

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

---

# BÀI 16: VẼ ĐỒ HỌA VÀ JAVASCRIPT

# Mục tiêu

- Mô tả về vẽ đồ họa trong HTML5
- Cách vẽ đường thẳng
- Cách sử dụng màu và tỉ lệ trong suốt
- Cách vẽ các đối tượng khác nhau
- Làm việc với hình ảnh và văn bản
- Tạo các sự kiện trên trang với javascript và jquery
- Xử lý nội dung mở rộng trang trang web

# Phần tử Canvas 1-6

Các phần tử <canvas> trong HTML5 có thể được sử dụng để vẽ các mẫu hình trên các trang web cũng như để vẽ đồ họa động sử dụng JavaScript.

Các yếu tố <canvas> được biểu diễn như một hình chữ nhật trên một trang và cho phép người sử dụng để vẽ vòng cung, văn bản, hình dạng, độ dốc, và các mẫu.

Các <canvas> trong HTML5 là như <div>, <table>, hoặc <a> thẻ ngoại trừ các nội dung được sử dụng trong nó là ra thông qua JavaScript.

Các phần tử <canvas> không chứa bất kỳ khả năng vẽ, thay vào đó, các bản vẽ được thực hiện bằng cách sử dụng mã JavaScript

Để sử dụng phần tử <canvas>, người dùng phải thêm thẻ <canvas> trên trang HTML.

Sử dụng <canvas> với JavaScript cải thiện hiệu suất tổng thể của trang web và tránh yêu cầu tải hình ảnh từ các trang web.

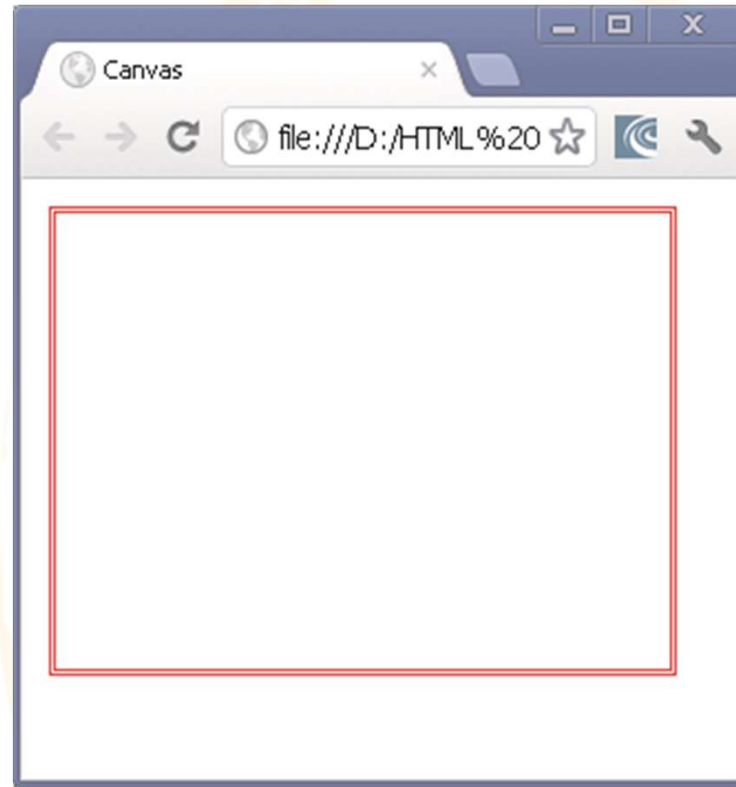
# Phần tử Canvas 2-6

- Ví dụ

```
<!DOCTYPE HTML>
<html>
  <head>
    <title> Canvas </title>
    <style>
      canvas{border: medium double red; margin:
        4px}
    </style>
  </head>
  <body>
    <canvas width="278" height="200"></canvas>
  </body>
</html>
```

# Phần tử Canvas 3-6

- Kết quả



Để vẽ một phần tử <canvas>, người dùng có thể sử dụng một đối tượng bối cảnh.

Đối tượng đó chứa các chức năng vẽ cho một phong cách cụ thể của đồ họa.

Ngữ cảnh hai chiều (2D) được sử dụng để làm việc với các hoạt động 2D.

# Phần tử Canvas 4-6

Các phần tử <canvas> trong DOM mở ra giao diện HTMLCanvasElement.

Giao diện này cung cấp các phương thức và thuộc tính để thay đổi trình bày và bố cục khung hình.

Đối tượng HTMLCanvasElement có phương thức getContext(context) và trả về ngữ cảnh vẽ đồ họa cho canvas.

- Ví dụ.

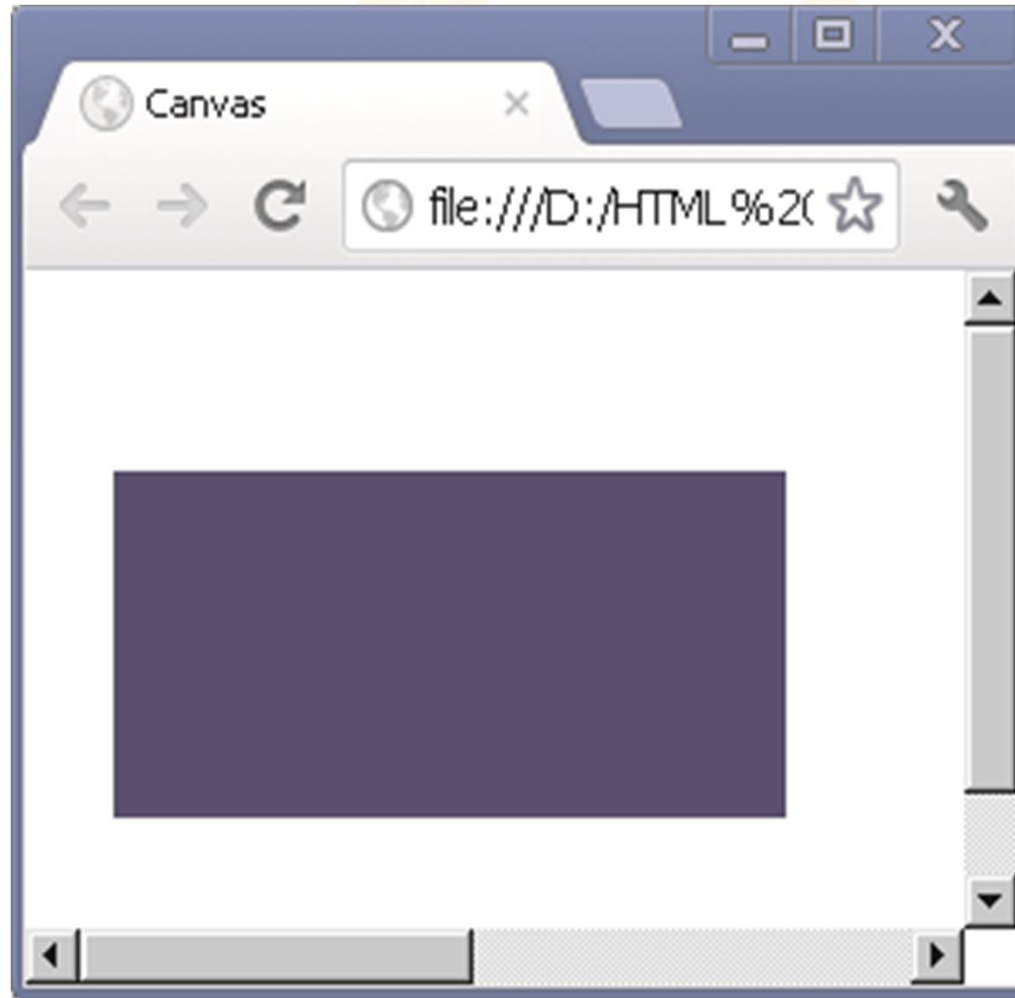
```
<!DOCTYPE HTML>
<html>
  <head>
    <title> Canvas </title>
    <script>
      window.onload = function()
      {
        var canvas = document.getElementById('mCanvas');
        var ctext = canvas.getContext('2d');
```

