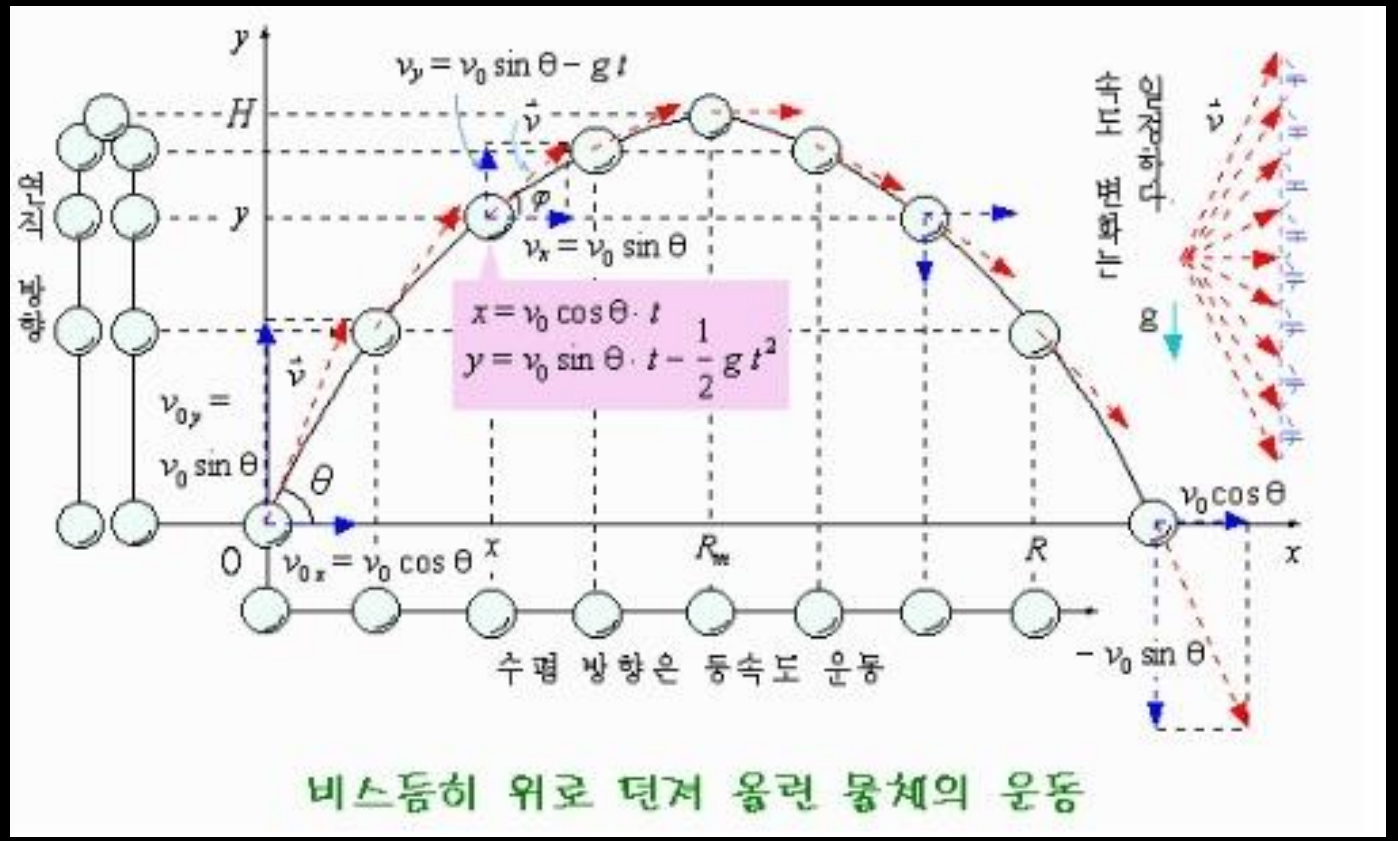
2018 Ajou2

Jonghwa Park
suakii@gmail.com

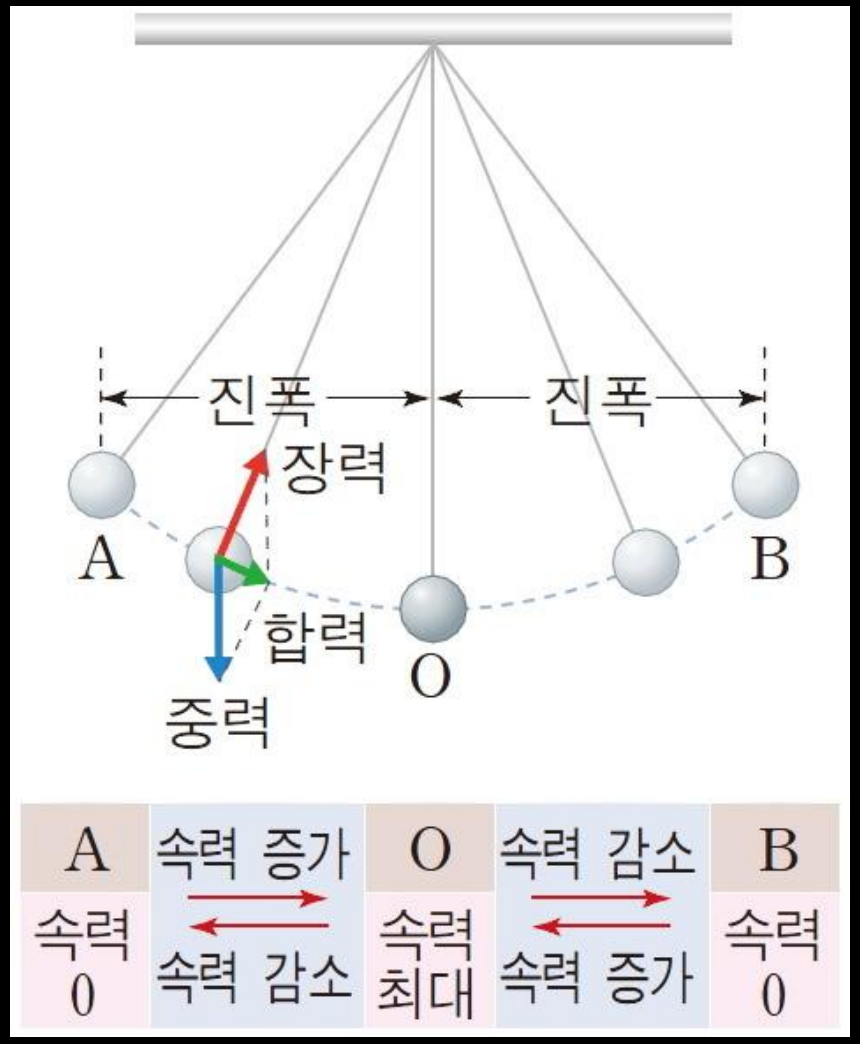
GYEONGGI SCIENCE HIGH SCHOOL

목적



출처: http://sciencelove.com/653

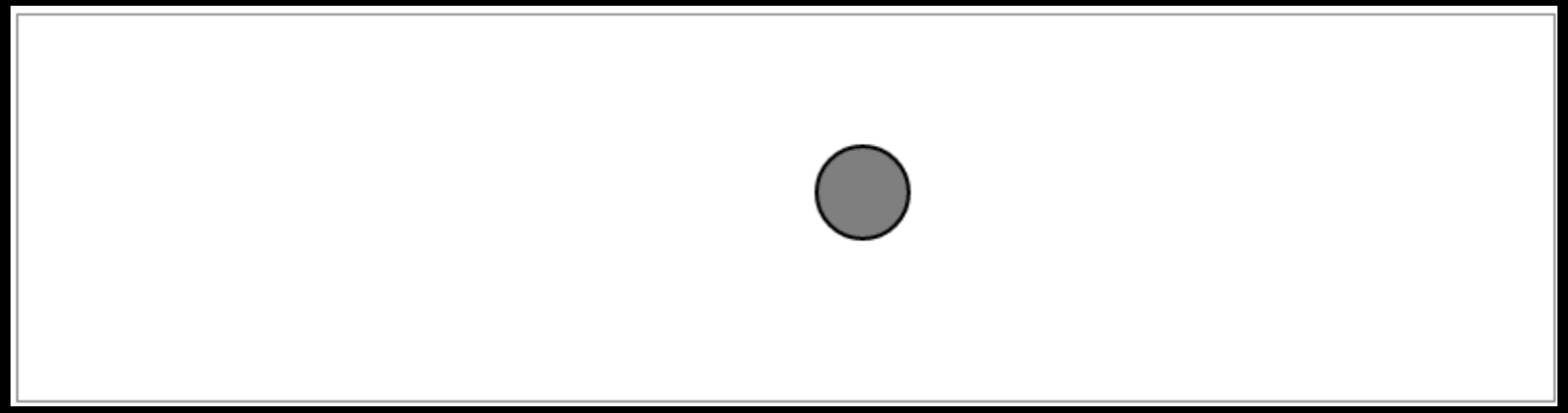
목적



목적

- 어떻게 이러한 요구 사항을 프로그래밍으로 표현할 수 있을까?

Bouncing Ball – No Vectors



Bouncing Balls – No Vectors

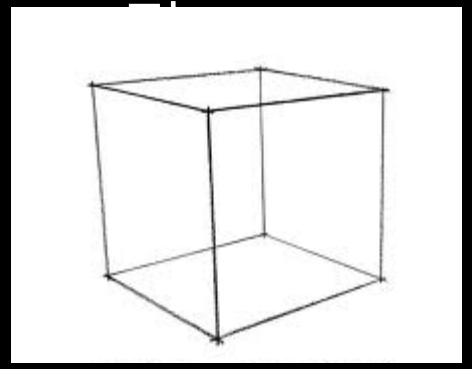
