag.

```
<figure>

<figcaption>This diagram shows the logo of a
  product.</figcaption>
</figure>
```

# CHÈN HÌNH ẢNH 6-6

- Thẻ <figure> cũng có thể chỉ định style và các thuộc tính khác cho các phần tử <figure>.
- Một tiêu đề duy nhất để nhóm các hình ảnh bằng cách sử dụng thẻ <figure>.
- Ví dụ.

```
<figure>  
  
  
  
<figcaption>The different types of flowers</figcaption>  
</figure>
```

- Kết quả



# KÍCH THƯỚC VÀ PADDING ẢNH TRONG CSS 1-2

- Kích thước của một hình ảnh được quy định tại điểm ảnh.
- Thuộc tính width và height thiết lập chiều cao và chiều rộng của hình ảnh.
- Người ta có thể xác định chiều rộng và chiều cao sẽ được thay đổi kích cỡ hoặc ngược lại.

- Ví dụ

```
p.ex
{
height:100px
;
width:100px;
}
```

- Bảng sau mô tả các thuộc tính và giá trị tương ứng

Thuộc tính	Mô tả	Các giá trị
height	Thiết lập chiều cao	<ul style="list-style-type: none"><li>• Auto</li><li>• Length</li><li>• %</li><li>• Inherit</li></ul>

# KÍCH THƯỚC VÀ PADDING ẢNH TRONG CSS 2-2

Property	Description	Values
max-height	Thiết lập chiều cao tối đa	<ul style="list-style-type: none"><li>• none</li><li>• length</li><li>• %</li><li>• inherit</li></ul>
max-width	Thiết lập chiều cao tối đa	<ul style="list-style-type: none"><li>• none</li><li>• length</li><li>• %</li><li>• Inherit</li></ul>
min-height	Thiết lập chiều cao tối thiểu	<ul style="list-style-type: none"><li>• length</li><li>• %</li><li>• Inherit</li></ul>
min-width	Thiết lập chiều rộng tối thiểu	<ul style="list-style-type: none"><li>• length</li><li>• %</li><li>• Inherit</li></ul>
width	Thiết lập chiều rộng	<ul style="list-style-type: none"><li>• auto</li><li>• length</li><li>• %</li><li>• inherit</li></ul>



# PADDING 1-3

- Thuộc tính Padding CSS được sử dụng để xác định không gian giữa viền các phần tử và nội dung của nó.
- Màu nền của các phần tử ảnh hưởng đến thuộc tính padding.
- Sử dụng thuộc tính riêng biệt như top, right, bottom, và left, giá trị padding khác nhau có thể được xác định và nó có thể được thay đổi một cách riêng biệt.
- Bảng sau liệt kê các giá trị khác nhau được sử dụng trong thuộc tính padding.

Giá trị	Mô tả
length	Chỉ ra giá trị cố định cho padding ở dạng pixels, pt, em.
%	Chỉ ra giá trị tỉ lệ phần trăm so với đối tượng chứa nó.

# PADDING 2-3

- Ví dụ.

```
padding-top:10px;  
padding-  
bottom:10px;  
padding-right:15px;  
padding-left:15px;
```

- Bạn cũng có thể chỉ rút gọn như sau:

```
padding:25px 50px 75px 100px;
```

Trong đó, top padding là 25px, right padding là 50px, bottom padding là 75px, và left padding là 100px.

# PADDING 3-3

- Bảng sau liệt kê tất cả các thuộc tính.

Thuộc tính	Mô tả
padding	Định nghĩa padding
padding-bottom	Định nghĩa padding dưới
padding-left	Định nghĩa padding trái
padding-right	Định nghĩa padding phải
padding-top	Định nghĩa padding trên

# HÌNH ẢNH THU NHỎ (THUMBNAIL) 1-5

- Một thumbnail là một hình ảnh nhỏ, hoặc một phần của một hình ảnh lớn hơn.
- Nhấn vào hình ảnh thu nhỏ sẽ liên kết đến hình ảnh ban đầu lớn hơn, có thể xem và tải về.
- Ngay cả một hiệu ứng di chuột có thể được đưa ra thông qua CSS và JavaScript.
- Ví dụ.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Thumbnail</title>
  <style>
    /* general */
    body {
      margin:0;
      padding:40px 80px;
      background:#fff;
      font:70% Arial, Helvetica, sans-serif;
      color:#555;
      line-height:100%;
    }
```

## Hình ảnh thu nhỏ (Thumbnail) 2-5

