

# BÀI 4

## LÀM VIỆC VỚI CÁC THÀNH PHẦN VIDEO, AUDIO, CANVAS CỦA HTML5

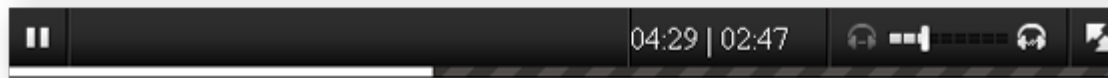
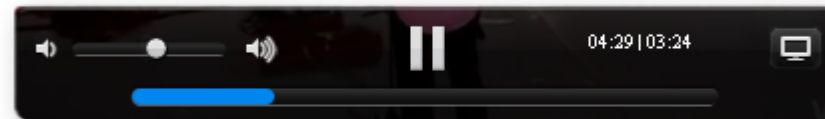
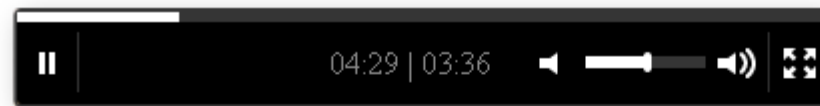
## NHẮC LẠI BÀI TRƯỚC

- ✓ Tổng quan về Javascript và JQuery
- ✓ Làm việc với Javascript
- ✓ Làm việc với thư viện JQuery
- ✓ Học Javascript, jQuery với w3schools

# MỤC TIÊU BÀI HỌC

- ✓ Chèn các thành phần video, audio vào trang
- ✓ Điều khiển video với Javascript
- ✓ Làm quen với thành phần canvas
- ✓ Sử dụng thành phần canvas để thực hiện:
  - Vẽ hình và đường
  - Áp dụng màu và cộ fradient
  - Thực hiện tạo hình động

# THÀNH PHẦN VIDEO, AUDIO TRONG HTML5



## THÀNH PHẦN VIDEO, AUDIO TRONG HTML5

- ✓ Tại sao phải thêm video, âm thanh vào trang web?
  - Tạo nên trang web hấp dẫn
  - Thu hút sự truy cập của người duyệt với website
  - Là một cách tiếp cận người dùng
- ✓ Trước khi có HTML5, làm cách nào để chèn video, âm thanh vào trang web?
  - Phải sử dụng plug-ins của các công ty thứ 3 (third party)



Microsoft®  
**Silverlight™**

## THÀNH PHẦN VIDEO, AUDIO TRONG HTML5

✓ HTML5 chèn video, âm thanh vào trang web như thế nào?

- Cung cấp thành phần HTML video chạy trong trình duyệt
- Tích hợp thêm với Javascript
- Câu lệnh:


```
<video src= 'video/BigBuck.ogg' autoplay></video>
```

- **Autoplay**: thuộc tính quy định với trình duyệt đoạn video sẽ được chơi ngay khi trang được load

## THÀNH PHẦN VIDEO, AUDIO TRONG HTML5

- ✓ HTML5 cung cấp thêm các điều khiển cho video:

```
<video src= 'video/BigBuck.ogg' controls poster= 'poster854.jpg'  
width= '320' height= '180' ></video>
```

- Controls: cung cấp trình điều khiển video trên các trình duyệt khác nhau
- 
- Poster: thuộc tính chỉ định cho trình duyệt load hình ảnh ban đầu của video
  - Width, height: thuộc tính chỉ định kích thước của video

## THÀNH PHẦN VIDEO, AUDIO TRONG HTML5

- Audio: thuộc tính này cho phép tắt tiếng của video, giá trị '**muted**'
- Loop: thuộc tính này sẽ kích hoạt đoạn video phát lại
- Preload: cho phép tải đoạn video ngay khi tải trang web, giảm thời gian tải video



## THÀNH PHẦN VIDEO, AUDIO TRONG HTML5

### ✓ Thêm các thuộc tính để video được hỗ trợ trên nhiều trình duyệt:

- Có nhiều định dạng, codec cho video nhưng không phải lúc nào cũng hỗ trợ HTML5 video hiển thị trên các trình duyệt
- **Định dạng Ogg:**
  - Định dạng theo chuẩn mã nguồn mở
  - Được hỗ trợ bởi các trình duyệt Chrome, Firefox, Opera



- **Định dạng MP4:**
  - Sử dụng **codec H.264** và âm thanh sử dụng **codec AAC**
  - Được hỗ trợ bởi IE, Safari, iOS và Android

