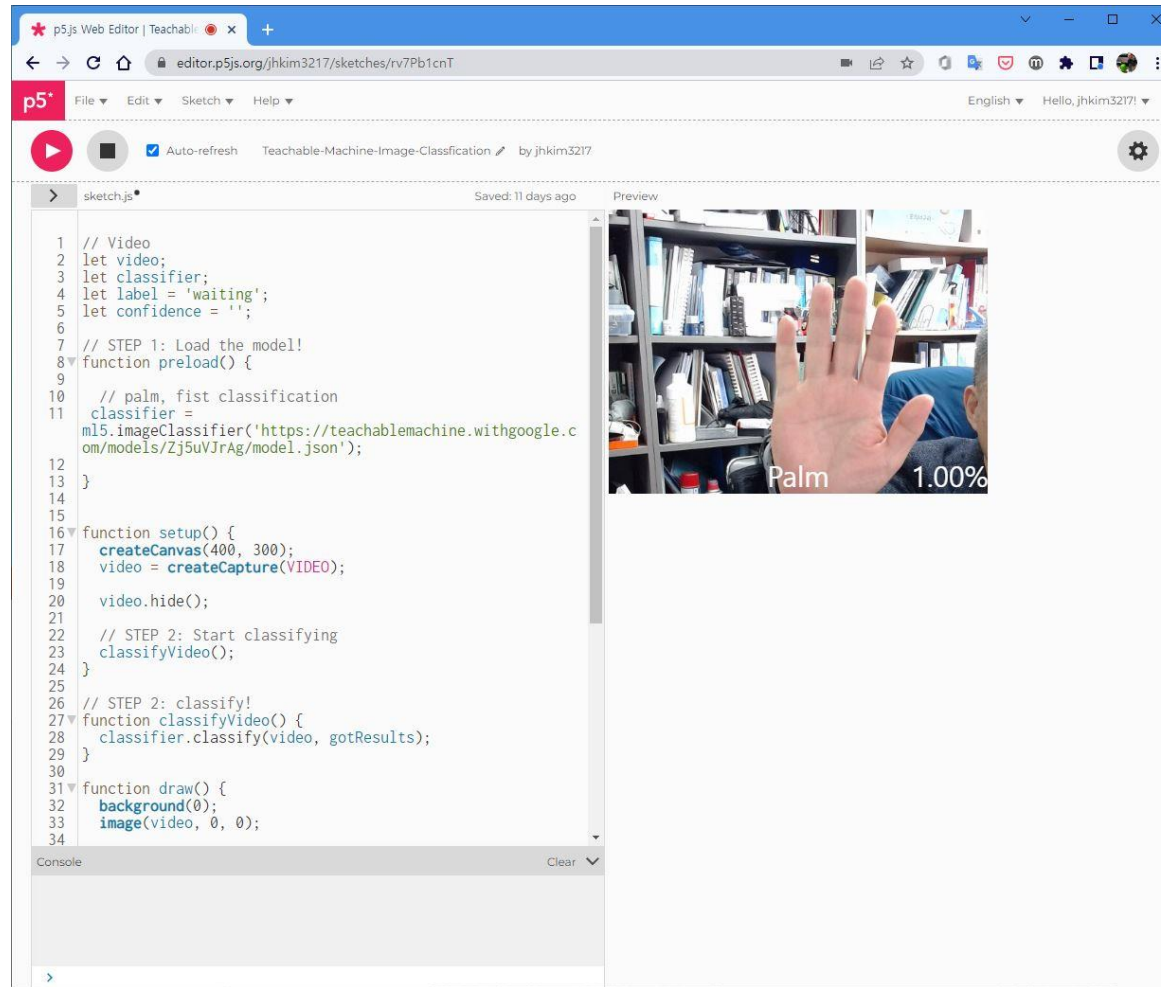


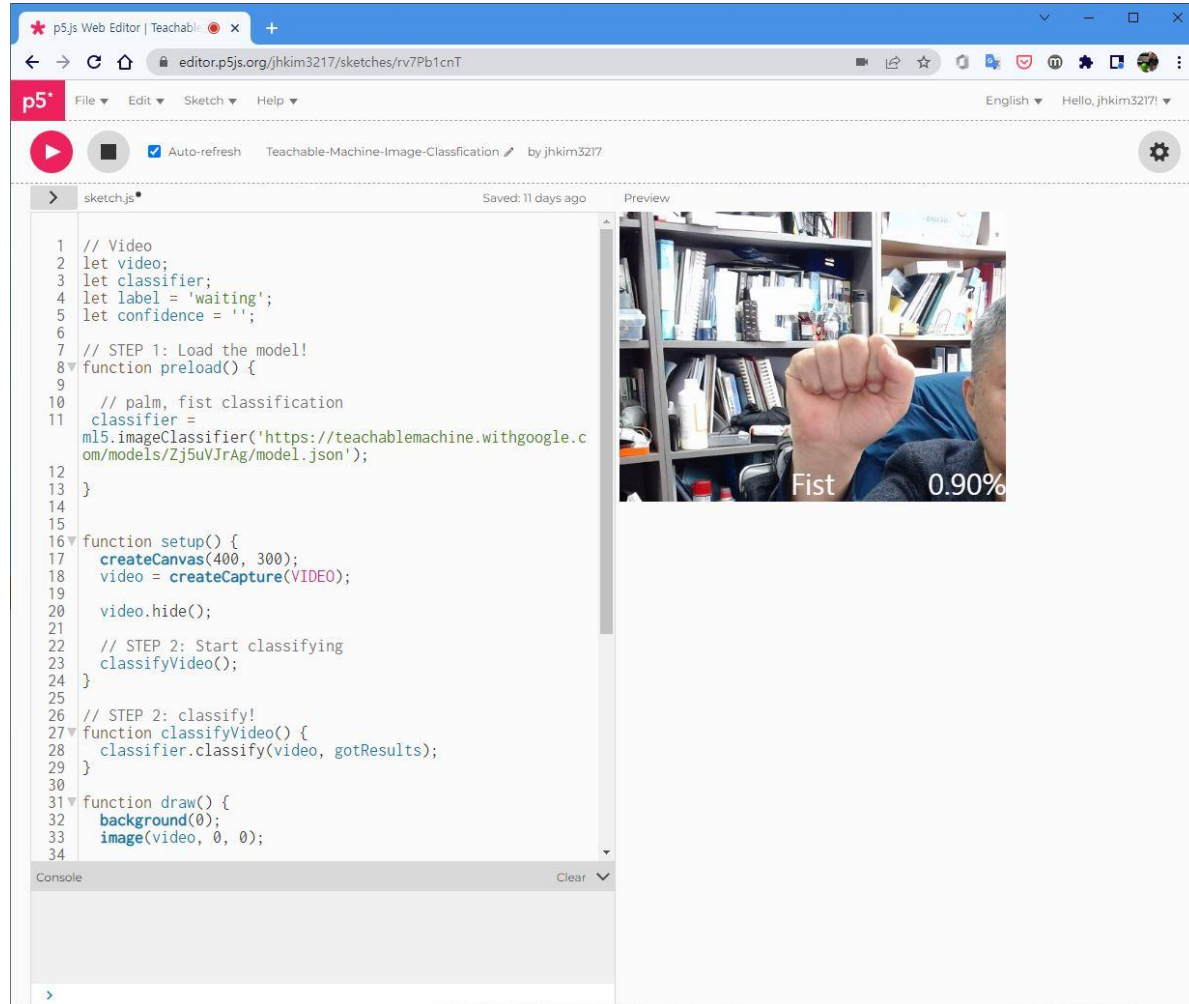
p5.js

Teachable Machine을 이용한 이미지 분류

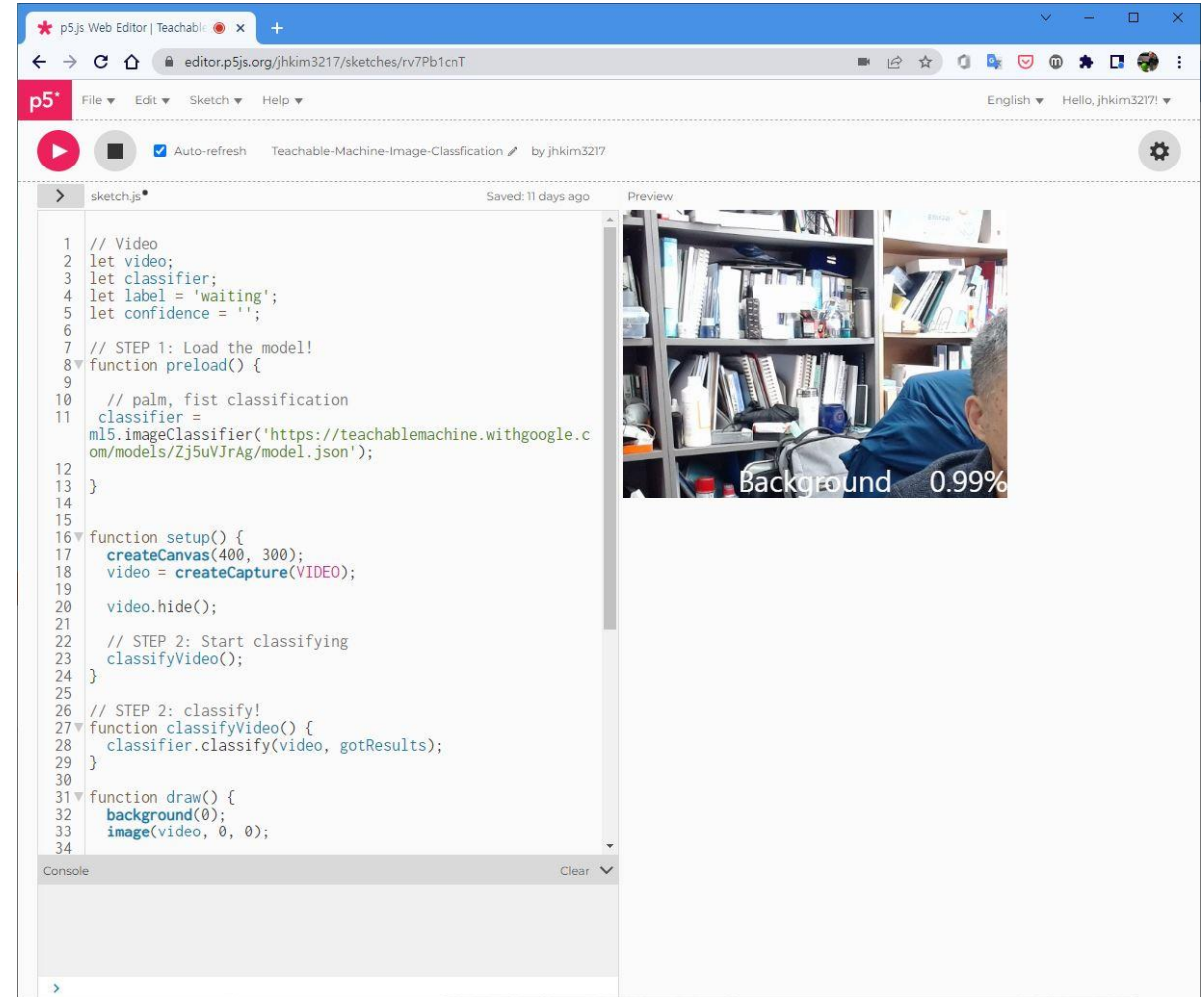
- class 01 : palm



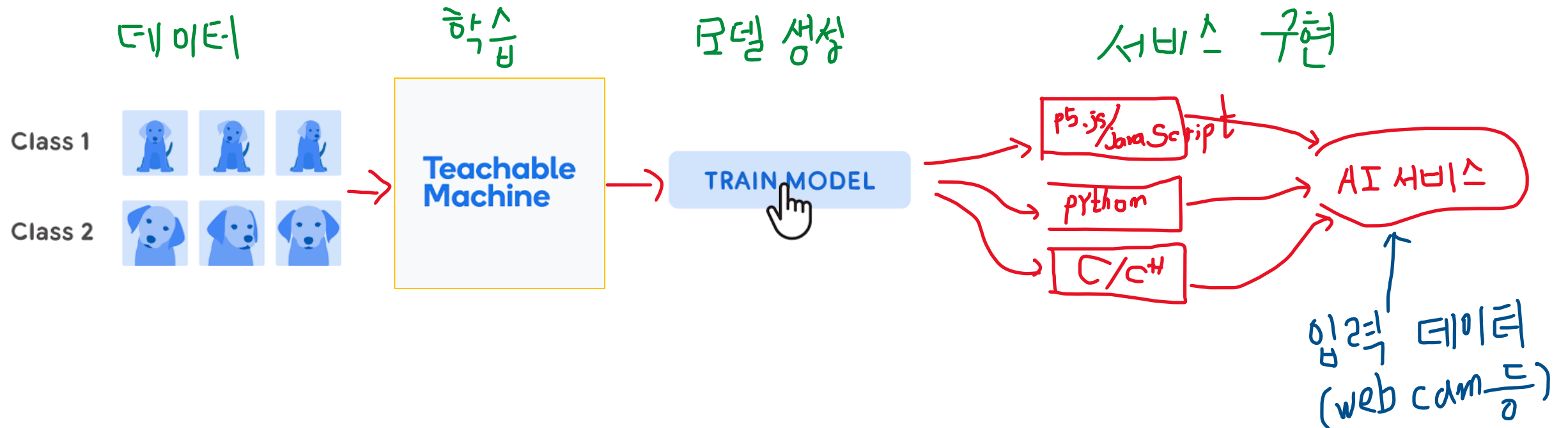
- class 02 : fist



- class 01 : background



Teachable Machine을 이용한 AI 서비스 만들기



모델을 이용한 컴퓨터 프로그래밍

```
//Video;
let video;
let classifier;
let label = "waiting";
let confidence = "";

// STEP 1: Load the model!
function preload() {
  // palm, fist classification
  classifier = ml5.imageClassifier(
    "https://teachablemachine.withgoogle.com/models/Zj5uVJrAg/model.json"
  );
}

function setup() {
  createCanvas(400, 300);
  video = createCapture(VIDEO);
  video.hide();

  // STEP 2: Start classifying
  classifyVideo();
}
```

```
// STEP 2: classify!
function classifyVideo() {
  classifier.classify(video, gotResults);
}

function draw() {
  background(0);
  image(video, 0, 0);

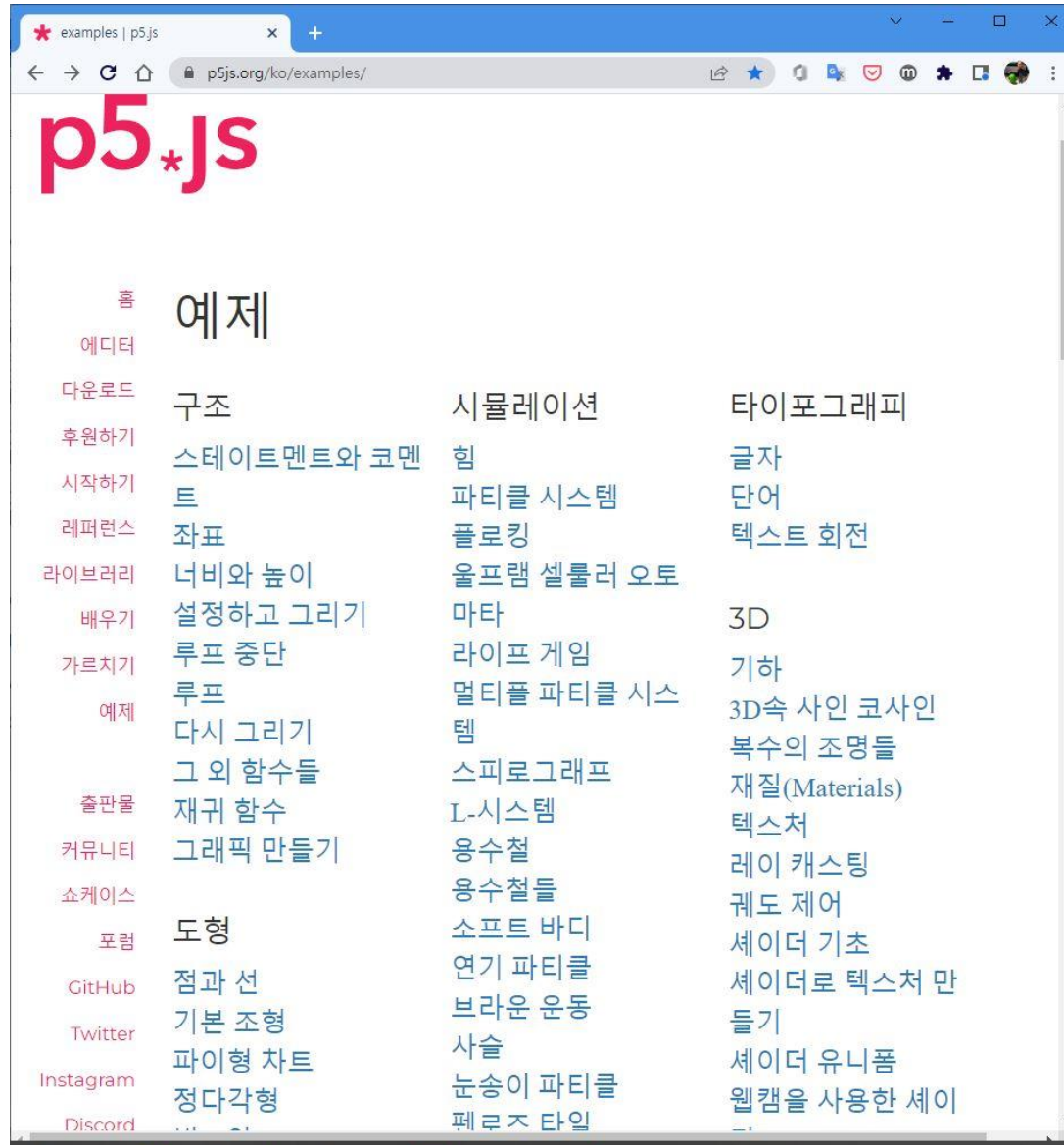
  // STEP 4: Draw the label
  textSize(30);
  textAlign(CENTER, CENTER);
  fill(255);
  text(label, width / 2, height - 16);
  text(confidence, width - 40, height - 16);
}

// STEP 3: Get the classification!
function gotResults(error, results) {
  label = results[0].label;
  confidence = results[0].confidence;
  confidence = confidence.toFixed(2) + "%";
  classifyVideo();
  //classify video again
}
```

- JavaScript 라이브러리
- 텍스트, 비디오, 입력, 비디오, 웹캠, 사운드 등을 비롯한 각종 HTML 요소 사용 가능
- 예술가, 디자이너, 교육자, 입문자에서 오픈소스 도구 제공

- 대표 사이트 : <https://p5js.org/ko/>