```
float x = 100;
float y = 100;
float xspeed = 2.5;
float yspeed = 2;

void setup() {
  size(800, 200);
  smooth();
}
```

```
void draw() {
  background(255);


  // Add the current speed to the location.
  x = x + xspeed;
  y = y + yspeed;

  if ((x > width) || (x < 0)) {
    xspeed = xspeed * -1;
  }
  if ((y > height) || (y < 0)) {
    yspeed = yspeed * -1;
  }


  // Display circle at x location
  stroke(0);
```

변수들

x, y : 위치
xSpeed, ySpeed : 속도

+) 힘, 바람, 마찰 → 현실

x, y, z → 

3D

.
.
.

변수 너무 많다

- Now Change to Vectors

벡터란 무엇인가?

벡터 (물리)

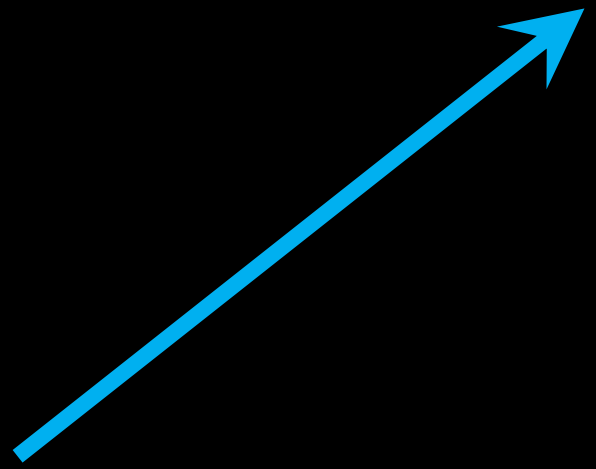
위키백과, 우리 모두의 백과사전.

벡터(vector)는 방향과 크기의 의미를 모두 포함하는 표현 도구로서 주로 힘이나 자기장, 전기장 등의 물리적 개념을 설명할 때 이용된다. 크기만을 의미하는 스칼라량과 비교되는 양이다. 물리적 현상을 나타낼 때는 2차원 또는 3차원 방향의 벡터량을 쓴다.