## THÀNH PHẦN VIDEO, AUDIO TRONG HTML5

- Để đoạn video ở cả 2 định dạng Ogg và MP4 sẽ giúp hiển thị tốt trên các trình duyệt và thiết bị

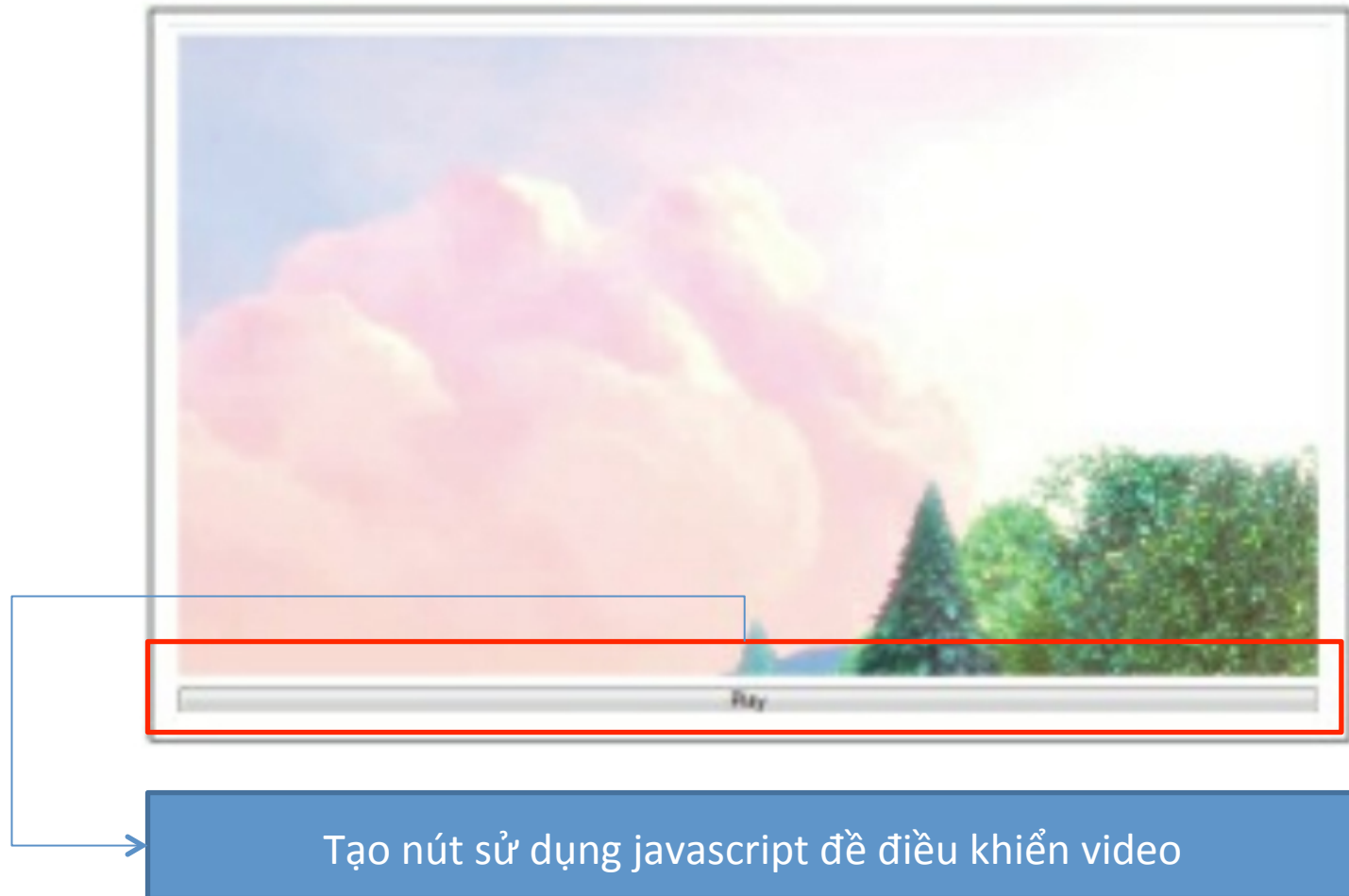
```
<video controls poster= 'poster320.jpg' width= '320' height= '180' >  
    <source src= 'video/BigBuck.ogg' >  
    <source src= 'video/BigBuck.mp4' >  
</video>
```

