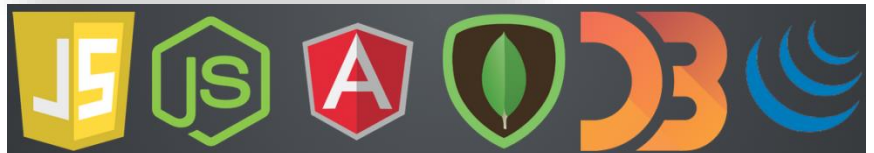
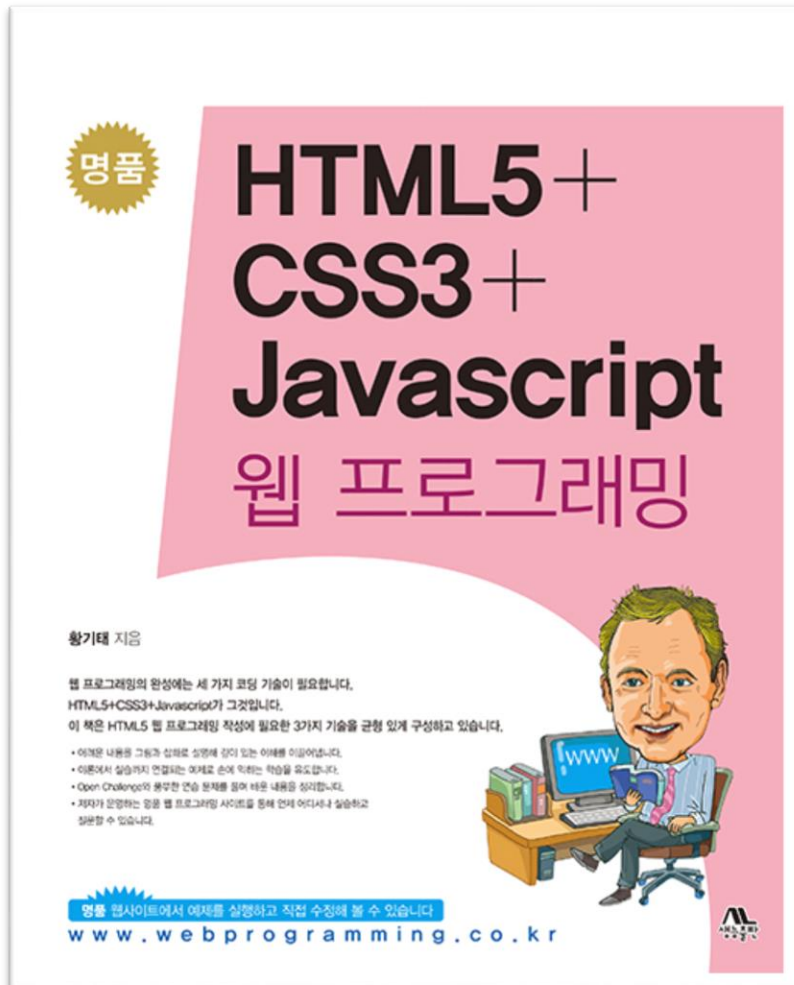


# Mobile Simulation

2017-2

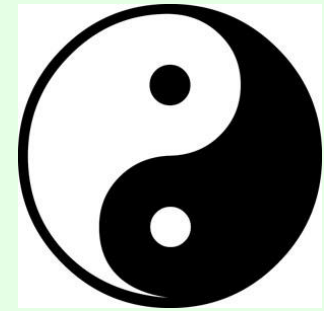
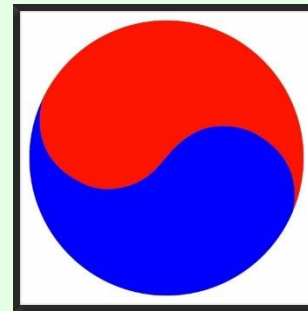


# Weekly plan (HTML5, 1<sup>st</sup> semester 2017)

- **wk01 : Introduction to curriculum & current state of HTML5**
- **wk02 : Making HTML5 documents**
- **wk03 : Table, iframe and media**
- **wk04 : Semantic tag and Form**
- **wk05 : CSS3 I. Basic**
- **wk06 : CSS3 II. Advanced**
- **wk07 : CSS3 III. Animation**
- **wk08 : Mid-term Exam.**
- **wk09 : Javascript : Data types & operators**
- **wk10 : Javascript : Loop & functions**
- **wk11 : Javascript : Core objects**
- **wk12 : Javascript : DOM**
- **wk13 : Javascript : Event handling I**
- **wk14 : Javascript : Event handling II**
- **wk15 : Final exam.**

# Weekly plan (Mobile Simulation, 2<sup>nd</sup> semester 2017)

- **wk01 : Introduction to curriculum & current state**
- **wk02 : Browser Object Model (BOM), installing Brackets editor**
- **wk03 : Canvas graphics I. Basic**
- **wk04 :**
- **wk05 :**
- **wk06 :**
- **wk07 :**
- **wk08 : Mid-term Exam.**
- **wk09 :**
- **wk10 :**
- **wk11 :**
- **wk12 :**
- **wk13 :**
- **wk14 :**
- **wk15 : Final exam.**



# 과제01. msnn\_rpt01.zip

4

## [실습과제01] “사갈의 눈 내리는 마을” 완성

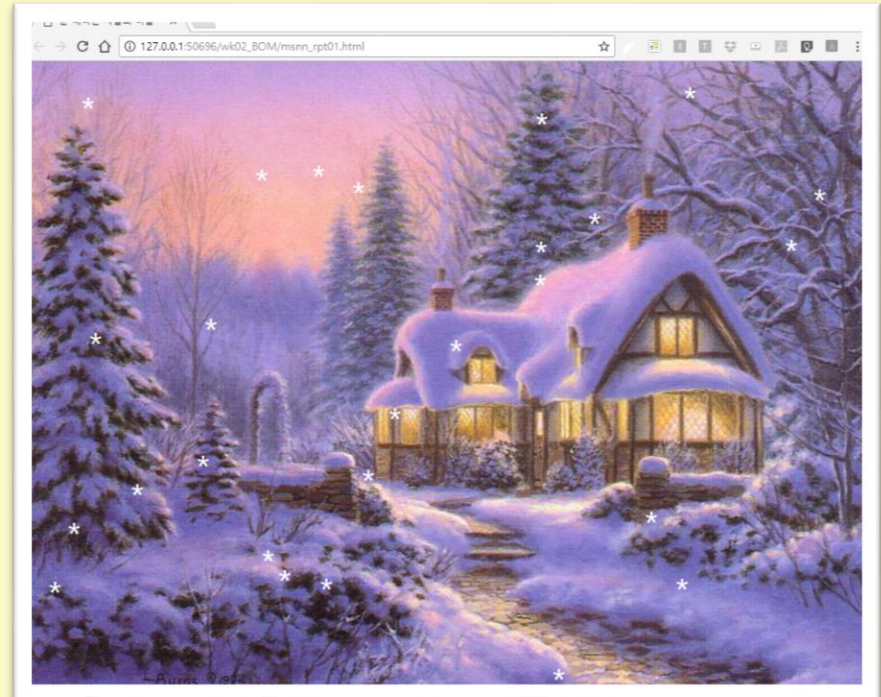
1. [교재 461쪽 Open challenge-10]  
window 객체의 속성을 이용해서 눈들이 땅에 떨어지면 새 눈들이 내리는 javascript code를 완성하시오.

파일명: msnn\_rpt01.html

가점: Javascript 프로그래밍 응용 능력.

[제출파일] msnn\_rpt01.zip

msnn\_rpt01.html 과 media folder를  
msnn\_rpt01.zip 으로 압축해서 제출



# 과제01. msnn\_rpt01.zip: code hint-1

5

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>사갈의 눈 내리는 마을</title>
<style>
body {
    background-image : url("media/snow.jpg");
    background-size : 100%;
    background-repeat : no-repeat;
}
div.snow {
    position : absolute;
    font-size : 40px;
    color : white;
    padding : 0px;
    margin : 0px;
}
</style>
```

# 과제01. msnn\_rpt01.zip: code hint-2

초기에 만든 30개의 눈송이들이 땅에 떨어질 때마다 새로운 눈송이의 위치를 새로 정해서 눈이 계속 내리는 시뮬레이션

6

```
<script>
var snow = new Array(30);    // 눈송이 30개에 대한 배열
var x = new Array(30);       // 눈송이 각각의 x좌표
var y = new Array(30);       // 눈송이 각각의 y좌표
var speedOfFall = new Array(30); // 눈송이의 낙하하는 속도
var speedOfwind = new Array(30); // 떨어지며 풍속의 영향을 받는 정도
var step = new Array(30);    // 풍속영향력의 변화도

function fall() {
    for(i=0; i<30; i++){
        //눈송이의 다음 y좌표
        y[i] += speedOfFall[i]
        //눈송이의 다음 x좌표, 코사인함수곡선을 이용
        x[i] += Math.cos(speedOfwind[i]);

        //땅에 닿은 눈송이에 대해 새로운 눈송이를 생성
        if(y[i] >= window.innerHeight-60) {
            x[i] = Math.floor(Math.random()*window.innerWidth);
            y[i] = 0;
            speedOfFall[i] = Math.random()*2+2;
        }

        //브라우저의 크기를 넘어가는 눈송이에 대한 처리
        if(x[i] >= window.innerWidth-50) x[i] = window.innerWidth - 50;
        else if(x[i] < 0) x[i] = 50;

        snow[i].style.top = y[i] + "px";
        snow[i].style.left = x[i] + "px";

        speedOfwind[i] += step[i];
    }
    setTimeout("fall()",50);
}

</script>
</head>
```

# 과제01. msnn\_rpt01.zip: code hint-3

7

30개의 눈송이를 생성 – random()로 위치 및 속도 변화 설정

```
<body onload="fall()">
<script>
makeSnow(); // 문서가 로딩되는 단계에서 눈(*)을 출력하는 <div> 객체 생성

function makeSnow() {
    // window.innerHeight는 현재 윈도우의 브라우저 영역의 높이
    // window.innerWidth는 현재 윈도우의 브라우저 영역의 폭

    for(var i=0; i<30; i++) { // 30 개의 눈송이 생성
        x[i] = Math.floor(Math.random()*window.innerWidth-10);    //눈송이의 x좌표 지정
        y[i] = Math.floor(Math.random()*window.innerHeight+10);    //최초 눈송이의 y좌표 지정
        speedOfFall[i] = Math.random()*2+2;    //눈송이의 낙하속도 지정
        speedOfwind[i] = 1;    //최초 바람의 영향
        step[i] = Math.random()*0.1+0.05;    // 눈송이가 받는 바람의 영향의 변화도