```
h1, h2{
  font-size:180%;
  font-weight:normal;
  color:#555;
}
p{
  margin:1em 0;
}
p.text{
  width:500px;
}
a{
  color:#f20;
  text-decoration:none;
}
a:hover{
  color:#999;
}
img{
  border:none;
}

/* // general */
/* thumbnail list */
ul#thumbs, ul#thumbs li{
  margin:0;
  padding:0;
  list-style:none;
}
ul#thumbs li{
  float:left;
  margin-right:0px;
  border:1px solid #999;
  padding:2px;
}
ul#thumbs a{
  display:block;
  float:left;
  width:125px;
  height:135px;
  line-height:50px;
  overflow:hidden;
  position:relative;
  z-index:1;
}
```

## Hình ảnh thu nhỏ (Thumbnail) 3-5

```
ul#thumbs a img{
  float:left;
  position:absolute;
  top:0px;
  left:0px;
}
/* mouse over */
ul#thumbs a:hover{
  overflow:visible;
  z-index:1000;
  border:none;
}
ul#thumbs a:hover img{
  border:1px solid #999;
  background:#fff;
  padding:2px;
}
/* // mouse over */
/* clearing floats */
```

```
ul#thumbs:after,
li#thumbs:after{
  content:".";
  display:block;
  height:0;
  clear:both;
  visibility:hidden;
}
ul#thumbs, li#thumbs{
  display:block;
}
ul#thumbs, li#thumbs{
  min-height:1%;
}
* html ul#thumbs, * html
li#thumbs{
  height:1%;
}
/* // clearing floats */
/* // thumbnail list */
```

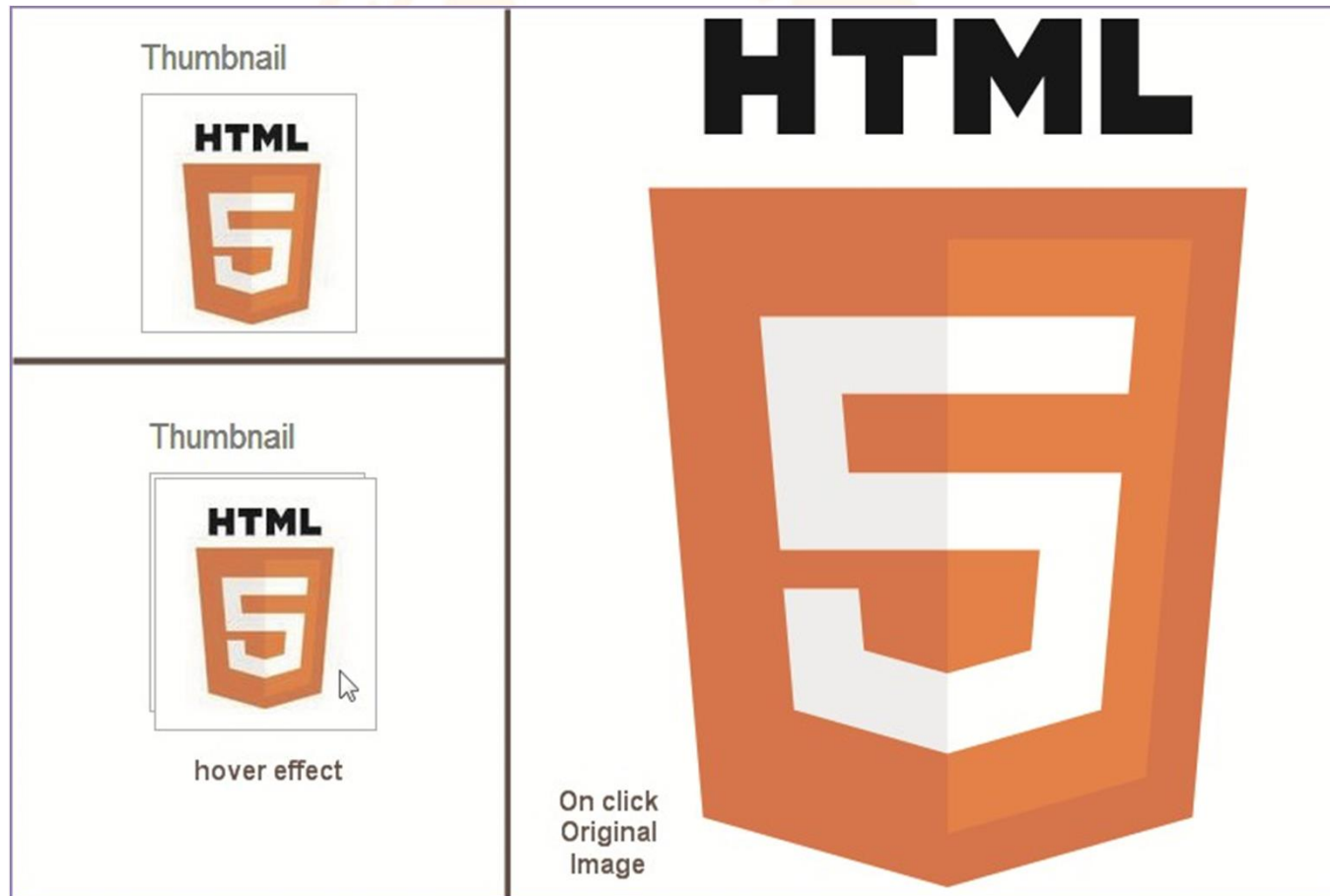
<style>

# Hình ảnh thu nhỏ (Thumbnail) 4-5