- p5 전용 에디터 : <https://editor.p5js.org/>
 - 데스크탑 에디터 : VS Code, Atom 등
- p5 시작하기 : <https://p5js.org/ko/get-started/>
- P5 예제 : <https://p5js.org/ko/examples/>

P5 예제 : <https://p5js.org/ko/examples/>



- **p5.js란?**
 - 에디터랑 친해지기
- **Drawing Shapes**
 - 도형 그리기
- **Color**
 - 색상 입히기
- **Variables**
 - 전역 변수 사용하기
- **random() 함수**
 - 임의의 위치에 도형 그리기
- **Functions**
 - 공 그리기

- **Arrays**
- **Loops**
 - arrayTest() 함수 작성
- **Objects**
 - 공을 객체로 만들어 출력
 - 객체를 배열로 만들어 출력
- **If-else**
- **Events**
 - 마우스 클릭 이벤트
- **Video, Image, Audio 처리하기**
- **Test 처리하기**
- **코드 활용하기**

p5.js Web Editor

editor.p5js.org

p5*

File Edit Sketch Help

English Log in or Sign up

Auto-refresh Easy diaschia

Sketch Files

- index.html
- sketch.js
- style.css

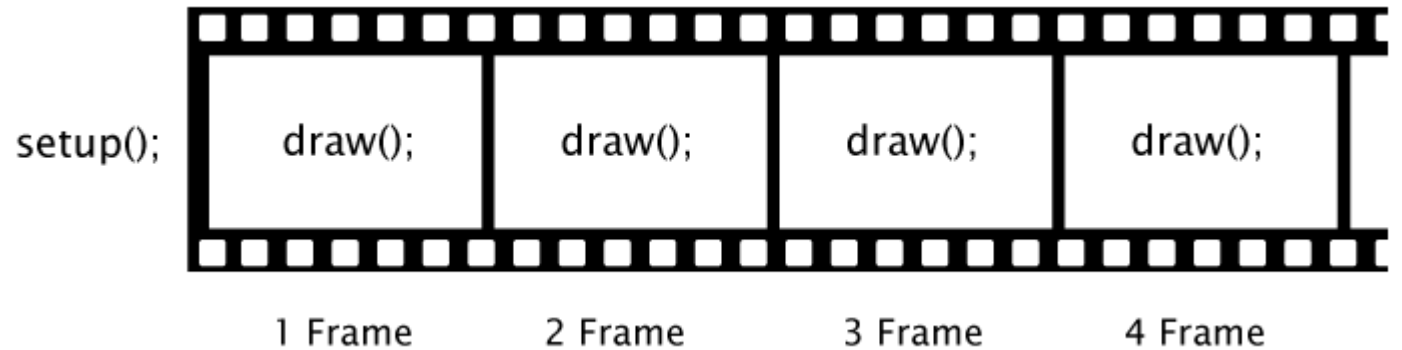
```
1 function setup() {  
2   createCanvas(400, 400);  
3 }  
4  
5 function draw() {  
6   background(220);  
7 }
```

Preview

Console Clear

프로그램 구조

```
1 ▼ function setup() {  
2   createCanvas(400, 400);  
3 }  
4  
5 ▼ function draw() {  
6   background(220);  
7 }
```



- `setup()` : 페이지가 실행될 때 맨 처음 **한번**만 실행되는 함수
- `draw()` : 페이지가 실행될 때 매 프레임마다 실행되는 **반복** 실행 함수

함수	기능 및 사용법
<code>createCanvas(width, height)</code>	<ul style="list-style-type: none">• 그리기 캔버스를 만들고 작업중인 html 페이지에 추가• html 페이지 본문에 다른 내용이 있으면 그 후에 페이지에 추가되므로 창 상단에 표시되지 않을 수 있음
<code>background(color)</code>	<ul style="list-style-type: none">• 배경색 설정• color : 'red', (255, 0,0), '#f00', '(rgb(255, 0,0, 0.5))' 등