        // <div id="snowobj012...">*</div> 생성
        var divtag = "<div class='snow' id=snowobj" + i + " style='top:" + x[i] + "px;left:" + y[i] +
"px'>*</div>";

        document.write(divtag);

        // <div> DOM 객체 기억
        snow[i] = document.getElementById("snowobj"+i)
    }
}
</script>
</body>
</html>
```







# 강의 목표

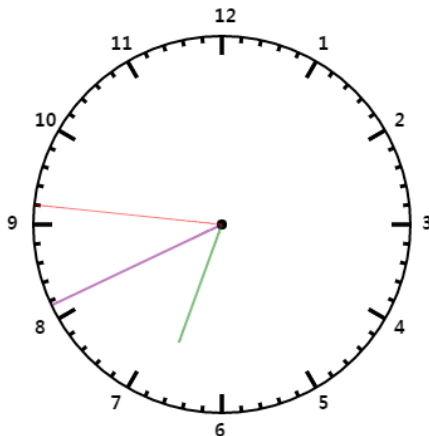
1. HTML5에서 웹 페이지에 그래픽으로 그림을 그리는 방법을 안다.
2. <canvas> 태그로 웹 페이지에 캔버스를 만들 수 있다.
3. 캔버스에 직선, 사각형, 원호 등의 도형을 그릴 수 있다.
4. 경로(path)에 대한 개념을 이해하고 경로를 이용한 그리기를 할 수 있다.
5. 캔버스에 사각형, 원호 등의 닫힌 도형의 내부를 칠할 수 있다.
6. 캔버스에 텍스트를 출력할 수 있다.
7. 캔버스에 이미지를 출력할 수 있다.
8. 캔버스에 마우스 드래깅으로 그림을 그리는 응용프로그램을 작성할 수 있다.
9. 그래디언트 효과를 추가할 수 있다.

# HTML5와 캔버스 그래픽스

10



The `<canvas>` element is used to draw graphics, on the fly, on a web page.



Canvas simulation : My Game



# HTML5와 캔버스

11

- 웹 페이지에 그래픽을 출력하는 방법
  - ▣ 사진이나 그림 : <img> 태그 이용
  - ▣ 그래픽 : 자바 애플릿이나 플래시 등 플러그인 이용
- HTML5에서 캔버스 도입
  - ▣ 도입 배경
    - 플러그인 없이 자바스크립트 코드로 웹 페이지에 자유롭게 그래픽
    - 모바일 단말기/PC를 포함하여 HTML5 표준 브라우저에서 작동
  - ▣ 그래픽 기능
    - 선, 원, 사각형, 곡선, 이미지, 2차원 문자
    - 이미지 합성 및 변환
  - ▣ 활용
    - 웹 페이지에 실시간 그래프, 애니메이션, 대화형 게임, 지도
  - ▣ 의미
    - 웹이 문서를 보여주는 수준을 넘어 응용 프로그램으로 진화하는 계기

# <canvas> 태그

12

## □ 웹 페이지에 캔버스 영역 만들기

```
<canvas id="캔버스 객체 id"
      style="CSS3 스타일 시트"
      width="캔버스 영역의 폭"
      height="캔버스 영역의 폭">
```

이 태그를 지원하지 않는 브라우저가 출력할 HTML 텍스트

```
</canvas>
```

- width, height : 캔버스가 만들어지는 영역의 크기(픽셀 단위)로 생략 가능. 각각 디폴트 300, 150 픽셀

- 웹 페이지에 빈 캔버스 공간 할당
- 예) 300x400 크기의 캔버스 공간 할당

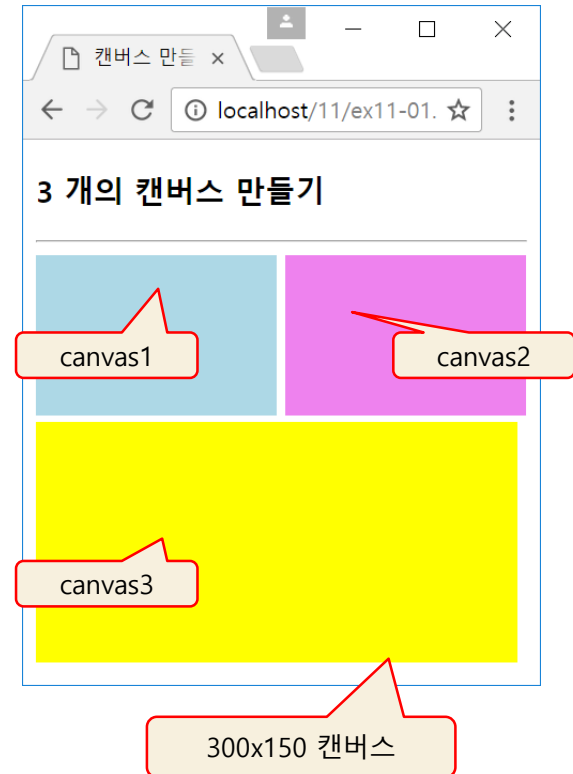
```
<canvas id="grade" width="300" height="150">
  <p>canvas가 지원되지 않네요. 죄송합니다.</p>
</canvas>
```

# 예제 11-1 캔버스 만들기

13

<canvas> 태그를 이용하여 3 개의 캔버스를 가진 HTML 페이지를 만들어라.

```
<!DOCTYPE html>
<html>
<head> <title>캔버스 만들기</title> </head>
<body>
<h3>3 개의 캔버스 만들기</h3>
<hr>
<canvas id="canvas1" width="150" height="100"
  style="background-color:lightblue"> </canvas>
<canvas id="canvas2" width="150" height="100"
  style="background-color:violet"> </canvas> <br>
<canvas id="canvas3" width="300" height="150"
  style="background-color:yellow"> </canvas>
</body>
</html>
```





# 캔버스(canvas) 객체 다루기

14

## □ 캔버스 공간 할당

```
<canvas id="myCanvas" width="300" height="150"> </canvas>
```

## □ 캔버스 객체 찾기

```
var canvas = document.getElementById("myCanvas");
```

## □ 캔버스 컨텍스트 얻어내기 : 캔버스에 그림 그리는 도구(컨텍스트) 얻어내기

```
var context = canvas.getContext("2d");
```

## □ 캔버스에 사각형 그리기

```
context.rect(60, 60, 50, 50); // context에 (60, 60)에서 50x50 크기의 사각형 그리기
context.strokeStyle = "blue"; // 선 색을 파란 색으로 설정
context.stroke();             // context 내에 구성된 도형을 캔버스에 그린다.
```

## □ 캔버스의 크기, canvas의 width와 height 프로퍼티

```
var canvas = document.getElementById("myCanvas");
var width = canvas.width;
var height = canvas.height;
alert("캔버스는 " + width + "x" + height);
```

localhost 내용:

캔버스는 300x150

확인

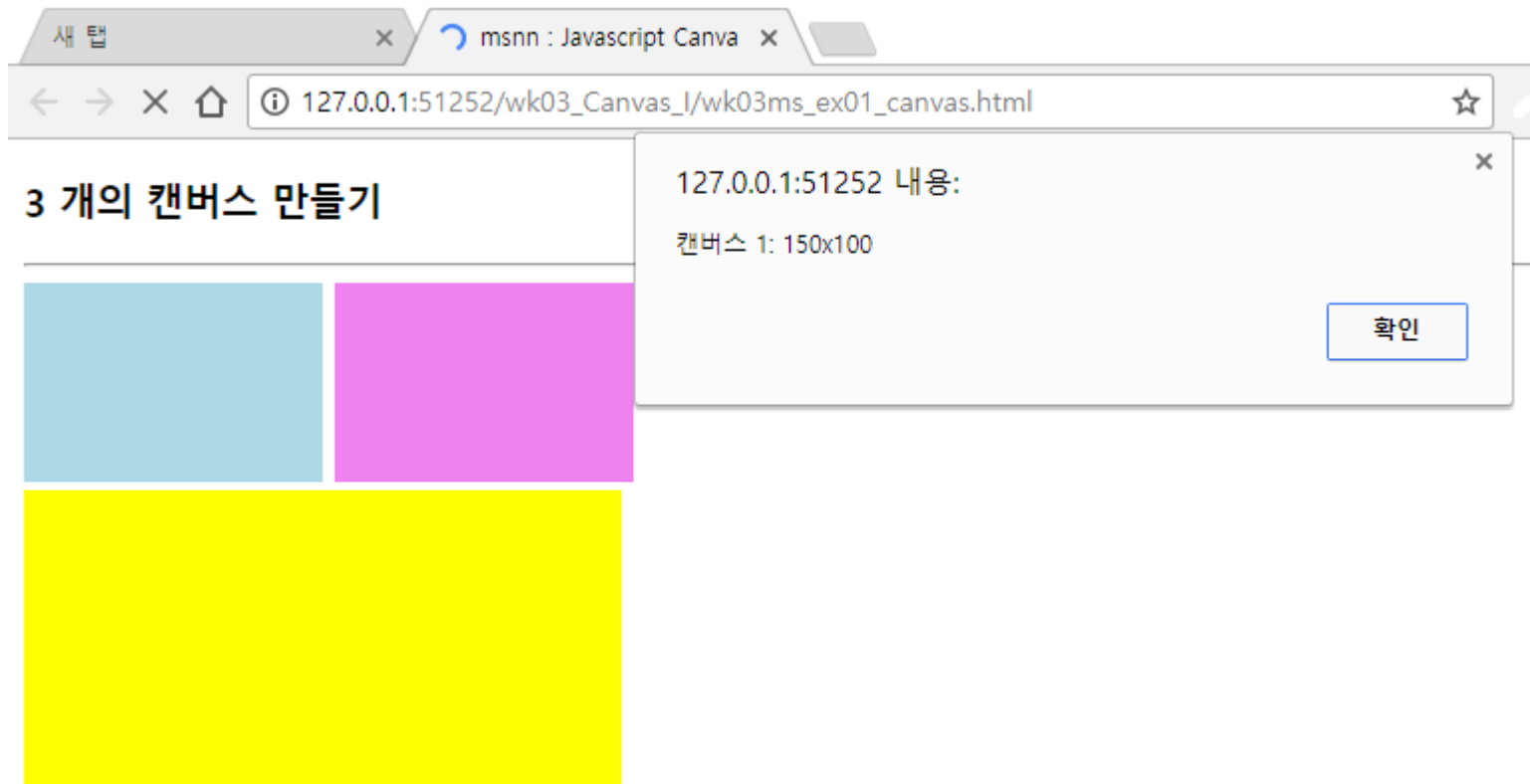
## □ 캔버스의 스타일 제어