```
</head>
<body>
  <h2>Thumbnail</h2>
  <ul id="thumbs">
    <li>
      <a href="HTML5.png" target="_blank">
        
      </a>
    </li>
  </ul>
</body>
</html>
```

# HÌNH ẢNH THU NHỎ (THUMBNAIL) 5-5

- Kết quả khi hover

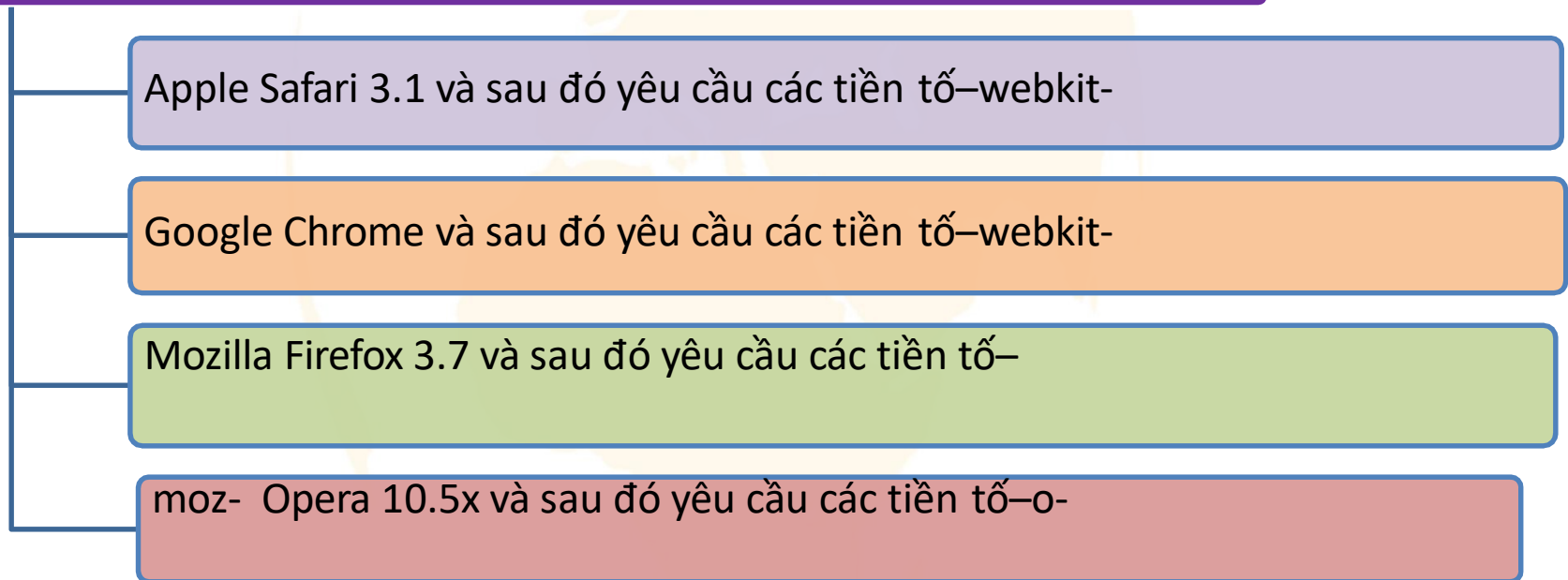




# LÀM VIỆC VỚI TRANSITIONS CSS3 1-6

- Trong năm 2007, Apple giới thiệu quá trình chuyển đổi CSS, mà sau này đã trở thành một tính năng độc quyền của Safari được gọi là CSS Animation.
- Đại diện của Apple và Mozilla đã bắt đầu bổ sung thêm các module chuyển đổi CSS vào các đặc điểm kỹ thuật CSS3, Apple đã thêm vào WebKit và moz.

Trình duyệt hỗ trợ CSS3 Transitions như sau:



- Hiện tại, Internet Explorer 9 không hỗ trợ CSS3 Transitions.

# LÀM VIỆC VỚI TRANSITIONS CSS3 2-6

Để thực hiện quá trình Transitions CSS cần hai yêu cầu kỹ thuật như sau:

Các thuộc tính CSS cần cho hiệu ứng

Thời gian hiệu ứng

- Ví dụ sau tạo hiệu ứng về độ rộng trong 3s.

