

2021 지역ICT 이노베이션 확산사업 인공지능 교육

반응속도 확인 게임 만들기 (p5.js)

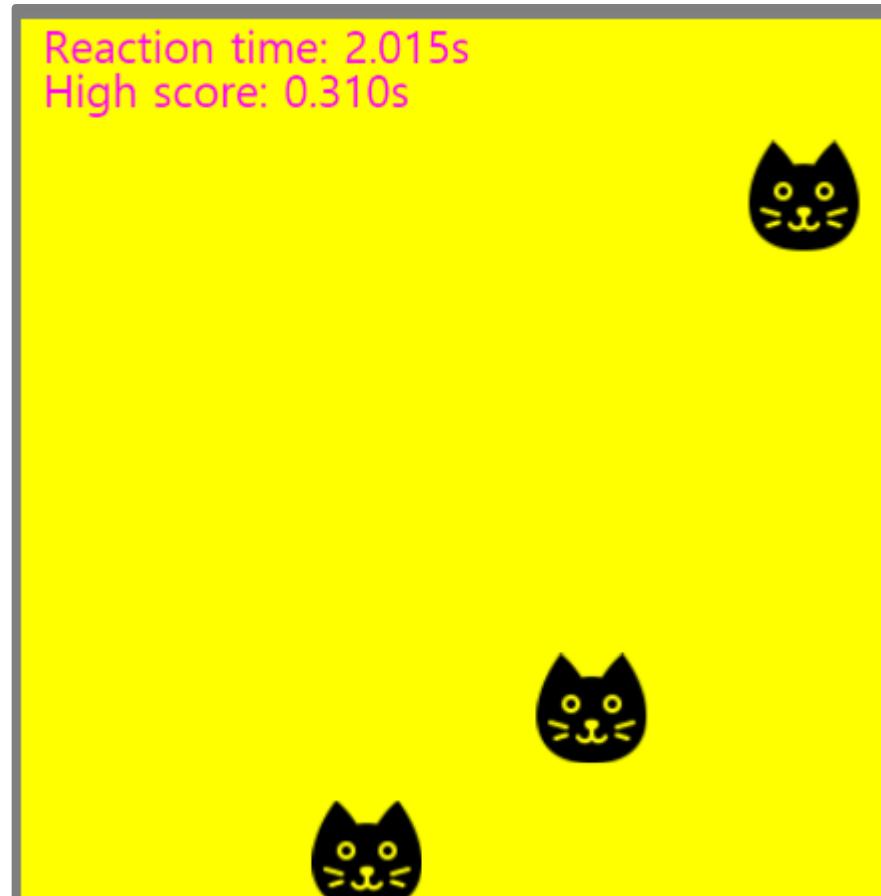
동의과학대학 컴퓨터정보과
김진숙

학습 내용

- **p5.js란?**
 - 실습1-에디터랑 친해지기
- **Drawing Shapes**
 - 실습2-도형 그리기
- **Color**
 - 실습3-색상 입히기
- **Variables**
 - 실습4-전역 변수 사용하기
- **random()함수**
 - 실습5-임의의 위치에 도형 그리기
- **Functions**
 - 실습6-공 그리기
- **Arrays**
 - 실습7-arrayTest()함수 작성
- **Loops**
 - 실습8-공을 객체로 만들어 출력
 - 실습9-객체를 배열로 만들어 출력
- **Objects**
 - 실습10-게임 규칙에 맞게 기능 추가
- **If-else**
 - 실습11-마우스 클릭이벤트 적용
- **Events**
 - 실습12-클릭된 공 객체 제거
 - 실습13-반응속도 관련 변수, 함수
- **Texts**
 - 실습14-displayScore함수 정의/호출
- **Images/Sound**
 - 실습15-게임에 그림과 소리 추가
- **코드 활용하기**
- **웹사이트에 게시하기**

학습 목표

- p5.js의 기본기 익히기
- p5.js로 반응 속도 확인 게임 만들기



p5.js 체험하기

- 예제 사이트 확인하기
 - <https://p5js.org/ko/examples/>

p5.js 란?

- 웹 상에서 영상, 인터랙션(상호작용) 등을 코딩으로 쉽게 구현할 수 있도록 기능을 제공하는 javascript 라이브러리
- Processing 제작진이 만들어 Processing과 비슷한 명령어 체계 가짐
 - 모이라 터너 [Moira Turner](#) 가 리드하며, 로렌 맥카시 [Lauren Lee McCarthy](#) 가 창안하고 협력자 커뮤니티와 함께 개발
- p5.js 등록하기

```
<head>
  <!--웹상의 p5.js 라이브러리 가져오기 -->
  <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.2.0/p5.js"></script>

  <!--같은 폴더에 있는 p5.js 라이브러리 가져오기 -->
  <script src="p5.js"></script>

</head>
```

cloudflare

CDN 서비스와 분산네임서버를 이용하여 사이트 성능과 속도, 보안을 향상시키는 서비스를 제공하는 기업

```
<script
  src="https://cdn.jsdelivr.net/npm/p5@[p5_version]/lib/p5.js">
</script>
```

jsdelivr

오픈 소스 프로젝트를 위한 무료 공개 CDN

편집 툴

- 온라인 웹 에디터 사이트
 - <https://editor.p5js.org/>
 - 회원가입
 - 구글이나 깃허브 계정으로도 로그인 가능
 - 최대 250MB 사용 가능
 - 레퍼런스 제공
 - <https://p5js.org/ko/reference/>
- 데스크톱 에디터
 - Visual Studio Code, Sublime, Brackets, Atom, Notepad++, Eclipse 등

웹 에디터 소개

The screenshot shows the p5.js Web Editor interface. Several UI elements are highlighted with colored boxes:

- A green box highlights the top navigation bar with the tabs "File", "Edit", "Sketch", and "Help".
- A blue box highlights the play/pause buttons in the toolbar.
- A purple box highlights the "Sketch Files" sidebar, which lists "index.html", "sketch.js", and "style.css".
- A red box highlights the code editor area containing the following P5.js code:

```
1 function setup() {  
2   createCanvas(400, 200);  
3 }  
4  
5 function draw() {  
6   background(255, 0, 0);  
7 }
```
- A yellow box highlights the "Preview" tab at the top right of the code editor.
- A pink box highlights the "Keyboard Shortcuts" dropdown menu in the top right corner.
- A brown box highlights the "Log in or Sign up" link in the top right corner.
- A magenta box highlights the settings gear icon in the top right corner.
- A blue box highlights the "Console" tab at the bottom left.
- A cyan box highlights the "Clear" button in the bottom right corner of the console.

[단축키]
Ctrl + Enter : 실행
Ctrl + Shift + Enter : 멈춤

English ▾ Log in or Sign up

Auto-refresh Easy diascia

Sketch Files

index.html

sketch.js

style.css

sketch.js

function setup() {
 createCanvas(400, 200);
}

function draw() {
 background(255, 0, 0);
}

Auto-refresh

Keyboard Shortcuts

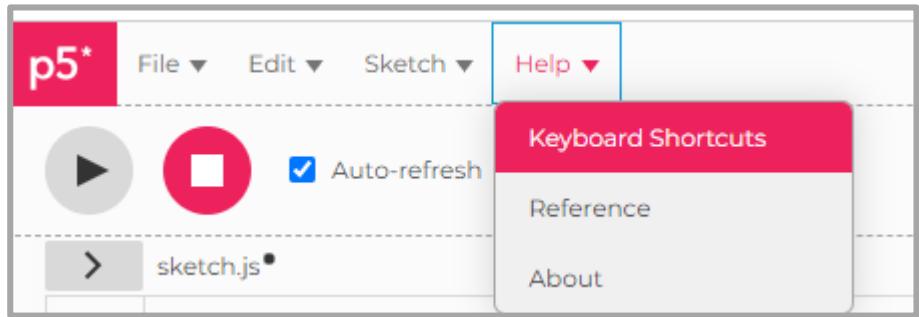
Reference

About

Console

Clear

웹 에디터 소개 - 단축키



김종현

Keyboard Shortcuts

Code Editing

Code editing keyboard shortcuts follow Sublime Text shortcuts.