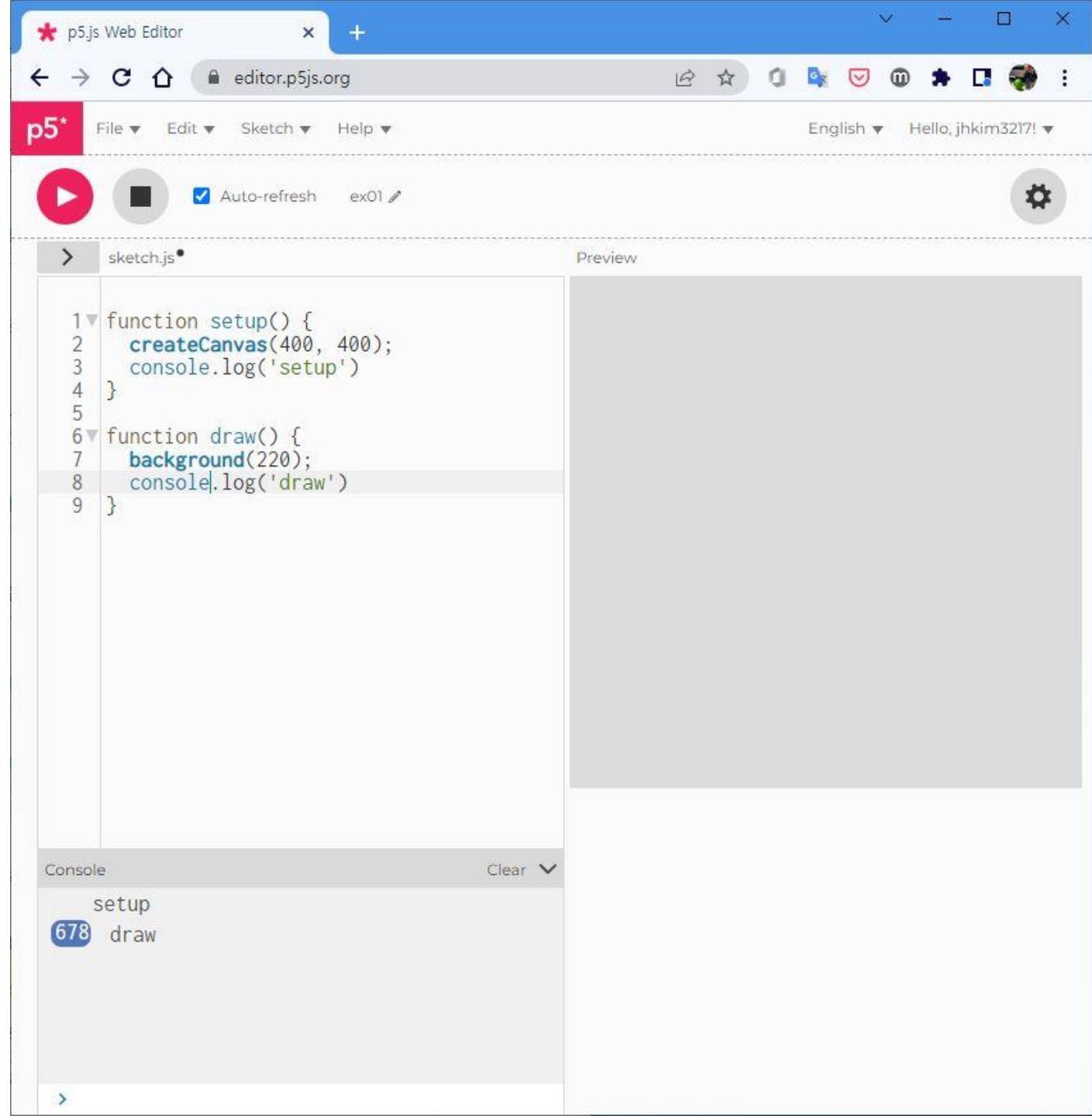
다양한 도형 그리기 함수

함수	기능 및 사용법
rect(x, y, width, height);	<ul style="list-style-type: none">• 사각형 그리기• 위치(x, y) / 크기(width, height)
square(x, y, s)	<ul style="list-style-type: none">• 정사각형 그리기• s : 한 변의 길이(픽셀)
ellipse(x, y, width, height)	<ul style="list-style-type: none">• 타원형 그리기
strokeWeight(weight)	<ul style="list-style-type: none">• 점, 선, 도형 윤곽선 함수인 stroke()의 결과값 두께를 픽셀 단위로 지정
print(contents) / console.log(contents)	<ul style="list-style-type: none">• 브라우저 콘솔 창에 출력

첫번째 스케치

```
function setup() {  
  createCanvas(400, 400);  
  console.log('setup')  
}
```

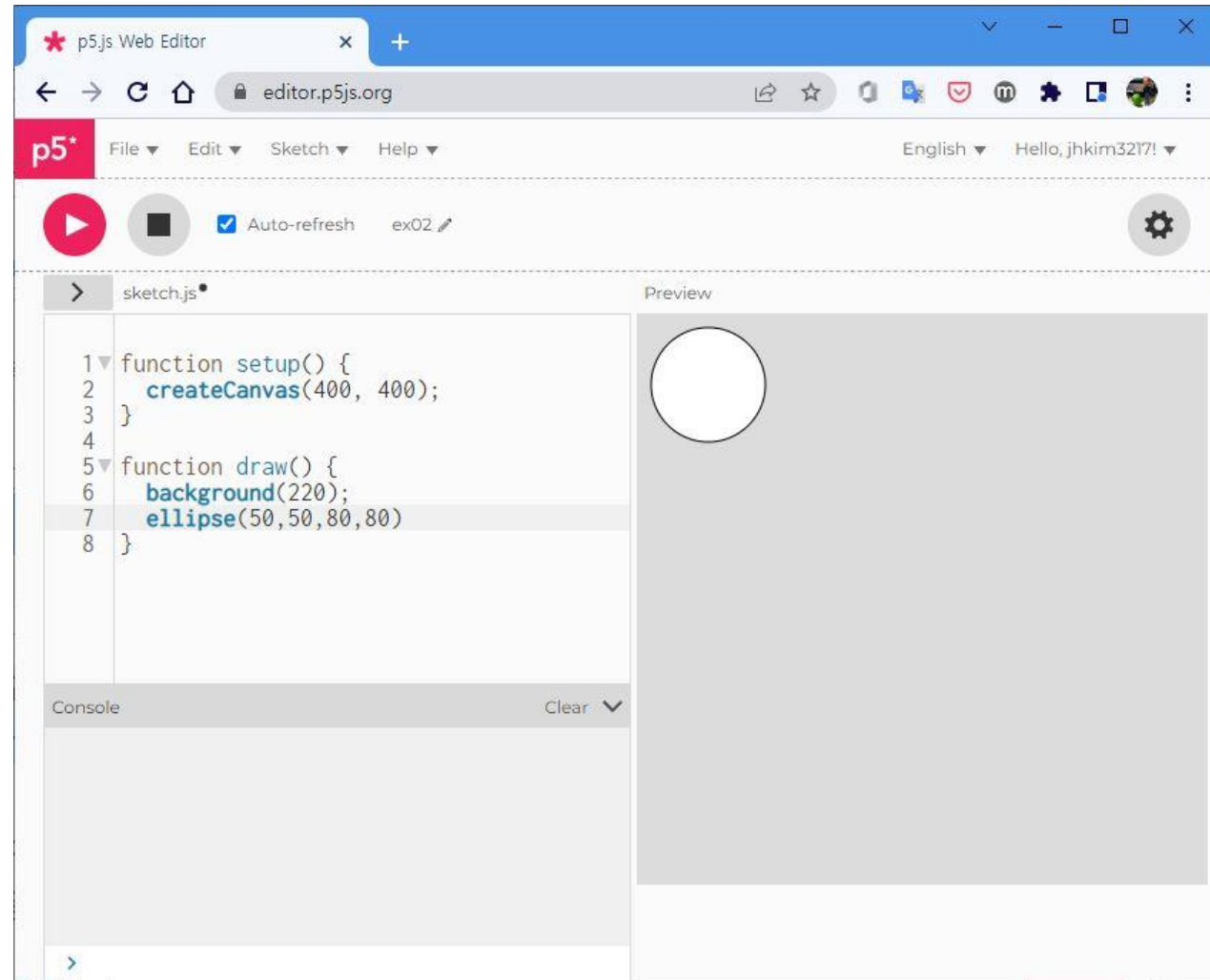
```
function draw() {  
  background(220);  
  console.log('draw')  
}
```



타원(ellipse) 그리기

```
function setup() {  
  createCanvas(400, 400);  
}
```

```
function draw() {  
  background(220);  
  ellipse(50,50,80,80);  
}
```

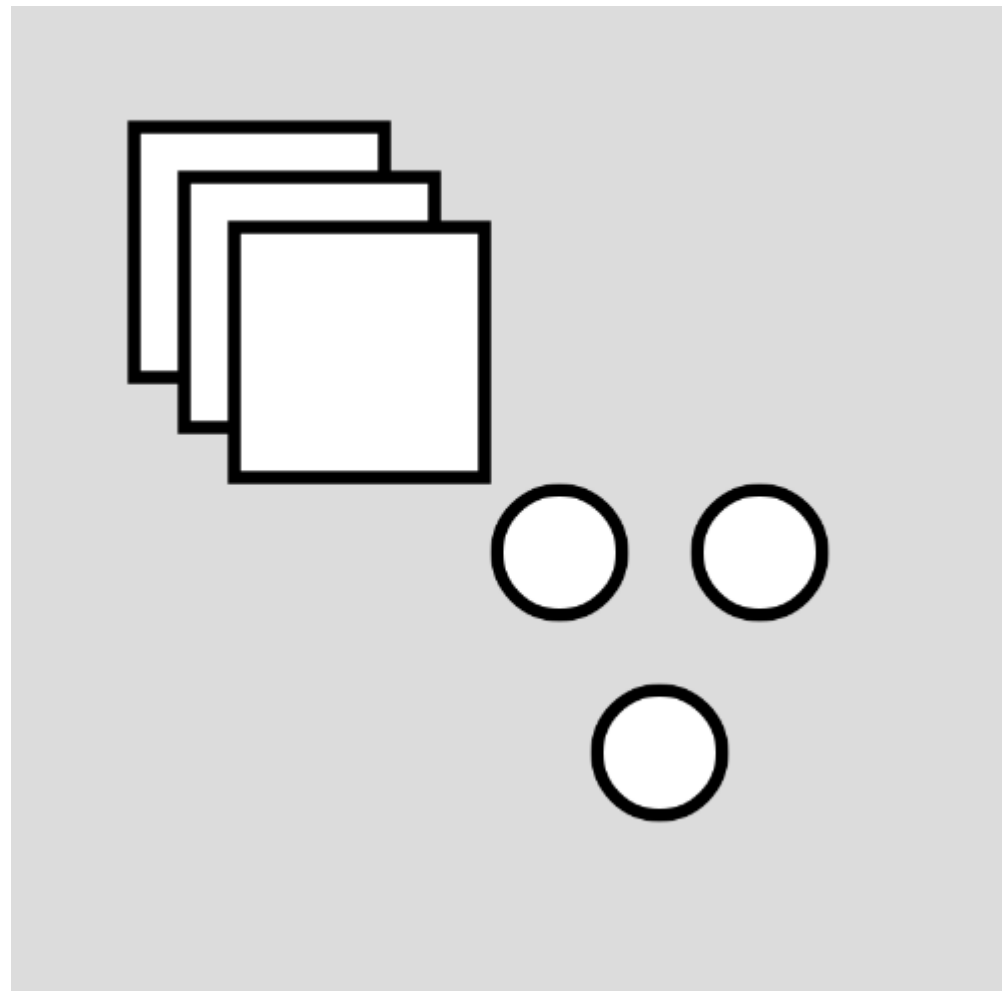


- 좌측 상단 모서리로부터 아래로 50px, 오른쪽으로 50px 떨어진 점을 중심으로 타원을 그린다.
- 타원의 너비와 높이는 모두 80px로 한다.

