

# HTML 5

강사 : 윤 차 주

# HTML5 새로운세계

1. html5의 동영상처리

2. canvas-기본개요

3. canvas-도형작성

4. canvas-이미지

## ➤ 동영상 = 비디오 파일 + 오디오 파일

### 1. 웹 상에서 사용 가능한 동영상 파일 형식

형식	설명
MPEG4(*.mp4, *.m4v)	영상과 음성을 디지털 데이터로 전송하고 저장하는 규격 중 하나로 mpeg-4규격으로 압축된 동영상 또는 음성파일.
플래쉬비디오(*.flv)	어도비사의 flash player를 통해 지원하는 플래쉬 규격
AVI(*.avi)	윈도우 운영체제에서 기본적으로 제공하는 동영상파일형식으로 대용량, 고화질 이므로 실시간에 적합하지 못함
WebM(*.webm)	구글이 발표한 공개소스 방식의 동영상파일 형식으로 대부분의 웹 브라우저 업 체에서 이 형식을 지원하겠다고 함
Ogg(*.ogg)	특허권으로 보호되지 않는 오픈 표준 파일 형식으로 무료로 사용할 수 있어서 여 러 무료 및 상용미디어 플레이어에서 ogg의 여러 코덱을 이용하여 사용

### 2. 비디오 코덱



## ▶ 코덱의 종류

종류	설명
H.264	고화질, 높은 압축률 지원. 애플의 아이폰, 안드로이드와 같은 모바일 장치를 위해 유튜브에서 H.264로 인코딩된 동영상 제공(주로 .MP4파일에 많이 사용)
OGG 테오라	테오라는 사용료가 없는 공개 코덱으로 주로 OGG파일에 사용(모질라의 파이어폭스 브라우저에서 플러그인 없이도 OGG테오라 재생가능)
VP8	구글에서 2010년 5월에 발표한 공개코덱으로 크롬브라우저 비롯한 IE9, 파이어폭스, 오페라와 같은 브라우저업체들이 지원하겠다고 발표함.-WEBM형식에 사용

\* 비디오 코덱의 브라우저별 지원현황

브라우저 코덱	IE 9	FireFox 3.6.17	Opera 11.11	Chrome 12.0.7	Safari 5.0.5
Ogg	X	O	O	O	X
MPEG 4	O	X	X	O	O
WebM	△	X	O	O	X

\* 오디오 코덱의 브라우저별 지원현황

브라우저 코덱	IE	FireFox	Opera	Chrome	Safari
<b>Vorbis(oga, ogg)</b>	X	O	O	O	X
<b>wav, wma</b>	O	O	O	X	O
<b>mp3</b>	O	X	X	O	O
<b>AAC</b>	O	X	X	O	O

- **Mp3** – 가장 많이 알려진 오디오 코덱으로 압축한 오디오 데이터 중 사람들이 들을 수 없거나 의미없는 소리는 버리고 인코딩
- **AAC** – 애플의 아이폰, 아이튠즈, 아이팟에서 사용되는 기본 오디오 형식
- **Vorbis** – 공개소스이며 누구나 무료로 사용할 수 있는 코덱으로 확장자는 .ogg, .oga를 사용

### > 오디오 재생을 위한 <audio>엘리먼트

#### ■기본형식

```
<audio src="재생파일 내용"></audio>
```

#### ■주요속성

속성	기능
src	Source- 재생할 파일을 표시
autoplay	페이지가 Load 되자마자 사운드를 재생할지를 지정
controls	플레이어를 표시
loop	audio 를 반복 재생할 횟수지정
preload	Page 가 열리면 audio 를 미리 Load 합니다

```
<audio src="아름다운세상.mp3" controls ></audio>
```

➔플레이어가 표시

### ➤ 동영상 재생을 위한 <video>엘리먼트

#### •기본사용방법

```
<video src="sample.mp4" loop="loop"> </video>
```

#### ▪주요속성

속성	설명
controls	재생 Control 표시.
autoplay	동영상이 Loading 되면 곧바로 재생되도록 함
autobuffer	이 속성을 사용하게 되면 사용하기 전부터 다운로드가 진행됩니다. 사용자가 재생할 때쯤이면 동영상이 어느 정도 다운로드가 된 상태일 것입니다.
poster	동영상이 Download 중이거나 Buffering 중에 나타낼 이미지를 지정합니다.
loop	동영상을 반복 재생
width	동영상의 너비지정
height	동영상의 높이지정

### ➤ <canvas>element 개요

- 단어에서 느낄 수 있듯이 도형, 직선 등의 그림을 그릴 수 있으며 사진등을 편집할 수 도 있음

#### 1. 기본 사용법

```
<canvas id="vas" width="200" height="300"></canvas>
```

- Id - 한 페이지에 여러 개의 캔버스를 만들 수 있으므로 각각을 구분짓기 위해 id속성을 지정
- Width,height속성을 지정하여 캔버스 크기 지정 - 지정하지 않으면 기본적으로 width=300, height=150 크기의 캔버스 지정