- ⌃ + ⌂ + F** Tidy
- ⌃ + F** Find Text
- ⌃ + G** Find Next Text Match
- ⌃ + ⌂ + G** Find Previous Text Match
- ⌃ + H** Replace Text Match
- ⌃ + [** Indent Code Left
- ⌃ +]** Indent Code Right
- ⌃ + /** Comment Line
- ⌃ + K** Show Inline Color Picker

General

- ⌃ + S** Save
- ⌃ + Enter** Start Sketch
- ⌃ + ⌂ + Enter** Stop Sketch

- ⌃ + ⌂ + 1** Turn On Accessible Output
- ⌃ + ⌂ + 2** Turn Off Accessible Output

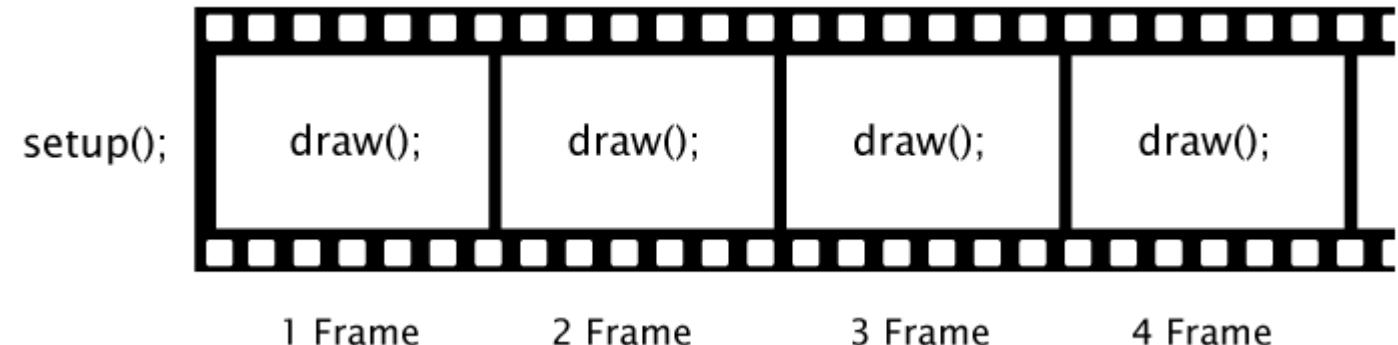
Program structure

- 기본 함수 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
createCanvas(width, height)	<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
background(color)	<ul style="list-style-type: none"> • 배경색 설정 • color : 'red', (255, 0, 0), '#f00', '(rgba(255, 0, 0, 0.5))' 등

실습1-에디터랑 친해지기

1. 웹 에디터를 실행 시켜 전체 구성 살펴보기
2. 제공된 다양한 예제 실행시켜 보기

Drawing Shapes

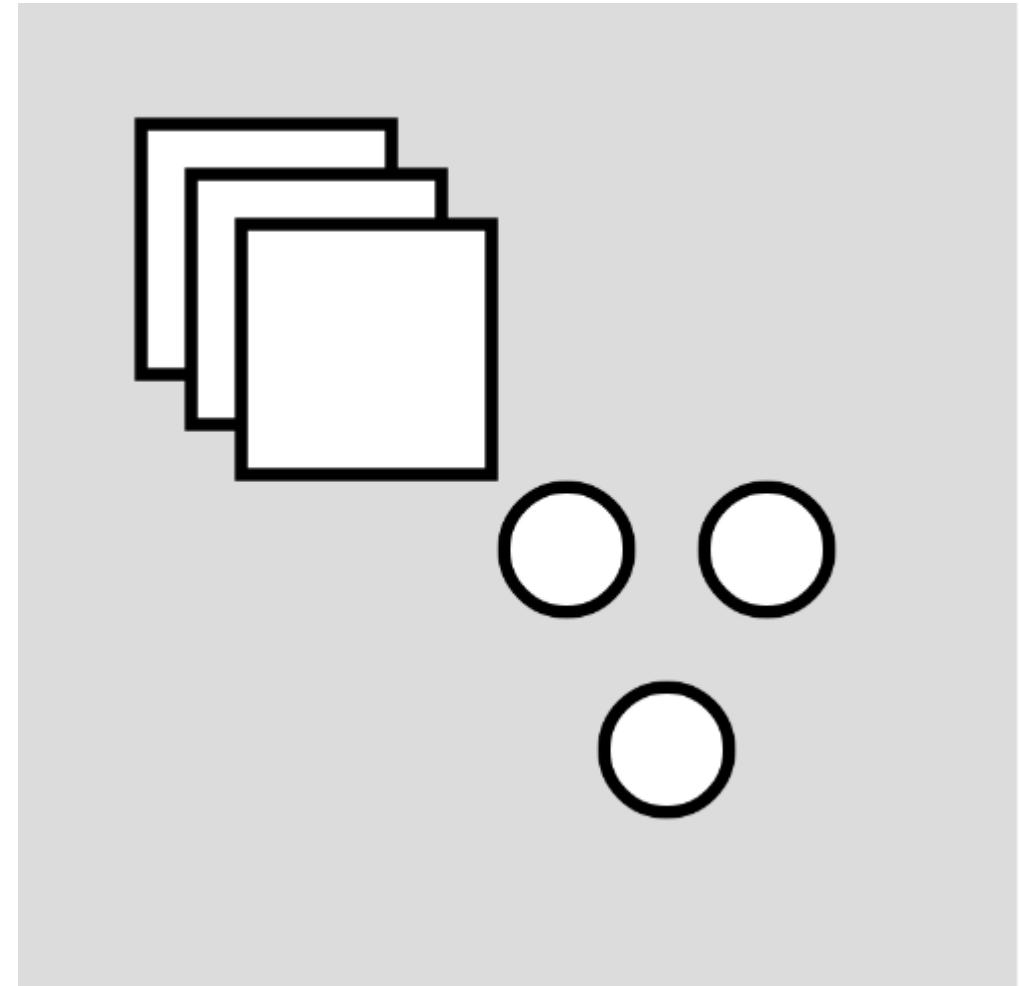
■ 다양한 도형 그리기 함수

함수	기능 및 사용법
rect(x, y, width, height);	<ul style="list-style-type: none">사각형 그리기위치(x, y) / 크기(width, height)
square(x, y, s)	<ul style="list-style-type: none">정사각형 그리기s : 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

실습2-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 }
```



Color

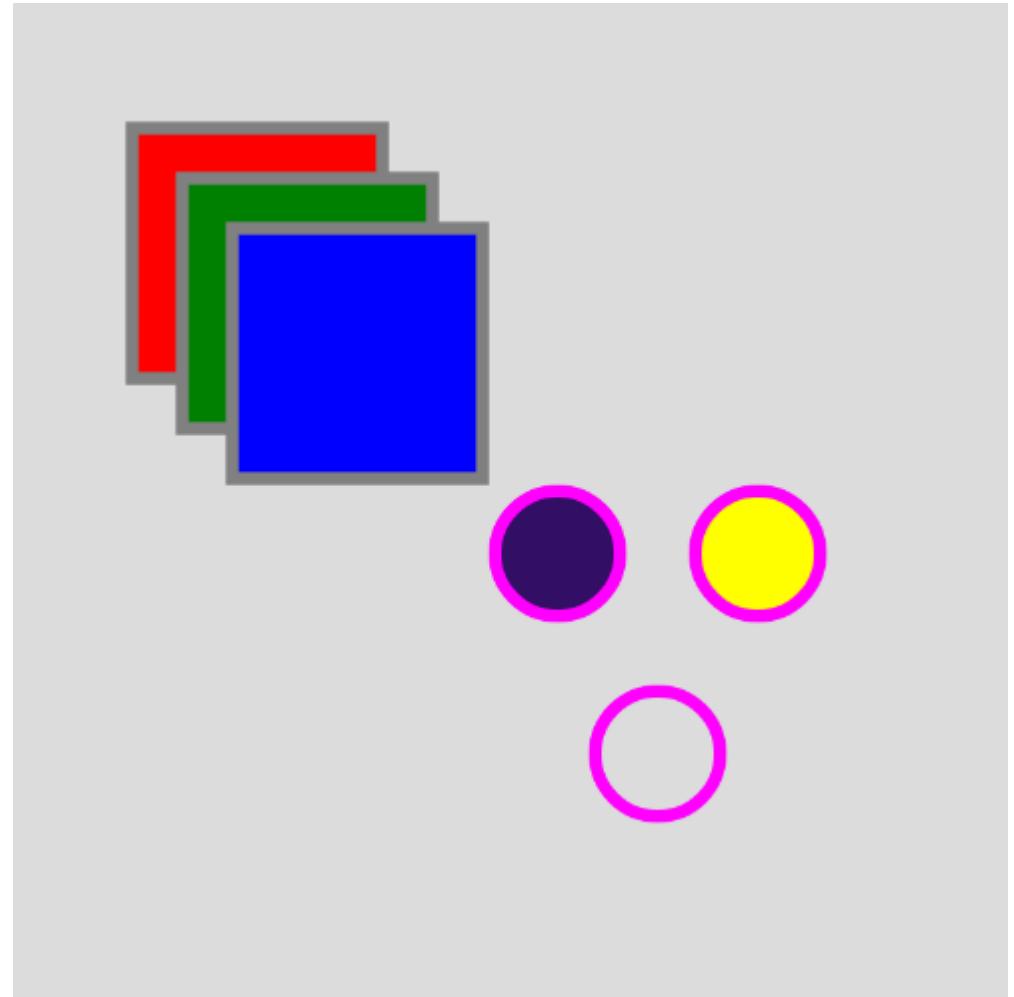
■ 색상 사용하기

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

실습3-색상 입히기

- 실습-2의 도형에게 다양한 색상을 입히세요.