실습 도형 그리기

- 다음 화면과 같이 도형을 그려 보세요.
- 도형의 위치는 도형에 따라 어떻게 결정되나요?

```
1 ▼ function setup() {  
2   createCanvas(400, 400);  
3  
4 }  
5 ▼ function draw() {  
6   background(220);  
7  
8   strokeWeight(5);  
9   square(50, 50, 100);  
10  
11  square(70, 70, 100);  
12  square(90, 90, 100);  
13  
14  ellipse(220, 220, 50, 50);  
15  ellipse(300, 220, 50, 50);  
16  ellipse(260, 300, 50, 50);  
17 }
```

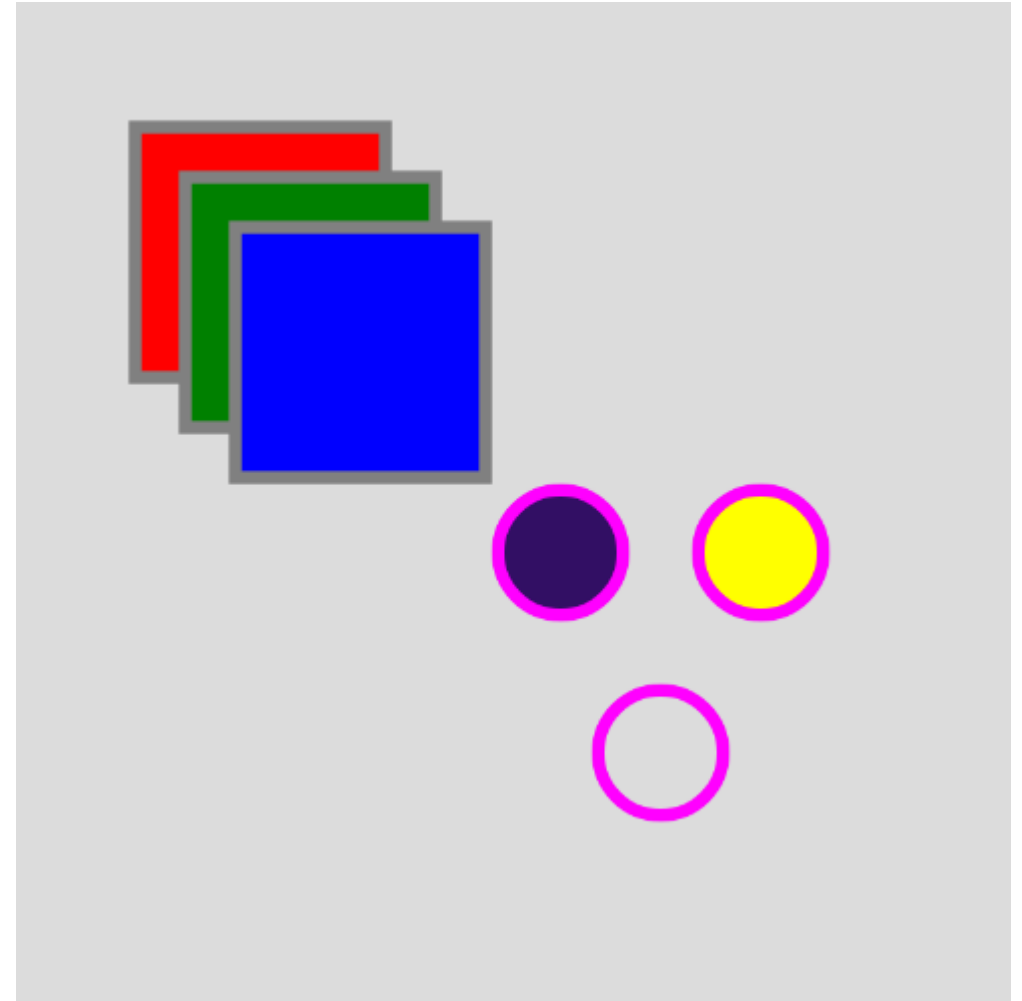


색상 사용하기

함수	기능 및 사용법
fill(color) / noFill()	<ul style="list-style-type: none">• 도형의 면을 채울 색상을 지정
stroke(color)	<ul style="list-style-type: none">• 그려질 선 또는 도형 윤곽선의 색상을 설정
strokeWeight(weight)	<ul style="list-style-type: none">• 점, 선, 도형 윤곽선 함수인 stroke()의 결과값 두께를 픽셀 단위로 지정

실습 도형에 다양한 색상 입히기

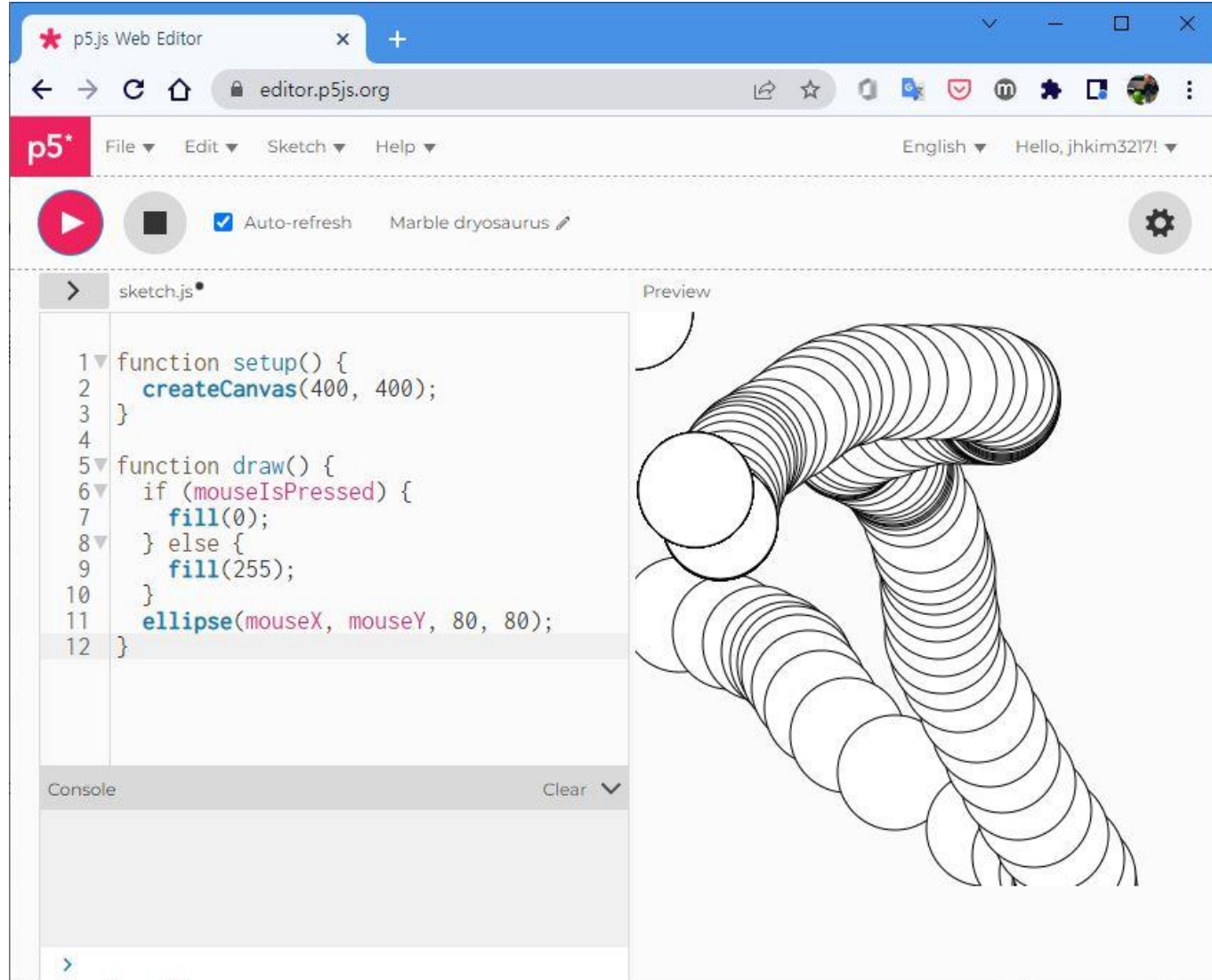
```
1 ▼ function setup() {  
2   createCanvas(400, 400);  
3  
4 }  
5 ▼ function draw() {  
6   background(220);  
7  
8   strokeWeight(5); stroke('grey'); fill('red')  
9   square(50, 50, 100);  
10  
11  fill('green'); square(70, 70, 100); fill('blue');  
12  square(90, 90, 100);  
13  
14  stroke(255, 0, 255); fill(50, 15, 100);  
15  ellipse(220, 220, 50, 50);  
16  
17  stroke(255, 0, 255); fill('yellow');  
18  ellipse(300, 220, 50, 50);  
19  
20  noFill();  
21  ellipse(260, 300, 50, 50);  
22 }
```



마우스 이벤트

```
function setup() {  
  createCanvas(400, 400);  
}
```

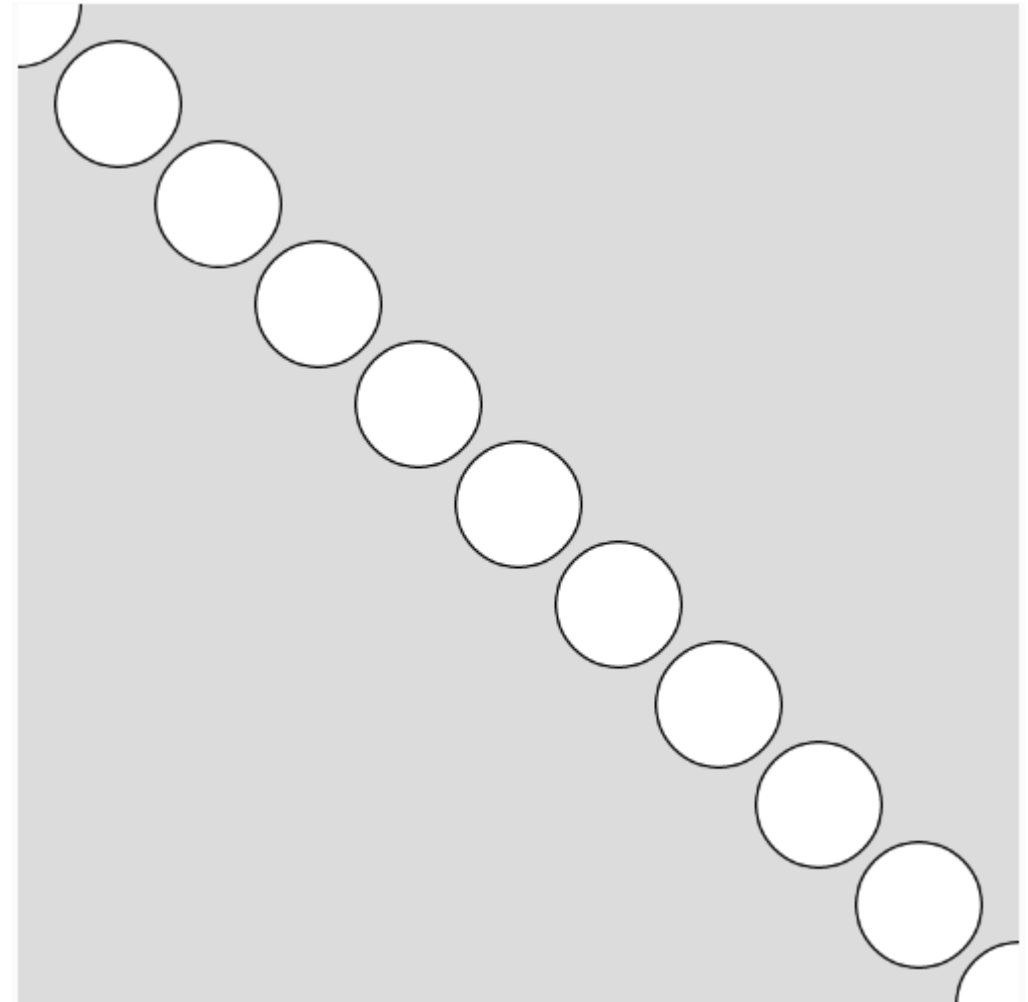
```
function draw() {  
  if (mouseIsPressed) {  
    // 마우스를 누르면 검은색 변경  
    fill(0);  
  } else {  
    fill(255);  
  }  
  ellipse(mouseX, mouseY, 80, 80);  
}
```



전역 변수 사용하기

- 위치를 지정하는 전역변수 x, y를 선언하고 이를 draw()함수에서 사용하여 다음과 같은 도형을 그리시오.

```
1  var x=0; //전역변수 x, y
2  var y=0;
3
4  ▼ function setup() {
5      createCanvas(400, 400);
6      background(220);
7  }
8
9  ▼ function draw() {
10     ellipse(x, y, 50, 50);
11     x = x + 40;
12     y = y + 40;
13 }
```



주석문(Comment) : 설명문

```
// This is a comment.
```

```
// Draw a purple ball
```

```
fill ( 200 , 30 , 255 ); // Make it purple
```

```
ellipse ( 100 , 100 , 50 , 50 ); // Draw the ball
```

```
/*
```

```
fill(200, 30, 255);
```

```
ellipse(100, 100, 50, 50);
```

```
*/
```

반복문 : for loop

```
for ( var number = 1 ; number <= 10 ; number ++ ) {  
    print ( number ); // body  
}  
print ( 'Done!' ) ;
```

문제) for문을 사용하여 $1+2+\dots+10$ 의 합을 구하시오

반복문 : while loop

```
var number = 1;
while ( number <= 10 ) { // condition
    print ( number); // body
    number ++; // updater
}

print ( 'Done!' );
```