# Phần tử Canvas 5-6

```
        ctext.beginPath();  
        ctext.rect(18, 50, 200,  
100);  ctext.fillStyle =  
        "DarkBlue";  ctext.fill();  
    };  
</script>  
</head>  
<body>  
    <canvas id="mCanvas" width="578"  
height="200"></canvas>  
</body>  
</html>
```

# Phần tử Canvas 6-6

- Kết quả.





# Vẽ đường thẳng trong Canvas 1-3

- Bạn có thể vẽ đường thẳng trong canvas sử dụng các phương thức như `stroke()`, `beginPath()`, `lineTo()`, và `moveTo()`.

## Cú pháp:

```
cctx.beginPath();
```

```
cctx.moveTo(x, y);
```

```
cctx.lineTo(x, y);
```

```
cctx.stroke();
```

Trong đó,

- `cctx` – đối tượng context
- `beginPath()` – bắt đầu vẽ
- `moveTo()` – Chỉ ra vị trí vẽ
- `lineTo()` – chỉ ra vẽ tới đâu
- `stroke()` – gán màu và hiển thị nó

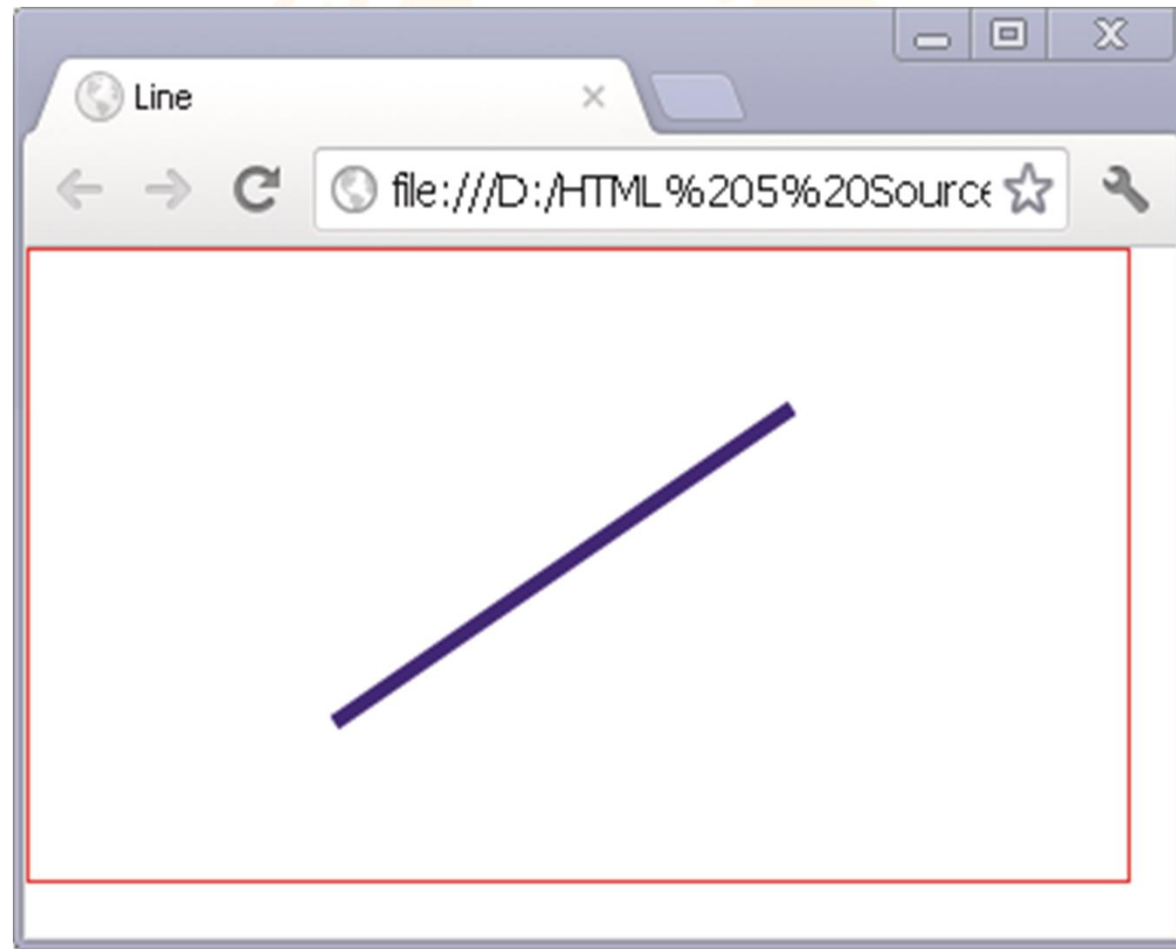
# Vẽ đường thẳng trong Canvas 2-3

- Ví dụ

```
<!DOCTYPE HTML>
<html>
<head>
  <title>Line</title>
  <style>
    body {margin: 0px; padding: 0px;}
    #mCanvas {border: 1px solid red;}
  </style>
  <script>
    window.onload = function() {
      var canvas = document.getElementById("mCanvas");
      var ctext = canvas.getContext("2d");
      ctext.beginPath(); ctext.moveTo(100, 150);
      ctext.lineTo(250, 50); ctext.lineWidth = 5;
      ctext.strokeStyle = "blue"; ctext.stroke(); };
    </script>
</head>
<body>
  <canvas id="mCanvas" width="360" height="200"></canvas>
</body>
</html>
```

# Vẽ đường thẳng trong Canvas 3-3

- Kết quả.



# Làm việc với các đối tượng đồ họa 1-17

- HTML5 canvas cho phép người dùng làm việc với các đối tượng đồ họa khác nhau, cụ thể như sau;

## ➤ Rectangle (hình chữ nhật)

- Với HTML5 canvas, người dùng có thể tạo một hình chữ nhật sử dụng phương thức `rect()`.
- Canvas HTML5 được đặt bằng cách sử dụng các tham số x và y và phù hợp kích thước thông qua thuộc tính chiều cao và chiều rộng.
- Bảng sau liệt kê các thuộc tính và phương pháp của hình dạng khác nhau phổ biến.