- Thêm thuộc tính type, codecs: mô tả cấu trúc kiểu của file ogg, mp4

```
<source src= 'video/BigBuck.ogg' type= 'video/ogg; codecs="theora, vorbis"' >  
<source src= 'video/BigBuck.mp4' type= 'video/mp4; codecs="avc1.42E01E, mp4a.40.2"' >
```

## THÀNH PHẦN VIDEO, AUDIO TRONG HTML5

### ✓ Điều khiển video với Javascript:



## THÀNH PHẦN VIDEO, AUDIO TRONG HTML5

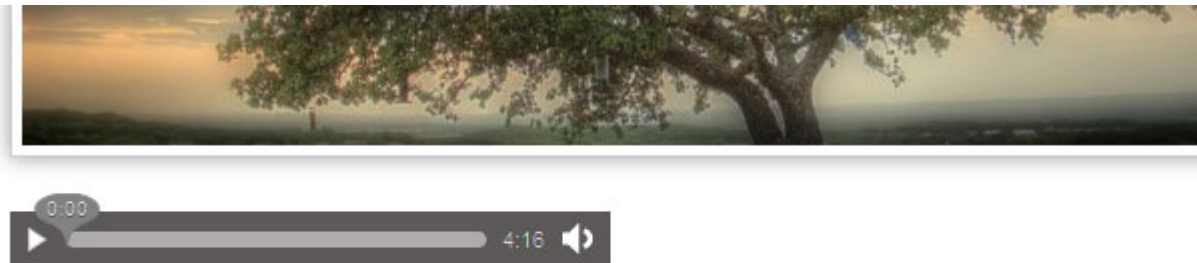
- Khởi tạo hàm trong javascript để thực hiện các sự kiện: Play, Pause, ...

```
<script type="text/javascript">
    var video = document.getElementsByTagName('video')[0];
    var playPause = document.getElementById('playPause');
    function setButtonWidth(e){
        playPause.style.width = video.videoWidth + 'px';
    }
    function setPlayPause(e){
        if(video.paused) {
            playPause.value = 'Play';
            playPause.onclick = function(e) { video.play(); }
        }else{
            playPause.value = 'Pause';
            playPause.onclick = function(e) { video.pause(); }
        }
    }
</script>
```

## THÀNH PHẦN VIDEO, AUDIO TRONG HTML5

### ✓ Chèn âm thanh vào trang web sử dụng HTML5:

```
<audio controls>  
  <source src= 'video/BigBuck.ogg' >  
  <source src= 'video/BigBuck.mp3' >  
</audio>
```



Song from Nivel Del Mar – “Sounds for Soulsurfers”

Nullam volutpat mollis ipsum, eget facilisis est accumsan pretium. Phasellus consectetur laoree pharetra. Quisque ac enim tempor eros bibendum fermentum eget id neque. Donec dignissim

# CANVAS



# CANVAS

- ✓ Canvas là hàm API vẽ 2 chiều của HTML5
- ✓ Hình vẽ sử dụng Canvas:
  - Cập nhật được trong thời gian thực
  - Lưu lại dưới định dạng .png
- ✓ Sử dụng các thành phần Canvas để xác định bề mặt vẽ, nhưng phải xác định hướng vẽ và dòng kết quả, hình dạng, màu sắc với Javascript
- ✓ Canvas luôn làm việc cùng với javascript
- ✓ Các bước làm việc với Canvas:
  - Định nghĩa thành phần canvas trong HTML
  - Tham chiếu bối cảnh vẽ cho các phần từ đó như một biến trong Javascript

# CANVAS

## ✓ Khởi tạo:

- Lệnh canvas:

```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #c3c3c3;"></canvas>
```

- Kết hợp vẽ với Javascript:

```
<script type="text/javascript">  
    var c=document.getElementById("myCanvas");  
    var ctx=c.getContext("2d");  
    ctx.fillStyle="#FF0000";  
    ctx.fillRect(0,0,150,75);  
</script>
```





# CANVAS

```
var ctx=c.getContext("2d");
```

- ✓ Đối tượng `getContext("2d")` là đối tượng HTML5 chứa các phương thức vẽ đường path, hình hộp, hình tròn, character, hình ảnh, v.v....

```
ctx.fillStyle="#FF0000";
```