```
canvas.style.backgroundColor = "yellowgreen";
```

# 예제 11-1 캔버스 만들기 (연습)

15

지정된 캔버스의 크기를 출력>



# 예제 11-2 캔버스 그리기 맛보기

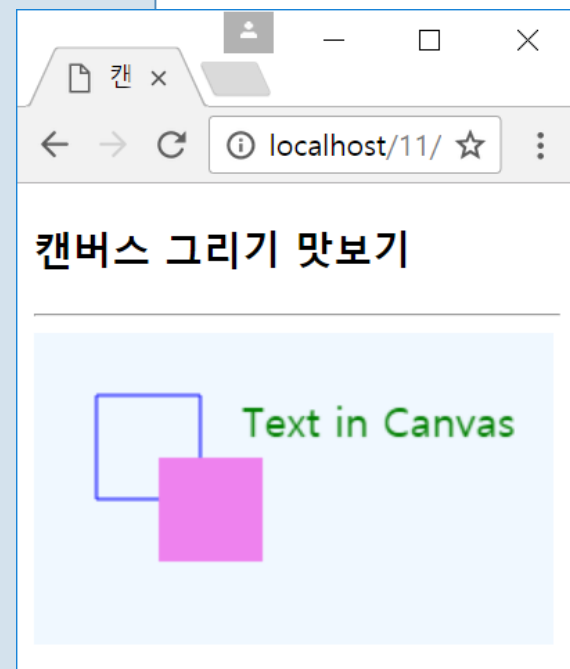
16

```
<!DOCTYPE html>
<html><head><title>캔버스 그리기 맛보기</title></head>
<body>
<h3>캔버스 그리기 맛보기</h3>
<hr>
<canvas id="myCanvas" style="background-color:aliceblue"
  width="250" height="150"></canvas>
<script>
  var canvas = document.getElementById("myCanvas");
  var context = canvas.getContext("2d");

  // 파란선으로 사각형 그리기
  context.beginPath(); // 빈 경로 만들기
  context.strokeStyle = "blue"; // 선 색 설정
  context.rect(30, 30, 50, 50); // (30,30)에서 50x50 크기 사각형을 경로에 삽입
  context.stroke(); // 경로에 있는 모든 도형의 외곽선 그리기

  // violet 색으로 채운 사각형 그리기
  context.beginPath(); // 빈 경로 만들기
  context.fillStyle = "violet"; // 채우기 색
  context.rect(60, 60, 50, 50); // (60,60)에서 50x50 크기 사각형을 경로에 삽입
  context.fill(); // 경로에 있는 모든 도형의 내부만 채워 그리기

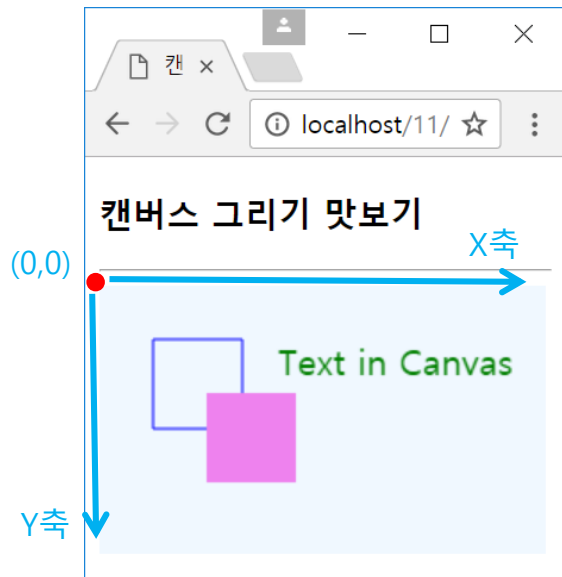
  // green 색으로 텍스트 내부만 그리기
  context.font = "20px Gothic";
  context.fillStyle = "green"; // 채우기 색
  context.fillText("Text in Canvas", 100, 50); // 텍스트를 경로에 넣지 않고 바로 그리기
</script></body></html>
```



# 캔버스 그래픽 좌표와 그래픽 기능

17

- 캔버스 그래픽은 픽셀 단위의 좌표와 크기



- 그래픽 기능
  - 도형 그리기와 칠하기
  - 글자 그리기
  - 이미지 그리기
  - 이미지 변환
  - 클리핑

캔버스

# 도형 그리기

18

- 도형 그리기
  - ▣ 캔버스가 지원하는 도형의 종류
    - 직선, 사각형, 원호
    - 외곽선 그리기, 내부색으로 채워 그리기 모두 가능
  - ▣ 도형 그리는 과정
    - 경로(path) 만들기
    - 캔버스에 경로에 담긴 도형 모두 그리기
- 경로(path) 만들기
  - ▣ 그리고자 하는 도형들을 컨텍스트 내 경로에 담는 과정
- 캔버스에 도형 그리는 순서
  - ▣ `beginPath()` - 새로운 빈 경로 만들기
  - ▣ `moveTo()`, `lineTo()`, `rect()`, `arc()` - 경로에 도형 담기
  - ▣ `Stroke()` - 경로 속의 도형을 캔버스에 그리기



# 도형 그리기 사례

19

## ▣ 경로 만들기

```
context.beginPath();           // 빈 경로 구성
```

## ▣ 시작점 설정

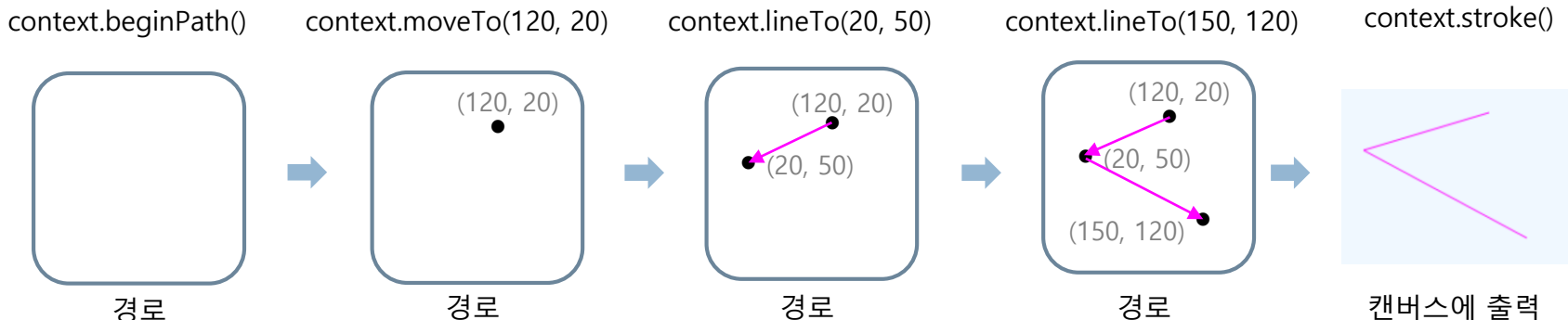
```
context.moveTo(120, 20);       // (120, 20)을 시작점으로 설정
```

## ▣ 경로에 도형 추가

```
context.lineTo(20, 50);         // (120, 20)에서 (20, 50)까지의 직선을 경로에 추가  
context.lineTo(150, 120);       // (20, 50)에서 (150, 120)까지의 직선을 경로에 추가
```

## ▣ 캔버스에 그리기

```
context.stroke();               // context의 경로 속 도형들을 캔버스에 모두 그린다.
```

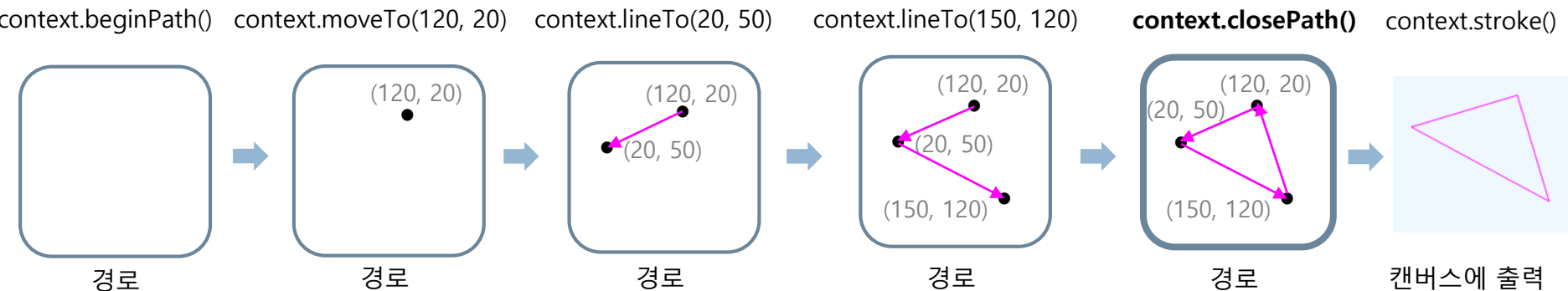


# 경로 닫기

20

## □ **closePath()**

- 경로의 마지막 점과 시작점을 연결하는 직선 추가
- 더 이상 경로에 새로운 도형을 추가할 수 없음
  - `beginPath()`를 호출하면 새로운 경로 시작



# 선 그리기와 사각형 그리기

21

## □ 선 그리는 컨텍스트 메소드

`moveTo(x, y)` 경로에 담긴 도형은 그대로 두고, 점  $(x, y)$ 를 새 시작점으로 삽입한다.  
`lineTo(x, y)` 경로의 끝 점에서  $(x, y)$ 까지 직선을 경로에 추가한다.

## □ 선그리기

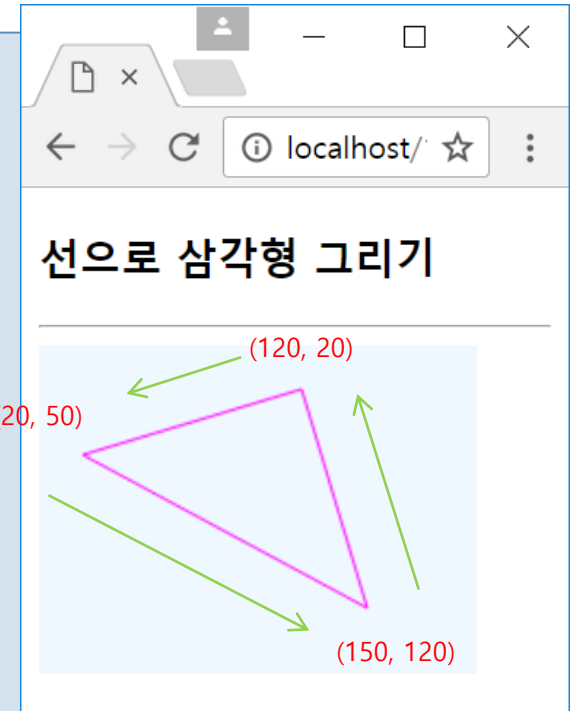
- `moveTo()`를 이용하여 시작점을 설정하고, `lineTo()`로 선을 연결해 나간다.
- `lineTo(x, y)`에 지정한  $(x, y)$ 가 끝점이 된다.

# 예제 11-3 선으로 삼각형 그리기

22