```
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 }
```



Variables

■ 변수 선언

```
var number;      var number = 1;
```

- javascript는 변수 선언 시 변수 타입을 지정할 필요 없음
- 변수 타입을 지정하지 않아도 되는 것은 유연하게 프로그래밍 할 수 있는 기능
이지만 메모리를 과도하게 사용할 우려 있음
- 변수명은 파일 내에서 unique해야 함
- 대소문자 구별
- 변수가 초기화 되지 않으면 그 값은 undefined 임
- 변수에 값을 넣고도 다시 같은 변수명으로 선언을 해도 값이 할당되기 전까지
는 위의 값을 유지

```
var language = "JavaScript";  
var language;
```

Variables

■ 변수 선언

```
var number;      var number = 1;
```

- javascript는 변수 선언 시 변수 타입을 지정할 필요 없음
- 변수 타입을 지정하지 않아도 되는 것은 유연하게 프로그래밍 할 수 있는 기능
이지만 메모리를 과도하게 사용할 우려 있음
- 변수명은 파일 내에서 unique해야 함
- 대소문자 구별
- 변수가 초기화 되지 않으면 그 값은 undefined 임
- 변수에 값을 넣고도 다시 같은 변수명으로 선언을 해도 값이 할당되기 전까지
는 위의 값을 유지

```
var language = "JavaScript";  
var language;
```

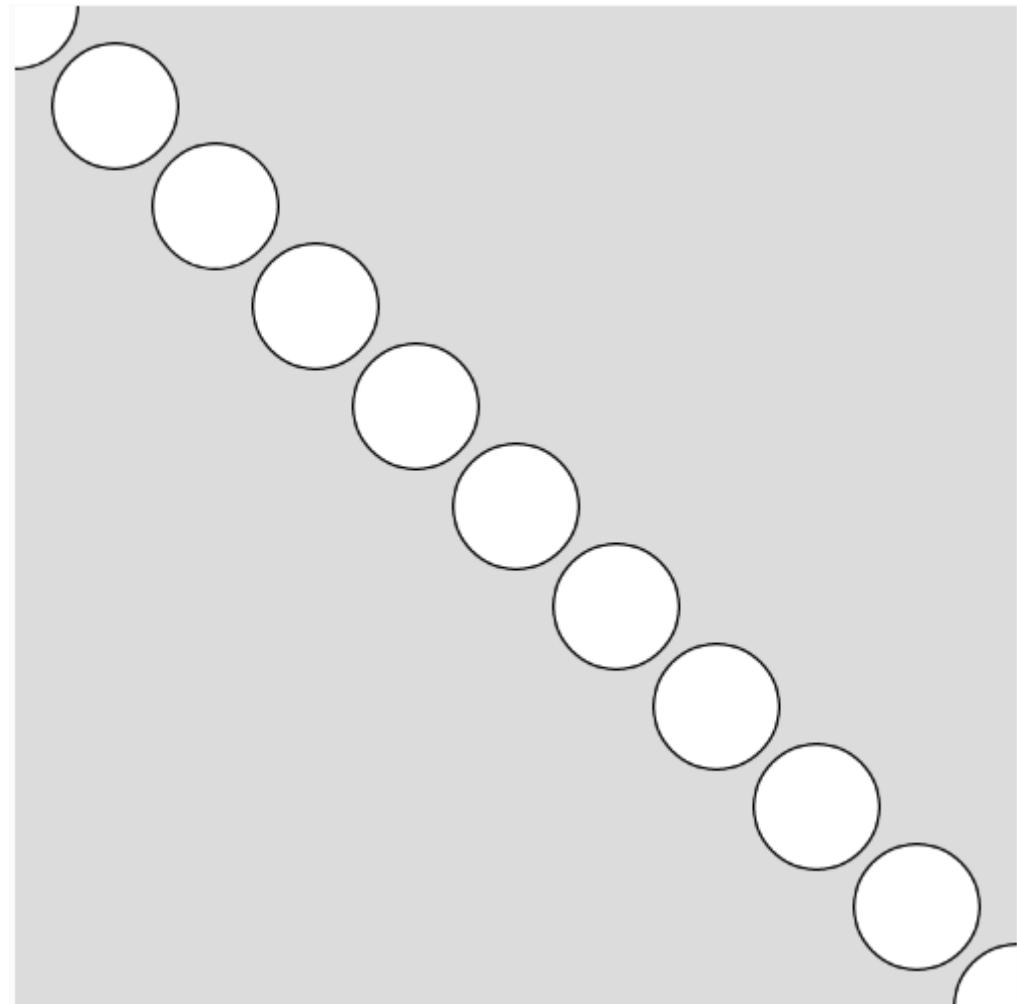
[참고]

Var 대신 let 키워드를 사용하면 같은 변수명을 재사용 할 수 없다.

실습4-전역변수 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 }
```



random([min], [max])

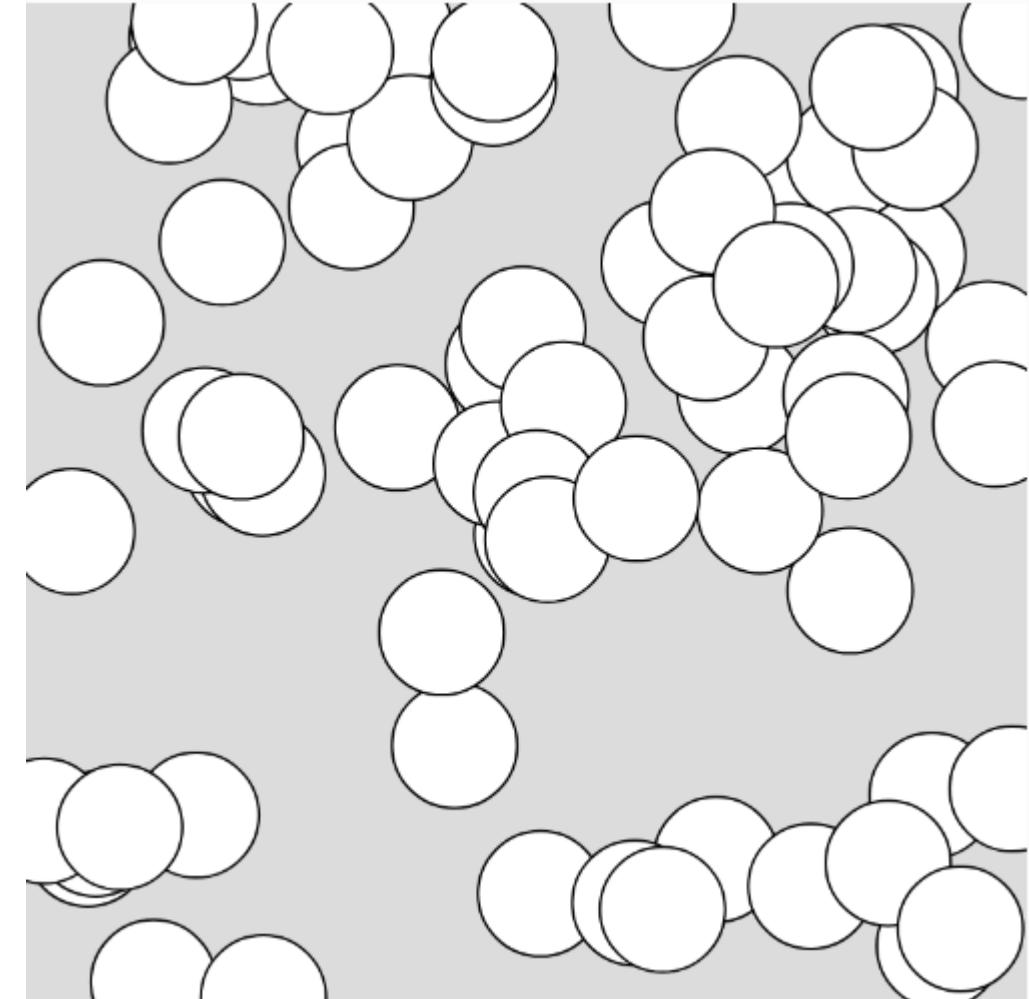
- 임의의 부동 소수점 숫자, 즉 실수(float)를 반환
- 별도의 인수를 지정하지 않을 경우, 0부터 1미만 사이의 난수 반환
 - random();
- 1개의 인수를 배열로 지정한 경우, 해당 배열로부터 임의의 요소 반환
 - var fruit = ['apple', 'bear', 'orange']
 - random(fruit);
- 2개의 인수를 지정한 경우, 1번째 인수에서 2번째 인수 미만 사이의 난수 반환
 - random(0, 10);

