

Week 05

인터랙션, 변수, 조건문

이 준 환 / 오 종 환

서울대학교 _ 연합전공 정보문화학

Quiz 1

Assignment 2 리뷰

- ♦ Short presentation
 - ♦ 자기 작품을 화면에 띄우면 10-60초간 짧은 설명
 - ♦ 자기 이름 소개
 - ♦ 인터랙션이 있을 경우 조작법
 - ♦ 왜 이 작품을 만들었는지
 - ♦ 기타 하고 싶은 말
 - ♦ 다른 사람 작품에 질문이 있을 경우 바로 하기

지난 시간 리뷰

- ✦ 좌표계: 픽셀단위, 좌측 상단 원점
- ✦ 그림을 그리는 함수들
 - ✦ 점, 선, 사각형, 타원, 다각형 등
- ✦ 주석문
- ✦ 콘솔에 표시하기: `println()`, `print()`
- ✦ `color`의 표현
 - ✦ `background()`, `stroke()`, `fill()`
 - ✦ 0 ~ 255
- ✦ `static mode`, `active mode`
 - ✦ `setup()`과 `draw()`

오늘 다룰 내용

- 인터랙션
- 변수
- 조건문

인터랙션 (interaction)

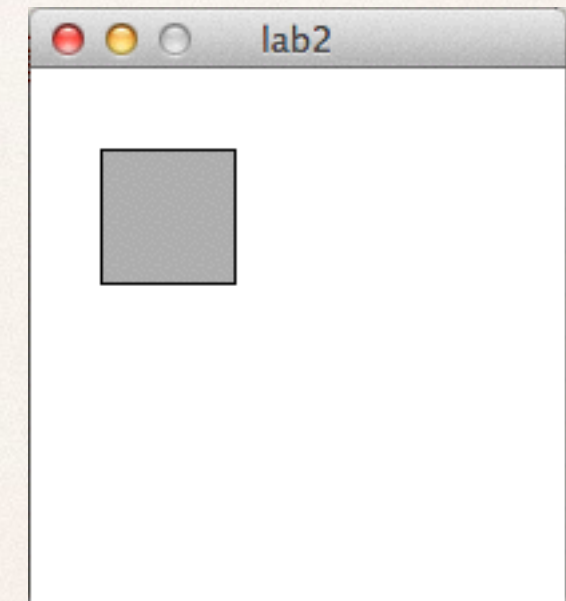
마우스 좌표 가져오기

- ◆ 프로세싱은 마우스 좌표를 가져올 수 있음
 - ◆ mouseX: 현재의 마우스 X 좌표
 - ◆ mouseY: 현재의 마우스 Y 좌표
 - 미리 시스템이 정해 놓은 변수로 고치거나 수정할 수 없음
- ◆ 사용법

```
rect(mouseX, mouseY, 50, 50);
```

Lab: 마우스를 따라 다니는 사각형 그리기

- ◆ 다음을 그리는 코드를 draw() 함수에 입력하자.
 - ◆ 배경색: 하얀색
 - ◆ 사각형 크기: 50 x 50
 - ◆ 사각형 외곽선: 검은색
 - ◆ 사각형 내부색: 50% 회색
 - ◆ rectMode: 사각형 중심
 - ◆ x, y 좌표로 마우스 좌표를 사용

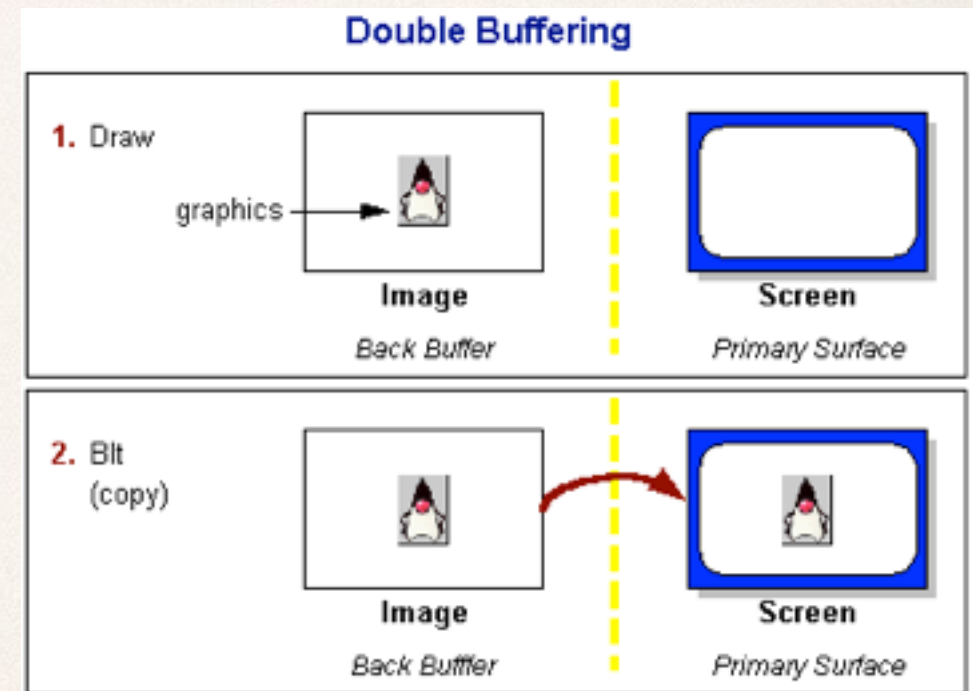


Lab : 마우스를 따라 다니는 사각형 그리기

- ◆ 코드를 다음과 같이 고치시오.
 - ◆ `draw()`에 있는 `background()` 함수를 `setup()`으로 옮기고 다시 실행.
 - ◆ 앞의 슬라이드와 결과가 어떻게 다른가?
 - ◆ 앞의 슬라이드와 결과가 다른 이유는?

더블버퍼링 (Double Buffering)

- ♦ 그림은 언제 그려지나?
 - ♦ processing is a procedural language → line by line?
 - ♦ 매 코드가 한 줄씩 실행결과를 스크린에 그린다면 매번 `background()` 함수에 의해서 화면이 지워지기 때문에 깜빡임(flickering)이 발생함.
 - ♦ 플리커링을 방지하기 위해서 프로그래밍할 때는 더블버퍼링이란 기법을 사용함.
 - ♦ 더블버퍼링: 보이지 않는 화면에 미리 그림을 다 그린 후 보이는 화면에 한번에 그림을 업데이트 하는 방법.
 - ♦ `draw()` 함수는 기본적으로 더블버퍼링을 제공.



마우스 좌표로 라인 그리기

- ◆ 포토샵 등의 그래픽 프로그램 - 라인그리기 툴이 제공 됨.