문제) for문을 사용하여 $1+2+...+10$ 의 합을 구하시오

random(*min*, *max*) : 난수 생성

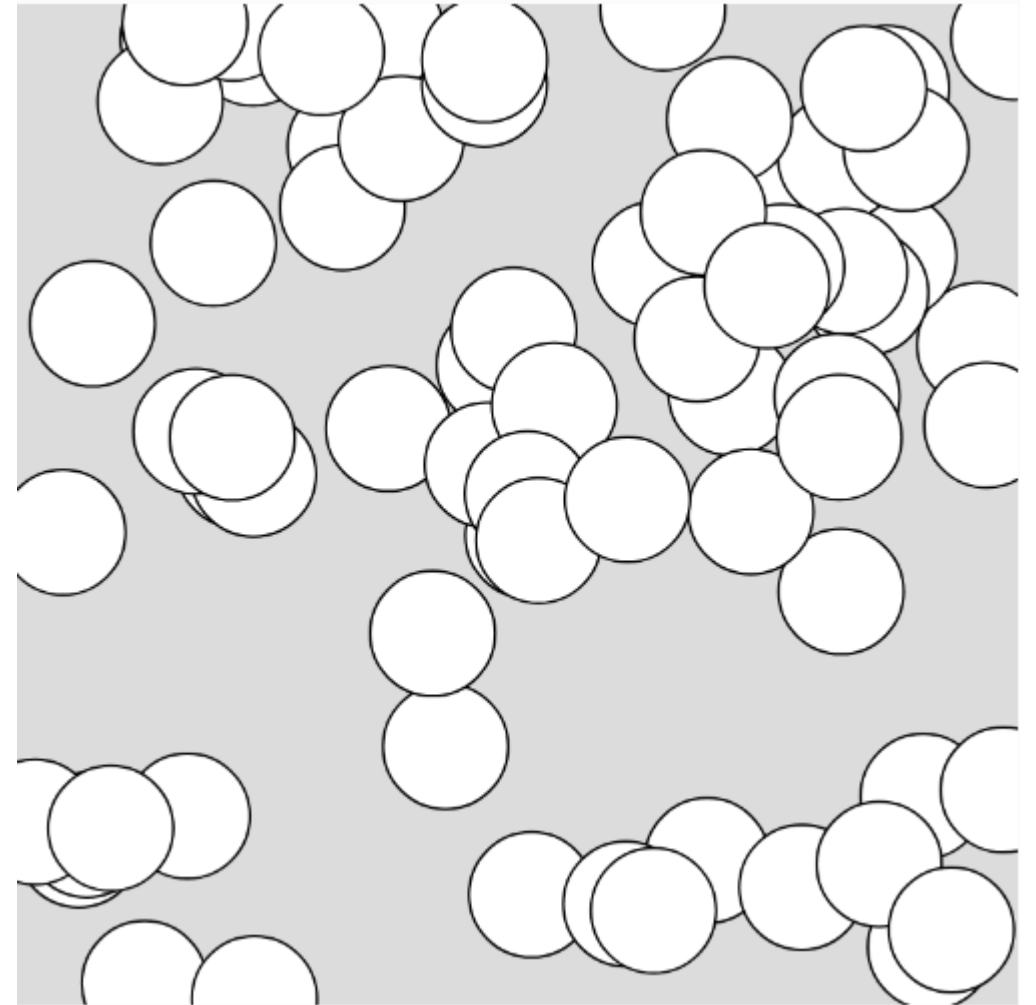
- 임의의 부동 소수점 숫자, 즉 실수(float)를 반환
- 별도의 인수를 지정하지 않을 경우, 0부터 1미만 사이의 난수 반환
 - random();
- 1개의 인수를 배열로 지정한 경우, 해당 배열로부터 임의의 요소 반환
 - var fruit = ['apple', 'bear', 'orange']
 - random(fruit);
- 2개의 인수를 지정한 경우, 1번째 인수에서 2번째 인수 미만 사이의 난수 반환
 - random(0, 10);

실습 : 임의의 위치에 도형 그리기

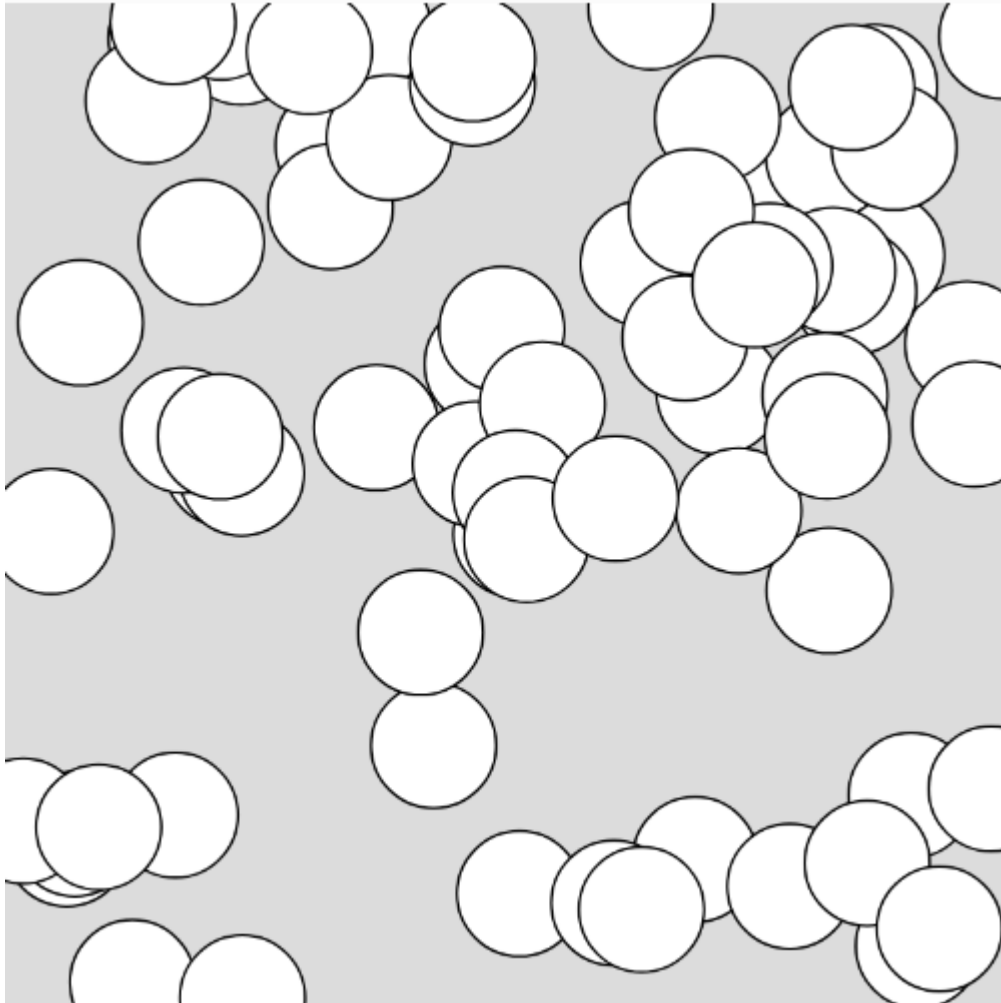
- 전역 변수 x, y 값을 캔버스 내 임의의 좌표로 할당하여 동그라미를 그리시오.

시스템 변수	기능
width	• 생성된 캔버스의 너비값을 저장한 시스템 변수
height	• 생성된 캔버스의 높이값을 저장한 시스템 변수

- random() 함수를 setup()과 draw()에서 실행해보고 차이를 설명하시오.
- 채움 색상과 테두리의 색도 변경해 보시오.



실습 : 임의의 위치에 도형 그리기

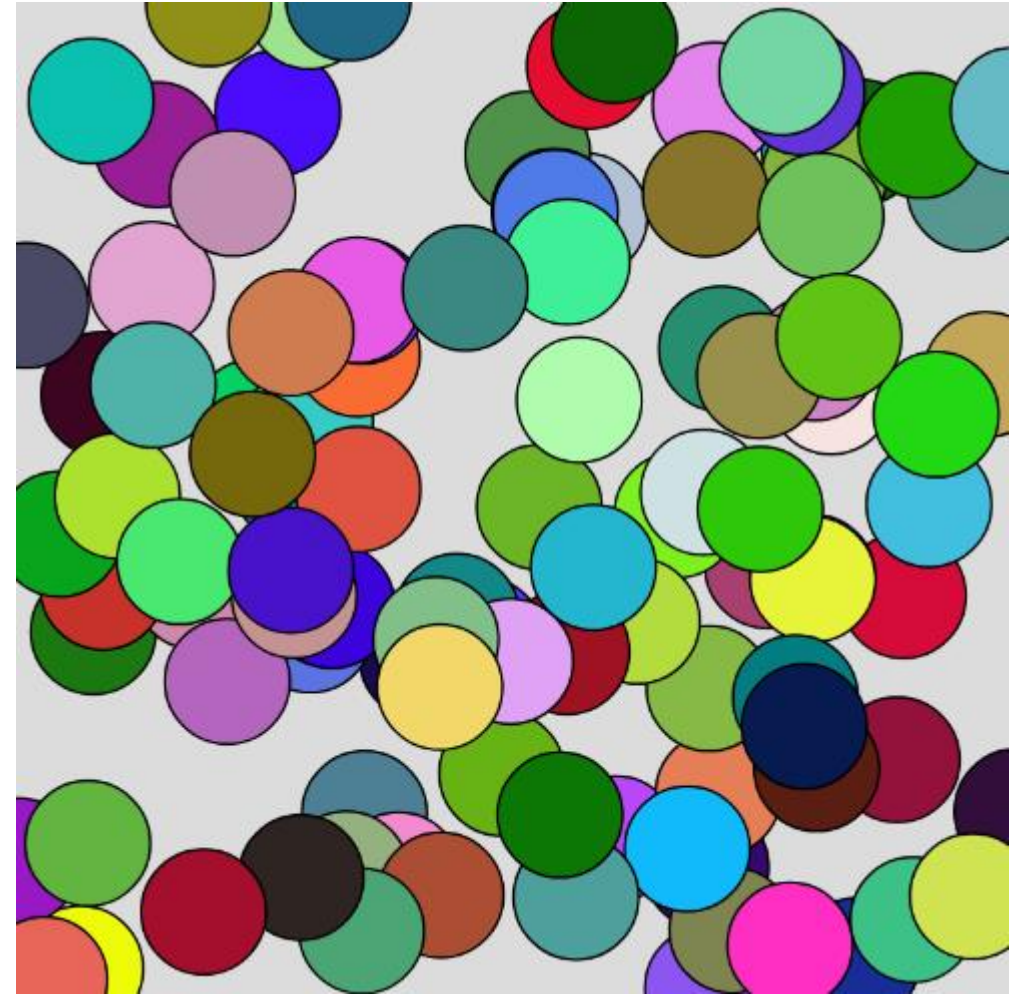


```
1  var x=0;
2  var y=0;
3  var c;
4
5  ▼ function setup() {
6      createCanvas(400, 400);
7      background(220);
8  }
9
10 ▼ function draw() {
11     x = random(0, width);
12     y = random(0, height);
13
14     ellipse(x, y, 50, 50);
15 }
```


실습

- random() 함수와 color(r, g, b) 함수를 사용하여 다음과 같은 도형을 그리시오.

```
1  var x=0;
2  var y=0;
3  var c;
4
5  ▼ function setup() {
6      createCanvas(400, 400);
7      background(220);
8  }
9
10 ▼ function draw() {
11     x = random(0, width);
12     y = random(0, height);
13     c = color(random(255), random(255), random(255));
14
15     fill(c);
16     ellipse(x, y, 50, 50);
17 }
```



함수(Function)

매개변수(parameter)

특정 함수에만 그 사용 범위가 지정된 변수를 뜻하며 함수 호출 시 그 값을 지정

- 함수의 특징
 - 특정 코드를 묶어 여러 곳에서 재사용하기 위해 작성
 - 함수명은 파일 내에서 unique해야 함
 - 함수 외부에서 데이터를 받기 위해 매개변수 지정할 수 있음
 - 함수 내부에서만 사용할 수 있는 변수를 지정할 수 있음
 - 함수 사용을 위해서 함수명으로 호출함
 - 함수 실행 결과를 return 키워드를 통해 함수 외부로 전달할 수 있음
 - return은 함수 실행을 멈추는 용도로도 사용
- 함수의 기본형

```
function myFunction(argument1, argument2, argument3) {  
    //some code to be executed  
}
```