실습5-임의의 위치에 도형 그리기

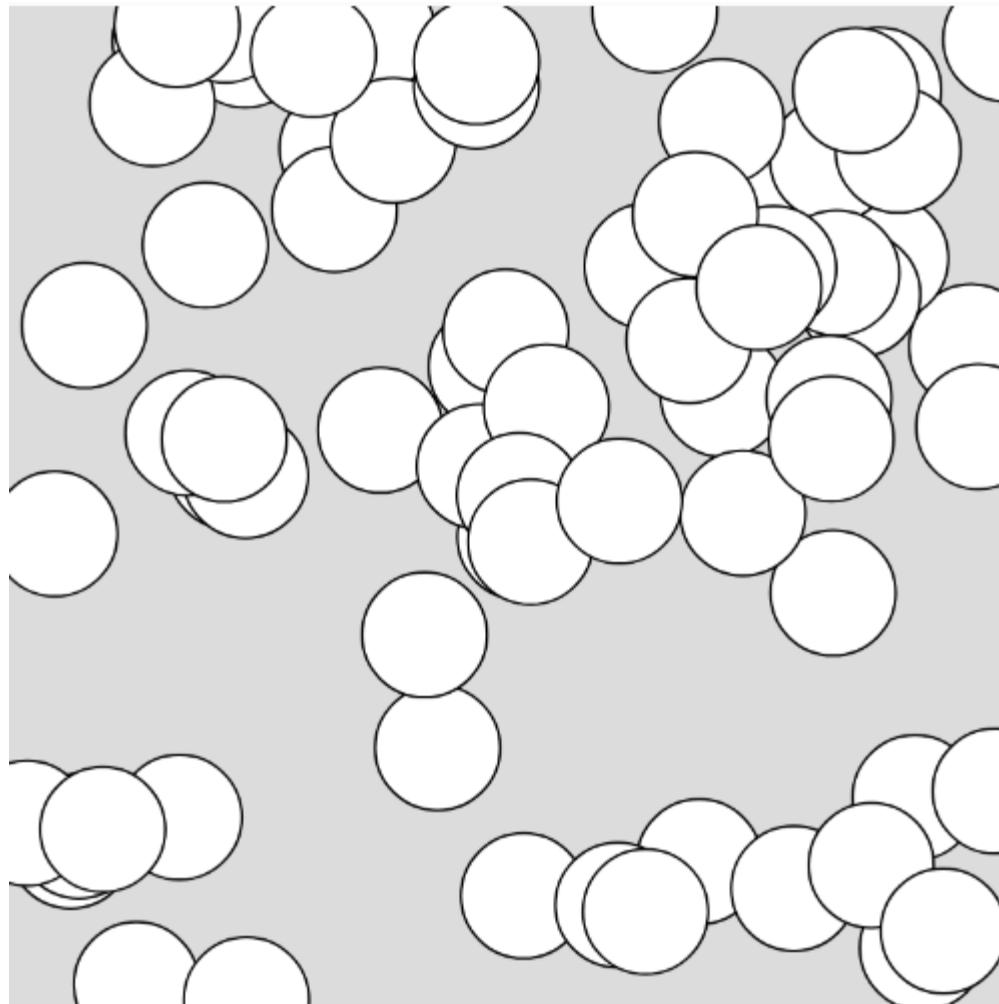
- 전역변수 x, y 값을 캔버스 내 임의의 좌표로 할당하여 동그라미를 그리시오.

시스템 변수	기능
width	<ul style="list-style-type: none">생성된 캔버스의 너비값을 저장한 시스템 변수
height	<ul style="list-style-type: none">생성된 캔버스의 높이값을 저장한 시스템 변수

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



실습5-임의의 위치에 도형 그리기

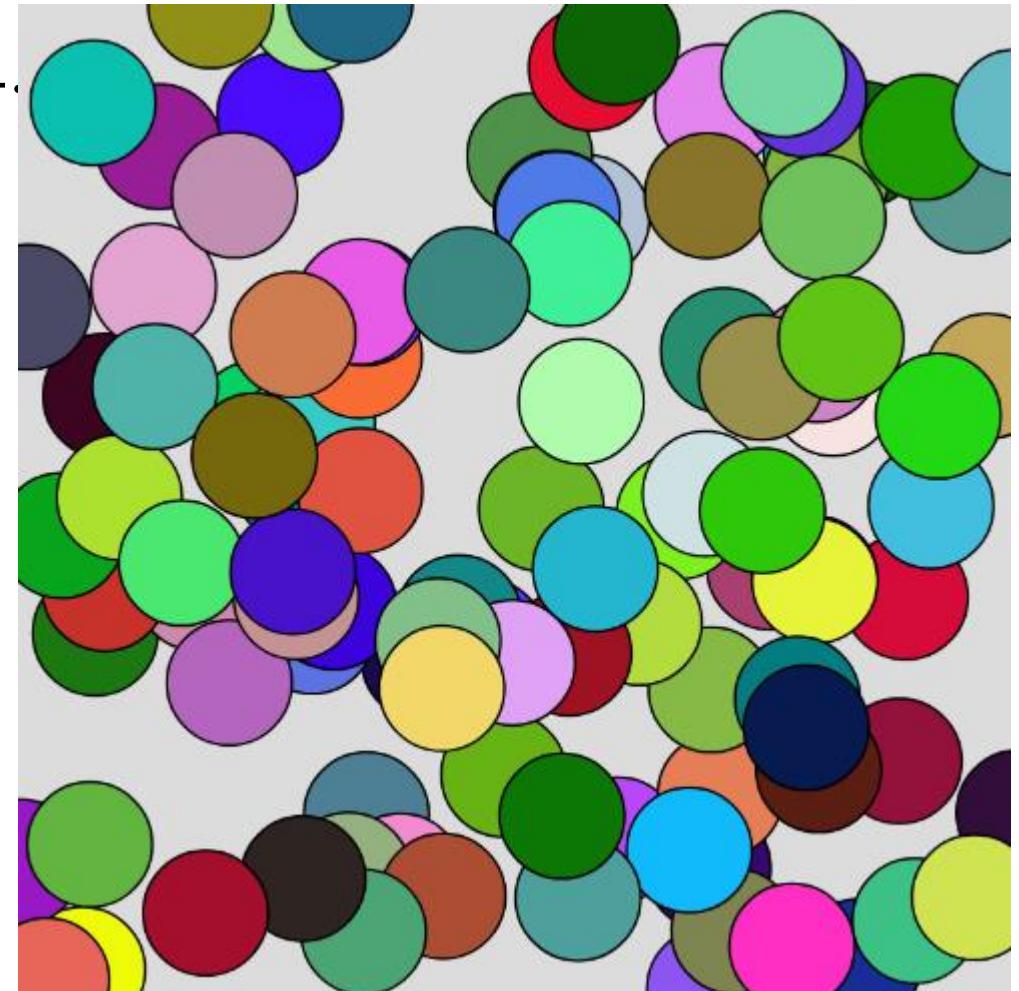


```
1 var x=0;
2 var y=0;
3 var c;
4
5▼ function setup() {
6   createCanvas(400, 400);
7   background(220);
8 }
9
10▼ function draw() {
11   x = random(0, width);
12   y = random(0, height);
13
14   ellipse(x, y, 50, 50);
15 }
```

실습-5-1

- 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 }
```



Functions

■ 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  
}
```

매개변수(parameter)

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

Functions

- 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;  
}  
  
var c = multiply(3, 4); //
```

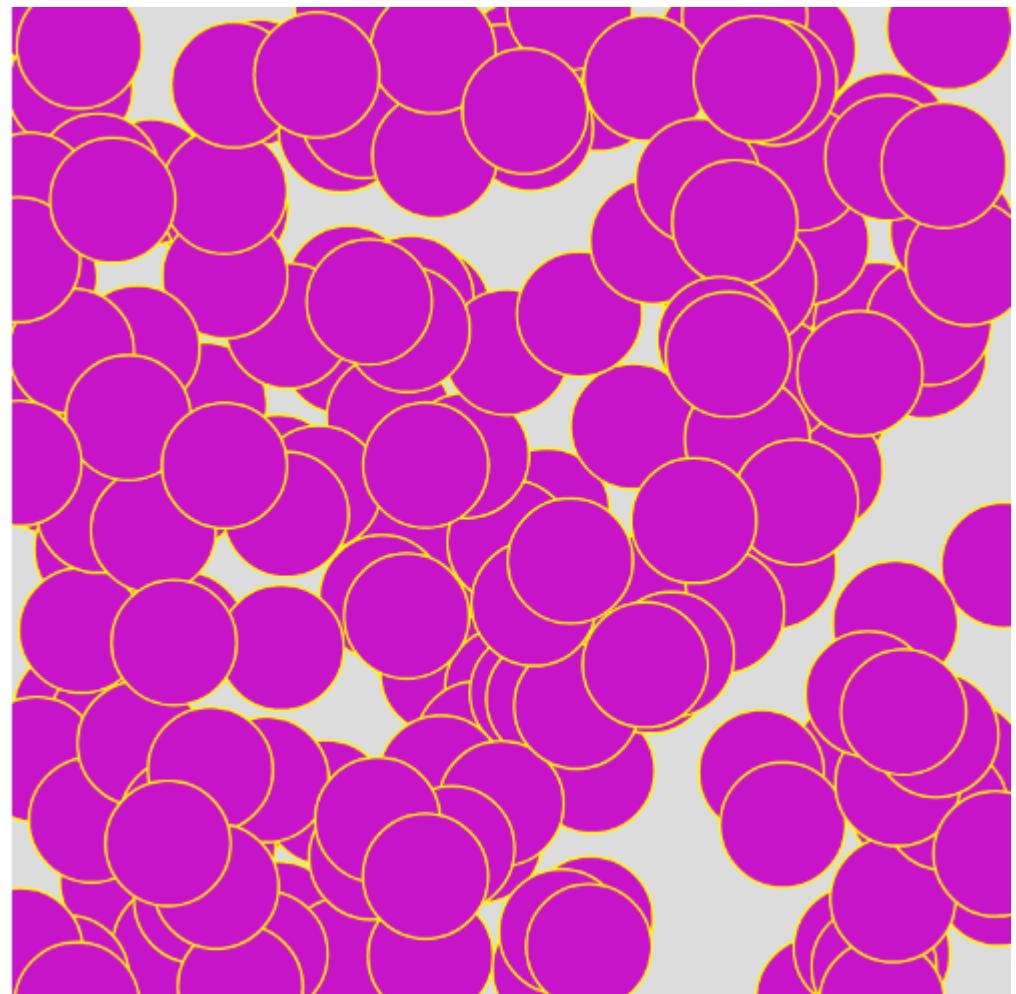
- 함수 호출(사용)

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

실습6-공 그리기

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

```
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 }
```



Arrays

- 배열은 한 번에 여러 값을 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'];
```

Arrays

■ 배열 관련 함수들

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

■ javascript 배열 관련 정보 살펴보기

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

Loops

- while loop

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

- for loop

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

실습7-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 출력 하시오.