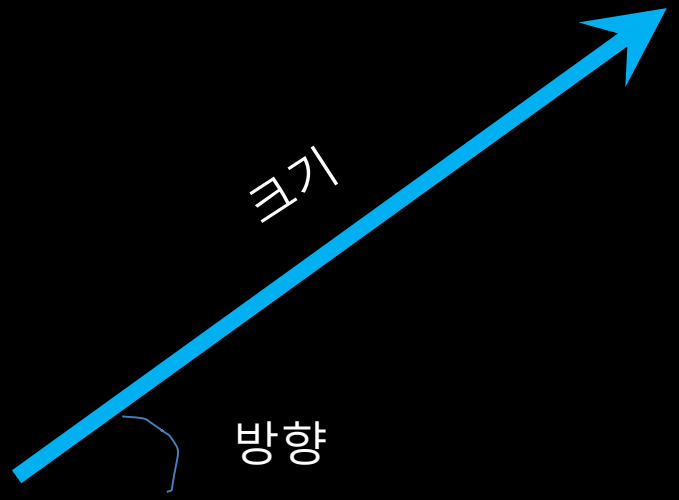
솔직히 다들 이미 알잖아?

벡터는 뭘까?

I’m a Vector

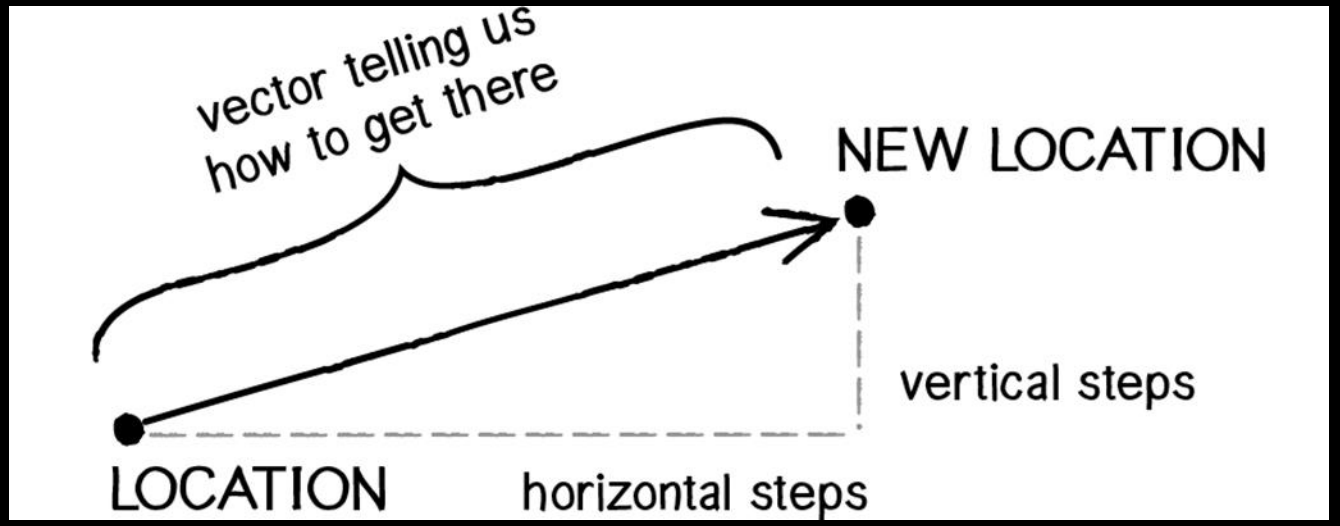


벡터는 뭘까



크기와 방향
동시에 갖는 것

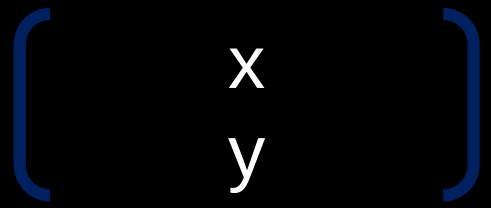
벡터는 무엇일까?


vector telling us
how to get there
NEW LOCATION
vertical steps
LOCATION
horizontal steps

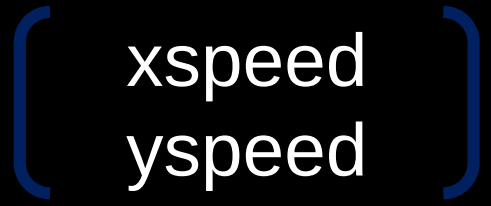
벡터로 바꾸면 변수가 줄어

```
float x;
float y;
float xspeed;
float yspeed;
```

```
Vector location;
Vector speed;
```



위치벡터



속도벡터

벡터는 무엇일까?