- ✓ Định nghĩa kiểu màu tô là màu đỏ

```
ctx.fillRect(0,0,150,75);
```

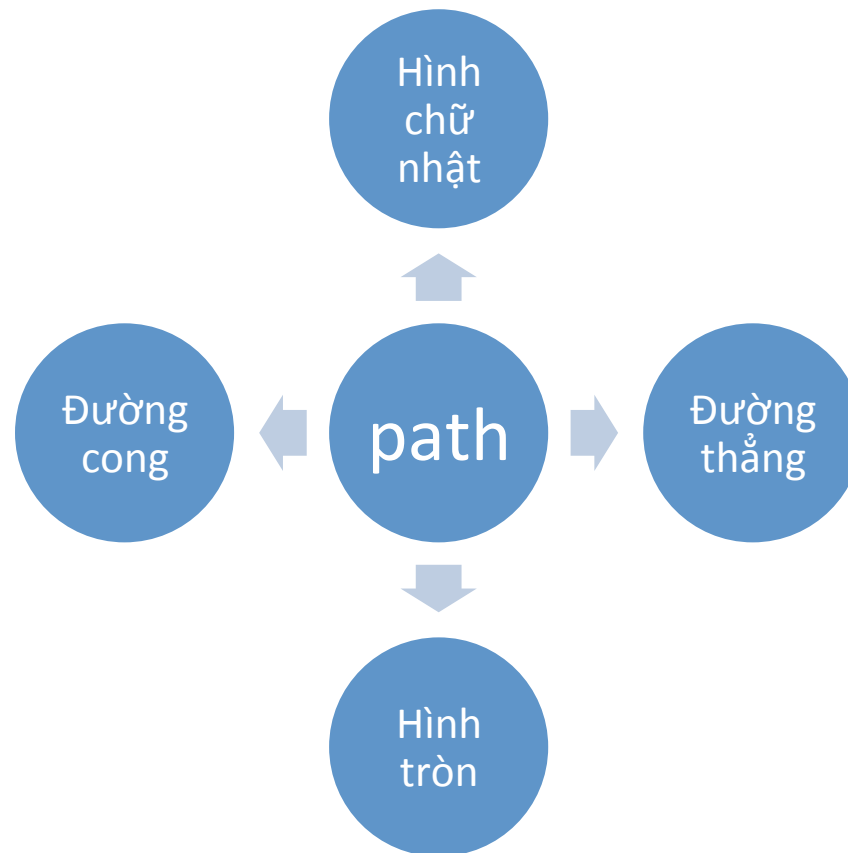
- ✓ Định nghĩa vẽ một hình chữ nhật kích thước 150x75, bắt đầu từ góc trên bên trái (0,0)

## ✓ Ưu điểm của thành phần Canvas:

- Cho phép tạo graphic, hình động (animation), gradient, các đối tượng đồ họa khác bằng mã
- Đã được các trình duyệt phổ biến hỗ trợ
- Khả năng mạnh mẽ: làm game, animation, chart, graph, vector, ...
- Không phải sử dụng thêm plugin

## ✓ Vẽ đường path:

- Đường path: tạo nên các hình dạng cơ sở

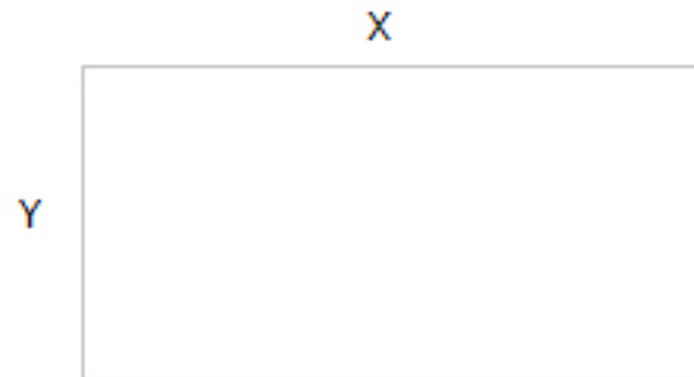


## ✓ Vẽ hình chữ nhật:

```
.fillStyle()  
.fillRect(x,y, width, height)  
.strokeStyle()  
.strokeRect(x, y, width, height)
```