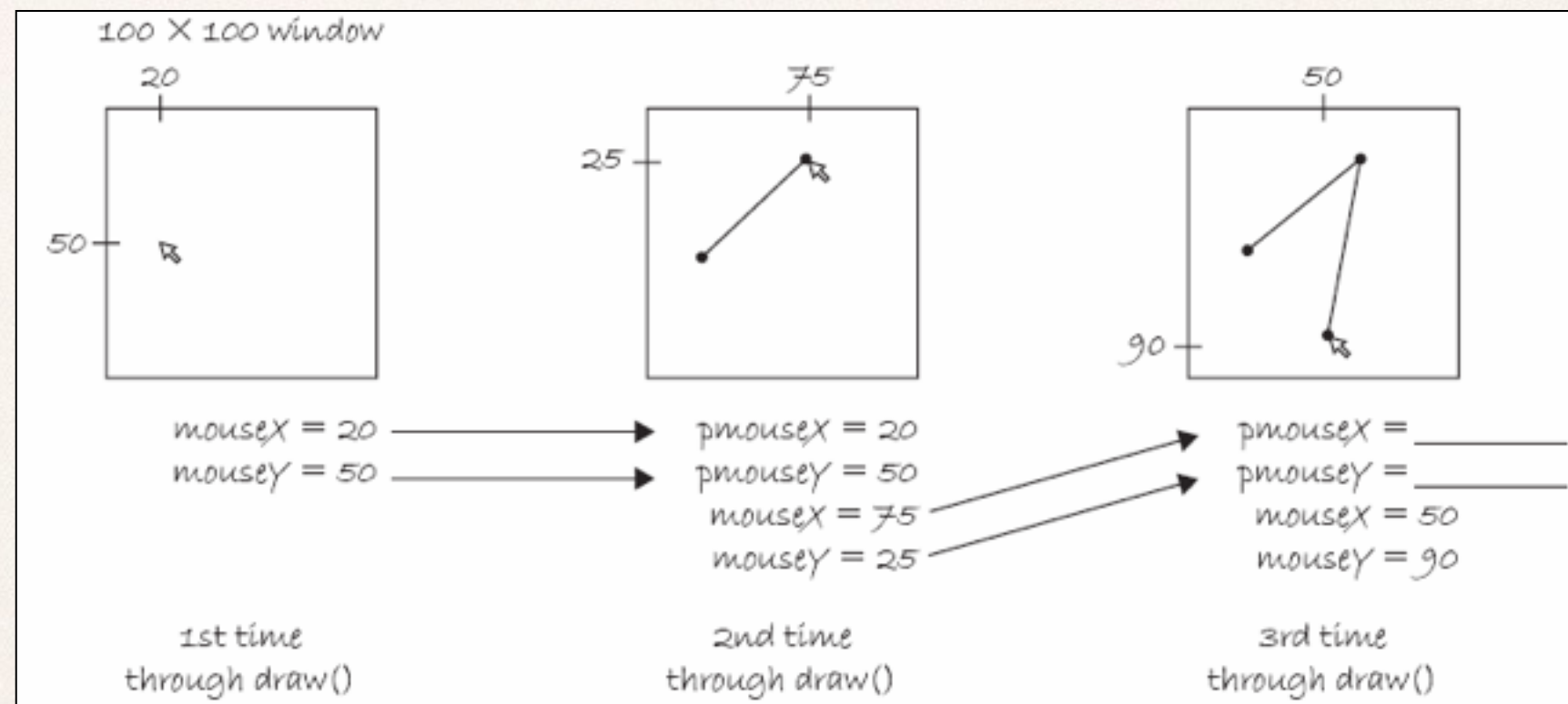


Adobe Photoshop Line Tool

- ◆ 프로세싱에서 mouseX와 mouseY를 line() 함수와 함께 사용하여 그리기
 - ◆ 어떻게?
 - 먼저 마우스를 찍어 한 점의 좌표를 기록하고, 마우스를 움직여서 두번째 좌표를 기록한 후, 그 두 좌표를 line 함수로 연결.
 - `line(oldX, oldY, newX, newY);`

마우스 좌표로 라인 그리기

- ✦ 새로운 변수 선언 없이 그리기
- ✦ 프로세싱은 draw() 함수의 이전 사이클에서 마우스 포인터가 있던 위치를 제공하고 있음
 - ✦ pmouseX: 이전 draw() 사이클에서의 x 좌표
 - ✦ pmouseY: 이전 draw() 함수 사이클에서의 y 좌표



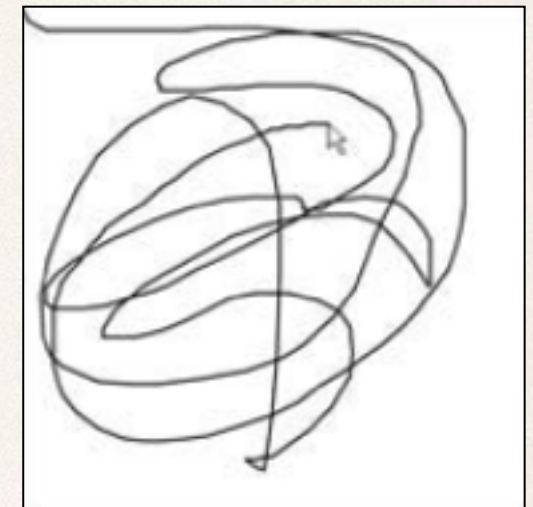
Lab: Line Drawing

- ✦ 마우스 움직임에 따라 그림을 그리는 프로그램을 만들어 보자.

- ✦ 화면크기: 200 x 200

- ✦ 배경색: 하얀색

- ✦ 라인색: 검은색



- ✦ pmouseX, pmouseY, mouseX, mouseY 사용

- ✦ 완성된 후에, setup()에 noSmooth()라는 함수를 넣어서 차이를 확인해 보자. (Anti-aliasing)

이벤트 핸들러

- ✦ 이벤트(event): 마우스 버튼 누름, 키보드 키 누름 등과 같은 사용자 입력
- ✦ 마우스와 키보드 입력은 반드시 `draw()` 함수가 있을 때에만 작동
 - ✦ `mousePressed()`: 마우스 버튼이 누르는 순간 이벤트가 호출되어 해당 함수 내의 코드가 실행됨.
 - ✦ `mouseButton` 변수 사용으로 왼쪽인지 오른쪽인지 알 수 있음
 - ✦ `keyPressed()`: 키버튼이 눌렸을 때 해당 함수 내의 코드가 실행 됨.
 - ✦ `key`, `keyCode` 변수로 어떤 버튼인지 알 수 있음
 - ✦ `mouseReleased()`, `mouseDragged()`, `keyReleased()` 등 키보드와 마우스 입력에 관한 함수들이 존재

Lab: 이벤트 핸들러의 사용

- ◆ 마우스와 키보드 이벤트를 활용한 프로그램 작성
 - ◆ 창의 크기는 400X400
 - ◆ 마우스를 클릭하면, 그 자리에 사각형을 그린다.
 - ◆ 키보드를 누르면, 화면을 모두 지운다.

2. 변수 (Variables)

변수(variable)란?

- ◆ 수학 공식에서의 변수

- ◆ $y = 2x + a$

- ◆ 문자 하나를 사용하여 값을 대입

- ◆ 프로그래밍에서의 변수

- ◆ (일반적으로) 긴 문자를 사용 (예: `sum`, `rgb_color`, etc.)

- ◆ 변수는 메모리에서 주소와 공간을 할당 받음

- ◆ 타입에 따라 다른 크기를 가지고 있음

- ◆ 숫자를 직접 사용하지 않고 변수에 대입하여 사용

변수의 타입(type)이란?

- ✦ 변수는 저장되는 데이터의 형태에 따라 다양한 타입이 준비되어 있음. 데이터의 형태에 맞게 타입을 선택하여 선언(declare)해야 함.
- ✦ 각각의 타입은 메모리에 정해진 크기가 있음.
- ✦ 기본적으로 다음의 세가지 타입이 존재
 - ✦ Integer: 1, 0, 29354, -383... (양/음의) 숫자
 - ✦ Float (Floating Point): 0.12, 3.45, 0.0, -291.332...
 - ✦ Character: a, B, e, %, #, 1, 2... 모든 문자

변수의 타입(type)

- ✦ Integer Types

- ✦ **byte**: A very small number (-127 to +128)
- ✦ **short**: A small number (-32768 to +32767)
- ✦ **int**: A large number (-2,147,483,648 to +2,147,483,647)
- ✦ **long**: A huge number

- ✦ Floating Point Types

- ✦ **float**: A huge number with decimal places
- ✦ **double**: Much more precise, for heavy math

- ✦ Other Types

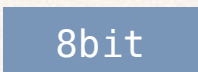
- ✦ **boolean**: true or false
- ✦ **char**: One symbol in single quotes 'a'

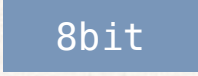
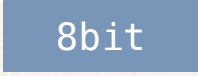
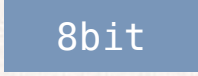
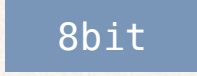
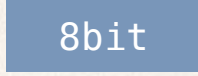
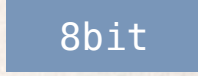
변수의 타입(type) 별 크기

♦ Integer Types

♦ **byte:**  = 1byte

♦ **short:**  

♦ **int:**    

♦ **long:**        

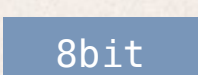
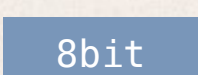
♦ Floating Point Types