함수

- 함수 정의

```
function helloWorld() {  
    print('Hello world!');  
}
```

```
function multiply(a, b) {  
    var answer = a * b;  
    print(answer);  
}
```

```
function multiply(a, b) {  
    var answer = a * b;  
    return answer;  
}
```

- 함수 호출(사용)

```
helloWorld(); // simply prints 'Hello world!'  
multiply(3, 4); // prints out 12;
```

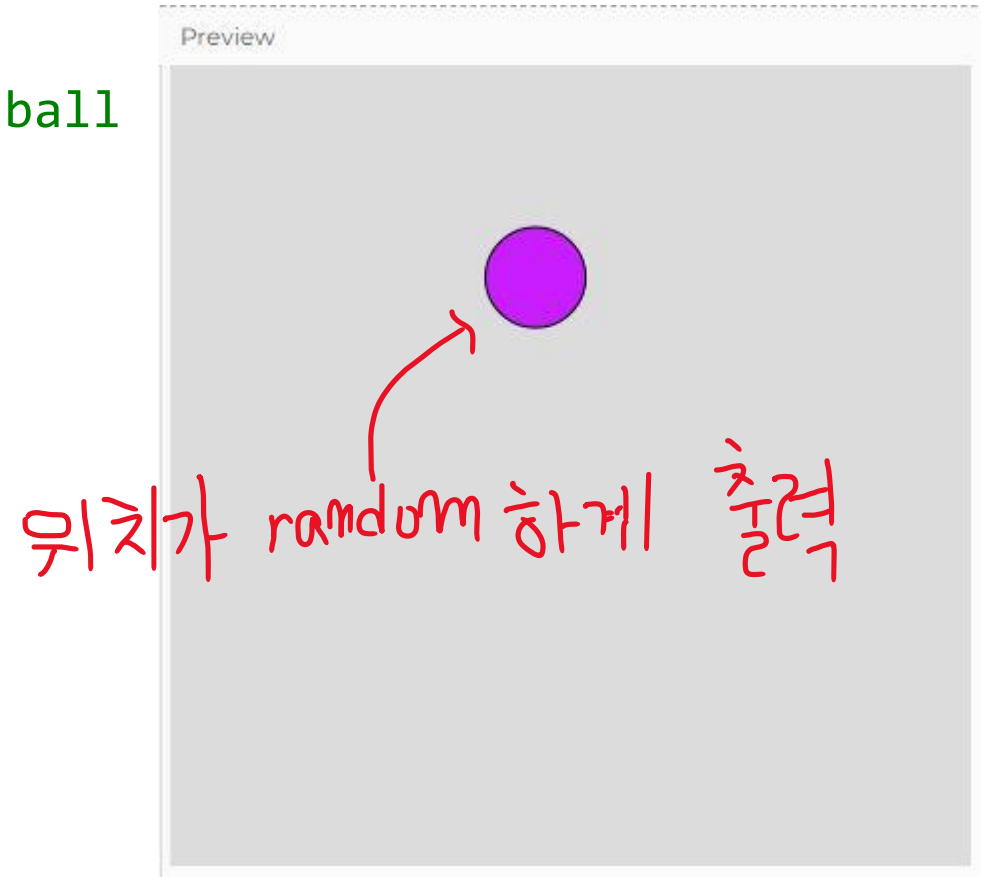
실습 : random 위치에 공그리기

```
var x ; // a global variable holding the x-position
var y ; // a global variable holding the x-position

function drawBall ( posX , posY ) {
  fill ( 200 , 30 , 255 ); // Make it purple
  ellipse ( posX , posY , 50 , 50 ); // Draw the ball
}

function setup () {
  createCanvas ( 400 , 400 );
  x = random ( 0 , width );
  y = random ( 0 , height );
}

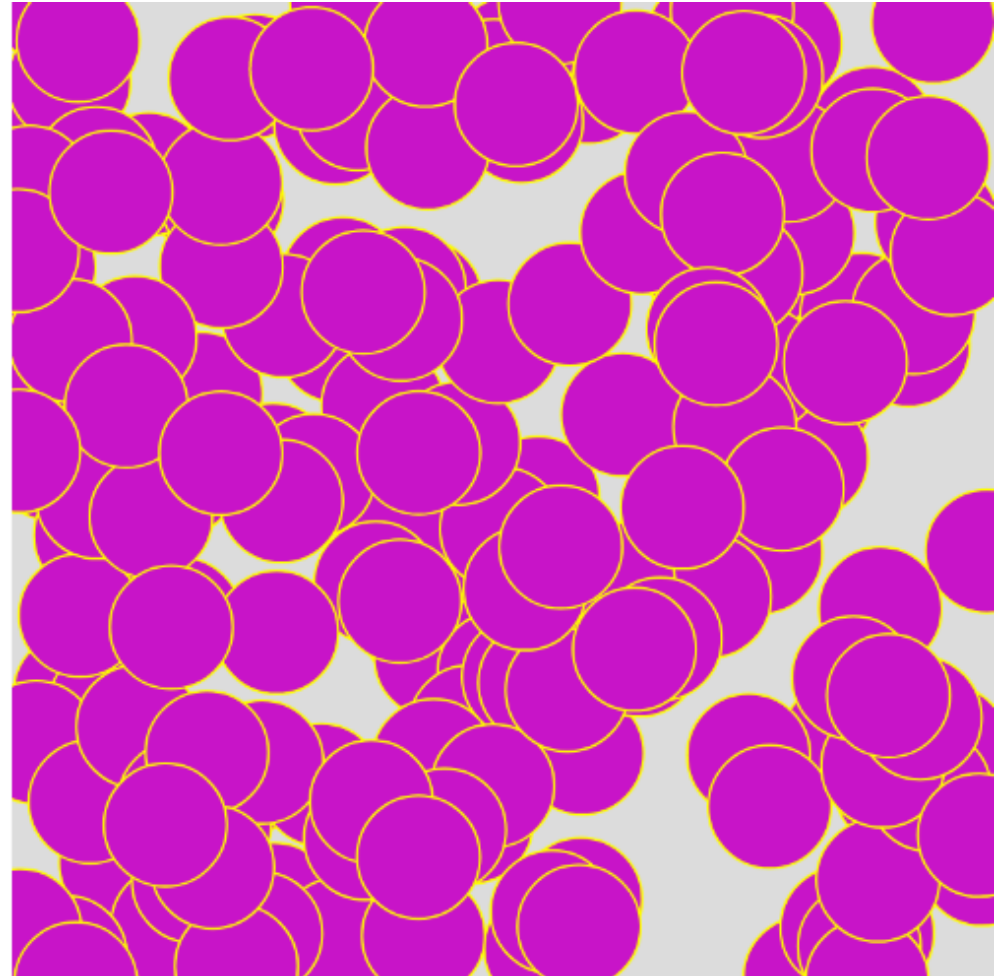
function draw() {
  background(220);
  drawBall ( x , y ); // Draw a ball
}
```



실습 : random 위치에 공 그리기

- 실습-5의 예제에서 공 모양을 그리는 코드를 함수로 작성하고 draw()에서 호출 하시오.

```
1  var x=0;
2  var y=0;
3
4  function setup() {
5    createCanvas(400, 400);
6    background(220);
7
8  }
9
10 function draw() {
11   x = random(0, width);
12   y = random(0, height);
13
14   drawBall(x, y); //함수 호출
15 }
16
17 function drawBall(posX, posY){
18   fill(200, 20, 200);
19   stroke('yellow');
20   ellipse(posX, posY, 50, 50);
21 }
```



배열 : Array

- 배열은 한 번에 여러 값들을 저장하는 특별한 변수
- 여러 개의 값이나 속성들을 하나의 목록에 저장하여 처리
- 배열 선언 또는 배열 선언과 초기화

```
var fruits = [];
```

```
var fruits = ['apple', 'pear', 'banana', 'cherry'];
```

Array 배열

- 배열 관련 함수들

함수	기능
<code>array.push(value)</code> <code>array[2] = 'value';</code>	<ul style="list-style-type: none">배열의 끝에 값을 추가하여 그 길이를 1씩 확장
<code>array.pop()</code>	<ul style="list-style-type: none">배열의 마지막 요소를 제거
<code>array.splice(index, num)</code>	<ul style="list-style-type: none">배열의 특정 index의 요소 개수 만큼 제거
<code>array.length</code>	<ul style="list-style-type: none">배열의 길이 출력

- javascript 배열 관련 정보 살펴보기

- https://www.w3schools.com/js/js_array_methods.asp

실습 : arrayTest()함수 작성

- sports를 전역 변수 배열로 선언하시오.
- arrayTest()함수를 작성하시오.
 - sports 배열에 다음의 값을 저장하시오.
 - soccer
 - volley ball
 - base ball
 - marathon
 - bicycle
 - sports 배열에서 배열 요소들을 console에 출력하시오.

```
Console
0 : soccer
1 : volley ball
2 : base ball
3 : marathon
4 : bicycle
```


실습 : arrayTest()함수 작성

Console

```
0 : soccer
1 : volley ball
2 : base ball
3 : marathon
4 : bicycle
```

```
1  var x=0;
2  var y=0;
3  var sports = [];
4
5  ▼ function setup() {
6      createCanvas(400, 400);
7      background(220);
8      arrayTest();
9      //noLoop();
10
11 }
12
13 ▼ function draw() {
14 }
15
16 ▼ function arrayTest(){
17     sports.push('soccer');
18     sports.push('volley ball');
19     sports.push('base ball');
20     sports.push('marathon');
21     sports.push('bicycle');
22
23     for(let i=0; i<sports.length; i++){
24         print(i + " : " + sports[i]);
25     }
26 }
```

Object(객체) : 객체지향 프로그래밍

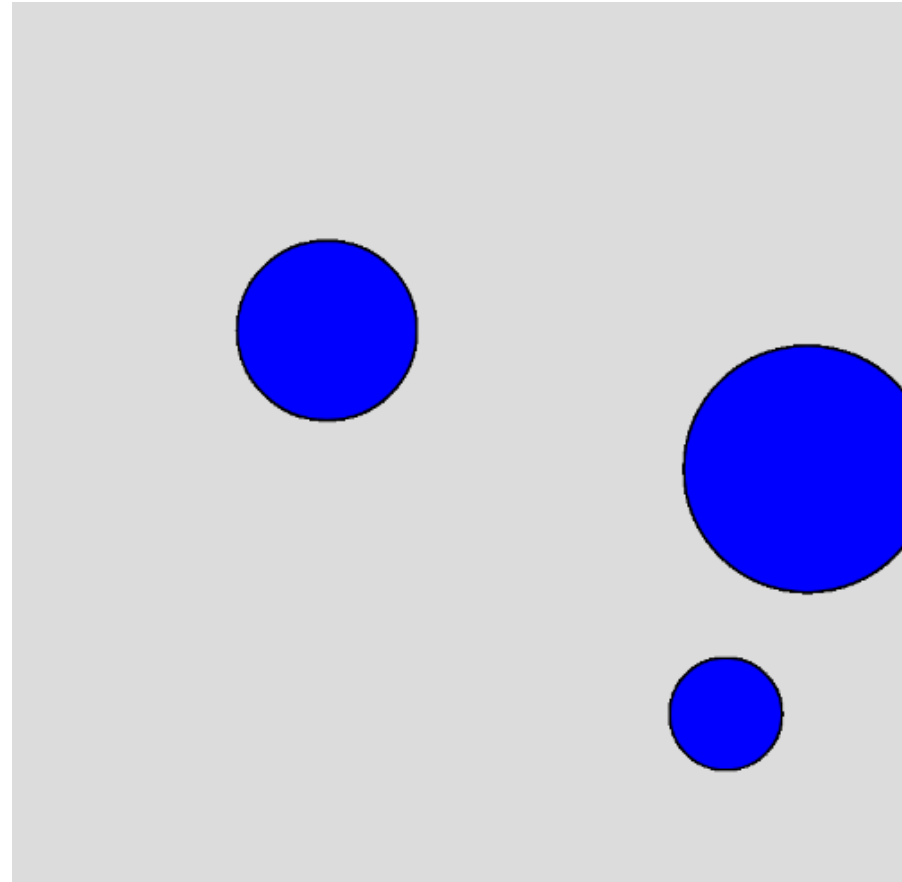
- 공을 그리는 Ball 객체를 작성한다.
 - 매개변수 : size
 - 멤버 속성
 - this.size
 - this.x
 - this.y
 - 멤버 메소드 :
 - this.display : 각자가 원하는 색상의 원을 그리는 메소드
 - this는 자기 자신의 객체를 가르킴

```
function Ball(size){  
  this.size = size;  
  this.x = random(0, width);  
  this.y = random(0, height);  
  
  this.display = function(){  
    fill(200, 20, 200);  
    stroke('yellow');  
    ellipse(this.x, this.y, this.size, this.size);  
  }  
}
```