```
<!DOCTYPE html>
<html>
<head> <title> 선으로 삼각형 그리기 </title> </head>
<body>
<h3> 선으로 삼각형 그리기 </h3>
<hr>
<canvas id="myCanvas"
        style="background-color:aliceblue"
        width="200" height="150"> </canvas>
<script>
    var canvas = document.getElementById("myCanvas");
    var context = canvas.getContext("2d");

    context.beginPath(); // 빈 경로 만들기
    context.moveTo(120, 20); // (120, 20)을 시작점으로 설정
    context.lineTo(20, 50); // 경로에 (120, 20)에서 (20, 50)까지 직선 추가
    context.lineTo(150, 120); // 경로에 (20, 50)에서 (150, 120)까지 직선 추가
    context.lineTo(120, 20); // 경로에 (150, 120)에서 (120, 20)까지 직선 추가
    context.strokeStyle="magenta"; // 선의 색
    context.stroke(); // 경로를 캔버스에 그린다
</script>
</body>
</html>
```



# 원호 그리기: arc()

23

## □ 원호를 그리는 arc() 메소드

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

- `x, y, radius` : `(x, y)`는 원호의 중심이고 `radius`는 반지름
- `startAngle` : 원호의 시작 각도. 3시를 기점으로 시계방향으로 각도 계산
- `endAngle` : 원호의 끝 각도. 3시를 기점으로 시계방향으로 각도 계산
- `anticlockwise` : `true`이면 반시계방향, `false`이면 시계방향으로 원호 그리기. 생략가능하며 디폴트는 시계 방향(`false`)

`(x, y)`를 중심으로 반지름이 `radius`이고, `startAngle`(시작 각도) 지점과 `endAngle`(끝 각도) 지점을 연결 하는 원호를 그린다. 원호는 `anticlockwise`의 값에 따라 반시계 방향이나 시계방향 중 하나로 그려진다.

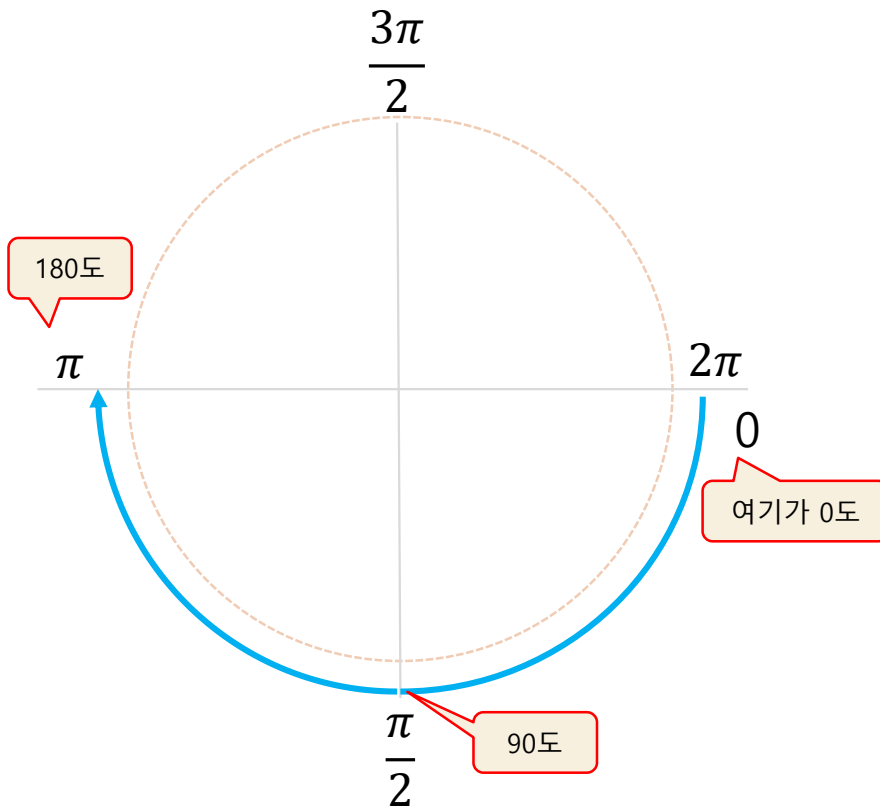
- 각도는 3시에서 0도 시작
- 각도는 원주율(`radian`)로 표현
  - $360^\circ$ 는  $2\pi$ ,  $90^\circ$ 는  $\pi/2$ 이고,  $180^\circ$ 는  $\pi$
  - $270^\circ \rightarrow 3\pi/2$  는 자바스크립트 코드 `1.5*Math.PI` 로 표현

원주률 `r = n*Math.PI/180`, 여기서 `n`은 각도(`0~360`)

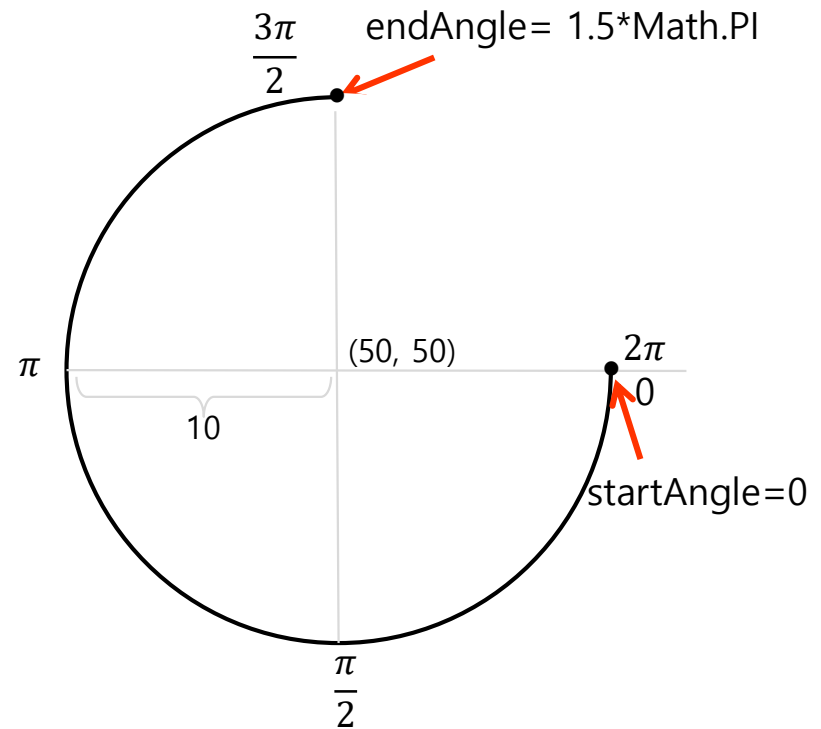


# startAngle(시작 각도)와 endAngle(끝 각도)

24



(a) 각도는 3시에서 시계방향으로 계산

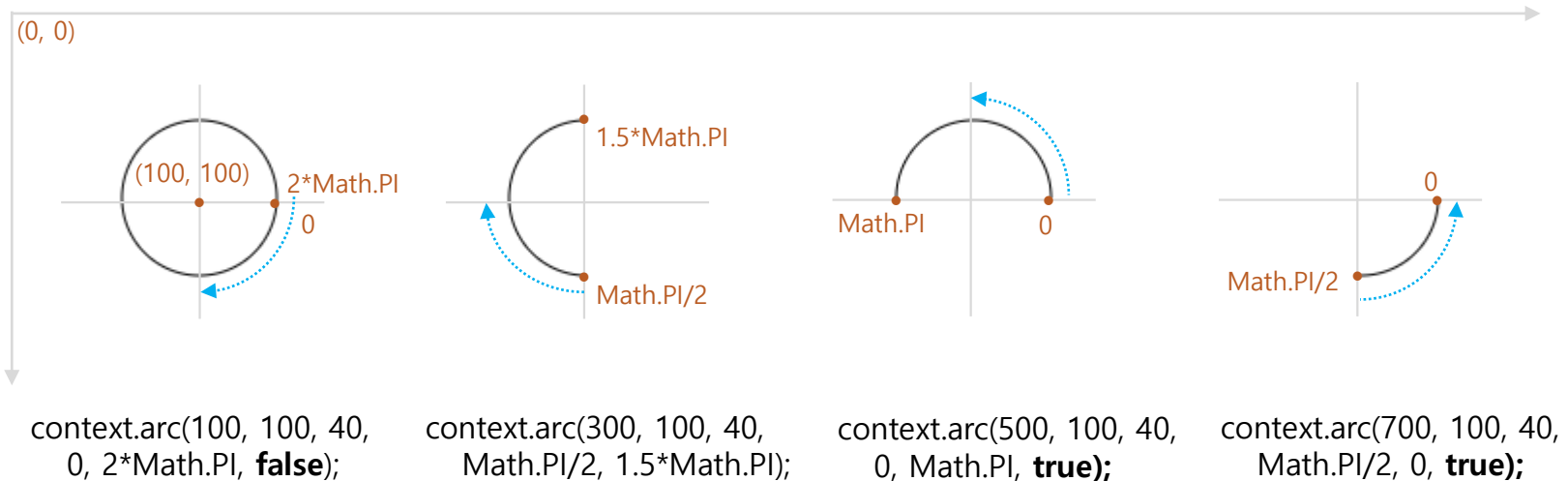