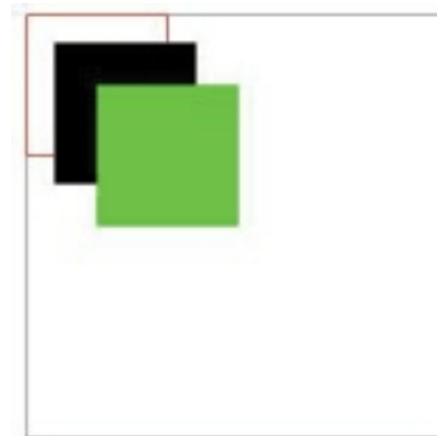
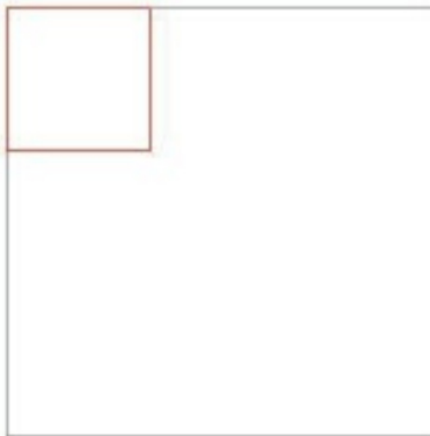
- X,y: tọa độ vẽ hình
- Width, height: kích thước hình

Tọa độ x,y trên màn hình



## ✓ Vẽ hình chữ nhật:

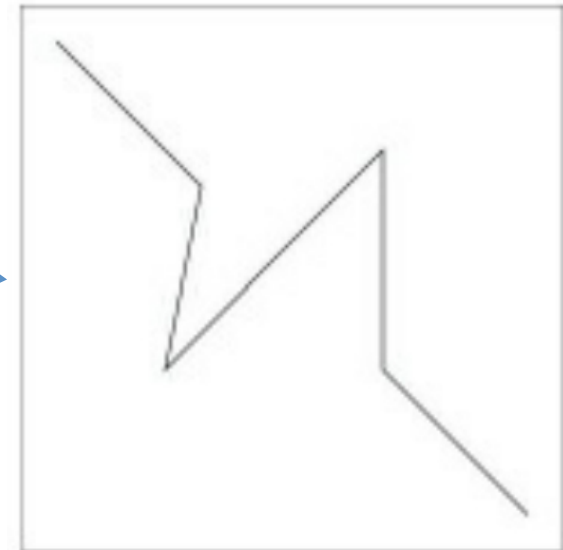
```
function setup() {  
  var canvas = document.getElementById('lessonCanvas');  
  if (canvas.getContext) {  
    var ctx = canvas.getContext('2d');  
    ctx.strokeStyle = 'rgb(255, 0, 0)';  
    ctx.strokeRect(0.5, 0.5, 100, 100);}  
}
```



# CANVAS

## ✓ Vẽ đường thẳng:

```
ctx.strokeRect(0, 0, 300, 300);  
ctx.moveTo(20, 20);  
ctx.lineTo(100, 100);  
ctx.lineTo(80, 200);  
ctx.lineTo(200, 80);  
ctx.lineTo(200, 200);  
ctx.lineTo(280, 280);  
ctx.stroke();
```

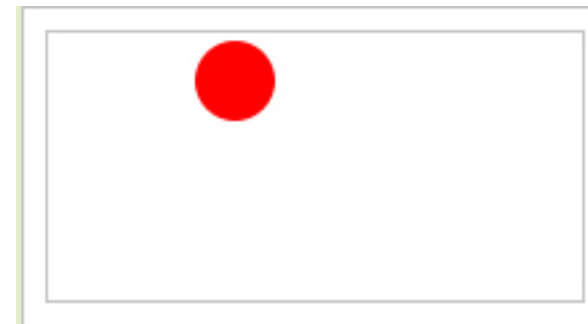


<b>moveTo(x, y)</b>	<b>Tạo ra một đường path phụ với tọa độ xác định</b>
<b>lineTo (x,y)</b>	<b>Thêm điểm point mới, kết nối với điểm trước đó bằng đường thẳng</b>

## ✓ Vẽ đường tròn:

```
<script type="text/javascript">  
    var c=document.getElementById("myCanvas");  
    var ctx=c.getContext("2d");  
    ctx.fillStyle="#FF0000";  
    ctx.beginPath();  
    ctx.arc(70,18,15,0,Math.PI*2,true);  
    ctx.closePath();  
    ctx.fill();  
</script>
```

Arc(x,y, bán kính , 2 πr)