실습 : 공을 객체로 만들어 출력

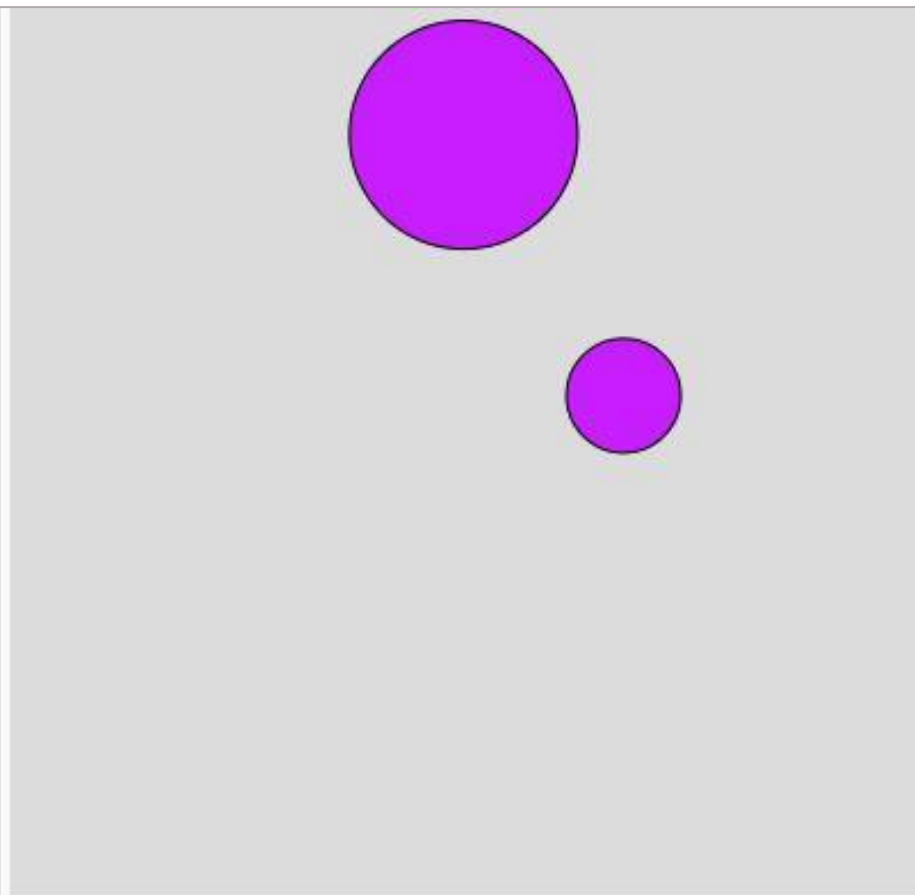
- 앞에서 선언한 객체 Ball을 크기를 달리하여 3개 생성하여 화면에 출력하시오.

```
1  var x;  
2  var y;  
3  var myBall1;  
4  var myBall2;  
5  var myBall3;  
6  
7  function setup() {  
8    createCanvas(400, 400);  
9    background(220);  
10  
11    myBall1 = new Ball(50);  
12    myBall2 = new Ball(80);  
13    myBall3 = new Ball(110);  
14  }  
15  function draw() {  
16    myBall1.display();  
17    myBall2.display();  
18    myBall3.display();  
19  }
```



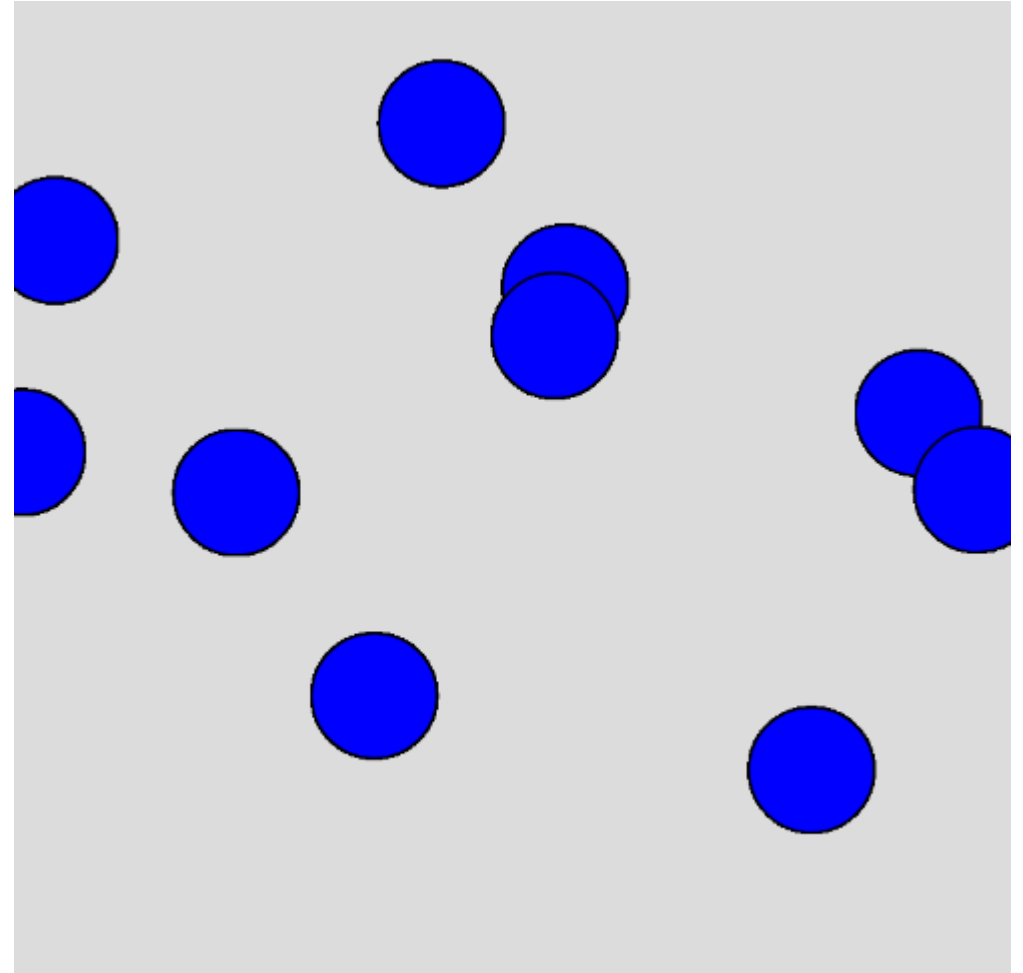
실습 : 공 2개를 랜덤 위치에 출력하기

```
1 var myBall ; // Declare a Ball object
2
3 ▼ function setup () {
4   createCanvas (400 , 400);
5   myBall = new Ball ( 50 ); // Initialize the Ball object
6   yourBall = new Ball(100);
7 }
8
9 ▼ function draw () {
10  background (220);
11  myBall.display (); // Display the ball
12  yourBall.display();
13 }
14
15 ▼ function Ball (s) {
16  this.size = s;
17  this.x = random (0 , width);
18  this.y = random (0 , height);
19  // Draw a purple ball
20
21 ▼ this.display = function () {
22   fill(200, 30, 255); // Make it purple
23   ellipse (this.x, this.y, this.size ,this.size ); // Draw the ball
24 }
25 }
```



실습 : 객체를 배열로 만들어 출력

- 배열을 전역 변수로 선언 : balls
- setup() 에서 Ball 객체 10개를 배열에 생성 하시오.
- draw() 에서 배열에 생성한 Ball객체를 화면에 출력 하시오



실습 : 객체를 배열로 만들어 출력

```
1 var x;  
2 var y;  
3 var balls = [];  
4  
5 ▼ function setup() {  
6     createCanvas(400, 400);  
7     background(220);  
8  
9 ▼     for(let i = 0; i < 10; i++){  
10         balls.push(new Ball(50));  
11     }  
12  
13 }
```

```
14 ▼ function draw() {  
15 ▼     for(let i = 0; i < 10; i++){  
16         balls[i].display();  
17     }  
18 }  
19  
20 ▼ function Ball(size){  
21     this.size = size;  
22     this.x = random(0, width);  
23     this.y = random(0, height);  
24  
25 ▼     this.display = function(){  
26         fill('blue');  
27         circle(this.x, this.y, this.size);  
28     }  
29 }
```

if-else

- 코드의 흐름 제어

```
let i = 10;  
if(i>0){  
    print('양수');  
}else{  
    print('음수');  
}
```

If-Else

비교 연산자	설명
==	• equal to
===	• equal value and equal type
!=	• not equal
!==	• not equal or not equal type
>	• greater than
<	• less than
>=	• greater than or equal to
<=	• less than or equal to
?	• 삼항 연산자.

논리 연산자	설명
&&	• and
	• or
!	• not

Events

- 이벤트(event)
 - 프로그램에 의해 감지되고 처리될 수 있는 동작이나 사건
 - 마우스를 움직이거나, 요소를 클릭하거나, 텍스트 박스에 글을 쓰는 등 수많은 종류의 동작(action)이 이벤트를 발생시킴
- 이벤트 핸들러(event handler)
 - 이벤트 발생시 이벤트를 처리하는 프로그램
 - 이벤트와 연결되어있음

Events

- reference

- <https://p5js.org/ko/reference/>

이벤트

가속도

deviceOrientation
accelerationX
accelerationY
accelerationZ
pAccelerationX
pAccelerationY
pAccelerationZ
rotationX
rotationY
rotationZ
pRotationX
pRotationY
pRotationZ
turnAxis
setMoveThreshold()
setShakeThreshold()
deviceMoved()
deviceTurned()
deviceShaken()

키보드

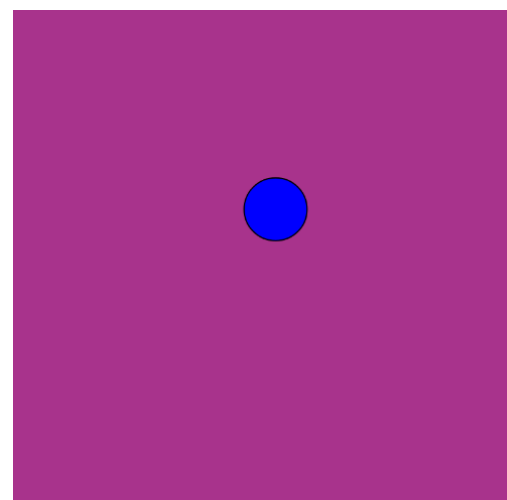
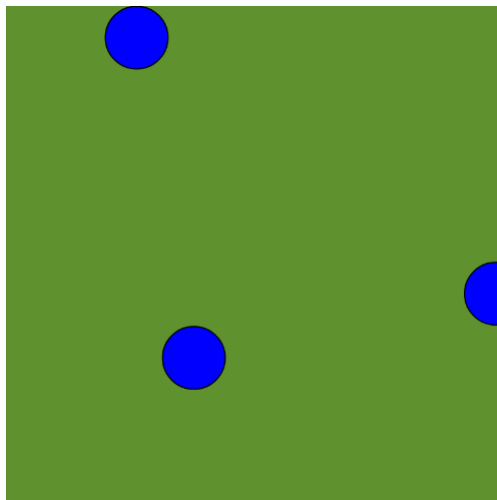
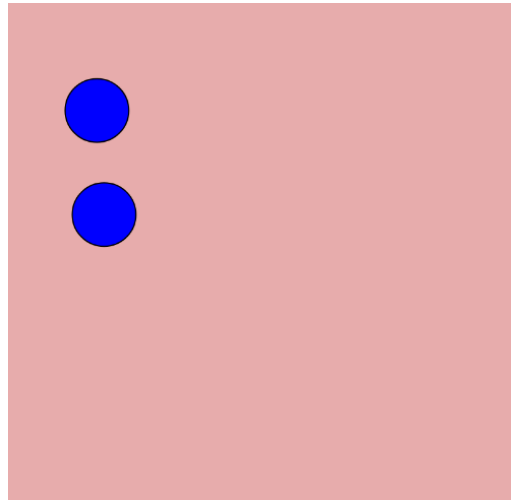
keyIsPressed
key
keyCode
keyPressed()
keyReleased()
keyTyped()
keyIsDown()

마우스

movedX
movedY
mouseX
mouseY
pmouseX
pmouseY
winMouseX
winMouseY
pwinMouseX
pwinMouseY
mouseButton
mouseIsPressed
mouseMoved()
mouseDragged()
mousePressed()
mouseReleased()
mouseClicked()
doubleClicked()
mouseWheel()
requestPointerLock()

실습 : 마우스 클릭 이벤트 적용

- `mouseClicked()` 이벤트를 사용하여 마우스를 클릭하면 배경색에 임의의 색으로 변하도록 추가하시오.
- `console` 창에는 색상이 변경되었음을 출력하시오.