Thuộc tính và phương thức	Mô tả
<code>fillStyle</code>	Các giá trị có thể gradient, pattern, or a CSS color. Thuộc tính mặc định là solid black,
<code>fillRect(x, y, width, height)</code>	Cho phép người sử dụng để vẽ một hình chữ nhật.
<code>strokeStyle()</code>	Thực thi style

# Làm việc với các đối tượng đồ họa 2-17

Thuộc tính và phương thức	Mô tả
<code>strokeRect(x, y, width, height)</code>	Cho phép người dùng vẽ hình chữ nhật với stroke style đã có sẵn.
<code>clearRect(x, y, width, height)</code>	Xóa hình chữ nhật

- Ví dụ

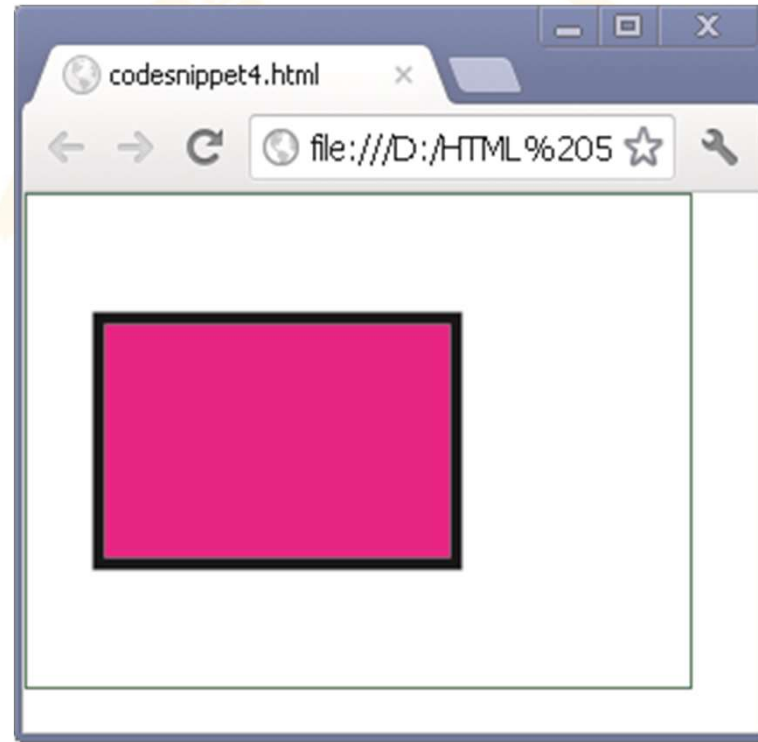
```
<!DOCTYPE HTML>
<html>
  <head>
    <style>
      #mCanvas {
        border: 1px solid green;
      }
      body {
        margin: 0px; padding: 0px;
      }
    </style>
```

# Làm việc với các đối tượng đồ họa 3-17

```
<script>
    window.onload = function() {
        var canvas =
            document.getElementById('mCanvas');  var ctext
            = canvas.getContext('2d');  ctext.beginPath();
            ctext.rect(30, 50, 150,
            100);  ctext.fillStyle =
            "Magenta";  ctext.fill();
            ctext.lineWidth = 5;
            ctext.strokeStyle =
            'black';  ctext.stroke();
        };
    </script>
</head>
<body>
    <canvas id="mCanvas" width="278"
    height="200"></canvas>
</body>
</html>
```

# Làm việc với các đối tượng đồ họa 4-17

- Kết quả



## ➤ Arcs

- Với HTML5 canvas, người dùng có thể tạo ra một vòng cung bằng cách sử dụng phương thức arc().
- Vòng cung được đại diện bằng cách sử dụng một góc bắt đầu, góc cuối, bán kính, một điểm trung tâm, và hướng bản vẽ (ngược chiều kim đồng hồ hoặc).

# Làm việc với các đối tượng đồ họa 5-17

## Cú pháp

`arc(x, y, radius, startAngle, endAngle, anticlockwise)`

Trong đó,

- `x, y` – chỉ ra tọa độ x,y
- `radius` – chỉ ra bán kính
- `startAngle, endAngle` – chỉ ra góc bắt đầu và góc kết thúc
- `anticlockwise` – vẽ theo chiều kim đồng hồ hay ngược (giá trị là `true` or `false`)

### • Ví dụ.

```
<!DOCTYPE HTML>
<html>
  <head>
    <style>
      body{
        margin: 0px; padding: 0px;
      }
    </style>
  </head>
  <body>
```

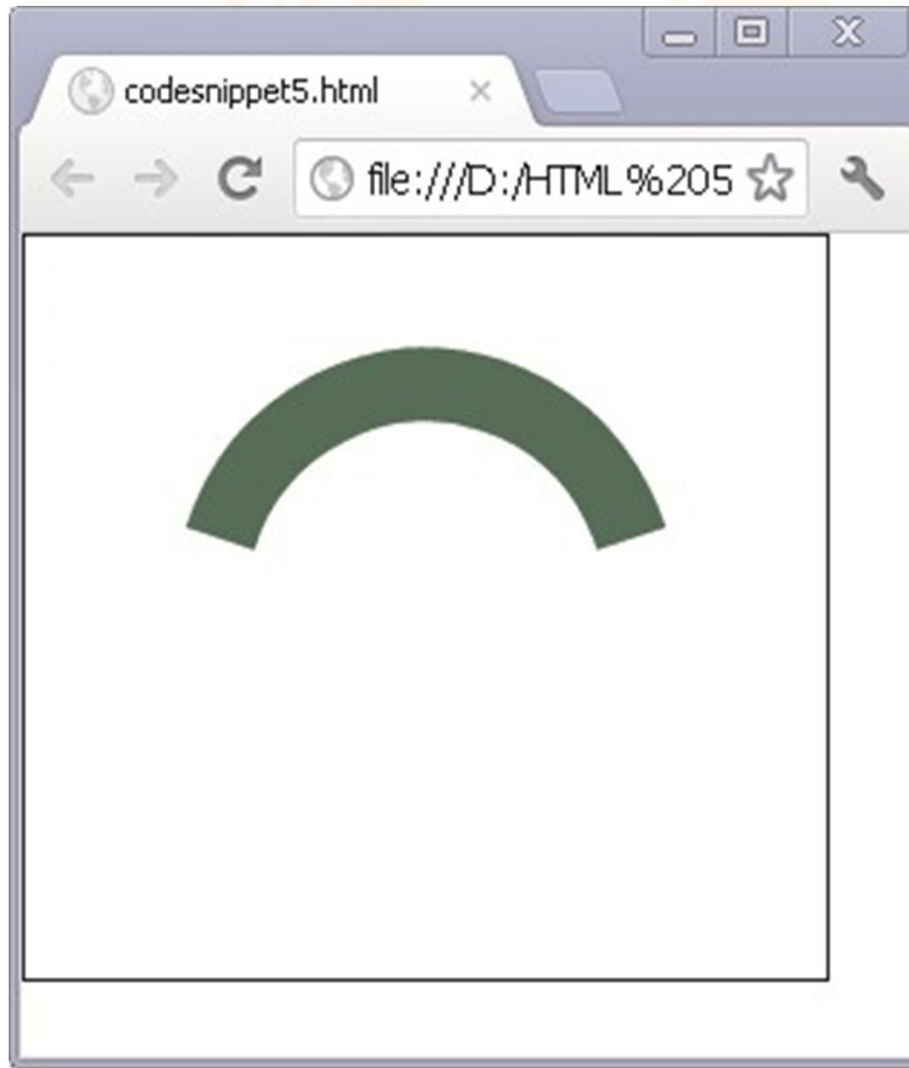


# Làm việc với các đối tượng đồ họa 6-17

```
#mCanvas {  
    border: 1px solid black; }  
</style>  
<script>  
    window.onload = function() {  
        var canvas = document.getElementById("mCanvas");    var  
        ctext = canvas.getContext("2d");  
        var x = canvas.width / 2;    var  
        radius = 75;  
        var startAngle = 1.1 * Math.PI;  
        var endAngle = 1.9 * Math.PI;  
        var ctrClockwise = false;  
        ctext.beginPath();  
        ctext.arc(x, y, radius, startAngle, endAngle, ctrClockwise);  
        ctext.lineWidth = 25;  
        // line color  
        ctext.strokeStyle = "DarkGreen";  
        ctext.stroke();  
    };  
</script> </head>  
<body>  
    <canvas id="mCanvas" width="278" height="250"></canvas>  
</body></html>
```