```
context.arc(50, 50, 10, 0, 1.5*Math.PI, false);  
context.stroke(); // 원호를 그린다.
```

(b) 원호 그리기 사례

# arc() 메소드로 그린 원호 사례

25



\* true 이면, 시계방향, false이면 반시계 방향. 생략되면 true

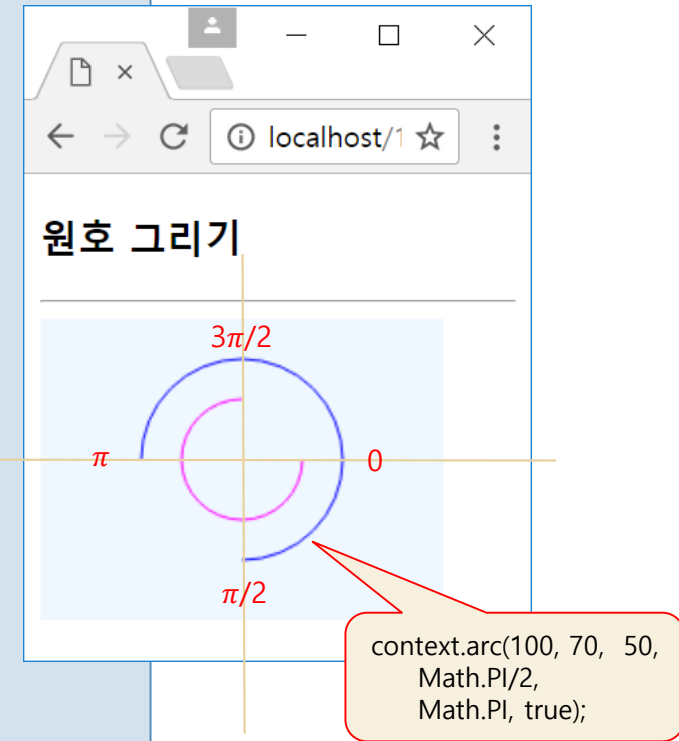
# 예제 11-4 원호 그리기

26

```
<!DOCTYPE html>
<html>
<head> <title> 원호 그리기 </title> </head>
<body>
<h3>원호 그리기</h3>
<hr>
<canvas id="myCanvas" style="background-color:aliceblue"
  width="200" height="150"> </canvas>
<script>
  var canvas = document.getElementById("myCanvas");
  var context = canvas.getContext("2d");

  context.beginPath(); // 빈 경로 구성
  context.strokeStyle="magenta";
  context.arc(100, 70, 30, 0, 1.5*Math.PI, false); // 시계 방향
  context.stroke(); // 경로에 있는 원호를 그린다

  context.beginPath(); // 여기서 반드시 빈 경로로 시작해야 함
  context.strokeStyle="blue";
  context.arc(100, 70, 50, Math.PI/2, Math.PI, true); // 반시계 방향
  context.stroke(); // 경로에 있는 한 개의 원호를 캔버스에 그린다
</script>
</body>
</html>
```



# 사각형 그리기: rect()

27

## □ 사각형 그리는 컨텍스트 메소드

`rect(x, y, w, h)`       $(x, y)$ 에서  $w \times h$  크기의 사각형을 경로에 삽입한다.  
`strokeRect(x, y, w, h)`       $(x, y)$ 에서  $w \times h$  크기의 사각형을 경로에 삽입하지 않고 캔버스에 직접 그린다.

### ▣ 예) (10, 10)에서 100x100 크기의 사각형 그리기

```
context.rect(10, 10, 100, 100); // (10, 10)에서 100x100 크기의 사각형을 경로에 추가
context.stroke();                // context에 구성된 사각형을 캔버스에 그린다.
```

### ▣ 예) strokeRect() 메소드 이용 – 캔버스에 바로 그리기

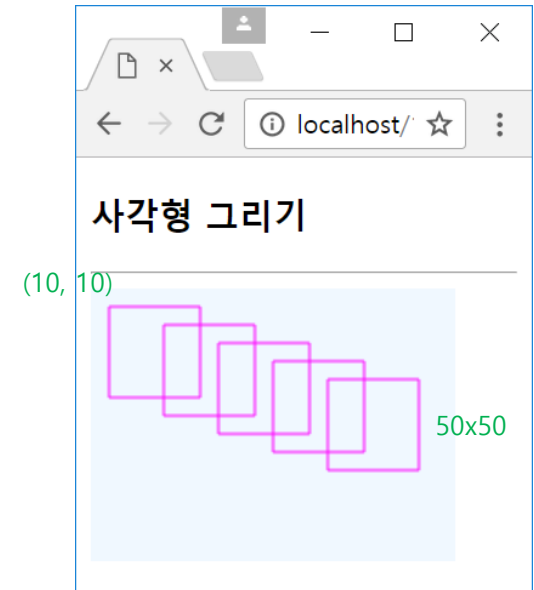
```
context.strokeRect(10, 10, 100, 100); // (10, 10)에서 100x100 크기의 사각형 그리기
```

# 예제 11-5 사각형 그리기

28

```
<!DOCTYPE html>
<html>
<head> <title>사각형 그리기</title> </head>
<body>
<h3>사각형 그리기</h3>
<hr>
<canvas id="myCanvas" style="background-color:aliceblue"
  width="200" height="150"> </canvas>
<script>
  var canvas = document.getElementById("myCanvas");
  var context = canvas.getContext("2d");

  context.beginPath(); // 빈 경로 구성
  for(var i=0; i<5; i++) {
    context.rect(10+i*30,10+i*10, 50,50);
  }
  context.strokeStyle="magenta"; // 선의 색
  context.stroke(); // 사각형을 캔버스에 그린다
</script>
</body>
</html>
```





# 캔버스 지우기

29

- 캔버스에 그리진 그래픽을 모두 지울 때

```
context.clearRect(0, 0, canvas.width, canvas.height);
```

- ▣ 이 코드는 경로를 지우지는 못함

- 캔버스의 그래픽과 경로(in context)를 모두 지울 때

```
context.clearRect(0, 0, canvas.width, canvas.height);  
context.beginPath();
```

# 도형 꾸미기

30

## □ 도형 꾸미기

- 선, 원호, 사각형, 글자 등의 색이나 굵기 조절
- 선 색 : **strokeStyle** 프로퍼티 이용

```
context.strokeStyle = "blue";  
context.strokeStyle = "#0000FF";  
context.strokeStyle = "rgb(0, 0, 255)";
```

- 채우기 색 : **fillStyle** 프로퍼티

```
context.fillStyle = "red";
```

- 선 굵기 : **lineWidth** 프로퍼티

```
context.lineWidth = 20;           // 선 굵기를 20픽셀로 지정
```

# 예제 11-6 선의 색과 굵기

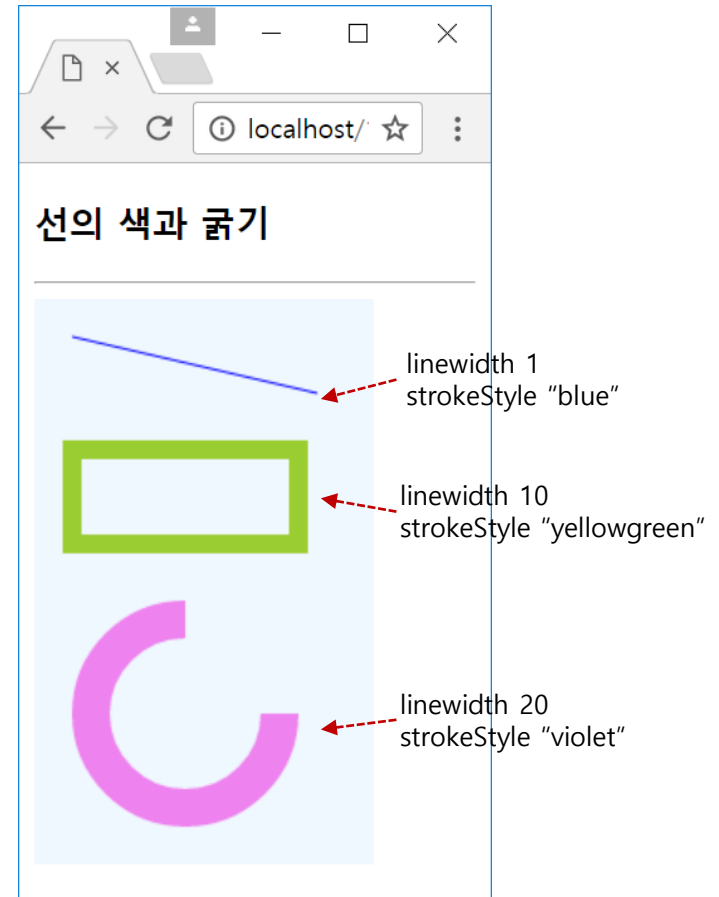
31

```
<!DOCTYPE html>
<html>
<head><title>선의 색과 굵기</title></head>
<body>
<h3>선의 색과 굵기</h3>
<hr>
<canvas id="myCanvas" style="background-color:aliceblue"
        width="180" height="300"></canvas>
<script>
  var canvas = document.getElementById("myCanvas");
  var context = canvas.getContext("2d");

  // 1 픽셀의 blue 직선 그리기
  context.beginPath();
  context.moveTo(20, 20);
  context.lineTo(150, 50);
  context.strokeStyle = "blue";
  context.stroke();

  // 10 픽셀 yellowgreen 사각형 그리기
  context.beginPath();
  context.rect(20, 80, 120, 50);
  context.lineWidth = 10; // 선 굵기 10픽셀
  context.strokeStyle = "yellowgreen"; // 선 색
  context.stroke();

  // 20 픽셀의 violet 색 원호 그리기
  context.beginPath();
  context.arc(80, 220, 50, 0, 1.5*Math.PI, false);
  context.lineWidth = 20; // 선 굵기 20픽셀
  context.strokeStyle = "violet"; // 선 색
  context.stroke();
</script>
</body>
</html>
```



# 칠하기

32

- 도형 내부를 칠하는 기능
  - ▣ 원호 내부, 사각형 내부, 텍스트 내부 칠하기
- 칠하는 여러 방법
  - ▣ **fillStyle** 프로퍼티 : 원호나 사각형, 텍스트의 내부를 칠할 색 지정

```
context.fillStyle = "violet";
```

- ▣ 캔버스에 바로 칠하기 : fillRect()
  - fillStyle의 색으로 사각형 내부 채우기