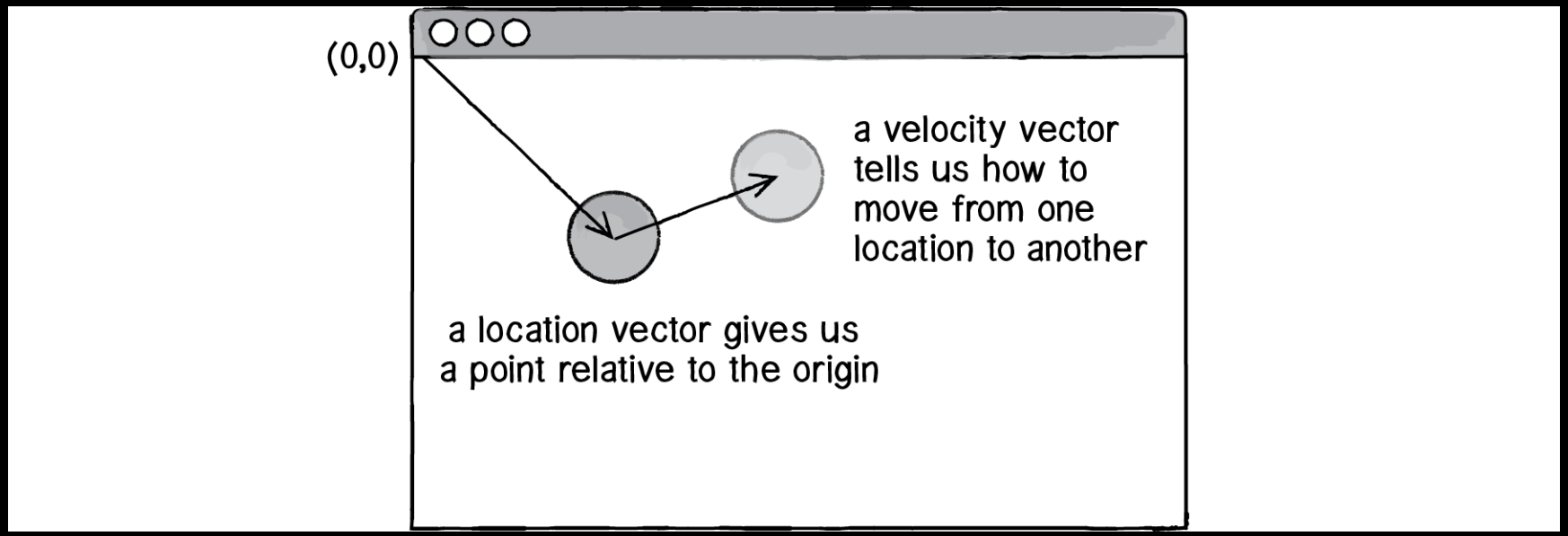
- 속도: 한 점을 어느 방향/크기로 움직여야 하는지 결국 속도는 벡터

- 위치도 벡터: 원점에서 한 위치까지 보면 결국 위치도 벡터

위치

원점

다음위치=현재위치+속도


(0,0)
a velocity vector
tells us how to
move from one
location to another
a location vector gives us
a point relative to the origin

벡터 클래스

클래스
변수 (데이터)
함수 (메소드)

벡터 클래스

필요한 변수는?