```
div
{
  transition: width 3s;
  -moz-transition: width 3s; /* Firefox 4 */
  -webkit-transition: width 3s; /* Safari and Chrome */
  -o-transition: width 3s; /* Opera */
}
```

# LÀM VIỆC VỚI TRANSITIONS CSS3 3-6

- Thiết lập hover cho phần tử <div>.

```
div:hover
{
    width:200px;
}
```

- Bảng sau liệt kê các thuộc tính của transition.

| Thuộc tính                 | Mô tả  |
|----------------------------|--|
| transition                 | Thuộc tính viết tắt thể hiện 4 thông số của transition.  |
| transition-property        | Được sử dụng để xác định tên của CSS thuộc tính mà giá trị chuyển tiếp được thiết lập.                             |
| transition-duration        | Được sử dụng để xác định thời gian của quá trình chuyển đổi. Giá trị mặc định là 0.                                |
| transition-timing-function | Được sử dụng để mô tả tốc độ trong quá trình chuyển đổi sẽ được tính toán như thế nào. Giá trị mặc định là 'ease'. |
| transition-delay           | Được sử dụng để xác định sự bắt đầu của quá trình chuyển đổi. Giá trị mặc định là 0.                               |

# LÀM VIỆC VỚI TRANSITIONS CSS3 4-6

- Ví dụ.

```
<!DOCTYPE html>
<html>
  <head>
    <style type="text/css">
      div {
        width:100px;
        height:100px;
        background:#000000;
        transition-property:width;
        transition-duration:2s;
        transition-timing-function:linear;
        transition-delay:1s; /* Firefox 4 */
        -moz-transition-property:width;
        -moz-transition-duration:2s;
        -moz-transition-timing-function:linear;
        -moz-transition-delay:1s; /* Safari and Chrome */
        -webkit-transition-property:width;
        -webkit-transition-duration:2s;
        -webkit-transition-timing-function:linear;
        -webkit-transition-delay:1s;
      }
    </style>
  </head>
</html>
```

# LÀM VIỆC VỚI TRANSITIONS CSS3 5-6