- 웹 브라우저가 <canvas>를 제공하지 않는 경우

```
<canvas id="vas" width="200" height="300">  
  
</canvas>
```

```
<BODY>  
<canvas id="a1" width="300" height="300"></canvas>  
  
</BODY>
```



실제로는 브라우저상에 아무것도 나타나지 않음  
➔ Stylesheet에서 border와 테두리 색상 지정하여 표시된 결과



## ➤ 자바스크립트의 접근

➤ <canvas>는 앞서 설명대로 작업공간을 의미하며 실제로 도형을 그리는 작업은 자바스크립트가 담당

1단계 : `var a1=document.getElementById('a1');`

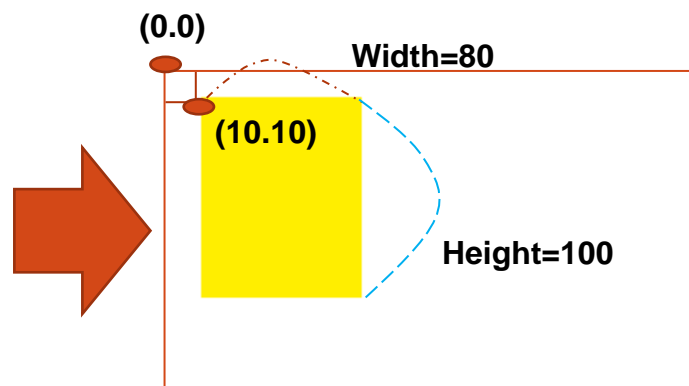
2단계 : `var draw2d = a1.getContext('2d');`

1. Id를 검색하여 객체를 불러오는 `getElementById` 메소드를 이용하여 a1 객체를 불러와서 a1 변수에 저장
2. <canvas> 엘리먼트는 도형을 그리는 데 필요한 `getContext` dom 메소드를 갖고 있으므로 "2D" 컨텍스트에 접근

### • 기본적인 스크립트 작업

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="UTF-8">
    <title>HTML 5 Canvas sample</title>
    <script type="text/javascript">
      function aa() {
        var canvas1=document.getElementById('Sample');
        var c=canvas1.getContext('2d');
        c.fillStyle='#FFee00';
        c.fillRect(10,10,80,100);
      }
    </script>
  </head>
  <body onload="aa()">
    <canvas id="Sample" width="300" height="300"></canvas>

  </body>
</html>
```



## ➤ 도형그리기 - 사각형작성

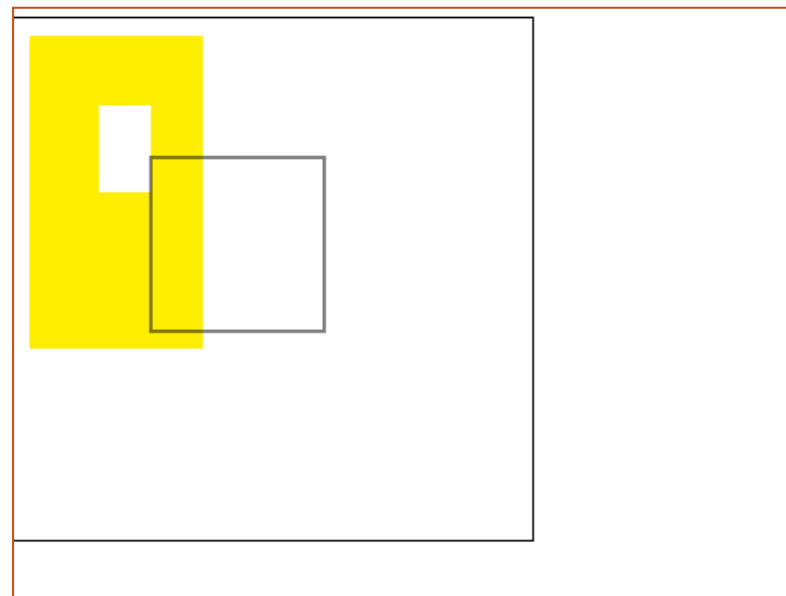
### 1. 주요 메소드

메소드	설명
fillRect(x,y,width,height)	Xy좌표를 시작점으로 하여 너비와 높이에 해당하는 색상이 채워진 사각형 그림
strokeRect(x,y,width,height)	Xy좌표를 시작점으로 하여 너비와 높이에 해당하는 비어있는 사각형 그림
clearRect(x,y,width,height)	X,y좌표를 시작점으로 하여 너비와 높이에 해당하는 사각형영역을 지우고 투명하게 함