X좌표, Y좌표, (Z좌표)

벡터 클래스

변수값

```
class PVector {

  float x;
  float y;

  PVector(float x_, float y_) {
    x = x_;
    y = y_;
  }

}
```

벡터 클래스

변수값

```
float x = 100;
float y = 100;
float xspeed = 1;
float yspeed = 3.3;
```

```
PVector location = new PVector(100,100);
PVector velocity = new PVector(1,3.3);
```

위치벡터에 (100, 100)을
속도벡터에 (1, 3.3)을 저
장

벡터 덧셈

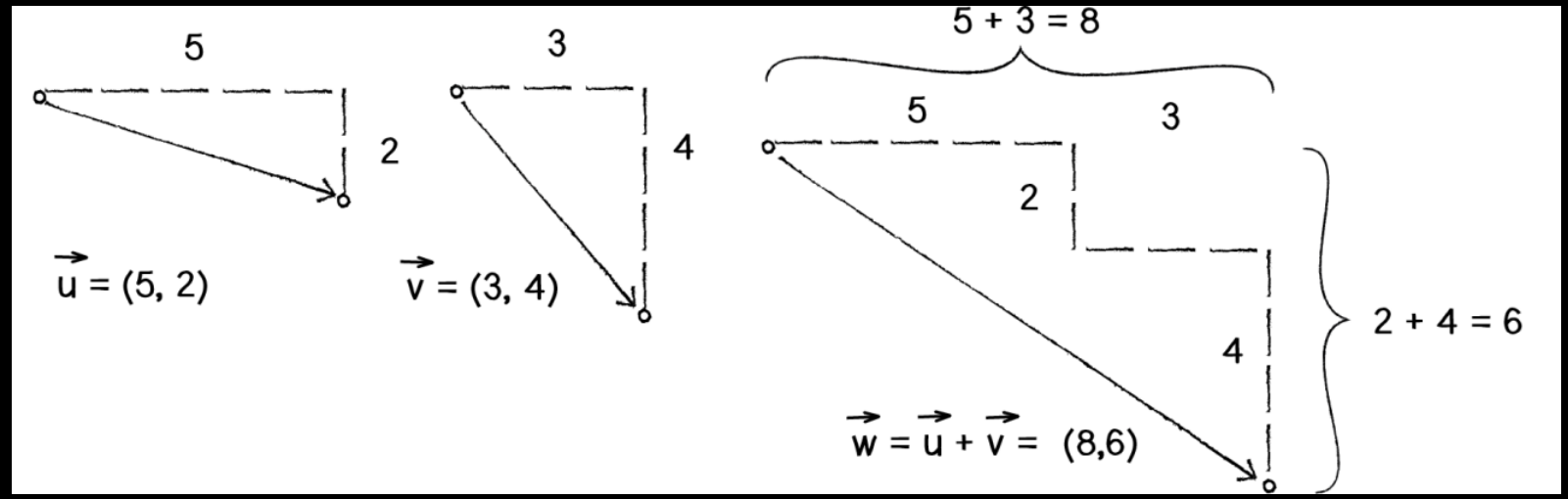
x = x + xspeed
y = y + yspeed

location = location + velocity (?)

벡터 덧셈

- 프로세싱에서 ‘+’는 벡터 덧셈을 지원하지 않는다.
- 벡터 더하는 함수를 만들자

벡터 덧셈



벡터 덧셈

```
class PVector {

  float x;
  float y;

  PVector(float x_, float y_) {
    x = x_;
    y = y_;
  }

  void add(PVector v) {
    y = y + v.y;
    x = x + v.x;
  }
}
```

PVector v를 더할때
원래 y에 v의 y를 더하고
원래 x에 v의 x를 더한다

x 성분은 x 성분끼리
y 성분은 y 성분끼리

벡터 덧셈

x = x + xspeed
y = y + yspeed

location = location + velocity (?)

location.add(velocity)

다른 함수들

- add() — add vectors
- sub() — subtract vectors
- mult() — scale the vector with multiplication
- div() — scale the vector with division
- mag() — calculate the magnitude of a vector
- setMag() - set the magnitude of a vector
- normalize() — normalize the vector to a unit length of 1
- limit() — limit the magnitude of a vector
- heading() — the 2D heading of a vector expressed as an angle
- rotate() — rotate a 2D vector by an angle

다른 함수들

- lerp() — linear interpolate to another vector
- dist() — the Euclidean distance between two vectors (considered as points)
- angleBetween() — find the angle between two vectors
- dot() — the dot product of two vectors
- cross() — the cross product of two vectors (only relevant in three dimensions)
- random2D() - make a random 2D vector
- random3D() - make a random 3D vector