# Làm việc với các đối tượng đồ họa 7-17

- Kết quả



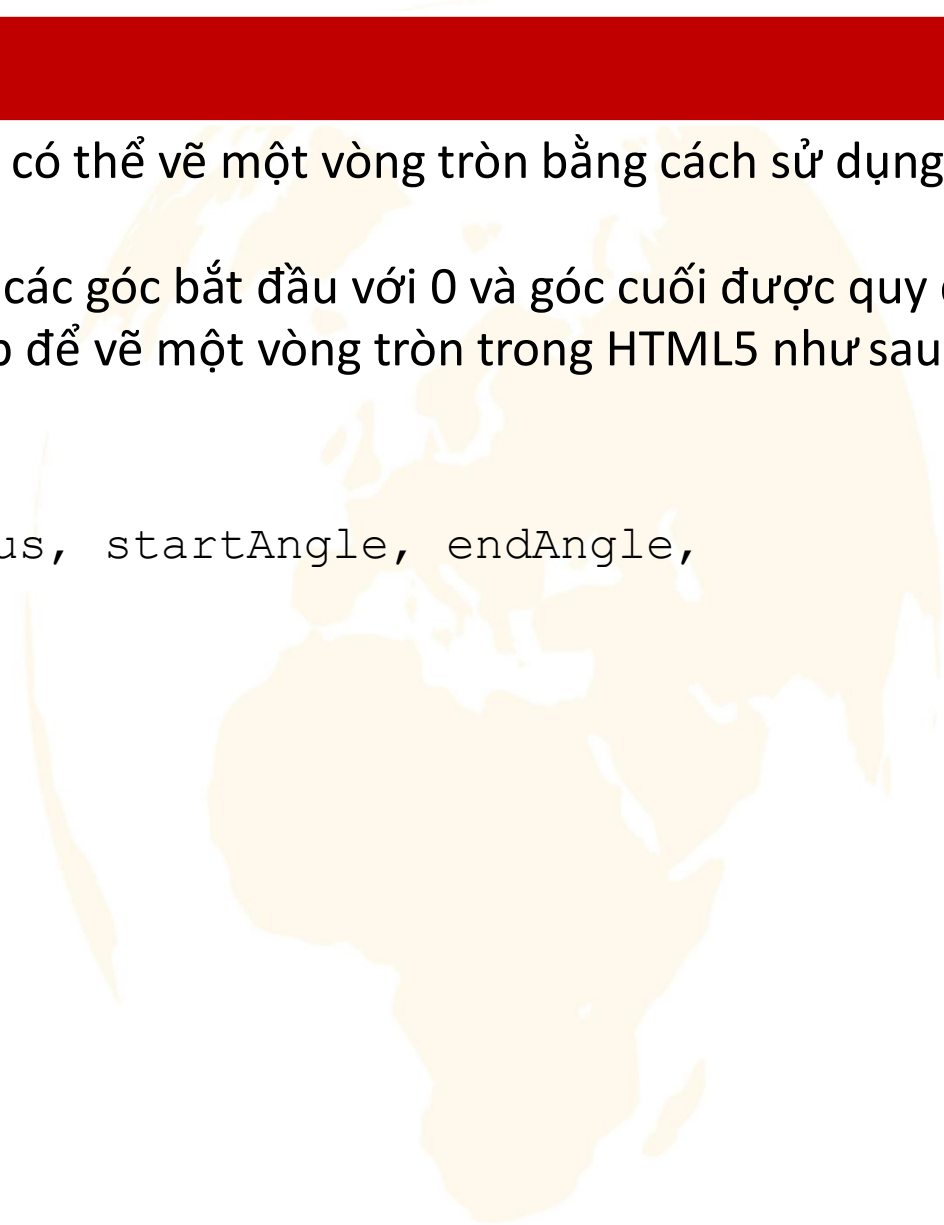
# Làm việc với các đối tượng đồ họa 8-17

## ➤ Circle

- Trong HTML5, bạn có thể vẽ một vòng tròn bằng cách sử dụng phương thức `arc()` cung.
- Bạn phải thiết lập các góc bắt đầu với 0 và góc cuối được quy định như  $2 * \text{PI}$ .
- Sau đây là cú pháp để vẽ một vòng tròn trong HTML5 như sau:

### Cú pháp:

```
arc(x, y, radius, startAngle, endAngle,  
anticlockwise)
```



# Làm việc với các đối tượng đồ họa 9-17

- Ví dụ

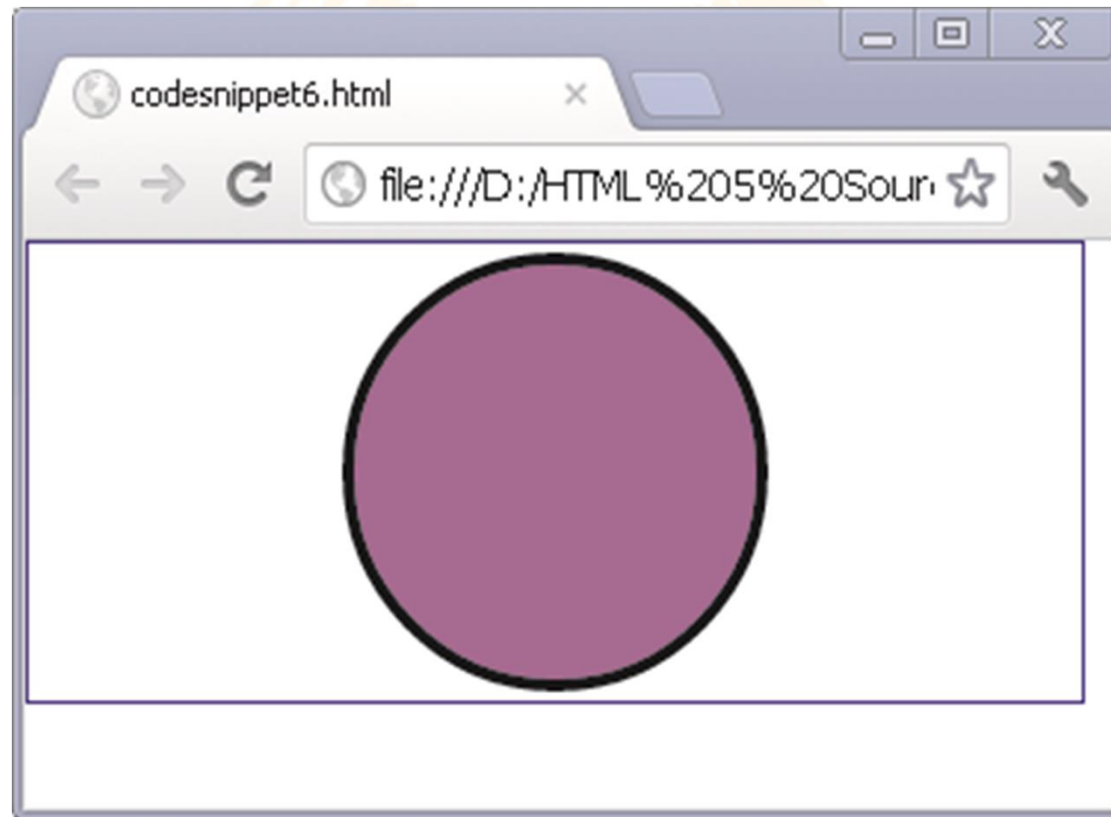
```
<!DOCTYPE HTML>
<html>
<head>
  <title>Circle</title>
  <style>
    body {
      margin: 0px;
      padding: 0px;
    }
    #mCanvas {
      border: 1px solid blue;
    }
  </style>
<script>
  window.onload = function() {
    var canvas = document.getElementById("mCanvas");
    var ctext = canvas.getContext("2d");
    var ctrX = canvas.width / 2;
    var ctrY = canvas.height / 2;
    var radius = 70;
```