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

실습7-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 }
```

Objects

- 객체는 기본적으로 변수와 같은 데이터 타입
- 객체는 데이터(속성)와 그 데이터(속성)를 조작하는 메소드로 구성
- 객체 선언은 함수와 같음

```
function Obj(param1, param2){  
    this.param1 = param1;  
    this.param2 = param2;  
  
    this.display = function(){  
        }  
}
```

- 객체 생성

```
myObj = new Obj(50, 50); //객체 생성  
  
myObj.display(); //객체의 메소드 실행
```

Objects

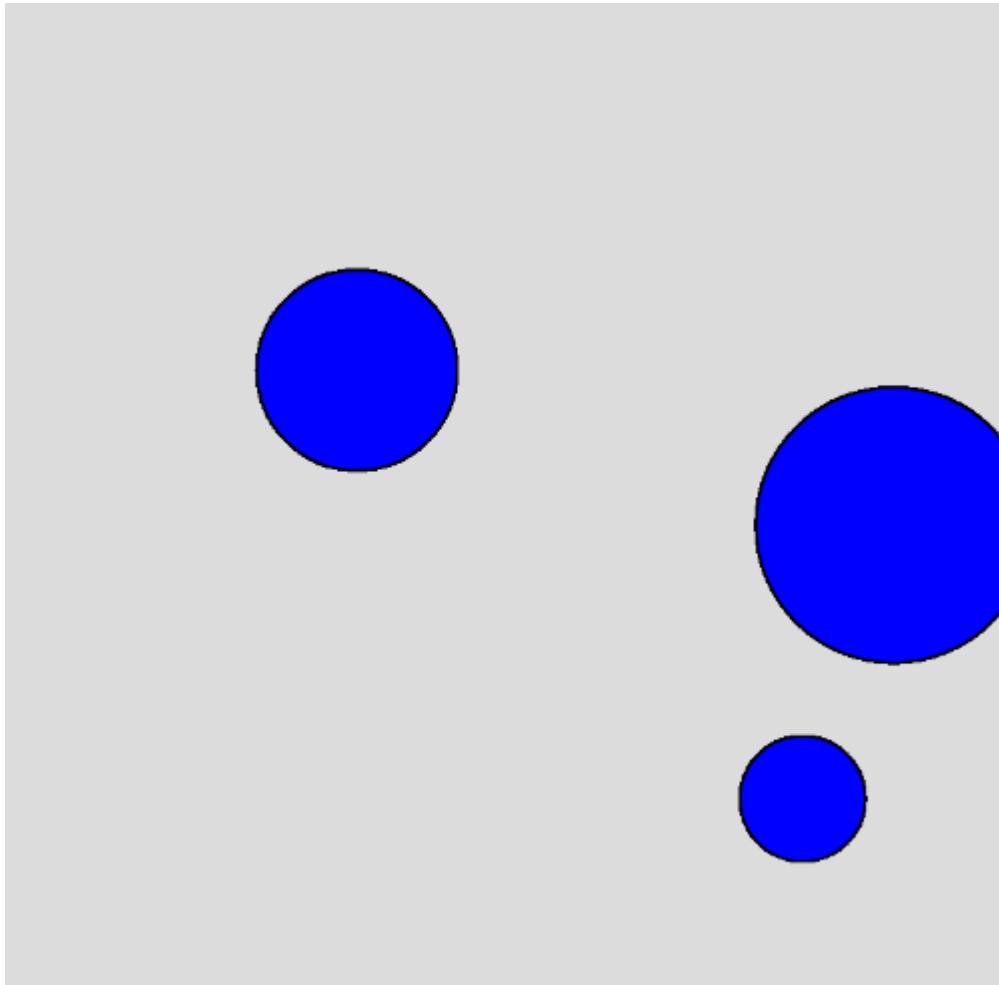
- 공을 그리는 Ball 객체를 작성한다.
 - 매개변수 : size
 - 멤버 속성
 - this.size
 - this.x
 - this.y
 - 멤버 메소드 :
 - this.display : 각자가 원하는 색상의 원을 그리는 메소드

```
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);  
    }  
}
```

실습8-공을 객체로 만들어 출력

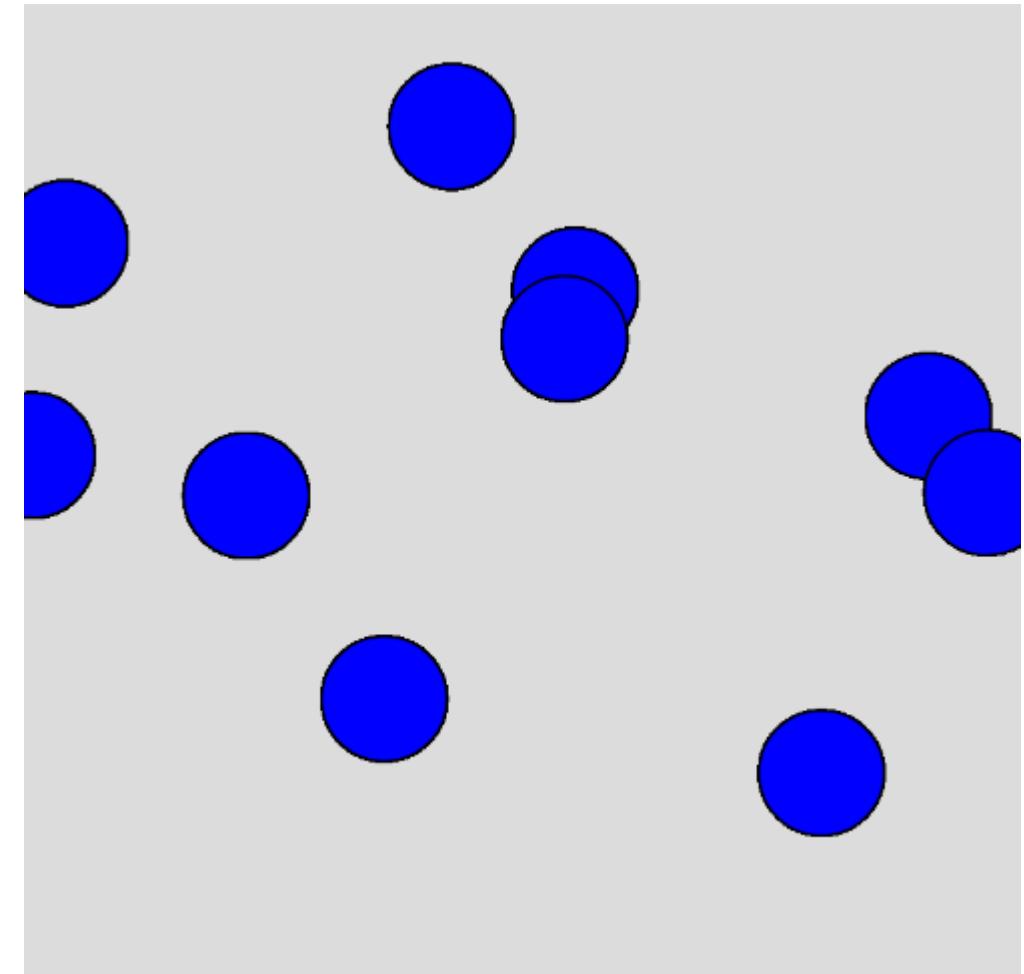
- 앞에서 선언한 객체 Ball을 크기 를 달리하여 3개 생성하여 화면에 출력하시오.