♦ **float:**    

♦ **double:**        

♦ Other Types

♦ **boolean:**  1bit

♦ **char:**  

그 외에 유용한 타입

♦ color

- ♦ 색깔을 변수로 지정해서 사용
- ♦ red(), green(), blue(), hue(), saturation(), brightness()

```
color c1 = color(204, 153, 0);  
color c2 = #FFCC00;
```

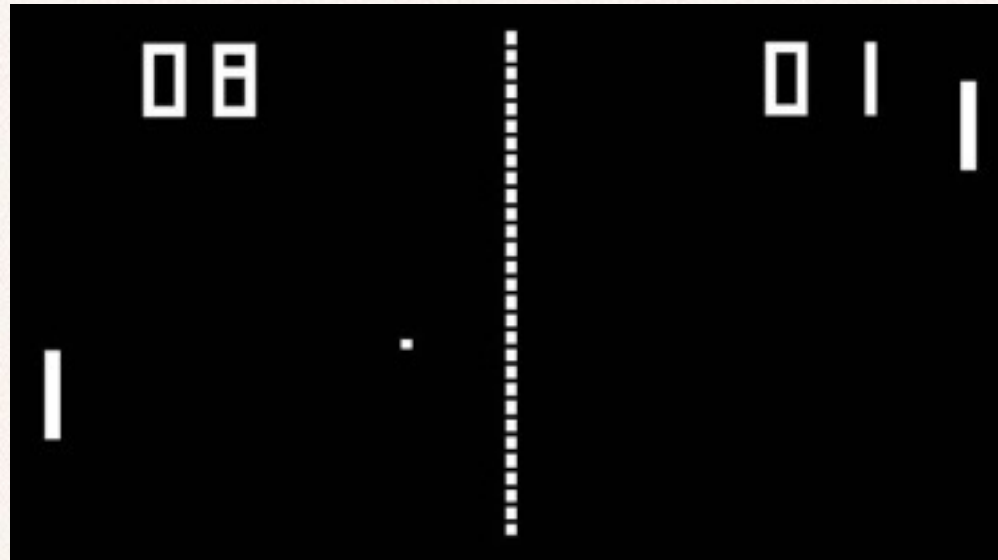
♦ String

- ♦ 문자열을 표시하는 자료형. 쌍따옴표와 사용.
- ♦ 다른 타입과 달리 “객체”이기 때문에 첫 글자가 대문자로 쓰임

```
String myString = “Hello!”;
```

Pong Game

♦ 다음과 같은 Pong 게임의 개발에 사용한 변수는?



- Scores (player 1 & 2)
- Paddle locations (player 1 & 2)
- Ball location
- Net location
- Screen Size

변수의 선언

- ◆ 사용할 변수는 반드시 “선언(declare)”되어야 함
 - ◆ 변수의 종류, 이름, 크기를 컴파일러에 알려주는 작업
 - ◆ 메모리(RAM)에 저장될 변수를 준비하는 과정
 - ◆ 예:

```
int total_number;
```

↓
type

↓
name

변수의 선언과 초기화

◆ 초기화

- ◆ 변수가 선언된 후 (값을 저장할 메모리 공간을 확보한 후), 변수의 초기 값을 지정하는 작업.
- ◆ 변수는 사용하기 전에 반드시 초기화를 해야 함

◆ 초기화 방법

- ◆ 변수의 선언과 동시에 초기화
`int count = 50;`
- ◆ 변수의 선언 후에 초기화
`int count;
count = 50;`
- ◆ 초기화 과정에서 계산하거나, 여러 변수를 한번에 초기화 가능
`int max = 100;
int count = max-min;
int var1, var2 = 0;`

변수 이름을 만드는 법

- ✦ 변수이름은 자유롭게 만들 수 있으나 몇가지 제약이 있음.
 - ✦ 문자, 숫자, 밑줄(`_`) 등을 자유롭게 조합해서 사용 가능
 - ✦ 숫자로 시작해서는 안됨
 - ✦ 예약어(reserved keyword)는 사용해서는 안됨
 - ✦ `mouseX`, `pmouseY`, `int`, `size...`
 - ✦ 소문자로 시작할 것 (강제사항 아님)
 - ✦ `count (o)`, `Count (x)`
 - ✦ 두 단어 이상일 때 두번째 단어는 대문자로 (강제사항 아님)
 - ✦ `currentTime (o)`, `currenttime(x)`
 - ✦ 변수의 내용을 설명할 수 있는 단어의 사용하는 것이 좋음
 - ✦ `imageScale = 1.5; (o)`, `happyEvening = 253; (x)`

변수의 선언과 초기화의 예

- ✦ Type name (optional initialization) ;

```
int count    = 0;           // Declare an int , initialized to 0
char letter  = 'a';         // Declare a char, initialized to 'a'
double d     = 132.32;      // Declare a double, initialized to 132.32
boolean happy = false;      // Declare a boolean, initialized to false
float x      = 4.0;         // Declare a float, initialized to 4.0
float y;                // Declare a float (no assignment)
float z = x * y + 15.0;      // Declare a float, initialize it to
                             // x times y plus 15.0.
```

- ✦ After declaration Assignments:

```
count  = 1;
letter = 'b';
happy  = true;
y = x + 5.2;    // Assign the value of x plus 5.2
```

변수의 선언과 초기화에서 주의할 점

- ♦ 프로세싱에서는 변수를 선언한 후 초기화 할 때, 선언된 type과 할당한 값의 type이 일치해야 함.

- ♦ 다음 중 잘못된 것은?