# Làm việc với các đối tượng đồ họa 10-17

```
        ctext.beginPath();
        ctext.arc(ctrX, ctrY, radius, 0, 2 * Math.PI, false);
        ctext.fillStyle = "DarkOrchid";
        ctext.fill();
        ctext.lineWidth = 4;
        ctext.strokeStyle = "black";
        ctext.stroke();
    };
</script>
</head>
<body>
    <canvas id="mCanvas" width="356" height="150"></canvas>
</body>
</html>
```

# Làm việc với các đối tượng đồ họa 11-17

- Kết quả



# Làm việc với các đối tượng đồ họa 12-17

## ➤ Bezier Curves

- Sử dụng HTML5 canvas, bạn có thể tạo ra một đường cong Bezier sử dụng phương pháp () bezierCurveTo.
- Bezier đường cong được thể hiện với hai điểm kiểm soát, điểm ngưỡ cảnh, và một điểm kết thúc.
- Mã Snippet cho thấy làm thế nào để tạo ra một đường cong Bezier sử dụng HTML5.

```
<!DOCTYPE HTML>
<html>
  <head>
    <style>
      body
      {
        margin: 0px;
        padding: 0px;
      }
      #mCanvas
      {
        border: 1px solid maroon;
      }
    </style>
```

# Làm việc với các đối tượng đồ họa 13-17

```
<script>
    window.onload = function()
    {
        var canvas =
            document.getElementById("mCanvas");    var ctext
            = canvas.getContext("2d");
            ctext.beginPath();
            ctext.moveTo(188,
            130);
            ctext.bezierCurveTo(140, 10, 388, 10, 288, 100);
            ctext.lineWidth = 15;
            // line color
            ctext.strokeStyle =
            "purple";    ctext.stroke();
        };
    </script>
</head>
<body>
    <canvas id="mCanvas" width="378"
    height="200"></canvas>
</body>
</html>
```



# Làm việc với các đối tượng đồ họa 14-17

- Kết quả



# Làm việc với các đối tượng đồ họa 15-17

## ➤ Quadratic Curves

- HTML5 canvas cho phép người dùng tạo ra các đường cong bậc hai bằng cách sử dụng phương thức `quadraticCurveTo()`.
- Đường cong bậc hai được thể hiện qua các điểm ngữ cảnh, một điểm kết thúc, và một điểm kiểm soát.
- Mã Snippet cho thấy làm thế nào để tạo ra một đường cong bậc hai sử dụng HTML5

```
<!DOCTYPE HTML>
<html>
  <head>
    <style> body
    {
      margin: 0px; padding: 0px;
    }
    #mCanvas
    {
      border: 1px solid #9C9898;
    }
  </head>
  <body>
```

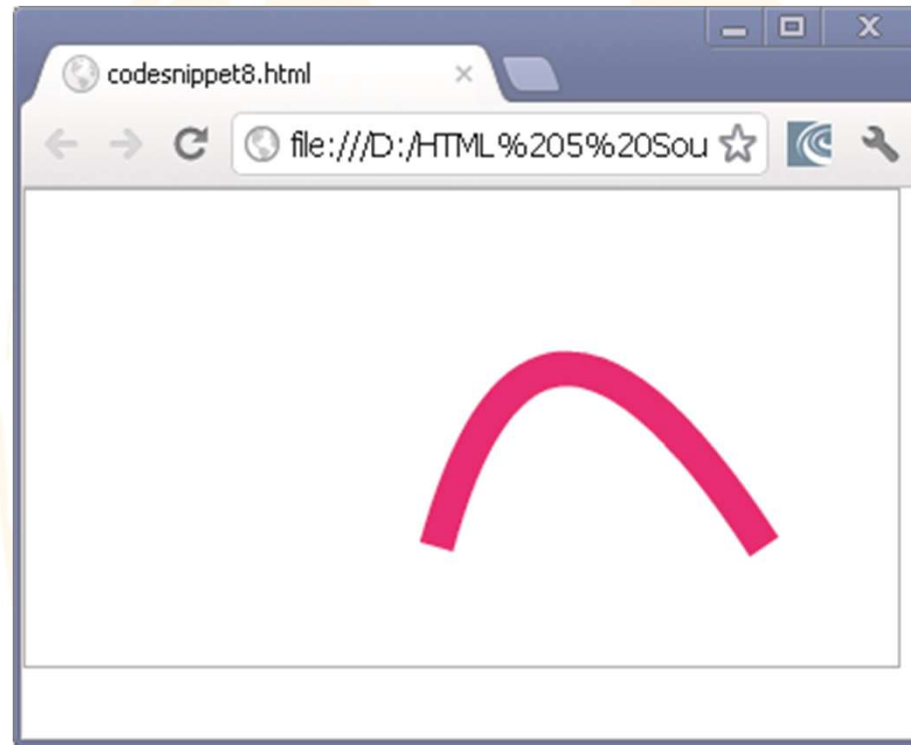
# Làm việc với các đối tượng đồ họa 16-17

```

window.onload = function() {
var canvas =
document.getElementById("mCanvas");  var ctext
= canvas.getContext("2d");  ctext.beginPath();
ctext.moveTo(178, 150);
ctext.quadraticCurveTo(220, 0, 320, 150);
ctext.lineWidth = 15;
// line color
ctext.strokeStyle = "Fuchsia";
ctext.stroke();
};
</script>
</head>
<body>
  <canvas id="mCanvas" width="378"
  height="200"></canvas>
</body>
</html>
```

# Làm việc với các đối tượng đồ họa 17-17

- Kết quả



# Làm việc với hình ảnh 1-3

- Trong HTML5, người dùng có thể vẽ các đối tượng hình ảnh trên canvas bằng cách sử dụng phương thức drawImage().
- Phương pháp drawImage() cũng có thể vẽ các bộ phận của một hình ảnh và tăng hoặc giảm kích thước của hình ảnh.
- Phương pháp này chấp nhận 9 thông số, tùy thuộc vào chỉnh sửa những gì cần thiết vào hình ảnh.
- Các đối tượng hình ảnh có thể là một video, hình ảnh, hoặc một phần tử canvas.
- Mã Snippet cho thấy làm thế nào để tạo ra một hình ảnh bằng cách sử dụng HTML5

```
<!DOCTYPE HTML>
<html>
  <head>

    <style>  body {
              margin: 0px;  padding: 0px;
            }
            #mCanvas {
              border: 1px solid #9C9898;
            }
    </style>
```