Vector Subtraction

```
void setup() {
  size(640,360);
}

void draw() {
        background(255);
        PVector mouse  = new PVector(mouseX,mouseY);
        PVector center = new PVector(width/2,height/2);
        mouse.sub(center);
        translate(width/2,height/2);
        line(0,0,mouse.x,mouse.y);
}
```

Normalizing a vector

```
void setup() {
  size(400,400);

}

void draw() {
  background(255);

  PVector mouse = new PVector(mouseX,mouseY);
  PVector center = new PVector(width/2,height/2);
  mouse.sub(center);

  mouse.normalize();
  mouse.mult(50);
  translate(width/2,height/2);
  line(0,0,mouse.x,mouse.y);

}
```

벡터의 쓰임

벡터를 이용해

움직이는 클래스,

Mover클래스를 만들어 보자

벡터의 쓰임

```
class Mover {

  PVector location;
  PVector velocity;

  Mover() {
    location = new PVector(random(width),random(height));
    velocity = new PVector(random(-2,2),random(-2,2));
  }

  void update() {
    location.add(velocity);
  }

  void display() {
    stroke(0);
    fill(175);
    ellipse(location.x,location.y,16,16);
  }

  void checkEdges() {
    if (location.x > width) {
      location.x = 0;
    } else if (location.x < 0) {
      location.x = width;
    }

    if (location.y > height) {
      location.y = 0;
    } else if (location.y < 0) {
      location.y = height;
    }
  }
}
```

위치 벡터와 속도 벡터 선언

update 함수
위치 벡터에 속도 벡터를 더한다

display 함수
위치 벡터가 가리키는 지점에 타원을 그린다

checkEdges 함수
타원이 창을 벗어나면 창 안으로 되돌린다

벡터의 쓰임

```
Mover mover;

void setup() {
  size(640,360);
  mover = new Mover();
}

void draw() {
  background(255);

  mover.update();
  mover.checkEdges();
  mover.display();
}
```

Mover 클래스를 토대로
mover 객체를 만들었다.

Mover 클래스 안의
update, checkEdges, display 함수를 이용한다

가속시키기

가속도와 벡터를 사용한 이동

- 가속도: 속도가 변하는 비율
- 속도: 위치가 변하는 비율
- 가속도는 속도에 속도는 위치에...
- Velocity.add(acceleration);
- Location.add(velocity);

벡터의 쓰임

가속도

- 일정한 가속
- 랜덤 가속
- 마우스를 따라 가속

벡터의 쓰임

일정한 가속

```
class Mover {

  PVector location;
  PVector velocity;
  PVector acceleration;
```

가속도 벡터를 추가한다

벡터의 쓰임

일정한 가속

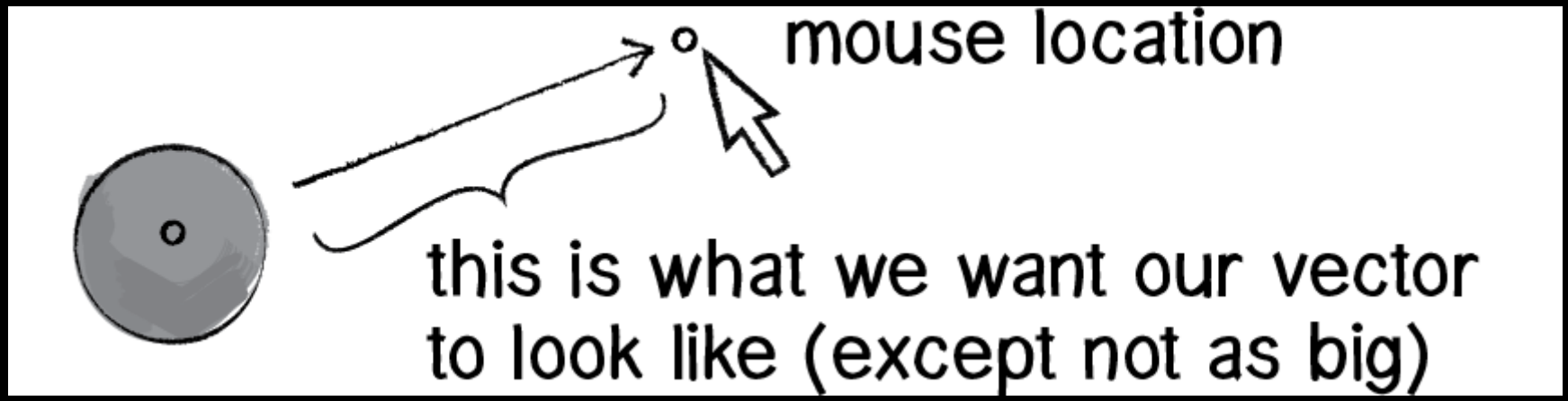
```
void update() {
    velocity.add(acceleration);
    location.add(velocity);
}
```