실습 : 마우스 클릭 이벤트 적용

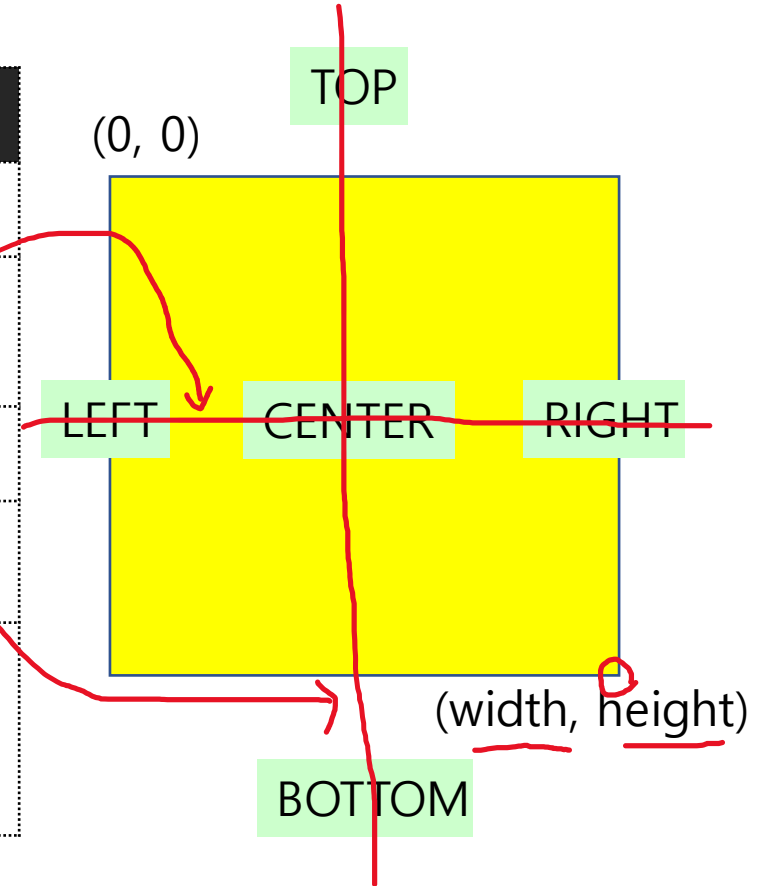
```
1 var x;  
2 var y;  
3 var balls = [];  
4 var c='yellow';
```

```
32 ▼ function mouseClicked() {  
33     c = color(random(0,255),random(0,255),random(0,255))  
34     print("배경색이 변했어요^^");  
35 }
```

Text

■ 브라우저 화면에 문자열 출력 함수

함수	기능 및 사용법
text(str, x, y, width, height)	<ul style="list-style-type: none">화면에 텍스트 출력
textAlign(a, b)	<ul style="list-style-type: none">a : LEFT / CENTER / RIGHTb : TOP / BOTTOM / CENTER
textSize(pixel)	<ul style="list-style-type: none">폰트 사이즈
str(value)	<ul style="list-style-type: none">값(Boolean, string, number)을 문자열 표현으로 변환
nf(num, left, right)	<ul style="list-style-type: none">숫자를 문자열로 형식화(실수, 정수)left : 소수점 기준 왼쪽 자리 수(정수 부분)right ; 소수점 기준 오른쪽 자리 수(소수 부분)



Video 파일(mp4) 출력

Auto-refresh

Paint hero

Sketch Files

index.html

sketch.js

style.css


video

test.mp4

sketch.js

```
1 let video;
2 let playing = false;
3
4 function setup() {
5   createCanvas(400, 300);
6   video = createVideo("video/test.mp4");
7   video.size(400, 300);
8   video.hide();
9 }
10
11 function draw() {
12   background(50);
13   image(video,0,0);
14 }
15
16
17 function mousePressed() {
18   if (!playing) {
19     video.play();
20     playing = true;
21   }
22   else {
23     video.pause();
24     playing = false;
25   }
26 }
```

Preview



Console

Clear

Image, Text 출력



cat_dog.jpg 파일 업로드

p5* File Edit Sketch Help

Auto-refresh image_cat_dog by jhkim3217


Sketch Files

- images
 - cat_dog.jpg
- index.html
- sketch.js
- style.css

```
1 let img;
2
3 function preload() {
4   img = loadImage('/images/cat_dog.jpg')
5   print('image loaded!!!')
6 }
7
8 function setup() {
9   createCanvas(400, 400);
10 }
11
12 function draw() {
13   background(220);
14   imageMode(CENTER);
15   image(img, 200, 200);
16
17   textSize(30);
18   textAlign(LEFT, CENTER);
19   fill(255)
20
21   print('width', width, 'height', height)
22   text("Hello AI", width-400, height-380);
23
24   textAlign(RIGHT, CENTER);
25   text("P5JS Coding!", width-8, height-380);
26 }
```

Saved: just now Preview

Hello AI P5JS Coding!



Console

image loaded!!!

715 width 400 height 400

Video, Text 출력



☒ Auto-refresh

Metal sousaphone



sketch.js

Preview

```
1 let video;
2
3 function setup() {
4   createCanvas(windowWidth, windowHeight);
5   print('windowWidth', windowHeight)
6   // createCanvas(400, 300);
7   video = createCapture(VIDEO);
8   video.hide();
9
10 }
11
12 function draw() {
13   background(220);
14   image(video, 0, 0)
15
16   textSize(30);
17   textAlign(LEFT, CENTER);
18   fill(255)
19
20   print('width', width, 'height', height)
21   text("Hello AI", width-620, height-830);
22
23   textAlign(RIGHT, CENTER);
24   text("P5JS Coding!", width-30, height-830);
25 }
```

Console

Clear

5000 width 633 height 858

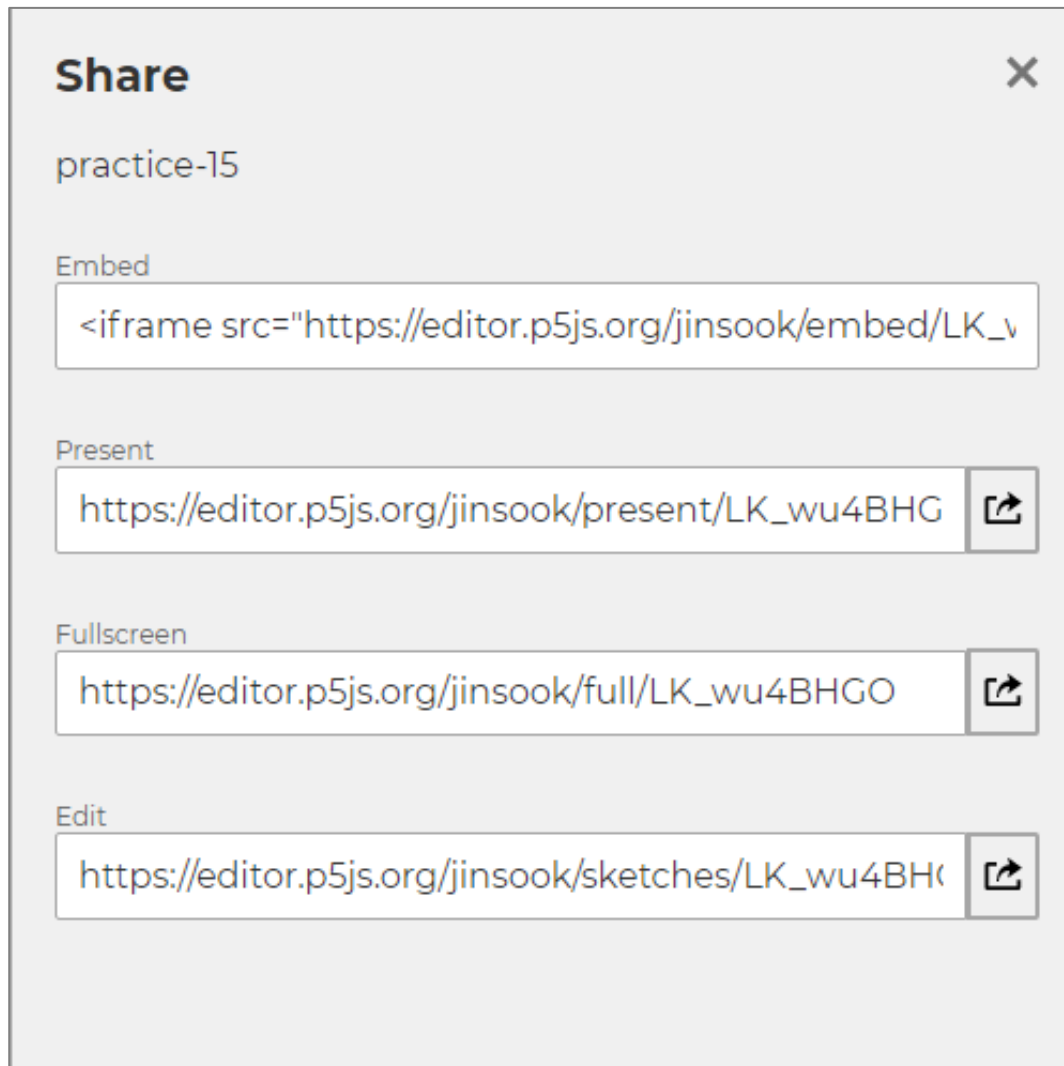


Images / Sound

- 이미지와 사운드를 위한 함수

함수	기능 및 사용법
function preload()	<ul style="list-style-type: none">• setup() 함수 실행 전에 외부 파일들을 비동기적으로 로딩• preload()함수는 loadImage, loadJSON, loadFont, loadString 등 load calls를 하는데 만 사용
loadImage(img)	<ul style="list-style-type: none">• 지정된 경로에서 이미지를 로드하고 p5.Image 생성• 외부 파일을 로드 할 때는 로컬 서버를 사용하는 것이 좋음
image(img, x, y, width, height)	<ul style="list-style-type: none">• 캔버스에 이미지를 그림
soundFormats(format)	<ul style="list-style-type: none">• 삽입하고자하는 소리 파일 형식 리스트• format : 'mp3', 'wav', 'ogg'• p5.sound 라이브러리를 포함해야 함• <script src="path/to/p5.sound.js"> </script>
loadSound()	<ul style="list-style-type: none">• 지정된 경로에서 새 p5.SoundFile을 반환• 외부 파일을 로드 할 때는 로컬 서버를 사용하는 것이 좋음
sound.play()	<ul style="list-style-type: none">• 소리 파일 실행

코드 활용 하기



HTML 페이지에 `<iFrame>`으로 삽입하기

웹 에디터에서 보기

- 참고 자료

- p5.js 레퍼런스

- <https://p5js.org/ko/reference>

- An Introduction to P5.js

- https://stijnwolters.nl/resources/files/P5_Manual.pdf