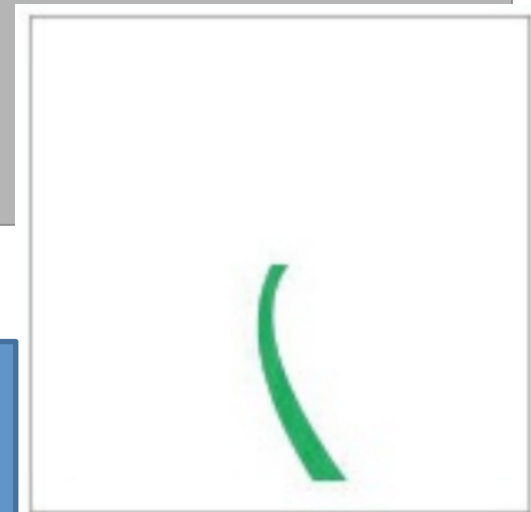


# CANVAS

## ✓ Vẽ đường cong:

```
ctx.fillStyle = 'rgb(0,173,104)';  
ctx.moveTo(145, 150);  
ctx.quadraticCurveTo(120, 200, 170, 280);  
ctx.lineTo(190, 280);  
ctx.quadraticCurveTo(125, 190, 155, 150);  
ctx.fill();  
var pt1 = { x: 155, y: 145 };  
var pt2 = { x: 93, y: 106 };  
var cp1 = { x: 111, y: 154 };  
var cp2 = { x: 4, y: 131 };
```

```
quadraticCurveTo(cpx, cpy, x, y)  
bezierCurveTo(cp1x, cp1y, cp2x, cp2y, x, y)
```





# CANVAS

## ✓ Vẽ text:

```
ctx.font = "bold 1.8em sans-serif";  
ctx.fillText(text,x,y,maxWidth);
```

```
ctx.font = "bold 1.8em  
sans-serif";  
ctx.fillText('HELLO MY  
NAME IS', 12, 40);
```



## ✓ Tô màu gradient:

```
var c=document.getElementById("myCanvas");  
var ctx=c.getContext("2d");  
  
var grd=ctx.createLinearGradient(0,0,170,0);  
grd.addColorStop(0,"black");  
grd.addColorStop(1,"white");  
  
ctx.fillStyle=grd;  
ctx.fillRect(20,20,150,100);
```

Phương thức xác định màu sắc và vị trí của đối tượng gradient  
Thường được sử dụng cùng `createLinearGradient()` và `createRadialGradient()`



## ✓ Chèn thêm hình ảnh:

```
context.drawImage(img,sx,sy,swidth,sheight,x,y,width,height);
```

Tham số	Diễn tả
Img	Chỉ định thành phần hình ảnh sử dụng
Sx	Tùy chọn. Tọa độ x nơi tọa độ bắt đầu
Sy	Tùy chọn. Tọa độ y nơi tọa độ bắt đầu
Swidth	Tùy chọn. Chiều rộng của hình ảnh cắt bớt
Sheight	Tùy chọn. Chiều cao của hình ảnh cắt bớt
X, y	Tọa độ x, y để đặt hình ảnh
Width, height	Tùy chọn. Chiều dài, rộng của hình ảnh sử dụng

## ✓ Sử dụng biến đổi:

- Sử dụng kết hợp với khung hình vẽ sẽ tạo ra hình ảnh đẹp hơn
- 3 loại biến đổi:
  - Scaling
  - Rotating
  - Translating
- Sử dụng kết hợp với hàm: **save, restore**

```
ctx.fillRect(30, 120, 40, 40);
```

```
ctx.translate(0, 20);
```

```
ctx.fillRect(80, 120, 40, 40);
```

## ✓ Vẽ hình chuyển động lặp (loop):

- Kết hợp sử dụng với hàm javascript (*setInterval*), sẽ tạo ra được những chuyển động lặp

```
function setup() {  
    var canvas =  
document.getElementById('lessonCanvas');  
    if (canvas.getContext) {  
        ctx = canvas.getContext('2d');  
  
        img = new Image();  
        img.onload = function(){  
  
            setInterval(function () { draw(); }, 36);  
  
        }  
        img.src = "";  
    }  
}
```

```
function draw() {  
    drawBackground();  
  
    <!--ufo-->  
    ctx.drawImage(img, x, y);  
  
    x += 3;  
    if(x > 300){  
        x = -50;  
        y=Math.random()*300;  
    }  
}
```

## TỔNG KẾT

- ✓ Khi chèn video, âm thanh vào trang web nên khai báo các định dạng **.ogg, .mp4, .mp3** để được các trình duyệt phổ biến hỗ trợ tốt nhất
- ✓ Khi vẽ các hình cơ bản với Canvas, điều cần chú ý:
  - Tọa độ vẽ
  - Kích thước
  - Hàm tính đặc biệt (chu vi hình tròn)
- ✓ Sử dụng kết hợp Canvas với Javascript để được hình ảnh tốt nhất