# Làm việc với hình ảnh 2-3

```
<script>
    window.onload = function()
    {
        var canvas =
        document.getElementById("mCanvas");   var ctext
        = canvas.getContext("2d");
        var imgObj = new Image();
        imgObj.onload =
        function()
        {
            ctext.drawImage(imgObj, 69, 50);
        };
        imgObj.src = "bird.jpg";
    };
</script>
</head>
<body>
    <canvas id="mCanvas" width="368"
    height="300"></canvas>
</body>
</html>
```

# Làm việc với hình ảnh 3-3

- Kết quả



# Làm việc với văn bản 1-5

- HTML5 canvas cho phép bạn thiết lập font, kiểu dáng và kích thước của văn bản bằng cách sử dụng các thuộc tính font.
- Các kiểu font chữ có thể nghiêng, bình thường, hoặc in đậm.
- Để thiết lập màu chữ, thuộc tính fillStyle của canvas có thể được sử dụng.
- Mã Snippet cho thấy làm thế nào để thiết lập font, kích thước, phong cách, và màu sắc của các văn bản trên một tấm vải HTML5.

```
<!DOCTYPE HTML>
<html>
  <head>
    <style>
      body {
        margin: 0px;
        padding: 0px;
      }
      #mCanvas {
        border: 1px solid blue;
      }
    </style>
```

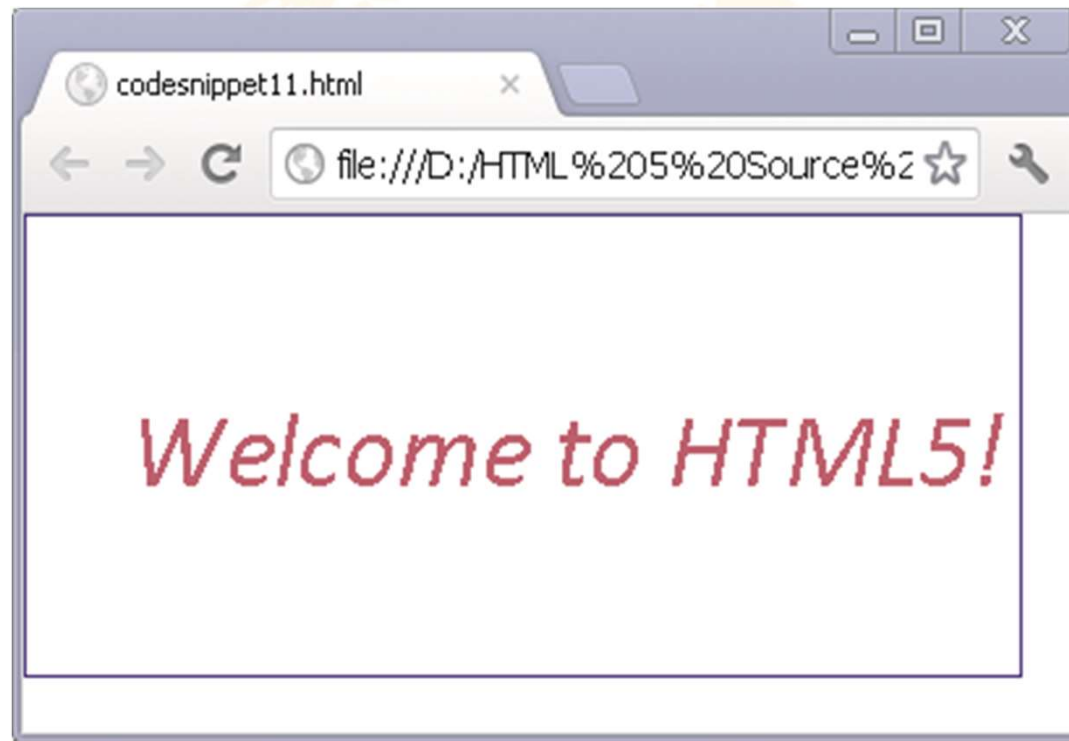


# Làm việc với văn bản 2-5

```
<script>
    window.onload = function() {
        var canvas = document.getElementById("mCanvas");
        var ctext =
            canvas.getContext("2d");
        ctext.font = "italic 30pt Calibri";
        ctext.fillStyle = "MediumVioletRed";
        ctext.fillText("Welcome to HTML5!", 40,
            100);
    };
</script>
</head>
<body>
    <canvas id="mCanvas" width="380"
        height="170"></canvas>
</body>
</html>
```

# Làm việc với văn bản 3-5

- Kết quả



- Trong HTML5 canvas, người dùng có thể thiết lập màu sắc stroke bằng cách sử dụng các phương thức strokeText () phương pháp và thuộc tính của ngữ cảnh canvas strokeStyle.

# Làm việc với văn bản 4-5

- Ví dụ

```
<!DOCTYPE HTML>
<html>
<head>
<title></title>
<style>
body { margin: 0px; padding: 0px; }
#mCanvas { border: 1px solid black; }
</style>
<script>
window.onload = function() {
    var canvas = document.getElementById("mCanvas");
    var ctext = canvas.getContext("2d"); var x = 80;
    var y = 110; ctext.font = "40pt Calibri";
    ctext.lineWidth = 2; ctext.strokeStyle = "Brown";
    ctext.strokeText("HTML5", x, y);
};
</script>
</head>
<body>
<canvas id="mCanvas" width="360" height="200"></canvas>
</body>
</html>
```

# Làm việc với văn bản 5-5

- Kết quả



# Sử dụng Transparency cho văn bản trong Canvas 1-3

- Có hai cách để thiết lập sự trong suốt cho các văn bản và các hình.
- Phương pháp đầu tiên là sử dụng `strokeStyle` và `fillStyle` bằng cách sử dụng hàm `rgb`.
- Phương pháp thứ hai là sử dụng thuộc tính state `globalAlpha` vẽ, có thể được áp dụng phổ biến.
- Thuộc tính `globalAlpha` là một giá trị dao động từ 0 (hoàn toàn trong suốt) và 1 (hoàn toàn mờ đục).
- Mã Snippet cho thấy việc sử dụng thuộc tính `globalAlpha`.