```
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 }
```



실습9-객체를 배열로 만들어 출력

- 배열을 전역 변수로 선언 : balls
- setup()에서 Ball 객체 10개를 배열에 생성하시오.
- draw()에서 배열에 생성한 Ball 객체를 화면에 출력하시오



실습9-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

▪ 조건절에 사용되는 연산자

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

논리 연산자	설명
<code>&&</code>	• and
<code> </code>	• or
<code>!</code>	• not

실습10-게임 규칙에 맞게 기능 추가

■ Ball 객체 속성 추가

- timeToCreate = millis(); // 해당 객체가 만들어진 시간
- timeToLive = 2000; // 객체 수명 설정 : 2초

[참조]

- [millis\(\)](#): 스케치의 시작(즉, setup() 함수 호출 시점) 이후의 시간을 밀리 세컨드(1/1000초)로 반환

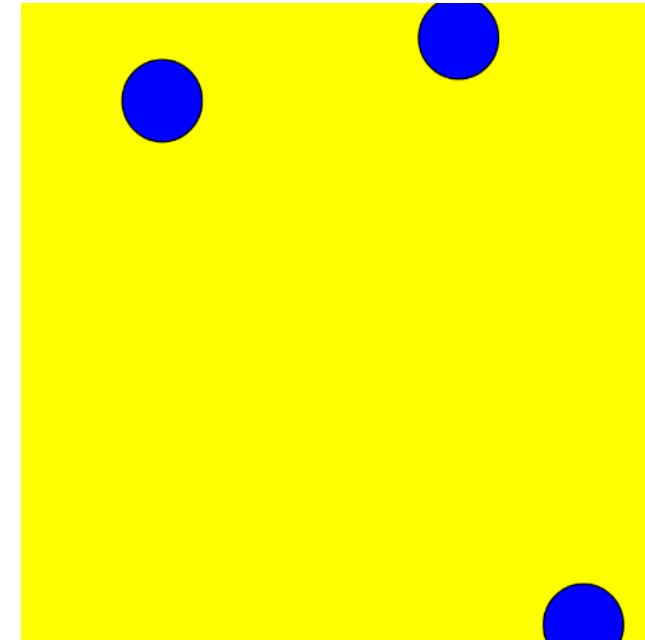
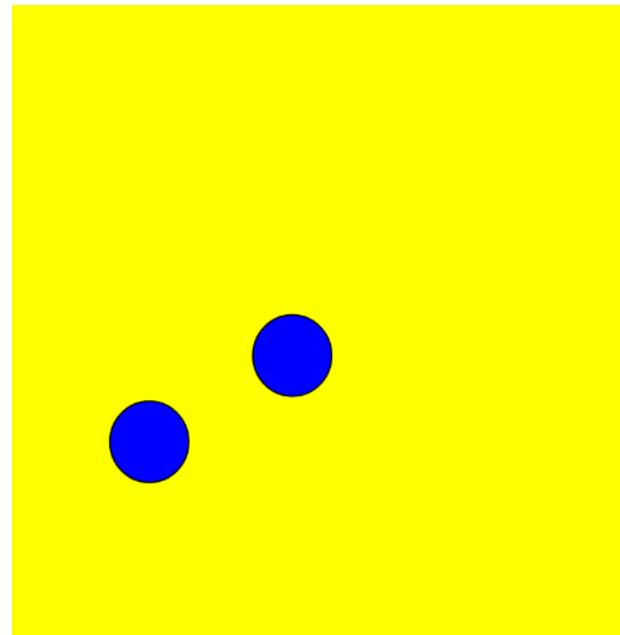
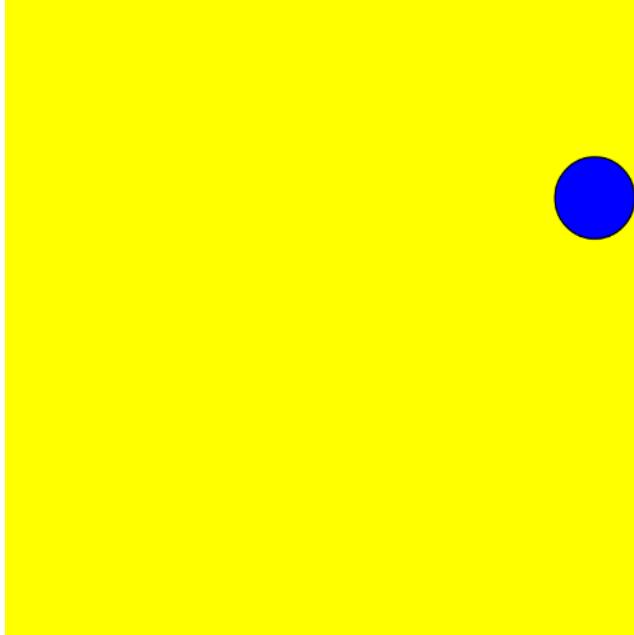
■ 게임 규칙

- 동시에 3개 이상의 Ball 객체가 나타나지 않도록 한다.
- 새로운 객체가 생성되는 조건
 1. 현재 Ball 객체가 3개 미만일 때만 하나의 Ball 객체 생성
 2. 2%의 확률로 객체 생성 : 생성 속도를 조절하기 위해 random() 활용

■ 객체가 삭제되는 조건

- 모든 Ball 객체는 만들어진 뒤 2초가 지나면 제거됨
 - (현재시간 – Ball 객체의 생성 시간) > (Ball 객체의 수명) : 해당 객체는 제거됨

실습10-게임 규칙에 맞게 기능 추가



실습10-게임 규칙에 맞게 기능 추가

```

1 var x;
2 var y;
3 var balls = [];
4
5 function setup() {
6   createCanvas(400, 400);
7 }
```

```

8 function draw() { //Ball을 계속 생성, 제거
9   background('yellow');
10
11   //새로운 Ball 객체 생성
12   //조건: Ball객체의 갯수가 3개보다 적고(AND) 2% 확률로 새 객체 생성(너무 빠르게 생성되지 않도록)
13 if(balls.length <3 && random(0, 100) < 2){
14   balls.push(new Ball(50));
15 }
16
17   //Ball 객체 출력
18 for(var i=0; i<balls.length; i++){
19   balls[i].display()
20
21   //오래된 Ball 객체는 제거
22   //현재초 - 해당 Ball의 생성초(timeToCreate) >= 해당 Ball의 수명(timeToLive)
23 if(millis()-balls[i].timeToCreate >= balls[i].timeToLive){
24   balls.splice(i, 1);
25 }
26 }
27 }
```