```
div:hover {  
    width:500px;  
}  
</style>  
</head>  
<body>  
    <p><b>Note:</b> The example</p>  
    <div></div>  
    <p>Hover over the div element above, to see the  
transition effect.</p>  
    <p><b>Note:</b> The transition effect will wait 1  
seconds before starting.</p>  
    </body>  
</html>
```

# LÀM VIỆC VỚI TRANSITIONS CSS3 6-6

- Kết quả

Note: The example



Hover over the div element above, to see the transition effect.

Note: The transition effect will wait 1 seconds before starting.

Before transition effect started

Note: The example



Transition on mouse-over



Hover over the div element above, to see the transition effect.

Note: The transition effect will wait 1 seconds before starting.

After transition effect started

# CSS3 ANIMATION (HOẠT ẢNH)

- Animations CSS3 có thể tạo hoạt ảnh transition một kiểu CSS cấu hình cho cái khác.

Hai thành phần của animations như sau:

Một kiểu animation mô tả hoạt ảnh.

Một tập hợp các khung hình mà xác định các trạng thái điểm bắt đầu và kết thúc của style CSS hoạt ảnh và các điểm trung gian có thể trên đường đi.

Những lợi thế của Animations CSS3 trên kỹ thuật hoạt hình dựa trên kịch bản như sau:

Dễ sử dụng và ai cũng có thể tạo ra chúng mà không cần kiến thức về JavaScript.

Thực hiện tốt ngay cả khi chịu tải hệ thống.

Cho phép trình duyệt để kiểm soát trình tự hình ảnh động, tối ưu hóa hiệu suất và hiệu quả.

# CẤU HÌNH ANIMATION 1-7

- Một chuỗi hình ảnh động CSS có thể được tạo ra bởi kiểu dáng các phần tử với thuộc tính `animation`.
- Thuộc tính này có thể được sử dụng để cấu hình thời gian, và trình tự của các hình ảnh động.
- `@keyframes` nguyên tắc xác định sự xuất hiện của các hình ảnh động.
- Keyframe được sử dụng để mô tả việc sinh ra các phần tử trong chuỗi hoạt hình.
- Bảng sau liệt kê các quy tắc của `@keyframes` và các thuộc tính `animation`.

Thuộc tính	Mô tả
<code>@keyframes</code>	Được sử dụng để xác định các hình ảnh động.
<code>animation</code>	Là thuộc tính ngắn gọn đại diện cho tất cả các thuộc tính hình ảnh động, ngoại trừ thuộc tính <code>animation-play-state</code> .
<code>animation-name</code>	Được sử dụng để xác định tên của <code>@keyframes</code> animation.
<code>animation-duration</code>	Được sử dụng để xác định thời gian của một chu kỳ hoạt hình trong vài giây hoặc mili giây. Giá trị mặc định là 0.
<code>animation-timing-function</code>	Được sử dụng để mô tả tiến độ của hình ảnh động trên một chu kỳ thời gian của nó. Giá trị mặc định là 'ease'.



# CẤU HÌNH ANIMATION 2-7

Thuộc tính	Mô tả
<code>animation-delay</code>	Được sử dụng để xác định giá trị bắt đầu của hình ảnh động. Giá trị mặc định là 0.
<code>animation-iteration-count</code>	Được sử dụng để xác định số lần một hình ảnh động được phát. Giá trị mặc định là 1.
<code>animation-direction</code>	Được sử dụng để xác định có hay không các hình ảnh động nên phát ngược lại trên chu kỳ thay thế. Giá trị mặc định là 'normal'.
<code>animation-play-state</code>	Được sử dụng để xác định tình trạng của các hình ảnh động, xem liệu nó có đang chạy hay dừng lại. Giá trị mặc định là 'running'.

## Cú pháp:

```
@keyframes myfirst
{
  from {background: red;}
  to {background: yellow;}
```

# CẤU HÌNH ANIMATION 3-7

```
}  
@-moz-keyframes myfirst /* Firefox */  
{  
  from {background: red;}  
  to {background: yellow;}  
}  
@-webkit-keyframes myfirst /* Safari and Chrome */  
{  
  from {background: red;}  
  to {background: yellow;}  
}
```

- Mã HTML

```
<!DOCTYPE html>  
<html>  
<head>  
<style type="text/css">
```

# CẤU HÌNH ANIMATION 4-7