(!) `int count = 'a';`

(!) `char letter = 0;`

(X) `double deposit = "Fred";`

(X) `boolean happy = 1;`

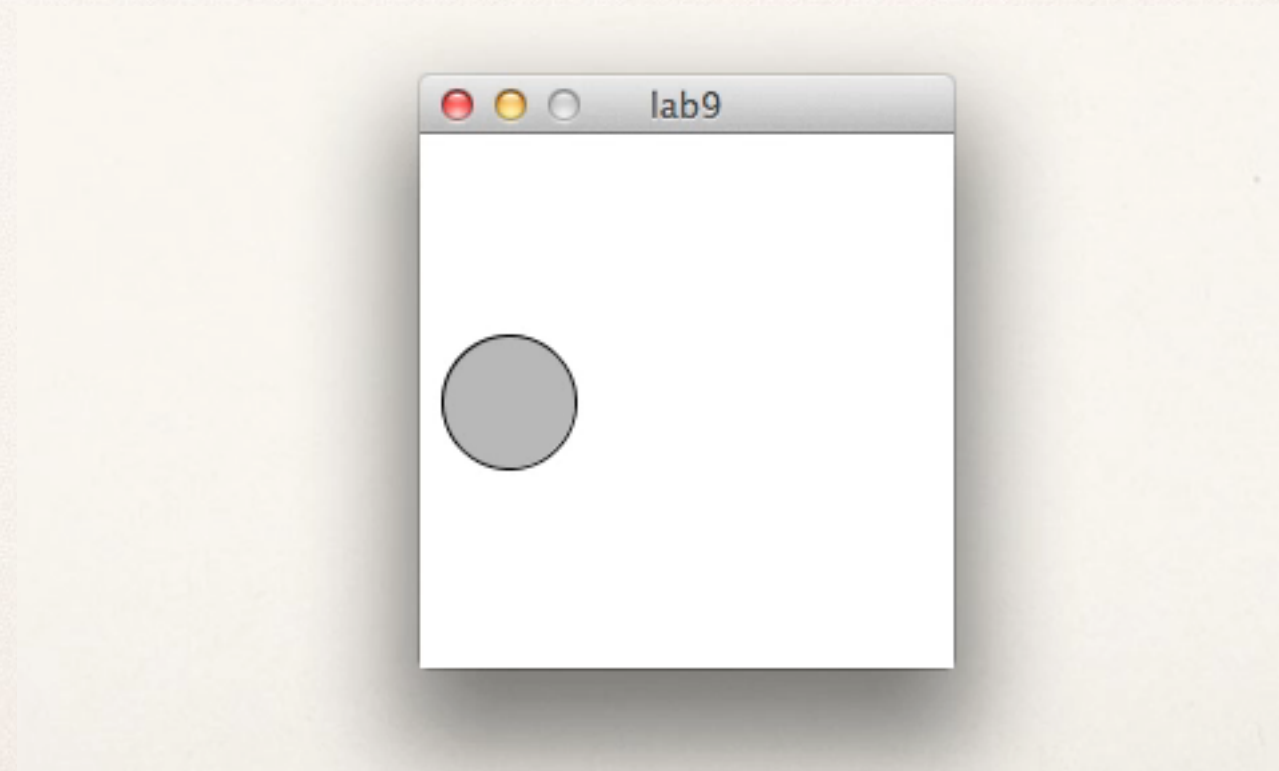
(0) `float feet = 6;`

(X) `int inches = feet * 12;`

(X) `long giant = feet * 3.0;`

Lab: 변수를 사용한 간단한 애니메이션

- ◆ 매번 `draw()` 함수가 호출될 때, 변수의 값을 바꾸어 주면 애니메이션을 만들 수 있다.
- ◆ 원의 `x`, `y` 좌표를 변수로 지정하고, `x`의 값을 증가시켜 보자.



변수의 사용

- ◆ 변수는 코드 어느 부분에서건 사용 가능하다.
- ◆ 블록 안에 선언된 변수는 그 블록 내에서만 사용가능하다.
(지역변수 - local variables)
- ◆ 블록의 가장 밖에서 선언된 변수는 모든 하위 블록에서도 사용가능하다. (전역변수 - global variables)

```
int a;  
void sampleFunction() {  
    int b;  
    b = a + 1;  
    a = b + 1;  
}  
a = 3;  
b = 4; // ← error
```

Lab: 변수를 사용한 간단한 애니메이션

- ◆ 앞의 예제를 이용하여 원이 움직이는 대신, 크기가 변하도록 바꾸어 보자.
 - ◆ 다음 세 개의 변수를 사용
`circleSize`, `circleX`, `circleY`
 - ◆ 원이 커지는 속도를 조절해보기

Lab: 더 많은 변수의 사용

다음의 코드를 입력하고
실행하십시오.

각각의 변수의 값은 어떻게
변했는가?

Call to draw()	circleX	circleY	circleW	circleH
1	0.0	0.0	50.0	100.0
2	0.5	0.5	50.5	99.5
3	1.0	1.0	51.0	99.0
10	4.5	4.5	54.5	95.5
30	14.5	14.5	64.5	85.5
100	49.5	49.5	99.5	50.5
200	99.5	99.5	149.5	0.5

```
float circleX = 0;
float circleY = 0;
float circleW = 50;
float circleH = 100;
float circleStroke = 255;
float circleFill = 0;
float backgroundColor = 255;
float change = 0.5;

void setup() {
  size(200, 200);
  smooth();
}

void draw() {
  // Draw the background and the ellipse
  background(backgroundColor);
  stroke(circleStroke);
  fill(circleFill);
  ellipse(circleX, circleY, circleW, circleH);

  // Change the values of all variables
  circleX = circleX + change;
  circleY = circleY + change;
  circleW = circleW + change;
  circleH = circleH - change;
  circleStroke = circleStroke - change;
  circleFill = circleFill + change;
}
```

시스템 변수

- ✦ 시스템이 기본으로 제공하는 변수
 - ✦ **mouseX, mouseY, pmouseX** and **pmouseY**
 - ✦ **width**: Width (in pixels) of sketch window
 - ✦ **height**: Height (in pixels) of sketch window
 - ✦ **frameCount**: Number of frames processed
 - ✦ **frameRate**: Rate (per sec.) that frames are processed
 - ✦ **screen.height, screen.width**: Entire screen
 - ✦ **key**: Most recent key pressed on keyboard
 - ✦ **keyCode**: Numeric code for which key pressed
 - ✦ **mousePressed**: True or false (pressed or not?)
 - ✦ **mouseButton**: Which button (left, right, center)

글자 표시하기

- ◆ 기본 지정된 폰트로 표시

- ◆ text() 함수만으로 가능
- ◆ 시스템마다 다르게 표시될 수도 있음

```
textSize(32);  
text("word", 10, 30);
```

- ◆ 폰트를 지정하여 표시

- ◆ 다양한 타이포그래피를 표현할 수 있음
- ◆ 시스템의 폰트를 불러와서 표현하는 방법
- ◆ 미리 작성한 폰트를 포함하는 방법

Lab: 시스템 변수의 사용

- ◆ 시스템 변수와 이벤트를 사용하여 Zoog를 그려보자.
 - ◆ 마우스 커서 위치에 Zoog를 표시한다.
 - ◆ frameCount에 따라 테두리의 변화시킨다.
 - ◆ 마우스가 눌리면 눈의 색깔을 변화시킨다.
 - ◆ 키보드가 눌리면 화면에 키를 표시한다.

좌표계의 변환 (Translate)

- ✦ 기존의 방법: 마우스의 위치에 (mouseX, mouseY)에 Zoog를 표시하는 법
- ✦ 좌표계의 원점을 mouseX, mouseY로 이동하는 법
 - ✦ 원점을 이동시키고, 원점 근처에 Zoog를 그린다.
 - ✦ pushMatrix() 로 현재 좌표계를 저장해두었다가 popMatrix() 로 꺼내어 다시 이전 좌표계로 복귀한다.
 - ✦ rotate()를 이용하여 좌표계를 회전시킬 수 있다.

다양한 글자 표시하기

- ✦ 시스템 폰트를 불러와서 사용
 - ✦ 실행하는 컴퓨터에 해당 폰트가 없을 경우 사용불가
 - ✦ 미리 폰트를 포함할 필요가 없이 간단하지만 배포용으로 부적절
- ✦ 다음과 같은 순서를 따름
 - ✦ Pfont의 객체를 전역변수로 선언
 - ✦ setup()함수 내에서 createFont() 함수로 폰트 로딩
 - ✦ textFont() 함수로 폰트와 사이즈를 불러옴
 - ✦ textAlign() 함수로 정렬 방향 지정
 - ✦ fill() 함수로 색을 지정
 - ✦ text() 함수로 글자 표시

다양한 글자 표시하기

- ◆ 미리 폰트를 불러와서 사용
 - ◆ 로딩 시간이 걸리고 용량을 많이 차지함
 - ◆ 배포시 저작권에 주의
 - ◆ 다음과 같은 순서를 따름
 - ◆ Tools > Create Font 메뉴에서 폰트 종류와 기본사이즈를 선택
 - ◆ 한글 폰트를 사용할 경우 Character... 버튼을 눌러서 **Latin, Hangul Compatibility, Hangul Syllables** 세 가지를 체크해줌
 - ◆ .vlw파일명을 기억함
 - ◆ 로딩 시간이 수 초에서 몇 분간 소요되고 .vlw 파일이 data폴더에 생성됨
 - ◆ setup()함수 내에서 loadFont() 함수로 .vlw 파일 로딩
 - ◆ 다른 과정은 createFont() 사용과 동일

난수(random number) 활용

- ✦ “Variety is the spice of life!”
- ✦ random() 함수
 - ✦ return의 자료형은 float
- ✦ Syntax

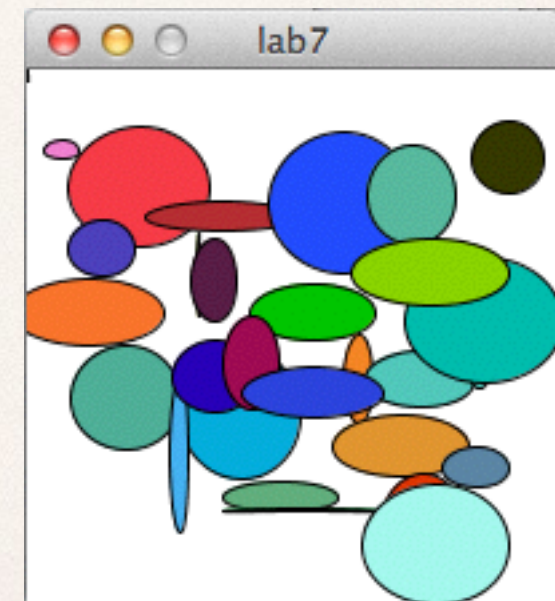
```
random(high)  
random(low, high)
```