```

29 function Ball(size){
30   this.size = size;
31   this.x = random(0, width);
32   this.y = random(0, height);
33
34   this.timeToCreate = millis();
35   this.timeToLive = 2000;
36
37 this.display = function(){
38   fill('blue');
39   circle(this.x, this.y, this.size);
40 }
41 }
```

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

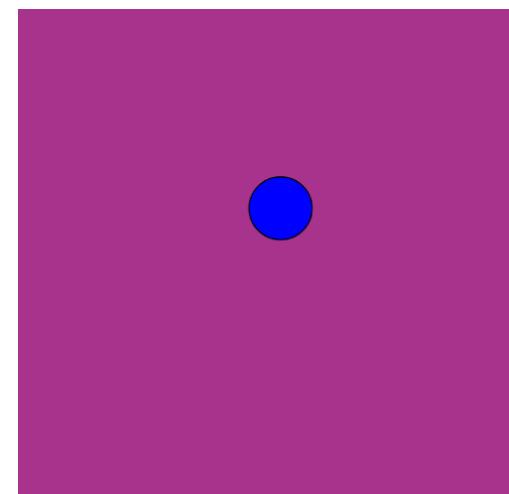
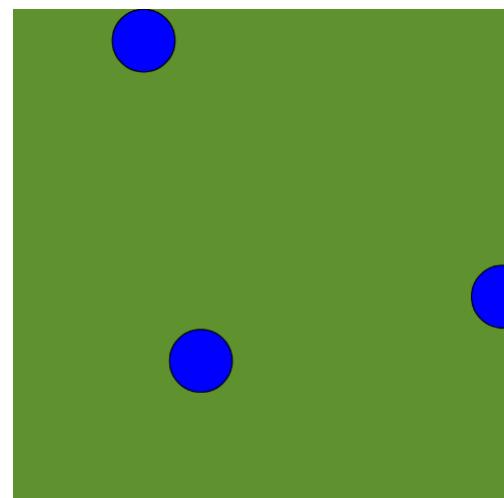
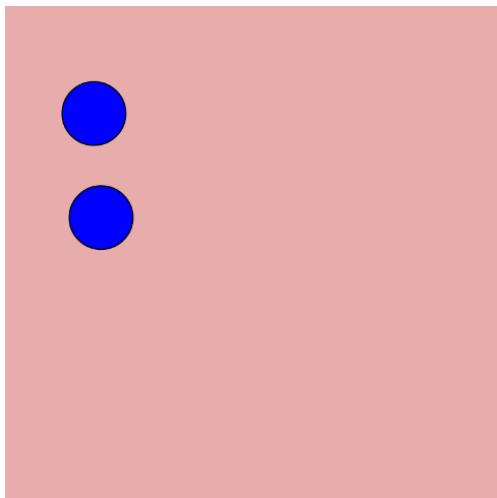
keyIsPressed
key
keyCode
keyPressed()
keyReleased()
keyTyped()
keyIsDown()

마우스

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

실습11-마우스 클릭 이벤트 적용

- 이전 실습(실습-10)에서 mouseClicked() 이벤트를 사용하여 마우스를 클릭하면 배경색에 임의의 색으로 변하도록 추가하시오.
- console 창에는 색상이 변경되었음을 출력하시오.



실습11-마우스 클릭 이벤트 적용

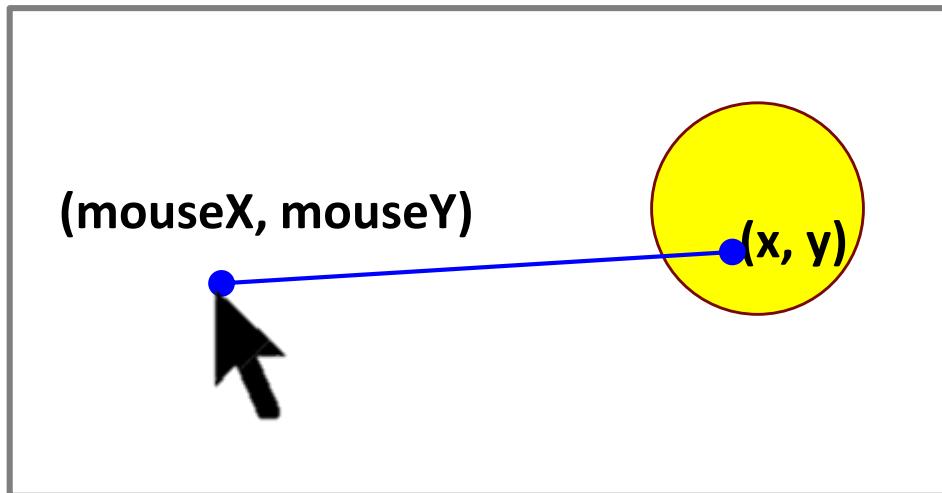
```
1 var x;  
2 var y;  
3 var balls = [];  
4 var c='yellow';
```

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

실습12-클릭된 공 객체 제거

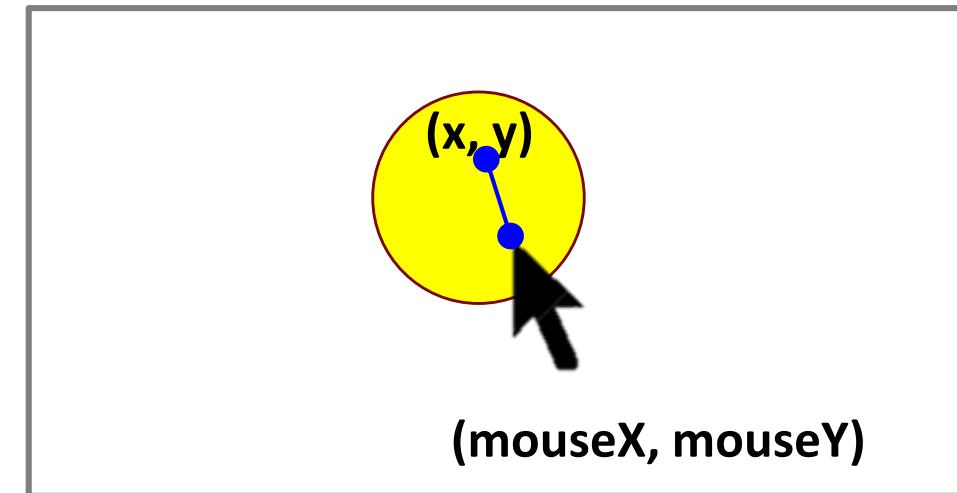
- 마우스가 Ball을 클릭했으면 해당 Ball 제거
 - mouseX, mouseY 는 시스템 변수

- 마우스가 원을 클릭하지 않았을 때



마우스의 위치와 Ball의 위치가 Ball의 크기보다 크다.

- 마우스가 원을 클릭했을 때 => 해당 볼 제거



마우스의 위치와 Ball의 위치가 Ball의 크기보다 작다.

[참고]

- [dist\(x1, y1, x2, y2\)](#)

2차원 또는 3차원 상 두 점 사이의 거리를 계산하는 함수

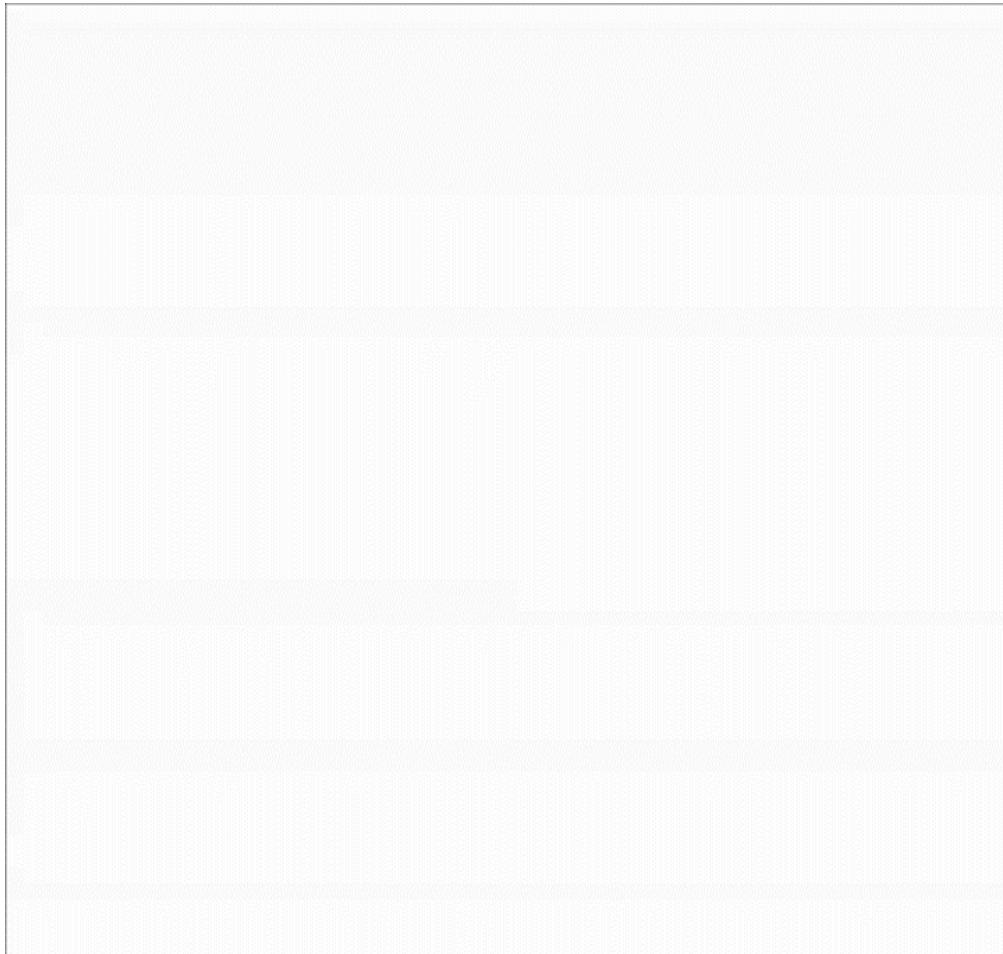
실습12-클릭된 공 객체 제거

- 마우스가 Ball을 클릭했으면 해당 Ball 제거
 - mouseClicked() 이벤트에서 해당 내용 프로그래밍

```
31▼ function mouseClicked() {  
32    var x = mouseX;  
33    var y = mouseY;  
34  
35▼  for(var i=0; i<balls.length; i++){  
36        var distance = dist(x, y, balls[i].x, balls[i].y);  
37  
38▼        if(distance < balls[i].size){  
39            balls.splice(i,1);  
40        }  
41    }  
42 }
```

반응 속도 관련 코드

- 반응 속도 게임은 현재의 반응 속도와 이제까지 가장 빠른 반응속도를 기록한다.



Console

```
score : 0.8965350000653416
highscore : 0.8965350000653416
score : 1.2873400000389665
highscore : 0.8965350000653416
score : 1.1486800001002848
highscore : 0.8965350000653416
score : 0.9898149999789894
highscore : 0.8965350000653416
score : 0.7912999999243766
highscore : 0.7912999999243766
score : 1.0575900000985712
highscore : 0.7912999999243766
```

실습13-반응속도 관련 변수, 함수

- 전역변수
 - score : 현재 반응 속도 저장
 - highscore : 이제까지 가장 빠른 반응 속도 저장
- 반응속도(reactionTime)
 - mouseClicked()에서 Ball 객체를 클릭해 Ball이 제거될 때 반응속도 챌 수 있음
 - 현재시간에서 Ball이 생성된 시간을 빼주면 반응 속도가 됨
 - reactionTime = (마우스클릭현재시간 – Ball의 생성시간)
- setScore(reactionTime) 함수 작성
 - 전역변수 score에 reactionTime 저장
 - 이전 highscore보다 현재 reactionTime이 더 빠르면 highscore에 새 값을 저장

실습13-반응속도 관련 변수, 함수

- 전역변수

```

1 var x;
2 var y;
3 var balls = [];
4 var score;
5 var highscore;
```

- 반응속도(reactionTime)

```

33▼ function mouseClicked() {
34    var x = mouseX;
35    var y = mouseY;
36
37▼ for(var i=0; i<balls.length; i++){
38    var distance = dist(x, y, balls[i].x, balls[i].y);
39
40▼   if(distance < balls[i].size){
41      //반응속도를 second로 변환
42      var reactionTime = (millis() - balls[i].timeToCreate)/1000;
43      setScore(reactionTime); //score에 반응속도값 설정 함수 호출
44
45      balls.splice(i,1); //마우스 클릭한 ball 제거
46    }
47  }
48 }
```

- setScore(reactionTime) 함수

```

50▼ function setScore(time){
51  score = time; //현재값으로 반응속도 저장
52▼  if(highscore==null || highscore> time){ //반응속도가 짧은 것이 highscore가 되어야 함
53    highscore = time;
54  }
55  print('score : ' + score);
56  print('highscore : ' + highscore);
57 }
```

Texts

■ 브라우저 화면에 문자열 출력 함수

함수	기능 및 사용법	
text(str, x, y, width, height)	<ul style="list-style-type: none"> 화면에 텍스트 출력 	(0, 0) TOP
textAlign(a, b)	<ul style="list-style-type: none"> a : LEFT / CENTER / RIGHT b : TOP / BOTTOM / CENTER 	LEFT RIGHT
textSize(pixel)	<ul style="list-style-type: none"> 폰트 사이즈 	
str(value)	<ul style="list-style-type: none"> value(Boolean, string, number)를 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) 	(width, height) BOTTOM

실습14-displayScore()함수 정의/호출

■ 화면 출력 내용

- score : 현재 반응 속도 저장
- highscore : 이제까지 가장 빠른 반응 속도 저장

■ displayScore() 함수 정의

- Canvas에 highscore와 score 를 다음과 같은 포맷으로 출력

Console

```
score : 0.8965350000653416
highscore : 0.8965350000653416
```

score : 0.896s
highscore : 0.896s

1. float 타입을 string으로 변환 : str() 함수 사용
2. 소수점이하 3자리수만 표현 : nf() 함수 사용
3. 초를 나타내는 's' 문자 추가 : + 연산자 사용

■ displayScore() 함수 호출

- draw() 의 마지막에 호출하여 계속 업데이트 되도록 한다.

실습14-displayScore()함수 정의/호출