```
div {  
  width:200px; height:200px; background:red; position:relative;  
  border-radius:100px; animation-name:myfirst;  
  animation-duration:4s; animation-timing-function:linear;  
  animation-delay:1s; animation-iteration-count:infinite;  
  animation-direction:alternate; animation-play-state:running;  
  /* Firefox: */  
  -moz-border-radius:100px;  
  -moz-animation-name:myfirst;  
  -moz-animation-duration:4s;  
  -moz-animation-timing-function:linear;  
  -moz-animation-delay:1s;  
  -moz-animation-iteration-count:infinite;  
  -moz-animation-direction:alternate;  
  -moz-animation-play-state:running;  
  /* Safari and Chrome: */  
  -webkit-border-radius:100px;  
  -webkit-animation-name:myfirst;  
  -webkit-animation-duration:4s;  
  -webkit-animation-timing-function:linear;  
  -webkit-animation-delay:1s;  
  -webkit-animation-iteration-count:infinite;
```

# CẤU HÌNH ANIMATION 5-7

```
-webkit-animation-direction:alternate;  
-webkit-animation-play-state:running;
```

```
}  
@keyframes myfirst {  
  0%    {background:red; left:0px; top:0px;}  
  25%   {background:yellow; left:300px; top:0px;}  
  50%   {background:blue; left:300px; top:300px;}  
  75%   {background:green; left:0px; top:300px;}  
  100%  {background:red; left:0px; top:0px;}  
} @-moz-keyframes myfirst { /* Firefox */  
  0%    {background:red; left:0px; top:0px;}  
  25%   {background:yellow; left:300px; top:0px;}  
  50%   {background:blue; left:300px; top:300px;}  
  75%   {background:green; left:0px; top:300px;}  
  100%  {background:red; left:0px; top:0px;}  
} @-webkit-keyframes myfirst { /* Safari and Chrome */  
  0%    {background:red; left:0px; top:0px;}  
  25%   {background:yellow; left:200px; top:0px;}  
  50%   {background:blue; left:200px; top:200px;}  
  75%   {background:green; left:0px; top:200px;}  
  100%  {background:red; left:0px; top:0px;}  
}
```

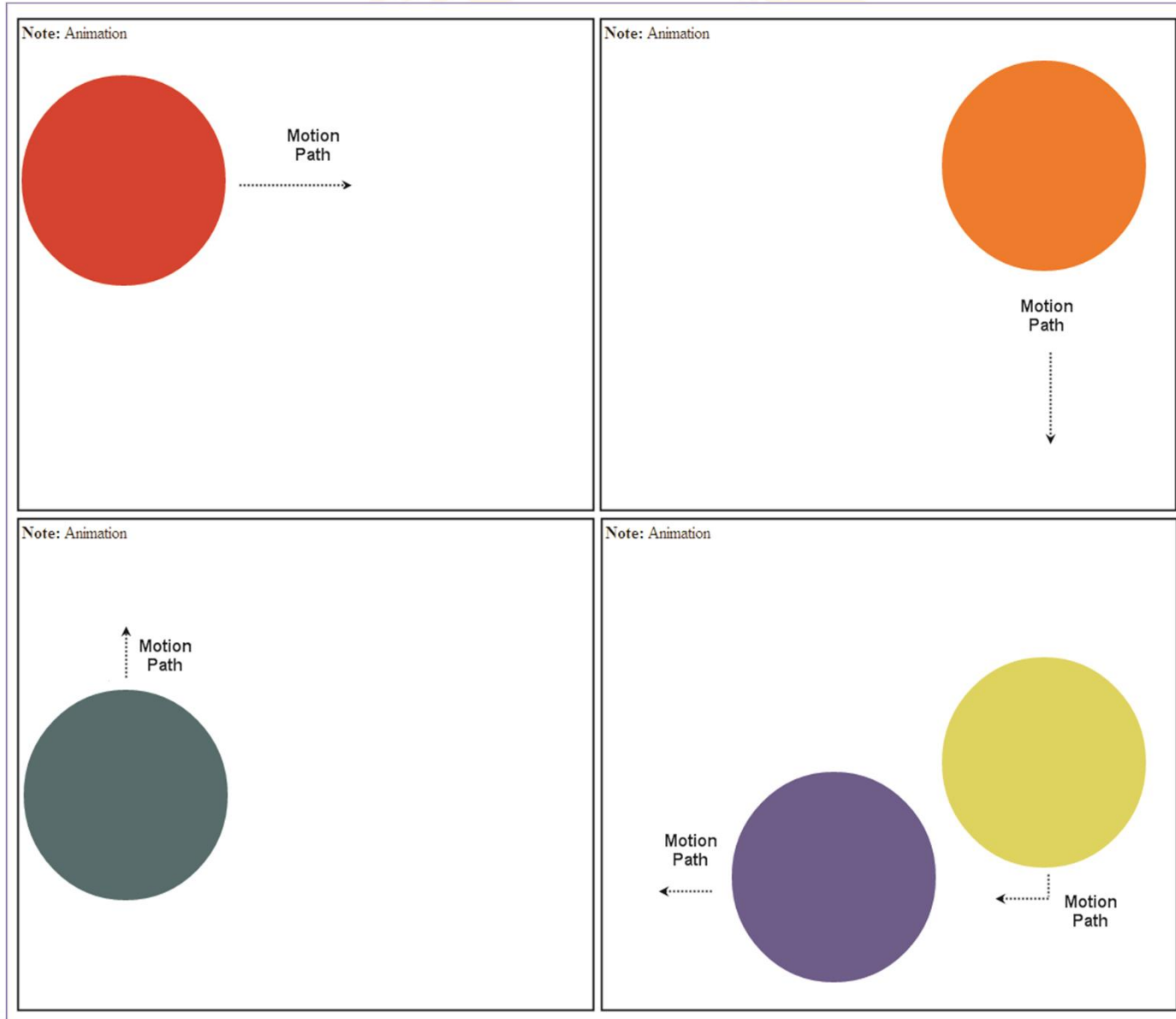
# CẤU HÌNH ANIMATION 6-7