- ✦ Example

```
random(10,200);  
background( random(0,255), random(0,255), random(0,255) );
```

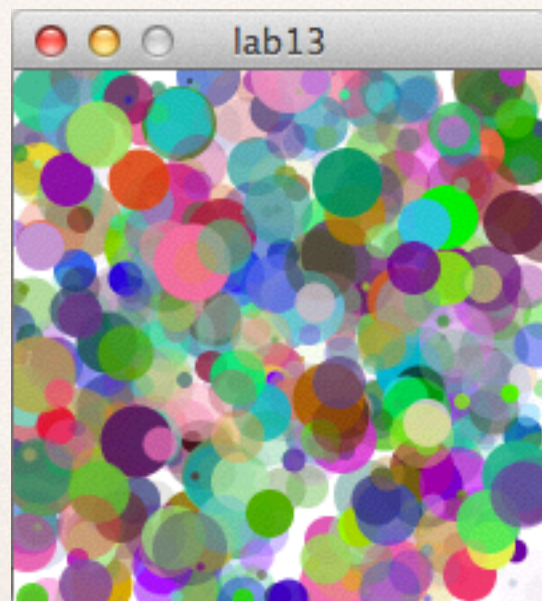
Lab: Random Art

- ✦ 마우스를 클릭할 때마다 랜덤한 원이 표시되는 프로그램을 작성해보자.
 - ✦ 사이즈를 random 하게
 - ✦ 색을 random 하게



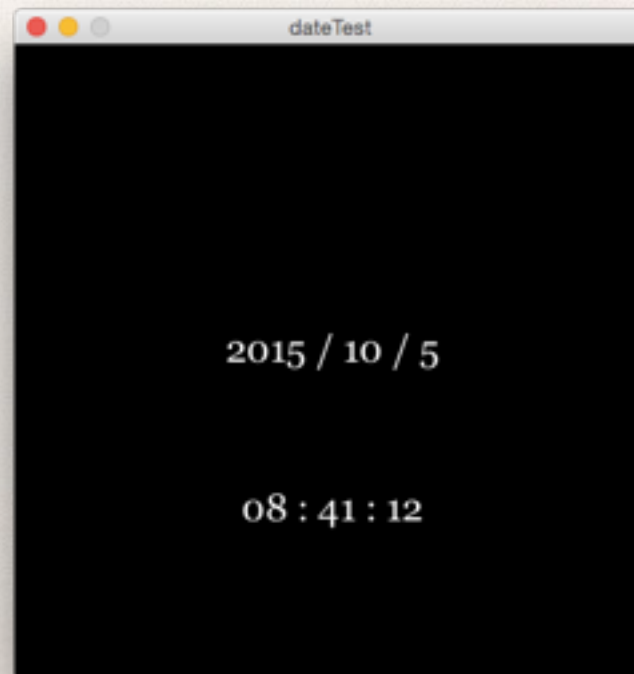
Lab: Screen Saver

- ✦ random 함수를 사용하여 다음과 같은 그림을 그리는 프로그램을 작성하시오.
 - ✦ RGB, Transparency 모두 random 값으로 지정할 것
 - ✦ 원의 위치, 크기도 모두 random 값으로 지정할 것
 - ✦ 마우스를 클릭했을 때, 화면을 모두 지울 것



현재 시간을 불러오는 함수

- ✦ 시스템 변수와 유사하게 사용할 수 있음
- ✦ 현재 연, 월, 일, 시, 분, 초를 불러오는 함수들
- ✦ `year()`, `month()`, `day()`, `hour()`, `minute()`, `second()`



조건문 (Conditionals)

Boolean Expression

- ♦ ‘참/거짓’ 혹은 ‘예/아니오’로 판별되는 문장
 - ♦ 15 는 20 보다 크다 → false
 - ♦ 5 는 5 와 같다 → true
 - ♦ 32 는 33 보다 작거나 같다 → true
- ♦ 논리 자료형으로 불리며 문장의 논리적 구조를 판별할 때 사용됨.

A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

AND table

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

OR table

Boolean Comparison Operators

- ✦ Similar to Algebra

- ✦ `>` greater than
- ✦ `<` less than
- ✦ `>=` greater than or equal to
- ✦ `<=` less than or equal to
- ✦ `==` equality (equal to)
 - ✦ Note: `'=`' is the 'assignment' operator: `x = 5;`
- ✦ `!=` inequality (not equal to)

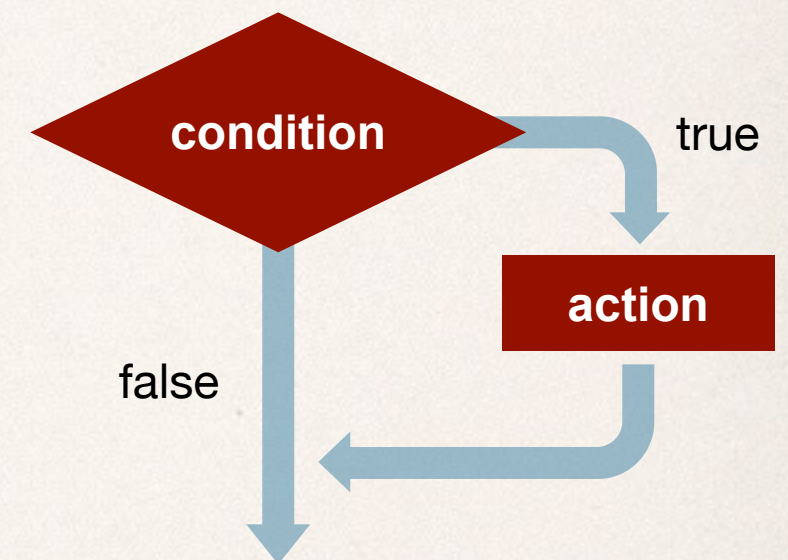
- ✦ Logical Operator

- ✦ `&&` AND
- ✦ `||` OR

if 문과 boolean

- 만약 주어진 조건문이 참(true)이라면 실행. 그렇지 않으면 (false) 실행하지 않음.

- ```
if (Boolean Expression) {
 // execute this line;
}
```



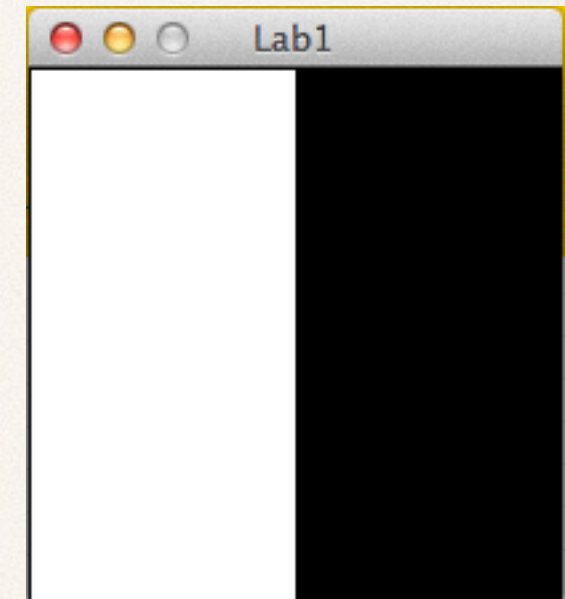
- Example

- ```
if (x < 125) {  
    background(x);  
}
```