```

<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Canvas sample</title>
    <style type="text/css">
      #sample { border:1px solid #000000;}
    </style>
  </head>
  <body>
    <canvas id="sample" width="300" height="300"></canvas>
    <script type="text/javascript">
      var canvas1=document.getElementById('sample');
      var c=canvas1.getContext('2d');
      c.fillStyle='#FFee00';
      c.fillRect(10,10,100,180);
      c.clearRect(50,50,30,50);
      c.strokeRect(80,80,100,100);
    </script>
  </body>
</html>

```



## ➤ 도형- 직선그리기

메소드	설명
stroke()	도형의 테두리를 그림
fill()	도형에 색을 채움
moveTo(x,y)	X,y좌표 위치로 시작점을 이동
lineTo(x,y)	X,y위치까지 직선을 그림
beginPath()	경로 시작
closePath()	경로를 종료

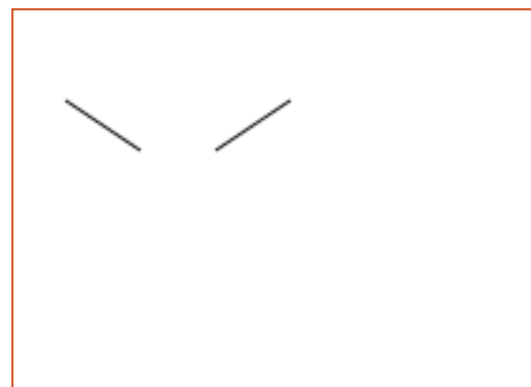
```

<body>
  <canvas id="sample" width="300" height="300"></canvas>
  <script type="text/javascript">

    var canvas1=document.getElementById('sample');
    var c=canvas1.getContext('2d');

    c.beginPath();
    c.moveTo(30,30);
    c.lineTo(60, 50);
    c.moveTo(120,30);
    c.lineTo(90, 50);
    c.stroke();
  </script>

```



### ➤ 도형- 원그리기

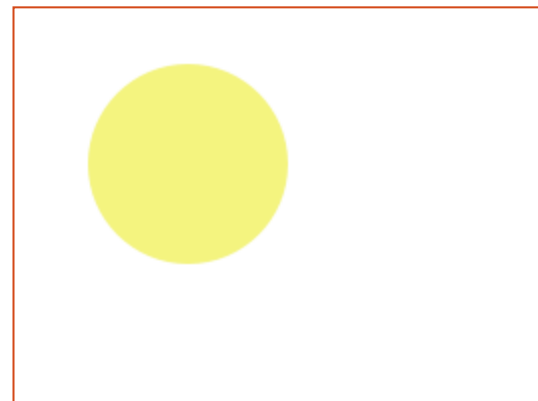
`arc(x,y,반지름,호의 시작점각도, 호의 끝점각도,anticlockwise)`

설명 - x,y좌표에서 반지름 원을 그리는데 몇 도에서 몇 도 까지 원을 그림

`anticlockwise="true"` 반시계 방향으로 그림

`anticlockwise="false"` 는 시계방향으로 그림

```
<canvas id="sample" width="300" height="300"></canvas>
<script type="text/javascript">
  var canvas1=document.getElementById('sample');
  var c=canvas1.getContext('2d');
  c.beginPath();
  c.arc(100,100,50,0,Math.PI*2,true);
  c.fillStyle="rgba(233,233,0,0.5)";
  c.fill();
  c.closePath();
</script>
```



### ➤ 캔버스에 글자넣기

- 캔버스에 텍스트를 넣는데 정렬이나 그림자 효과를 넣어 보다 세련된 효과를 표시할 수 있음.

#### ◆ 주요속성

속성/메소드	설명
fillText(텍스트,x,y,[드로잉최대너비])	캔버스에 표시할 텍스트를 xy좌표부터 시작하여 그림, 텍스트는 색상으로 채워지게 됨
strokeText(텍스트,x,y,[드로잉최대너비])	fillText 메소드와 같으나 다만,, 텍스트의 테두리만 색상으로 표시됨
font	글자크기와 글꼴을 지정
textAlign	텍스트 표시할 때 캔버스내 정렬 방법(left,right,center)
shadowOffsetX(Y)=값	객체로부터 그림자까지의 X(Y)거리지정
shadowColor=color지정	그림자 색상 지정

```

c.font = "30px tahoma";
c.fillStyle = "rgba(255,0,0,0.7)";
c.fillText("html5시작", 150, 30);
c.textAlign = "right";
c.fillText("웹 표준 종결자", 200, 70);
</script>

```

html5시작  
웹 표준 종결자

##### ➤ 캔버스에 이미지표시

- 캔버스에 이미지를 표시할 때는 자바스크립트에서 이미지 객체를 생성한 후 그 객체에 이미지를 할당하는 방식을 사용

**drawImage(image,x,y,[width],[height])**

캔버스에 이미지 객체를 xy좌표위치에서부터 표시(width,height 값 기술하여 이미지 크기를 조절)

```
<script type="text/javascript">
  function drawing() {
    var canvas1=document.getElementById('sample');
    var c=canvas1.getContext('2d');
    var img = new Image();
    img.src='images/back_2.gif';
    c.drawImage(img, 10, 10);
  };
</script>
</head>
<body onload="drawing()" >
  <canvas id="sample" width="300" height="300"></canvas>
```