```
</style>
</head>
<body>
    <p><b>Note:</b> Animation</p>
    <div></div>
</body>
</html>
```



# CẤU HÌNH ANIMATION 7-7

- Kết quả.



# SỬ DỤNG CSS3 TRÊN THIẾT BỊ MOBILE 1-2

There are different ways to provide Web pages for mobile devices.

The user can make use of style sheet for the handheld devices (all mobile browsers do not recognize it).

iPhone's Safari and Opera's Mini browsers support a new feature of CSS3 called media queries.

These queries allow the user to specify a conditional expression for media type.

- Following Code Snippet shows the use of a conditional expression for displaying a link element where the maximum screen width for mobile devices is 480 pixels.

```
<link rel="stylesheet" href="styles/mobile.css" media="only  
screen and (max-device-width: 480px)"/>
```

- The user can also specify another link element for screen media with a minimum screen width of 481 pixels.
- In other words, the style sheet for this element can be used for standard computer screens.

# SỬ DỤNG CSS3 TRÊN THIẾT BỊ MOBILE 2-2

- Hầu hết các trang web điện thoại di động được tạo ra để đứng trước tên miền của các trang web chính với m ví dụ **m.aptech-education.com**.
- Để phát hiện một thiết bị di động, một trang web có thể sử dụng JavaScript trên máy khách, một ngôn ngữ kịch bản trên máy chủ, hoặc file Wireless Universal Resource (WURFL) trên server.

Năm cách để cung cấp các trang web cho các thiết bị di động như sau:

Xác định một style sheet cho các thiết bị di động

Bao gồm liên kết đến một phiên bản di động của trang web

Sử dụng JavaScript để phát hiện các thiết bị di động và chuyển hướng

Sử dụng một ngôn ngữ kịch bản phía máy chủ để phát hiện và chuyển hướng

Sử dụng WURFL để phát hiện các thiết bị di động



# TỐI ƯU KHẢ NĂNG TƯƠNG THÍCH VỚI BROWSER 1-3

Các biện pháp tương thích trình duyệt web được thực hiện để cung cấp khả năng dự đoán và nhất quán trên các trình duyệt web nhằm mục tiêu phù hợp với người sử dụng cuối.

Khả năng tương thích qua các trình duyệt có nghĩa là một Website có thể hiển thị tốt trên các trình duyệt như cách bố trí, màu sắc, chức năng, tương tác, và tỷ lệ.

Khả năng tương thích qua trình duyệt là trên tất cả các trình duyệt web hiện tại, bất kể trình duyệt không phổ biến hoặc phổ biến và các phiên bản khác nhau.

Khả năng tương thích đa trình duyệt là không đổi và nó được sinh ra chức năng được sử dụng trên các trình duyệt phổ biến nhất trong thị trường mục tiêu của khách hàng.

HTML5 sử dụng các tiêu chuẩn khác nhau và được hỗ trợ bởi các trình duyệt khác nhau. Những trình duyệt này cung cấp phiên bản hỗ trợ khác nhau.