```
context.fillStyle = "violet";  
context.fillRect(20, 20, 100, 100);
```



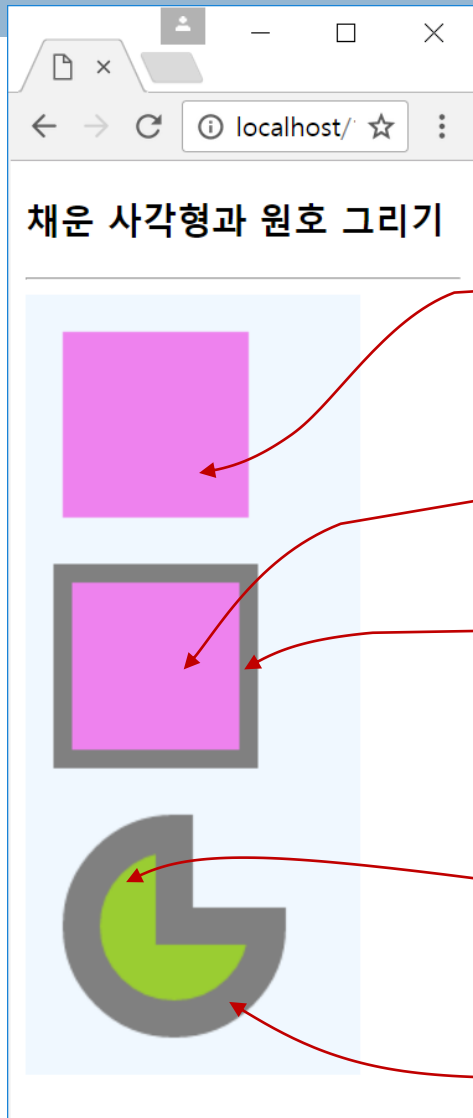
- ▣ 경로에 닫힌 도형 칠하기 : fill()
  - fill()은 사각형과 원호 모두 적용

```
context.fillStyle = "violet";  
context.rect(20, 20, 100, 100); // 경로에 사각형 삽입  
context.fill(); // 경로내 도형 내부 칠하기  
context.strokeStyle = "gray";  
context.lineWidth = 10;  
context.stroke(); // 경로내 도형 그리기(외곽선)
```



# 예제 11-7 칠하기

33



```
<!DOCTYPE html>
<html>
<head> <title>채운 사각형과 원호 그리기</title> </head>
<body>
<h3>채운 사각형과 원호 그리기</h3>
<hr>
<canvas id="myCanvas" style="background-color:aliceblue"
        width="180" height="420"> </canvas>
<script>
    var canvas = document.getElementById("myCanvas");
    var context = canvas.getContext("2d");

    // fillRect()로 외곽선 없이 색으로 채운 사각형 그리기
    context.fillStyle = "violet";
    context.fillRect(20, 20, 100, 100); // 채운 사각형 그리기

    // fill()로 사각형 내부 칠하기
    context.beginPath();
    context.rect(20, 150, 100, 100); // 경로에 사각형 삽입
    context.fillStyle = "violet";
    context.fill(); // 사각형 내부 칠하기

    // 사각형 외곽선 그리기
    context.strokeStyle = "gray";
    context.lineWidth = 10;
    context.stroke(); // 사각형 외곽선 그리기

    // fill()로 원호 내부 칠하기
    context.beginPath();
    context.moveTo(80, 340); // 원호의 중심을 시작점으로 설정
    context.arc(80, 340, 50, 0, 1.5*Math.PI); // 경로에 원호 삽입
    context.closePath(); // 원호의 끝점과 경로 시작점(원호중심)을
    // 연결하는 직선 자동 추가
    context.fillStyle = "yellowgreen";
    context.fill(); // 원호 내부 칠하기

    // 원호 외곽선 그리기
    context.strokeStyle = "gray";
    context.lineWidth = 20;
    context.stroke(); // 원호 외곽선 그리기
</script>
</body> </html>
```

# 텍스트 그리기

34

- 캔버스에 텍스트 그리기
  - ▣ 비트맵 이미지로 캔버스에 글자 그리기(출력)
  - ▣ 텍스트 그리기는 2가지 방법
    - 텍스트의 외곽선만 그리기 - `strokeText()`
    - 텍스트 외곽선 없이 내부를 채워 그리기 - `fillText()`

```
strokeText(text, x, y [, maxWidth])
```

```
fillText(text, x, y [, maxWidth])
```

- `text` : 출력하고자 하는 문자열 텍스트
- `x, y` : 텍스트가 출력되는 시작 점 (`x, y`)
- `maxWidth` : 텍스트가 출력되는 최대 폭. `text`가 이 값보다 크면 자동으로 다른 폰트로 대체됨

`strokeText()`는 `strokeStyle` 색으로 (`x,y`) 위치에 `text`의 외곽선만 그리며, `fillText()`는 외곽선없이 텍스트 내부를 `fillStyle` 색으로 칠한다.

# Quiz ??

35

## Definition and Usage

The `textBaseline` property sets or returns the current text baseline used when drawing text.

The illustration below demonstrates the various baselines supported by the `textBaseline` attribute:



**Note:** The `fillText()` and `strokeText()` methods will use the specified `textBaseline` value when positioning the text on the canvas.

<b>Default value:</b>	alphabetic
<b>JavaScript syntax:</b>	<code>context.textBaseline="alphabetic top hanging middle ideographic bottom";</code>

[1] What is the default value of `textBaseline` property?

- A. top      B. hanging      C. middle      D. alphabetic      E. ideographic

Bottom Middle Alphabetic  
Top Hanging

# 텍스트 그리기 사례

36

- ▣ 폰트 설정 : font 프로퍼티 이용
  - 예) context.font = "50px arial";
- ▣ 정렬 설정 : textAlign 프로퍼티 이용
  - 예) context.textAlign = "center";

## ▣ 텍스트 외곽선 그리기

```
context.font = "50px arial";  
context.strokeStyle = "blue";  
context.lineWidth = 1;  
context.strokeText("Javascript", 30, 100);
```

Javascript

## ▣ 텍스트 채워 그리기

```
context.font = "50px arial";  
context.fillStyle = "green";  
context.fillText("Javascript", 30, 200);
```

Javascript



# 예제 11-8 텍스트 그리기

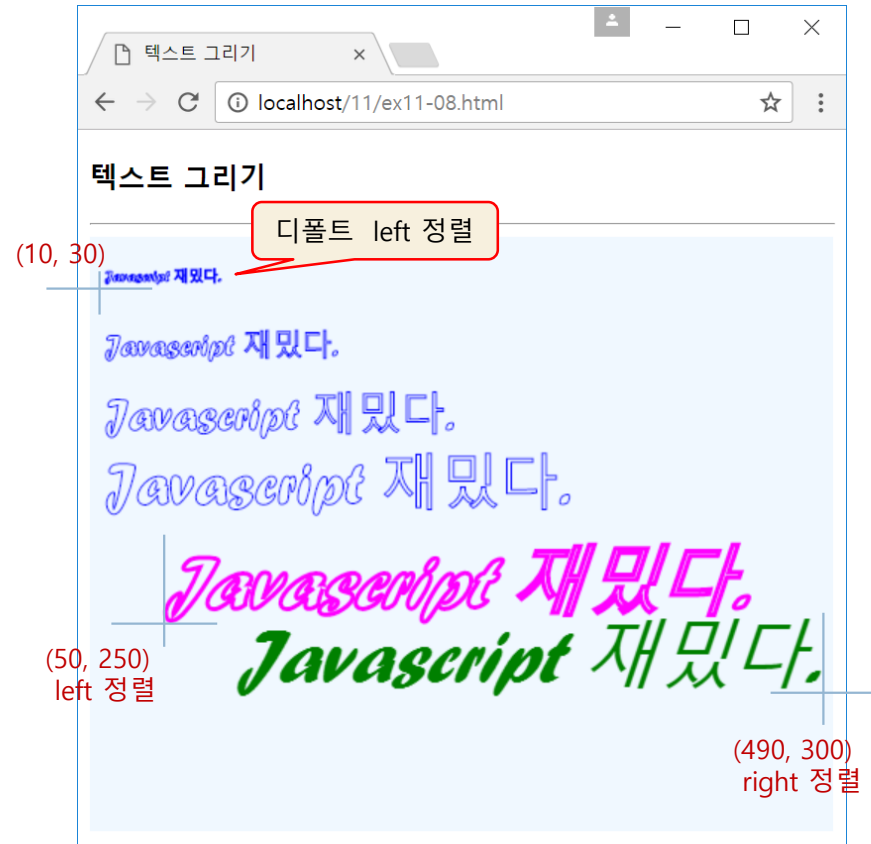
37

```
<!DOCTYPE html>
<html>
<head><title>텍스트 그리기</title></head>
<body>
<h3>텍스트 그리기</h3>
<hr>
<canvas id="myCanvas" style="background-color:beige"
  width="500" height="400"></canvas>
<script>
  var canvas = document.getElementById("myCanvas");
  var context = canvas.getContext("2d");
  context.strokeStyle = "blue";

  // font 프로퍼티로 다양한 크기와 서체 활용
  for(var i=0; i<4; i++) {
    context.font = (10 + i*10) + "px forte";
    context.strokeText("Javascript 재밌다.", 10, 30+i*50);
  }

  // 텍스트 외곽선만 그리기
  context.font = "italic 50px forte";
  context.strokeStyle = "magenta";
  context.lineWidth = 3;
  context.textAlign = "left";
  context.strokeText("Javascript 재밌다.", 50, 250);

  // 텍스트 채워 그리기
  context.fillStyle = "green";
  context.textAlign = "right";
  context.fillText("Javascript 재밌다.", 490, 300);
</script>
</body>
</html>
```



# 이미지 그리기

38

## □ 이미지 객체 생성

- ▣ 파일에서 읽은 이미지를 담을 객체 필요

```
var img = new Image();
```

## □ 이미지 그리는 과정

- ▣ 이미지 로딩과 onload

- 이미지 파일의 로딩이 완료된 후 이미지를 그린다.