Lab: if 문

♦ 다음을 입력하고 실행해 보자

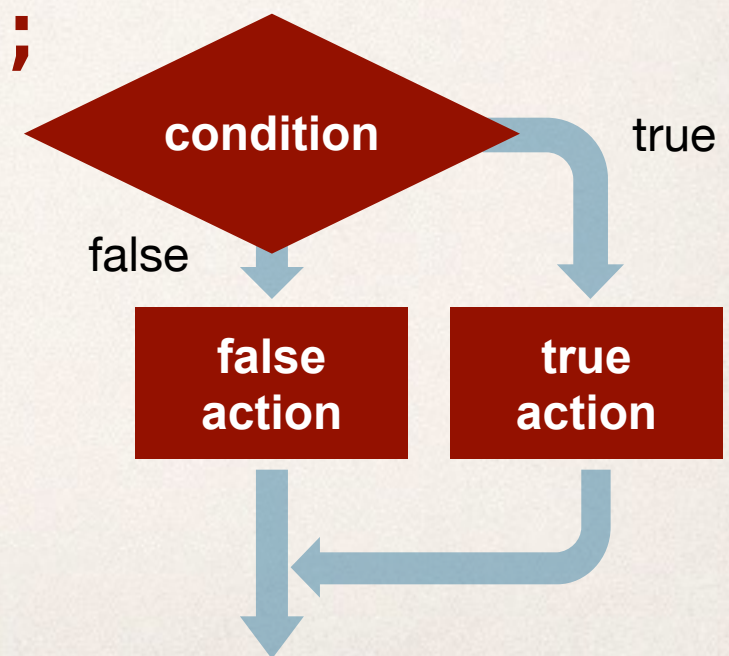
```
void setup() {  
  size(200, 200);  
  smooth();  
}  
  
void draw() {  
  background(0);  
  if (mouseX > width/2) {  
    fill(255);  
    rect(0, 0, width/2, height);  
  }  
}
```



두개의 조건: if...else

- ✦ 처음에 제시된 if 문의 조건을 만족하지 않는 경우 (false) else를 사용해서 false 조건일 경우 실행될 문장을 정의
- ✦

```
if (boolean expression) {  
    // execute this line #1;  
} else {  
    // execute this line #2;  
}
```



시스템 변수의 사용과 이벤트 사용

- ✦ draw() 함수 내에 mousePressed 함수 사용
 - ✦ if(mousePressed) { }
- ✦ mousePressed 함수를 이용한 이벤트의 사용
 - ✦ void mousePressed() { }

Lab: if...else clause

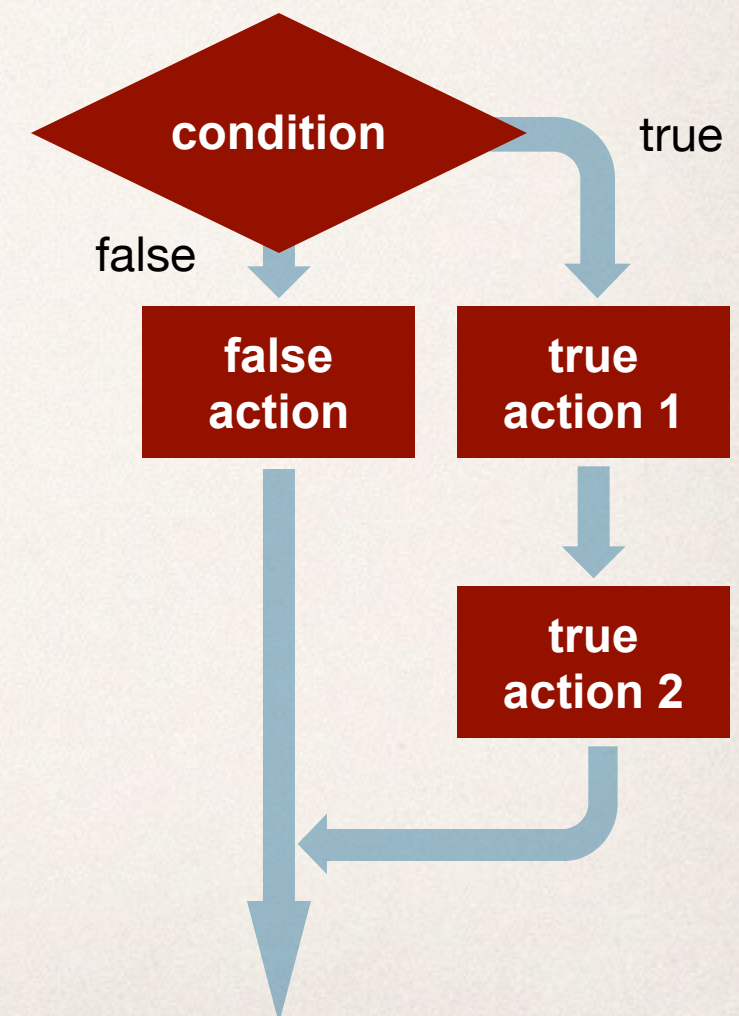
- ♦ 다음을 입력하고 실행해 보자

```
void setup() {  
    size(200, 200);  
    smooth();  
    background(255);  
}  
  
void draw() {  
    if (mouseX > width/2) {  
        background(255);  
    } else {  
        background(0);  
    }  
}
```

두개 이상의 조건: if...else if...else

- ♦ 조건을 여러 개 검증해야 할 때, else if 문을 사용한다.
- ♦

```
if (boolean expression) {  
    // execute this line #1;  
}  
else if (bool exp.) {  
    // execute this line #2;  
}  
else if (bool exp.) {  
    // execute this line #3;  
}  
else {  
    // execute this line #4;  
}
```



Lab: if...else if...else clause

- ◆ 다음의 조건을 만족하는 프로그램을 만들자.
 - ◆ 마우스 포인터가 움직일 때 마다 배경색이 바뀜
 - ◆ 1/3 영역 (왼쪽) → White
 - ◆ 2/3 영역 (중간) → Grey
 - ◆ 3/3 영역 (오른쪽) → Black
 - ◆ 구분선 역시 Grey(1/3), Black(2/3), White(3/3)으로 지정
 - ◆ 힌트: width 를 사용



Pop Quiz 1

♦ 다음의 코드가 출력할 내용은?

```
int x = 75;

if (x > 50) {
    println(x + " is greater than
    50!");
} else if (x > 25) {
    println(x + " is greater than
    25!");
} else {
    println(x + " is 25 or
    less!");
}
```

OUTPUT: 75 is greater than 50!

```
int x = 75;

if(x > 25) {
    println(x + " is greater
    than 25!");
} else if (x > 50) {
    println(x + " is greater
    than 50!");
} else {
    println(x + " is 25 or
    less!");
}
```

OUTPUT: 75 is greater than 25!

Pop Quiz 2

♦ 다음의 코드가 출력할 내용은?

```
int x = 5;

println("x is now: " + x);
if (x == 5) {
    x = 6;
}
if (x == 6) {
    x = 5;
}
println("x is now: " + x);
```

OUTPUT: x is now: 5
x is now: 5

```
int x = 5;

println("x is now: " + x);
if (x == 5) {
    x = 6;
} else if (x == 6) {
    x = 5;
}
println("x is now: " + x);
```

OUTPUT: x is now: 5
x is now: 6

Logical Operators: AND

- ✦ 경우에 따라 조건문을 두개 이상 “모두” 만족시켜야 하는 경우가 있음. 이런 경우 Logical Operator AND를 사용함.
- ✦ A **AND** B: 조건 A와 B가 모두 만족하는 경우 → 프로세싱에서는 A **&&** B 로 표현

A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

AND table

```
int age = 21;
int permit = 1;

if (age >= 20)
    if (permit == 1)
        print("OK to Drive");
    else
        print("Ride the bus");
else
    print("Ride the bus");
```

Nested ifs

```
int age = 21;
int permit = 1;

if (age >= 20 && permit == 1)
    print("OK to Drive");
else
    print("Ride the bus");
```

compound condition

Logical Operators: OR

- 만약 두개 이상의 조건문 중 하나만 만족해도 된다면, OR 를 사용
- A **OR** B: 조건 A와 B 중 하나만이라도 만족하는 경우 → 프로세싱에서는 A || B 로 표현

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

OR table

```
int age = 21;
int permit = 1;

if (age >= 20 || (age >= 20 && permit == 1))
    print("OK to Drive");
else
    print("Ride the bus");
```

괄호의 사용으로 AND 연산자와 OR 연산자를 섞어서 쓸 수 있다.
순서는 ()가 먼저.