Rendering engines là một bộ công cụ được sử dụng trong hầu hết các trình duyệt có hỗ trợ tính năng HTML khác nhau.

# TỐI ƯU KHẢ NĂNG TƯƠNG THÍCH VỚI BROWSER 2-3

- Một số trình rendering cung cấp trình duyệt khác nhau như sau:

## Gecko

- Engine Gecko là engine chính của Mozilla Firefox, và một số trình duyệt có liên quan.
- Nó có hỗ trợ cho các tính năng HTML5 khác nhau.

## Trident

- Engine Trident được sử dụng bởi các phiên bản khác nhau của Internet Explorer (IE).
- Hiện nay, HTML5 không chú trọng hỗ trợ cho Trident engine.

## WebKit

- Các công cụ WebKit được hỗ trợ chủ yếu cho trình duyệt Safari của Apple được sử dụng trong máy Mac, iPhone, iPad, và các sản phẩm khác của Apple.
- Động cơ này được dựa trên dự án mã nguồn mở KHTML.

## Presto

- Presto là công cụ được sử dụng trong trình duyệt Opera.
- Trình duyệt Opera được coi là một trình duyệt kỹ thuật cao, nhưng thị phần của trình duyệt Opera vẫn còn thấp.

# TỐI ƯU KHẢ NĂNG TƯƠNG THÍCH VỚI BROWSER 3-3

- Thực hiện những điều tốt nhất cho tối ưu khả năng tương thích với Browser như sau:

## Kiểm tra các trang web trên các trình duyệt khác nhau

- Xem xét sự xuất hiện của trang web và chức năng trên nhiều trình duyệt để đảm bảo rằng tất cả người dùng đang nhận được cùng trải nghiệm theo thiết kế.
- Tốt nhất kiểm tra trên các phiên bản khác nhau của cùng một trình duyệt cũng như chúng có thể hiển thị các trang web khác nhau.

## Viết HTML rõ ràng

- Để đảm bảo rằng trang trông giống nhau trong tất cả các trình duyệt là để viết các trang web sử dụng HTML hợp lệ và mã CSS, và sau đó thử nghiệm nó trong nhiều trình duyệt.
- Sử dụng CSS bên ngoài có thể giúp các trang hiển thị và tải nhanh hơn.

# TỔNG KẾT

- Có rất nhiều định dạng đồ họa có sẵn; thường được sử dụng nhất là chuyên (JPEG), (GIF), (PNG).
- Các phần tử IMG là một phần tử rỗng, cho phép người dùng để chèn một hình ảnh.
- Thuộc tính Padding CSS được sử dụng để xác định không gian giữa viền các phần tử và nội dung của nó.
- Một thumbnail là một hình ảnh nhỏ, hoặc một phần của một hình ảnh lớn hơn
- Trong năm 2007, Apple giới thiệu quá trình chuyển đổi CSS, mà sau này đã trở thành một tính năng độc quyền của Safari được gọi là CSS Animation
- Một chuỗi hình ảnh động CSS có thể được tạo ra bởi kiểu dáng các phần tử với thuộc tính `animation`
- Hầu hết các trang web điện thoại di động được tạo ra để đứng trước tên miền của các trang web chính với m ví dụ **m.aptech-education.com**

TRƯỜNG ĐÀO TẠO LẬP TRÌNH VIÊN VÀ QUẢN TRỊ MẠNG QUỐC TẾ BACHKHOA-APTECH

# THANK FOR WATCH !