```
1 img.onload = function () {  
2     ... // img 객체에 로드된 이미지를 그린다.  
3 }  
4 img.src = "test.png"; // img 객체에 test.png 파일로부터 이미지의 로딩 시작
```

- ▣ 이미지 그리기

- 컨텍스트 객체의 `drawImage()` 메소드 이용

# drawImage()로 이미지 그리기 사례

39

- (20, 20) 위치에 원본 크기로 그리기

```
var img = new Image();
img.onload = function () {
    context.drawImage(img, 20, 20);
}
img.src = "test.png";
```

- (20, 20) 위치에 100×200 크기로 그리기

```
var img = new Image();
img.onload = function () {
    context.drawImage(img, 20, 20, 100, 200);
}
img.src = "test.png";
```

- 캔버스에 꽉 차게 이미지 그리기

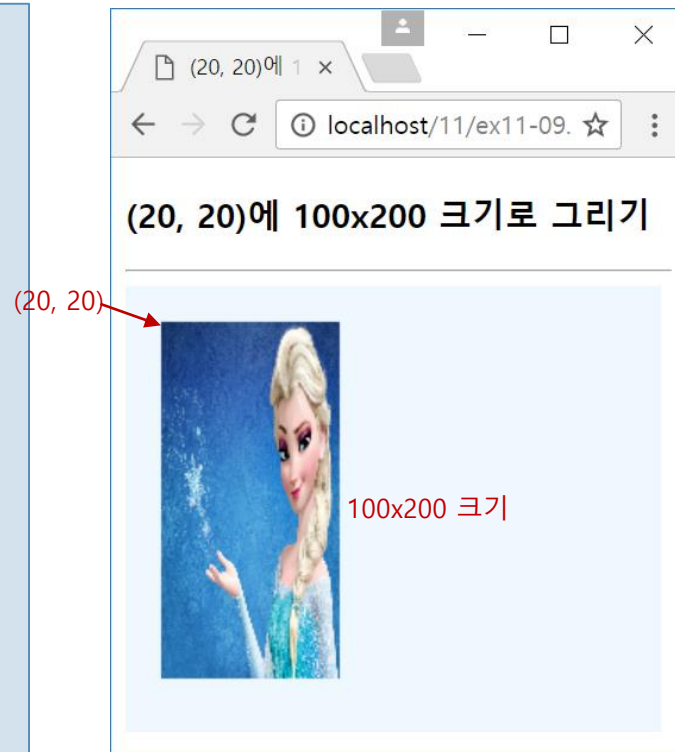
```
var img = new Image();
img.onload = function () {
    context.drawImage(img, 0, 0, canvas.width, canvas.height);
}
img.src = "test.png";
```

# 예제 11-9 캔버스의 (20, 20)에 100x200 크기로 변형하여 그리기

40

```
<!DOCTYPE html>
<html>
<head>
<title>(20, 20)에 100x200 크기로 그리기</title>
</head>
<body>
<h3>(20, 20)에 100x200 크기로 그리기</h3>
<hr>
<canvas id="myCanvas"
        style="background-color:aliceblue"
        width="300" height="250"> </canvas>
<script>
    var canvas = document.getElementById("myCanvas");
    var context = canvas.getContext("2d");

    var img = new Image();
    img.onload = function () {
        context.drawImage(img, 20, 20, 100, 200);
    }
    img.src = "media/elsa.png";
</script>
</body>
</html>
```

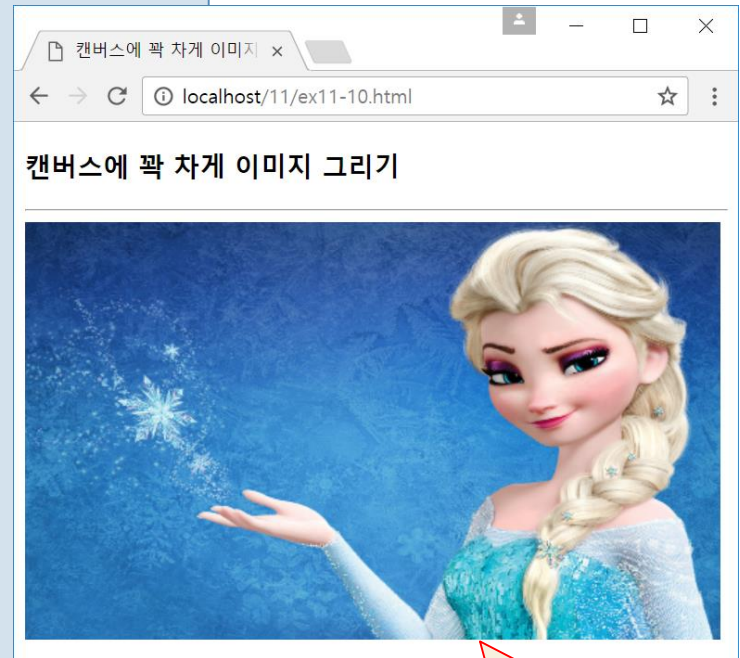


# 예제 11-10 캔버스에 꼭 차게 이미지 그리기

41

```
<!DOCTYPE html>
<html>
<head> <title>캔버스에 꼭 차게 이미지 그리기</title> </head>
<body>
<h3>캔버스에 꼭 차게 이미지 그리기</h3>
<hr>
<canvas id="myCanvas"
        style="background-color:aliceblue"
        width="500" height="300"> </canvas>
<script>
  var canvas = document.getElementById("myCanvas");
  var context = canvas.getContext("2d");

  var img = new Image();
  img.onload = function () {
    context.drawImage(img, 0, 0,
                     canvas.width, canvas.height);
  }
  img.src = "media/elsa.png";
</script>
</body>
</html>
```

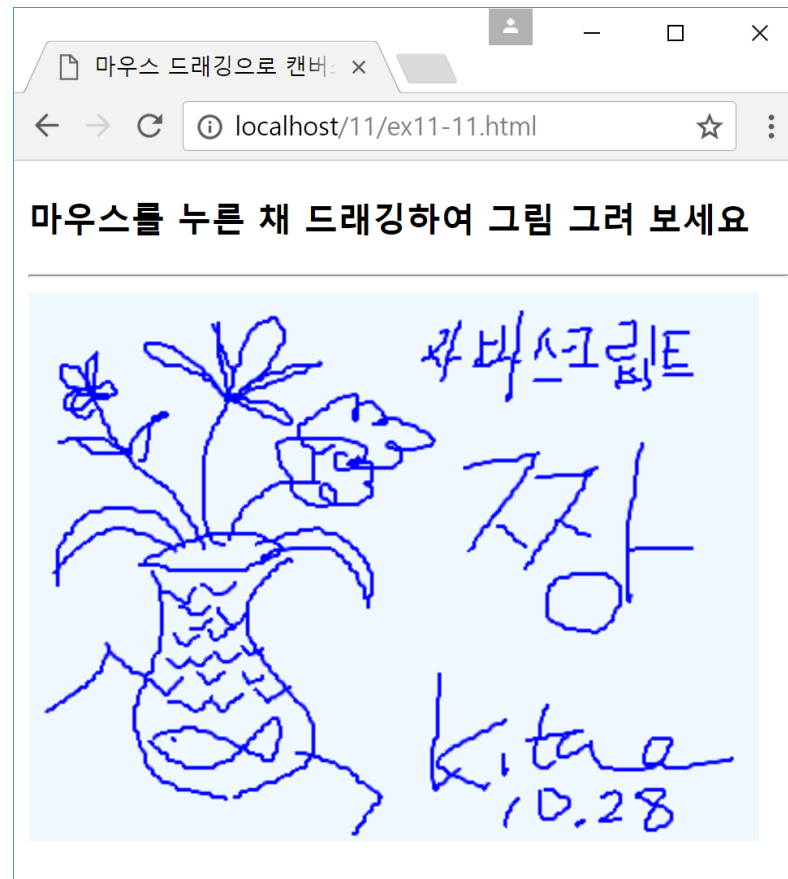


캔버스 크기 500x300

# canvas 객체와 마우스 이벤트 활용

42

- 사용자가 마우스로 드래킹하여 캔버스 위에 자유롭게 그림을 그리는 자바스크립트 응용 프로그램 작성



# 예제 11-11 마우스 드래킹으로 캔버스에 그림 그리기

43

```
<!DOCTYPE html>
<html>
<head><title>마우스 드래킹으로 캔버스에 그림 그리기</title></head>
<body onload="init()">
<h3>마우스를 누른 채 드래킹하여 그림 그려 보세요</h3>
<hr>
<canvas id="myCanvas" style="background-color:aliceblue" width="400" height="300">
</canvas>
<script>
```

```
var canvas, context;
```

```
function init() {
```

```
    canvas = document.getElementById("myCanvas");
```

```
    context = canvas.getContext("2d");
```

```
    context.lineWidth = 2; // 선 굵기 2
    context.strokeStyle = "blue";
```

마우스 리스너 등록.  
e는 MouseEvent 객체

```
    canvas.addEventListener("mousemove", function (e) { move(e) }, false);
    canvas.addEventListener("mousedown", function (e) { down(e) }, false);
    canvas.addEventListener("mouseup", function (e) { up(e) }, false);
    canvas.addEventListener("mouseout", function (e) { out(e) }, false);
```

```
}
```

```
var startX=0, startY=0; // 드래킹동안, 처음 마우스가 눌러진 좌표
```

```
var dragging=false;
```

```
function draw(curX, curY) {
```

```
    context.beginPath();
    context.moveTo(startX, startY);
    context.lineTo(curX, curY);
    context.stroke();
```

```
}
```

```
function down(e) {
```

```
    startX = e.offsetX; startY = e.offsetY; dragging = true;
```

```
}
```

```
function up(e) { dragging = false; }
```

```
function move(e) {
```

```
    if(!dragging) return; // 마우스가 눌러지지 않았으면 리턴
    var curX = e.offsetX, curY = e.offsetY;
    draw(curX, curY);
    startX = curX; startY = curY;
```

```
}
```

```
function out(e) { dragging = false; }
```

```
</script>
```

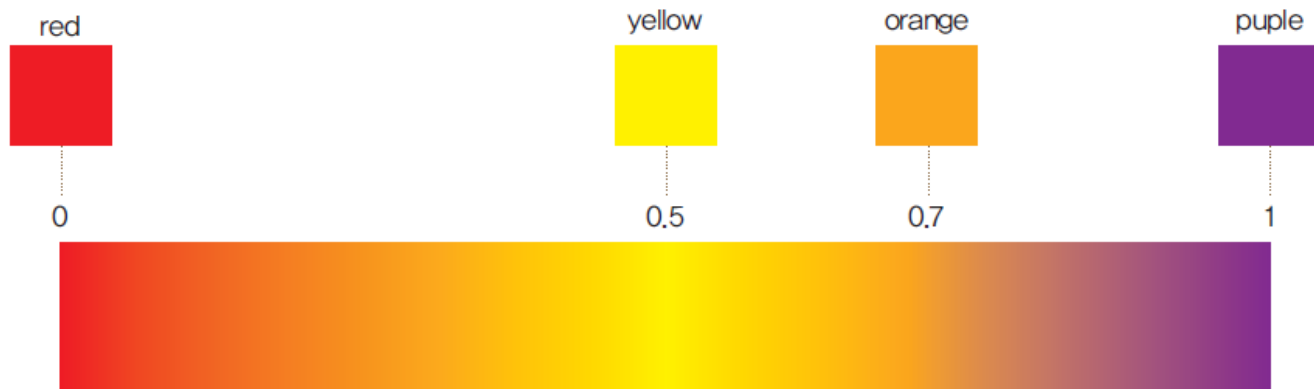
```
</body>
```

```
</html>
```

# 그라디언트 (Gradient)

44

- `createLinearGradient(x, y, x1, y1)` - 선형 그라디언트를 생성한다.
- `createRadialGradient(x, y, r, x1, y1, r1)` - 원형 그라디언트를 생성한다.

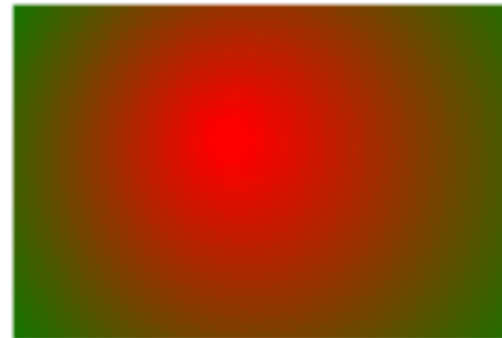




# 그라디언트 (Gradient)

45

Method	Description
<u><a href="#">createLinearGradient()</a></u>	Creates a linear gradient (to use on canvas content)
<u><a href="#">createPattern()</a></u>	Repeats a specified element in the specified direction
<u><a href="#">createRadialGradient()</a></u>	Creates a radial/circular gradient (to use on canvas content)
<u><a href="#">addColorStop()</a></u>	Specifies the colors and stop positions in a gradient object



# 선형 그라디언트 (Linear gradient)

46