update 함수
속도 벡터에 가속도 벡터를 더한다

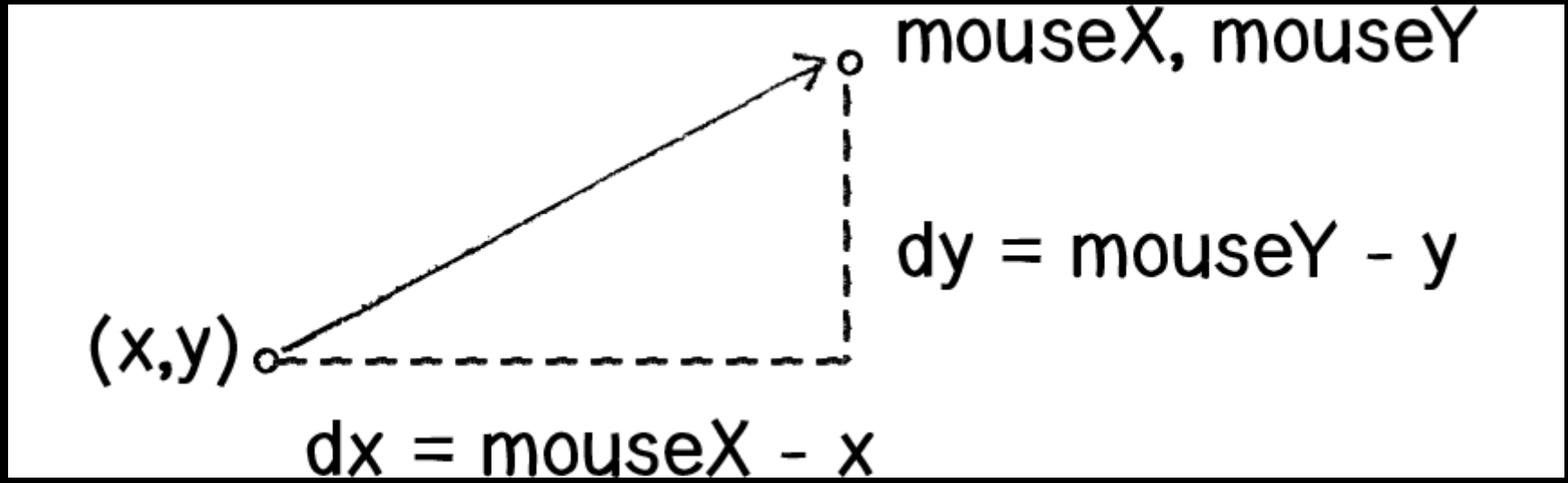
랜덤 가속

```
void update() {

    acceleration = PVector.random2D();
    acceleration.mult(random(2));

    velocity.add(acceleration);
    velocity.limit(topspeed);
    location.add(velocity);
  }
```

Interactivity with Acceleration



Interactivity with Acceleration



Interactivity with Acceleration

```
 void update() {

    PVector mouse = new PVector(mouseX,mouseY);
Step 1: Compute direction
    PVector dir = PVector.sub(mouse,location);

Step 2: Normalize
    dir.normalize();

Step 3: Scale
    dir.mult(0.5);

Step 4: Accelerate
    acceleration = dir;

    velocity.add(acceleration);
    velocity.limit(topspeed);
    location.add(velocity);

  }
```

Break…

- CU..