```
59▼ function displayScore(){
60    var hsText ='High score      : ' + nf(str(highscore),1,3) +' s';
61    var sText = 'Reaction Time : ' + nf(str(score), 1,3) + ' s';
62
63▼ if(highscore != null){
64    textSize(20);
65    fill(0);
66    text(hsText, 10, 20);
67    text(sText, 10, 40);
68 }
69 }
```

```
9▼ function draw() {//Ball을 계속 생성, 제거
10    background('lightgreen');
11
12    //새로운 Ball 객체 생성
13    //조건: Ball객체의 갯수가 3개보다 적고(AND)
14    // 2% 확률로 새 객체 생성(너무 빠르게 생성되지 않도록)
15▼ if(balls.length <3 && random(0, 100) < 2){
16    balls.push(new Ball(50));
17
18
19    //Ball 객체 출력
20▼ for(var i=0; i<balls.length; i++ ){
21        balls[i].display();
22
23        //오래된 Ball 객체는 제거
24        //현재시간 - 생성시간(timeToCreate) >= 수명(timeToLive)
25▼ if(millis()-balls[i].timeToCreate >= balls[i].timeToLive){
26            balls.splice(i, 1);
27        }
28    }
29 }
30 }
```

displayScore();

함수 호출

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 하는데만 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

실습15-게임에 그림과 소리 추가

- 동그라미 모양 공을 고양이 아이콘으로 대체함
- 아이콘을 마우스로 클릭하면 고양이 울음소리가 나도록 함

[유용한 사이트]

- 무료 아이콘 제공 : <https://iconmonstr.com/>
- 무료 소리창고 : <https://pgtd.tistory.com/110>
- MP3 파일을 잘라내기 편집 : <https://mp3cut.net/ko/>

실습15-게임에 아이콘/ 소리/ 버튼 추가

- 동그라미 모양 공을 고양이 아이콘으로 대체함
- 아이콘을 마우스로 클릭하면 고양이 울음소리가 나도록 함
- Pause / Resume 버튼 추가 등 완성도를 위해 다양한 아이디어로 프로그램 개선 할 것

전체 코드

```

1 var balls = [];
2 var score;
3 var highscore;
4 var img;
5 var snd;
6
7 function preload(){
8   img = loadImage('cat.png');
9
10  soundFormats('mp3');
11  snd = loadSound('cat.mp3');
12}
13
14 function setup() {
15   createCanvas(400, 400);
16
17   pauseButton = createButton('Pause');
18   pauseButton.position(0, 450);
19   pauseButton.mouseClicked(pause);
20
21   resumeButton = createButton('Resume');
22   resumeButton.position(60, 450);
23   resumeButton.mouseClicked(resume);
24 }

```

```

26 function pause(){
27   noLoop();
28 }
29 function resume(){
30   loop();
31 }
32
33 function draw() {//Ball을 계속 생성, 제거
34   background('lightgreen');
35
36   //새로운 Ball 객체 생성
37   //조건: Ball객체의 갯수가 3개보다 적고(AND)
38   // 2% 확률로 새 객체 생성(너무 빠르게 생성되지 않도록)
39 if(balls.length < 3 && random(0, 100) < 2){
40   balls.push(new Ball(50));
41 }
42
43   //Ball 객체 출력
44 for(var i=0; i<balls.length; i++){
45   balls[i].display();
46
47   //오래된 Ball 객체는 제거
48   //현재시간 - 생성시간(timeToCreate) >= 수명(timeToLive)
49 if(millis()-balls[i].timeToCreate >= balls[i].timeToLive){
50   balls.splice(i, 1);
51 }
52 }
53
54 displayScore();
55 }

```

```

56▼ function mouseClicked() {
57  var x = mouseX;
58  var y = mouseY;
59
60▼ for(var i=0; i<balls.length; i++){
61    var distance = dist(x, y, balls[i].x, balls[i].y);
62
63▼  if(distance < balls[i].size){
64    snd.play();
65    //반응속도를 second로 변환
66    var reactionTime = (millis() - balls[i].timeToCreate)/1000;
67
68    setScore(reactionTime); //score에 반응속도값 설정 함수 호출
69
70    balls.splice(i,1); //마우스 클릭한 ball 제거
71  }
72}
73}

```

```

75▼ function setScore(time){
76  score = time; //현재값으로 반응속도 저장
77
78  //반응속도가 짧은 것이 highscore가 되어야 함
79▼  if(highscore==null || highscore > score){
80    highscore = score;
81  }
82}

```

```

84▼ function displayScore(){
85  var hsText ='high score : ' + nf(str(highscore),1,3) + ' s';
86  var sText = 'reaction Time : ' + nf(str(score), 1,3) + ' s';
87
88▼  if(highscore != null){
89    textSize(15);
90    fill(255,0, 255);
91    text(hsText, 10, 20);
92    text(sText, 10, 40);
93  }
94 }

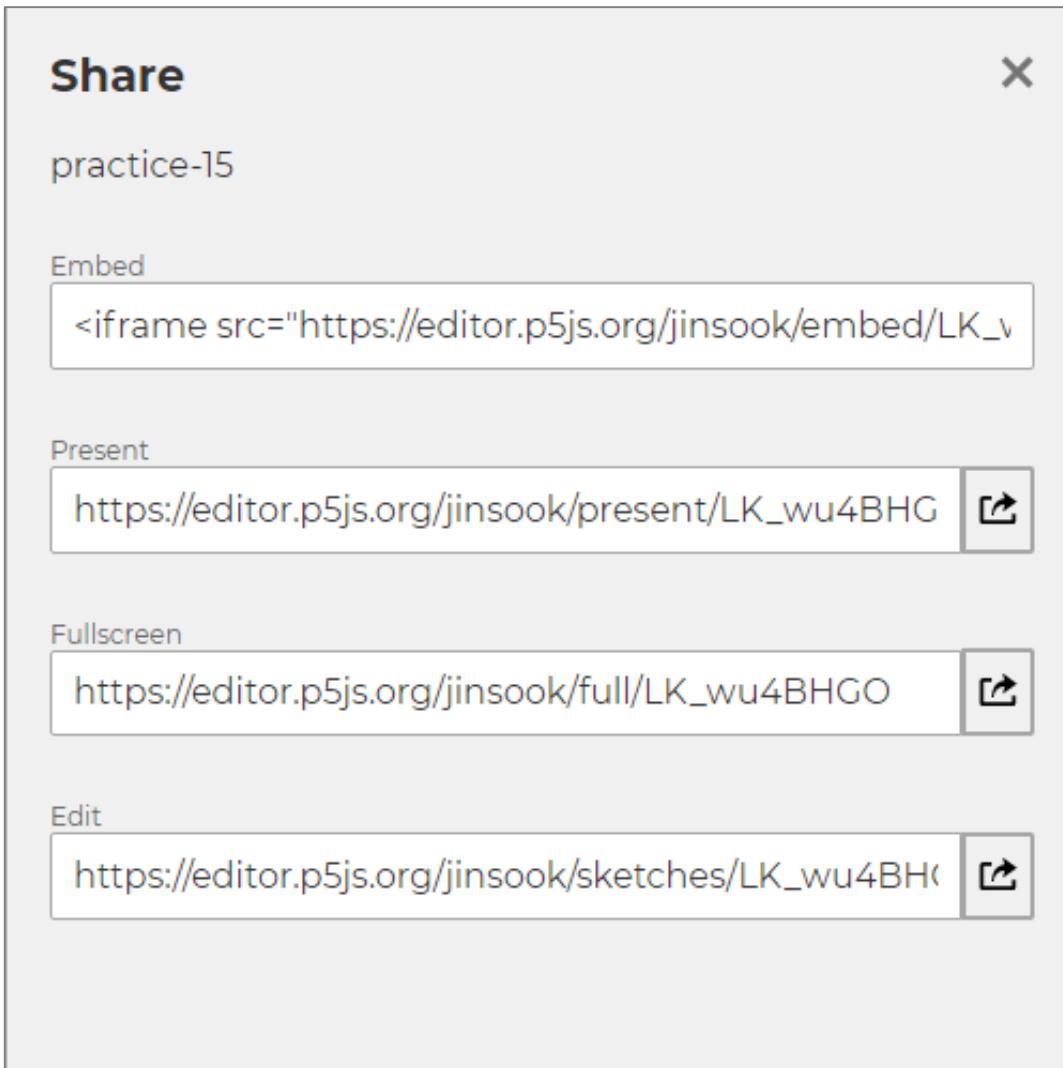
```

```

96▼ function Ball(size){
97  this.size = size;
98  this.x = random(0, width);
99  this.y = random(0, height);
100
101  this.timeToCreate = millis();
102  this.timeToLive = 2000;
103
104▼  this.display = function(){
105    //fill('blue');
106    //circle(this.x, this.y, this.size);
107    image(img, this.x, this.y, this.size, this.size);
108  }
109 }

```

코드 활용하기



HTML 페이지에 <iFrame>으로 삽입하기

웹 에디터에서 보기

웹사이트에 게시하기

1. 전체 폴더를 다운로드
2. 압축파일 풀기
3. 자신의 github 웹사이트에 게시하기

■ 참고 자료

- p5.js 레퍼런스

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

- An Introduction to P5.js

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

**Thank you for your
attention!!**