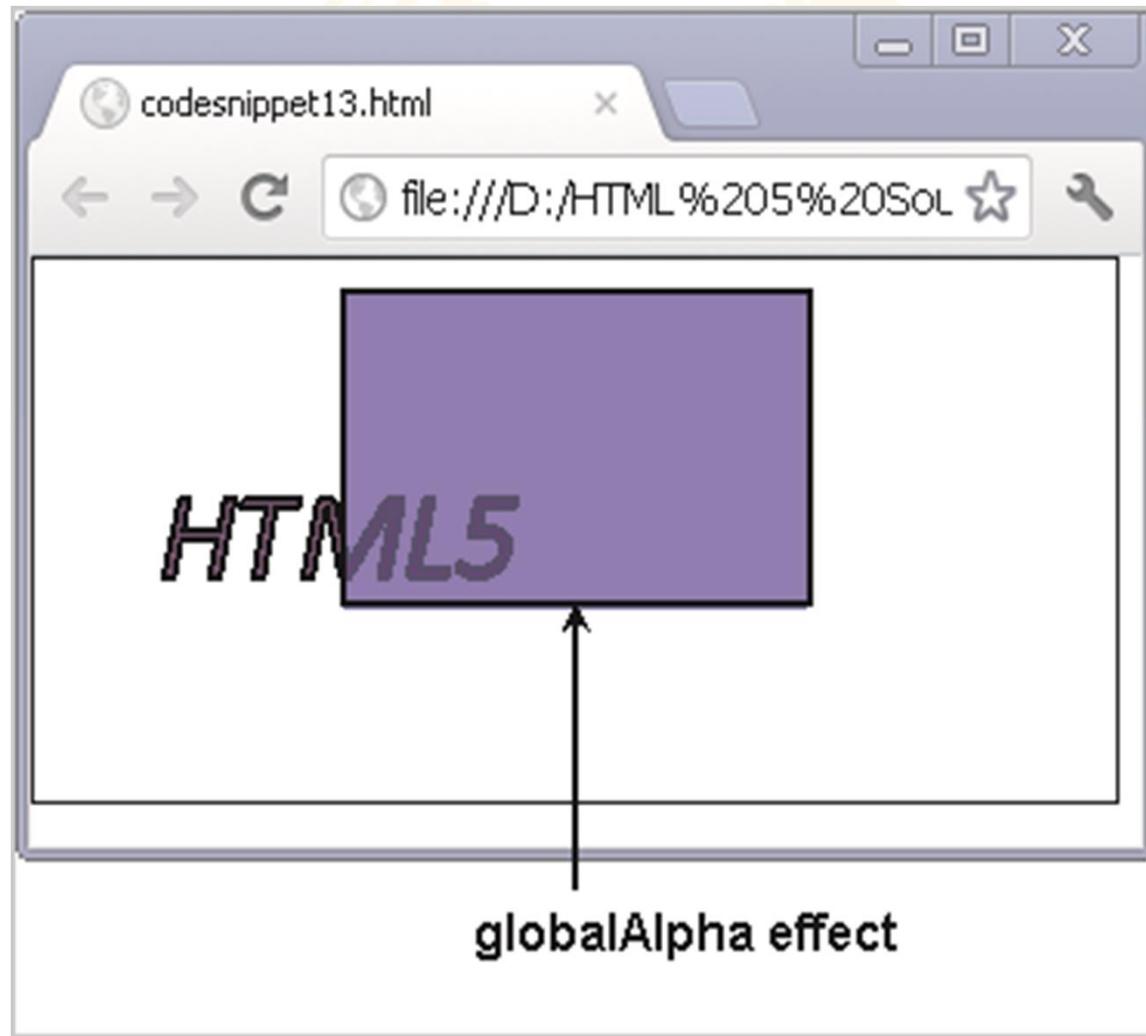
```
<!DOCTYPE HTML>
<html>
  <head>
    <style>
      body
      {
        margin: 0px;
        padding:
          0px;
      }
      #mCanvas {
        border: 1px solid black;
      }
    </style>
```

# Sử dụng Transparency cho văn bản trong Canvas 2-3

```
<script>
    window.onload = function()
    {
        var canvas = document.getElementById("mCanvas");   var
        ctext = canvas.getContext("2d");
        ctext.fillStyle      =    "Indigo";
        ctext.strokeStyle    =    "black";
        ctext.lineWidth=2;
        ctext.font = "italic 30pt Calibri";
        ctext.fillText("HTML5", 40, 100);
        ctext.strokeText("HTML5", 40, 100);
        ctext.fillStyle="blue";
        ctext.globalAlpha=0.5;
        ctext.fillRect(100, 10, 150, 100);
    };
</script>
</head>
<body>
<canvas id="mCanvas" width="350" height="170"></canvas>
</body>
</html>
```

# Sử dụng Transparency cho văn bản trong Canvas 3-3

- Kết quả



# Sử dụng các sự kiện với jQuery 1-6

- jQuery cũng cung cấp các sự kiện khác nhau để đối phó với các tương tác phổ biến khi người dùng di chuyển chuột hoặc thay đổi giữa hai hành động trong khi nhấn.
- Sau đây là các sự kiện:

## ➤ Sự kiện `hover()`

- Các `mouseenter` và `mouseleave` là hai sự kiện thường được sử dụng với nhau.
- jQuery cung cấp một `hover()` chức năng chấp nhận hai tham số.
- Tham số đầu tiên thực hiện khi di chuyển chuột trên các phần tử và chức năng thứ hai thực hiện khi chuột di chuyển ra khỏi các phần tử.
- Mã Snippet cho thấy các sự kiện di chuột.

```
<!DOCTYPE html>
<html>
  <head>
    <script src="jquery-1.7.2.min.js"></script>
    <script>

      $(document).ready(function() {
```