```
<!DOCTYPE HTML>
<html>
<head>
  <style>
    body {
      margin: 20px;
      padding: 0px;
    }
  </style>
</head>
<body>
  <canvas id="myCanvas" width="300" height="200"></canvas>
  <script>
    var canvas = document.getElementById('myCanvas');
    var context = canvas.getContext('2d');

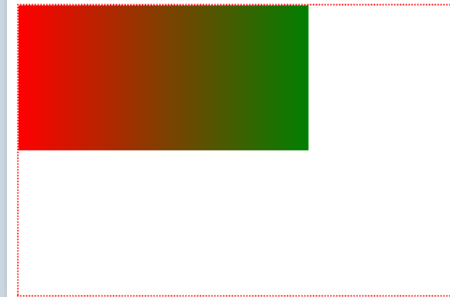
    var gradient = context.createLinearGradient(0, 0, 200, 0);
    gradient.addColorStop(0, "white");
    gradient.addColorStop(1, "red");

    context.fillStyle = gradient;
    context.fillRect(10, 10, 180, 90);
  </script>
</body>
</html>
```

## Linear gradient



## Canvas : 선형 그라디언트



# 원형 그라디언트 (Radial gradient)

47

```
<body>
  canvas id="myCanvas" width="300" height="300"
    style="border: 1px dotted red"> </canvas>

<script>

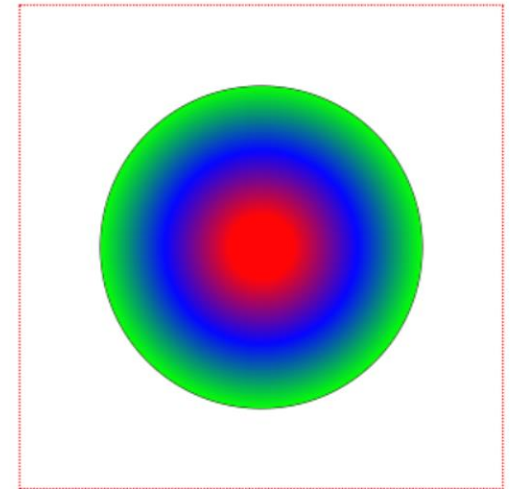
  var canvas = document.getElementById('myCanvas');
  var context = canvas.getContext('2d');

  // create a radial gradient
  var radGrd = context.createRadialGradient(150, 150, 20, 150, 150, 100);
  radGrd.addColorStop(0, "#f00");
  // start with red at 0
  radGrd.addColorStop(0.5, "#00f");
  // put blue at the halfway point
  radGrd.addColorStop(1, "#0f0");
  // finish with green
  context.fillStyle = radGrd;

  context.beginPath();
  context.arc(150, 150, 100, 0, 2 * Math.PI);
  context.stroke(); // draw circle
  context.fill(); // apply gradient

</script>
</body>
```

Radial gradient



# 과제02. msnn\_rpt02.zip

48

[실습과제02] Canvas에 태극 문양 그리기

- [1] 태극 문양에 색과 그라디언트 효과 추가.
- [2] 캔버스의 하단에 본인 ID를 텍스트로 그려서 추가.
- [3] 파일명: MSnn\_rpt02.html

파일명: [msnn\\_rpt02.html](#)

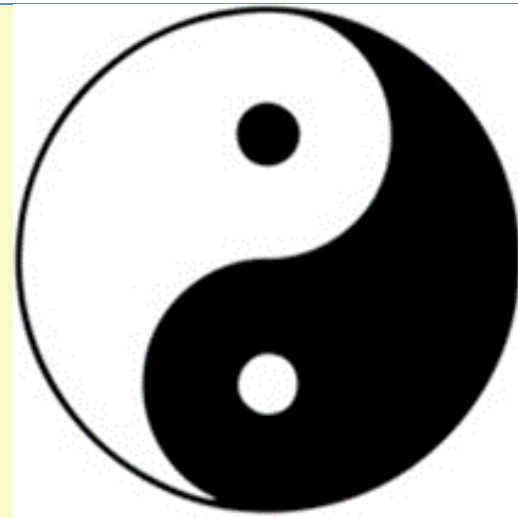
가점: Javascript 프로그래밍 응용 능력.

[제출파일] [msnn\\_rpt02.html](#)

[msnn\\_rpt02.html](#) 을 이메일로 제출

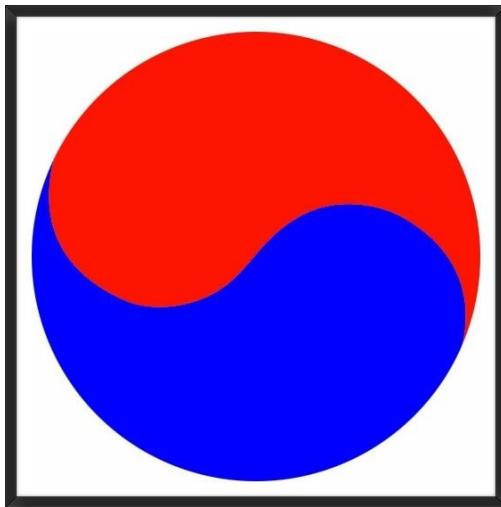
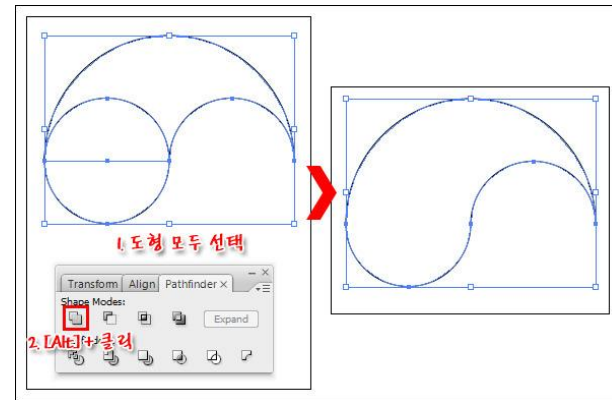
Email : [chaos21c@gmail.com](mailto:chaos21c@gmail.com)

[chaos21c@gmail.com](mailto:chaos21c@gmail.com)



## 과제02. hint

49



# 교재 WEB 강의 소개

← → ↻ ⓘ webprogramming.co.kr ☆

명품 **HTML5+  
CSS3+  
Javascript** 웹 프로그래밍

Home Introduction Notice Board Support Code

## 명품 **HTML5 + CSS3 + Javascript** 웹 프로그래밍

HTML5로  
여러분의 무한한  
상상력을  
표현해 보세요!



Sir Tim Berners-Lee  
(1955.6.8 ~)

명품 웹 프로그래밍 소개  
"웹 프로그래밍을  
가장 쉽게 익힐 수 있는 책"

처음 웹 프로그래밍을 공부하는 입문자들도  
모든 주제를 직관적으로 이해하고  
빠르게 파악할 수 있습니다.



자세히보기 →

강력한 Q&A 피드백 제공  
"빠르고, 간결하고, 정확한  
저자의 직접적인 답변"

'이거 이해가 잘 안되는데.. 물어볼 사람도 없고..'  
더이상 고민하지 마세요.  
명품 웹 프로그래밍 홈페이지에서는  
누구나 저자가 직접 답변해주는  
Q&A 게시판을 이용할 수 있습니다.



자세히보기 →

즉석 실행 가능한 예제 프로그램  
"백문이 불여일견, 백견이 불여일타(打)!"

코드로만 설명되어 있는 예제들,  
결과 화면이 있어도 이해가 잘 안되시죠?  
예제 소스를 바탕으로, 내맘대로 수정한  
코드를 즉석으로 웹 페이지로  
변환해주는 예제 프로그램을 통해  
모든 코드를 빠르고 쉽게  
이해할 수 있습니다.



자세히보기 →

Notice

Test

2017-01-16 15:32

Know-How

Test

2017-01-17 14:04 관리자

# 관련 WEB 강의 소개 – w3schools.com

← → ↻ 🔒 안전함 | https://www.w3schools.com

w3schools.com THE WORLD'S LARGEST WEB DEVELOPER SITE

TUTORIALS ▾ REFERENCES ▾ EXAMPLES ▾

HTML and CSS

- Learn HTML
- Learn CSS
- Learn W3.CSS
- Learn Colors
- Learn Bootstrap
- Learn Icons
- Learn Graphics
- Learn How To

JavaScript

- Learn JavaScript
- Learn W3.JS
- Learn jQuery
- Learn jQueryMobile
- Learn AppML
- Learn AngularJS
- Learn JSON
- Learn AJAX

Server Side

- Learn SQL
- Learn PHP
- Learn ASP

Web Building

- Web Templates
- Web Statistics
- Web Certificates

XML Tutorials

- Learn XML
- Learn XML AJAX
- Learn XML DOM
- Learn XML DTD
- Learn XML Schema
- Learn XSLT
- Learn XPath
- Learn XQuery

## HTML

The language for building web pages

LEARN HTML HTML REFERENCE

HTML Example:

```
<!DOCTYPE html>
<html>
<title>HTML Tutorial</title>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Try it Yourself »

## CSS

The language for styling web pages

LEARN CSS CSS REFERENCE

CSS Example:

```
body {
  background-color: lightblue;
}
h1 {
  color: white;
  text-align: center;
}
p {
  font-family: verdana;
  font-size: 20px;
}
```

Try it Yourself »

## JavaScript

The language for programming web pages

JavaScript Example:

```
<script>
function myFunction() {
  var x = document.getElementById("demo");
  x.style.fontSize = "35px";
}
```