Lab: 성적 처리 프로그램

- ✦ 다음의 조건을 만족하는 프로그램을 만드시오.
 - ✦ logical operator 를 이용해 조건을 연결
 - ✦ A: 90-100
 - ✦ B: 80-89.9999
 - ✦ C: 70-79.9999
 - ✦ D: 60-69.9999
 - ✦ F: below 60

```
float grade = random(0,100);  
if (_____) {  
    println("Assign letter grade A.");  
} else if (_____) {  
    println (_____);  
}
```

Lab: if...else example

♦ 다음을 입력하고 실행해 보자

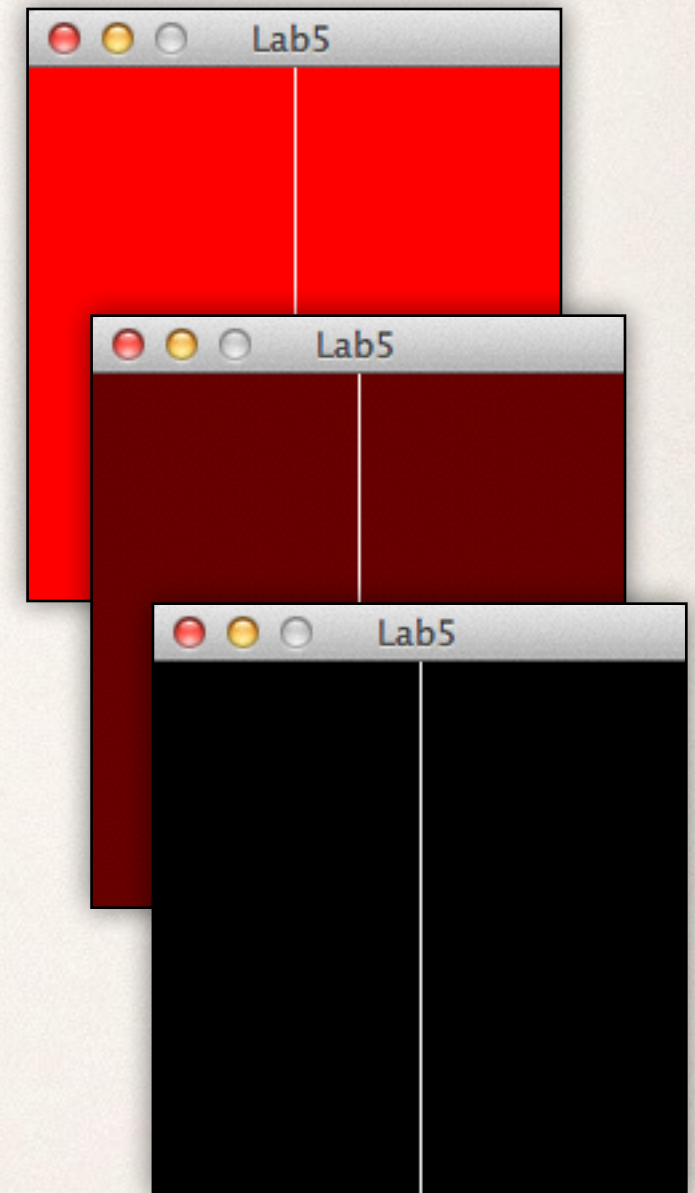
```
float r = 150; // variables
float g = 0;
float b = 0;

void setup() {
  size(200,200);
}

void draw() {
  background(r,g,b);
  stroke(255);          // Line down center
  line(width/2, 0, width/2, height);

  if (mouseX > width/2) // If right
    r = r + 1;          //   more red
  else                  // Else left
    r = r - 1;          //   less red

  if (r > 255)           // Range Check r
    r = 255;
  if (r < 0)             // Range Check r
    r = 0;
}
```



Lab: constrain()

♦ 다음을 입력하고 실행해 보자

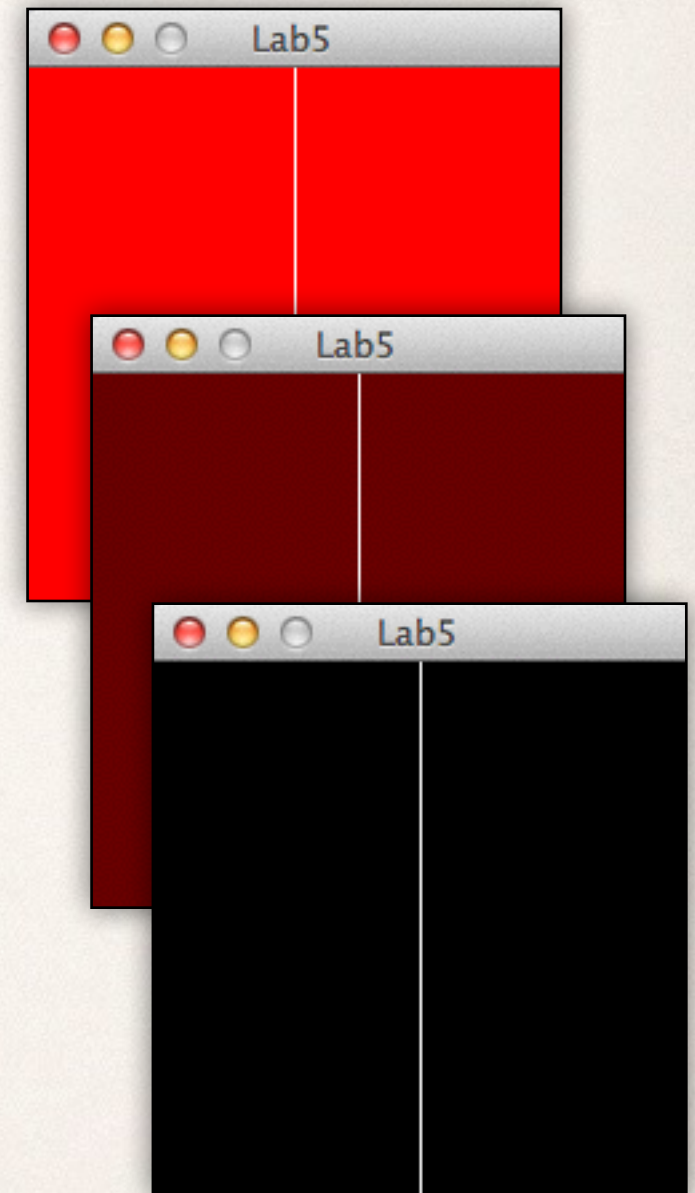
```
float r = 150; // variables
float g = 0;
float b = 0;

void setup() {
  size(200,200);
}

void draw() {
  background(r,g,b);
  stroke(255); // Line down center
  line(width/2, 0, width/2, height);

  if (mouseX > width/2) // If right
    r = r + 1;           // more red
  else // Else left
    r = r - 1;           // less red

  r = constrain(r, 0, 255);
}
```



boolean 변수

- ◆ 앞서 연습한 “driving” 예제는 다음과 같이 변경 가능

```
int age = 21;
int permit = 1;

if (age >= 20 && permit == 1)
    print("OK to Drive");
else
    print("Ride the bus");
```



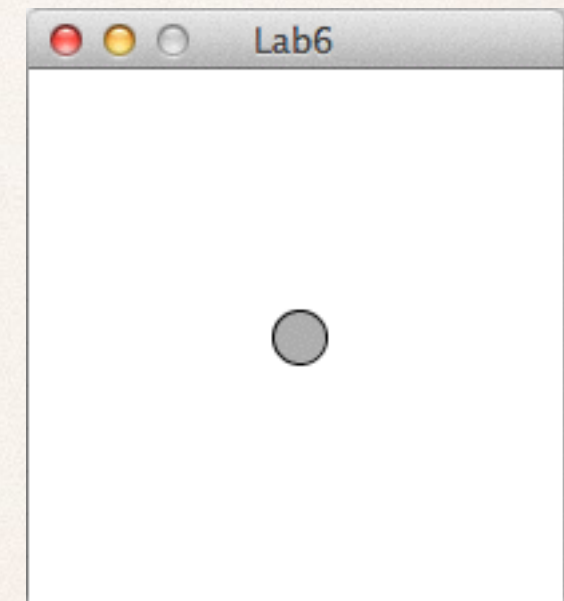
```
int age = 21;
boolean permit = true;

if (age >= 20 && permit == true)
    print("OK to Drive");
else
    print("Ride the bus");
```