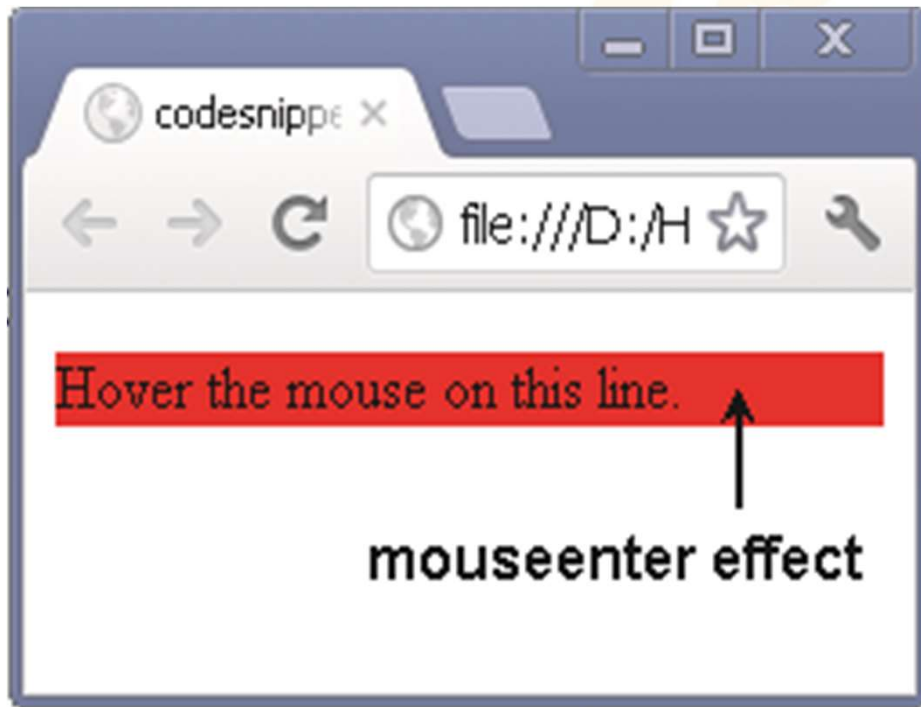


# Sử dụng các sự kiện với jQuery 2-6

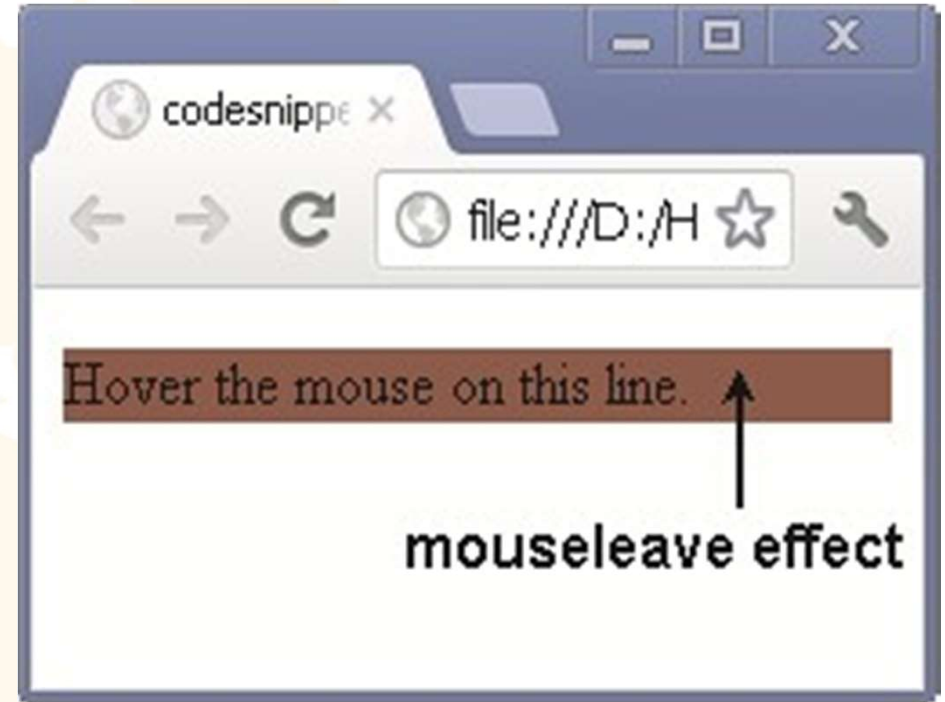
```
$ ("p").hover(function() {  
    $ ("p").css ("background-color","red");  
    },function() {  
    $ ("p").css ("background-  
color","maroon");  
    });  
});  
</script>  
</head>  
    <body>  
        <p>Hover the mouse on this line.</p>  
    </body>  
</html>
```

# Sử dụng các sự kiện với jQuery 3-6

- Khi chuột di vào



- Khi chuột rời ra



# Sử dụng các sự kiện với jQuery 4-6

## ➤ Sự kiện toggle()

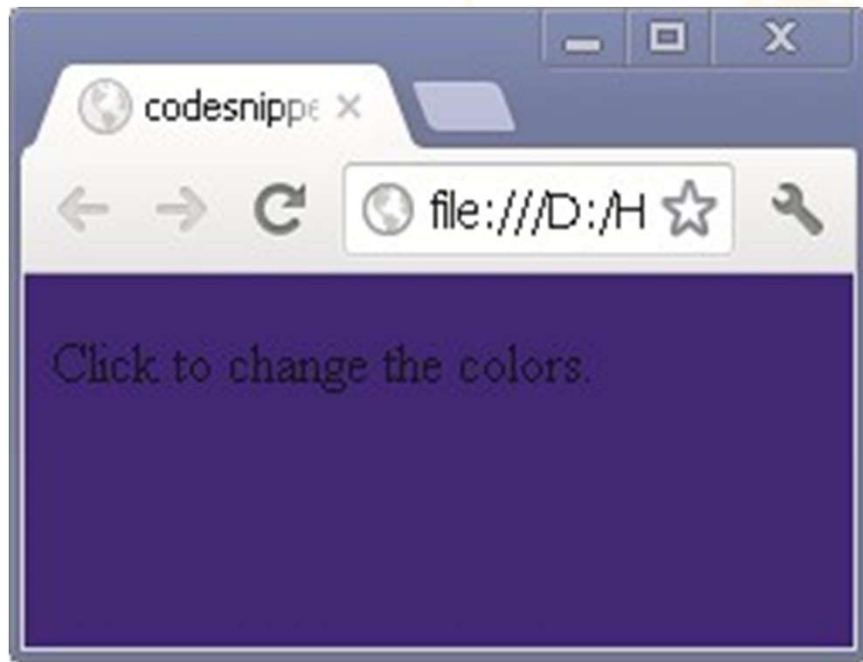
- Các sự kiện toggle () hoạt động theo cách tương tự như sự kiện hover (), ngoại trừ việc nó phản ứng với cú click chuột.
- Phương thức toggle () chấp nhận hơn hai chức năng như các đối số.
- Tất cả các chức năng chuyển tới sự kiện toggle() sẽ phản ứng với hành động nhấp chuột tương ứng của nó.
- Mã Snippet cho thấy các sự kiện toggle.

```
<!DOCTYPE html>
<html>
<head>
<script src="jquery-3.2.1.min.js"></script>
<script>
    $(document).ready(function() {
        $("p").toggle(function() {
            $("body").css("background-color", "blue");},
            function() {
                $("body").css("background-color", "pink");},
            function() {
                $("body").css("background-color", "grey");}
        );
    });
```

# Sử dụng các sự kiện với jQuery 5-6

```
</script>
</head>
<body>
    <p>Click to change the colors.</p>
</body>
</html>
```

- Hình dưới đây hiển thị các hiệu ứng chuyển đổi sang màu xanh.



- Hình dưới đây hiển thị các hiệu ứng chuyển đổi sang màu hồng.



# Sử dụng các sự kiện với jQuery 6-6

- hình dưới đây hiển thị các hiệu ứng chuyển đổi sang màu xám.



# Đính kèm các nội dung bên ngoài trang web

HTML5 giới thiệu thẻ <eventsource> cho phép người sử dụng để đẩy nội dung bên ngoài vào các trang web. Mô hình này được gọi là đẩy mô hình.

Kể từ khi thẻ <eventsource> không được hỗ trợ trong nhiều trình duyệt, người dùng sử dụng thẻ <embed> cho mục đích này.

Thẻ <embed> là một phần tử mới trong HTML5 và nó được biểu diễn như là một container cho một nội dung tương tác hoặc một ứng dụng bên ngoài.

Thẻ <embed> thường được sử dụng để thêm các phần tử như hình ảnh, âm thanh, hoặc video trên một trang web.

- Ví dụ

```
<embed src="mymovie.mp3" />
```

# Tổng kết

- Các phần tử <canvas> là một khu vực vẽ mà người dùng có thể vẽ đồ họa, sử dụng hình ảnh, thêm hình ảnh động, và cũng có thể thêm văn bản để nâng cao trải nghiệm người dùng trên các trang web.
- Để tạo ra một dòng, trên một bức vẽ có thể sử dụng các phương thức stroke (), beginPath (), lineTo (), và moveTo () .
- Vòng cung được đại diện bằng cách sử dụng một góc bắt đầu, góc cuối, bán kính, một điểm trung tâm, và hướng vẽ (ngược chiều kim đồng hồ hoặc).
- Với HTML5 canvas, người dùng có thể tạo ra một hình chữ nhật bằng cách sử dụng phương thức rect () .
- Đường cong Bezier được thể hiện với hai điểm kiểm soát, điểm ngưỡng cảnh, và một điểm kết thúc.
- HTML5 canvas cho phép người dùng tạo ra các đường cong bậc hai bằng cách sử dụng phương thức quadraticCurveTo () .
- HTML5 canvas cho phép người dùng vẽ đối tượng hình ảnh trên canvas bằng cách sử dụng phương thức drawImage () .



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

# THANK FOR WATCH !