- ◆ boolean 변수는 값을 예/아니오 형태로 저장함.
 - ◆ `boolean b = true;`
 - ◆ `println(b);`
 - ◆ `println(!b);`
- ◆ boolean 변수는 on/off 스위치 처럼 사용 가능
 - ◆ `switch = false;` // 초기값으로 switch 값이 off로 지정
 - ◆ `switch = true;` // 어떤 일을 수행한 후 switch 값을 on으로 지정

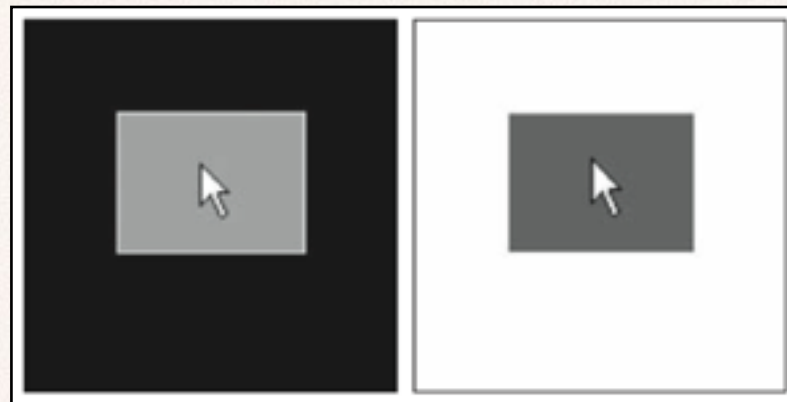
Lab: Pong Game! - bouncing ball

- ♦ 공이 창의 양쪽 끝을 만나면 되돌아오는 프로그램을 만들어 보자.
- ♦ 오른쪽으로 이동: $x = x + 1$
- ♦ 왼쪽으로 이동: $x = x - 1$
- ♦ goBack 이라는 boolean 자료형을 만들고 goBack 이 false 면 오른쪽으로 goBack 이 true 면 왼쪽으로 이동하게 한다.
- ♦ goBack 의 값은 공이 양쪽 벽에 닿았는지를 확인하여 결정한다.
- ♦ goBack 이 스위치 역할을 함.



Lab: Switch Button 을 만들어 보자.

- ◆ 다음의 조건을 만족하는 프로그램을 만들어 보자.



- ◆ 윈도우(200, 200) 안에 버튼을 하나 그린다.
- ◆ 버튼을 클릭하면 배경색이 변한다.
 - ◆ 한번 누르면 배경색이 white로 변하고
 - ◆ 다시 누르면 배경색이 black으로 변한다 (toggle button)
- ◆ 힌트
 - ◆ button 이라는 boolean 변수를 사용하여 현재 버튼의 상태를 저장
 - ◆ mousePressed () 함수를 사용하여 버튼 클릭을 감지

물리법칙을 적용한 애니메이션

- ✦ 아이폰 앱에서의 빠른 스크롤 → 가속도
- ✦ 매킨토시 아이콘 독의 bouncing 효과 → 중력가속도
- ✦ 가속도와 같은 물리 공식의 적용은 애니메이션을 보다 사실감 있게 만들어 준다.
- ✦ Disney - 12가지의 애니메이션 원칙을 만듦
 - ✦ “Illusion Of Life” by Frank Thomas & Ollie Johnston (pp.47-69)
 - ✦ http://en.wikipedia.org/wiki/12_basic_principles_of_animation

중력 가속도의 적용

- ♦ 앞선 예제에서 공을 움직이기 위해 다음과 같이 코딩
 - ♦ $xPos = xPos + 1 \rightarrow xPos$: 위치, 1: 속도
 - ♦ 위치 = 위치 + 속도
 - ♦ 이 예제는 등속운동. 가속도가 0.
 - ♦ 현실감이 없음.
- ♦ 가속도를 표현하기 위해서는 다음과 같이 계산함
 - ♦ 속도 = 속도 + 가속도 (중력)

물리학 관련 참조

- ♦ 역학적인 움직임을 물리법칙에 따라 구현
 - ♦ 인력/척력, 마찰, 탄성 등
 - ♦ 동물, 자동차, 유체 등 다른 물성의 움직임을 구현할 때 참조
- ♦ Nature of Code 참조
 - ♦ <http://natureofcode.com>
 - ♦ 온라인에서 무료로 읽을 수 있음
- ♦ 움직임 관련 라이브러리 이용
 - ♦ <https://processing.org/reference/libraries/>

Lab: 가속도를 가지고 떨어지는 사각형

```
float x = 100;    // x location of square
float y = 0;      // y location of square
float speed = 0;  // speed of square

float gravity = 0.1;    ← 가속도를 위한 변수

void setup() {
  size(200,200);
}

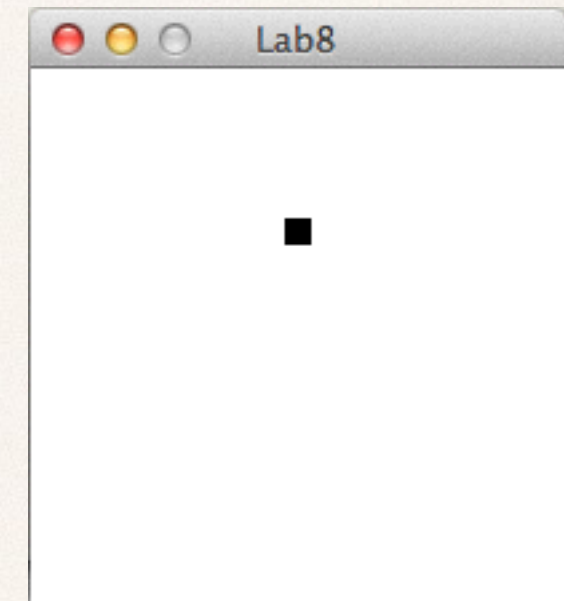
void draw() {
  background(255);

  // Display the square
  fill(0);
  noStroke();
  rectMode(CENTER);
  rect(x, y, 10, 10);

  y = y + speed;    ← 위치+속도 (등속운동)

  speed = speed + gravity;    ← 속도+가속

  // If square reaches the bottom
  // Reverse speed
  if (y > height) {
    speed = speed * -0.95;    ← 속도를 반대로, 즉 반대방향으로 이동하게 한다. -1보다 적을 수록 탄력이 줄어듦.
  }
}
```



Assignment 3

Assignment 3: 인터랙티브 아트

- ♦ 인터랙션을 이용한 작품만들기
 - ♦ 선택 1: 자신의 이전 과제의 정적인 결과물에 인터랙션을 추가하기
 - ♦ 마우스 또는 키보드 인터랙션이 반드시 들어가야 함
 - ♦ 선택 2: 새로운 개념의 시계 만들기 (이미 이전 과제에서 인터랙션을 이용한 경우)
 - ♦ 시간 함수를 불러와서 시각적으로 자유롭게 표현하기
 - ♦ 반드시 시간을 잘 읽을 수 있도록 해야할 필요는 없음
- ♦ 복수 제출시 높은 점수의 결과물로 채점

Assignment 3: 인터랙티브 아트

- ♦ 제출 기한: 2015-10-11 10:00pm
- ♦ 프로세싱 메뉴의 Archive Sketch... (.zip 으로 압축하기) 기능을 이용하여 “학번.zip”로 압축하여 제출
 - ♦ 예: 2014-31378.zip
 - ♦ 주의: 스케치명은 숫자로 시작할 수 없으므로, 자유롭게 지정

Questions?
