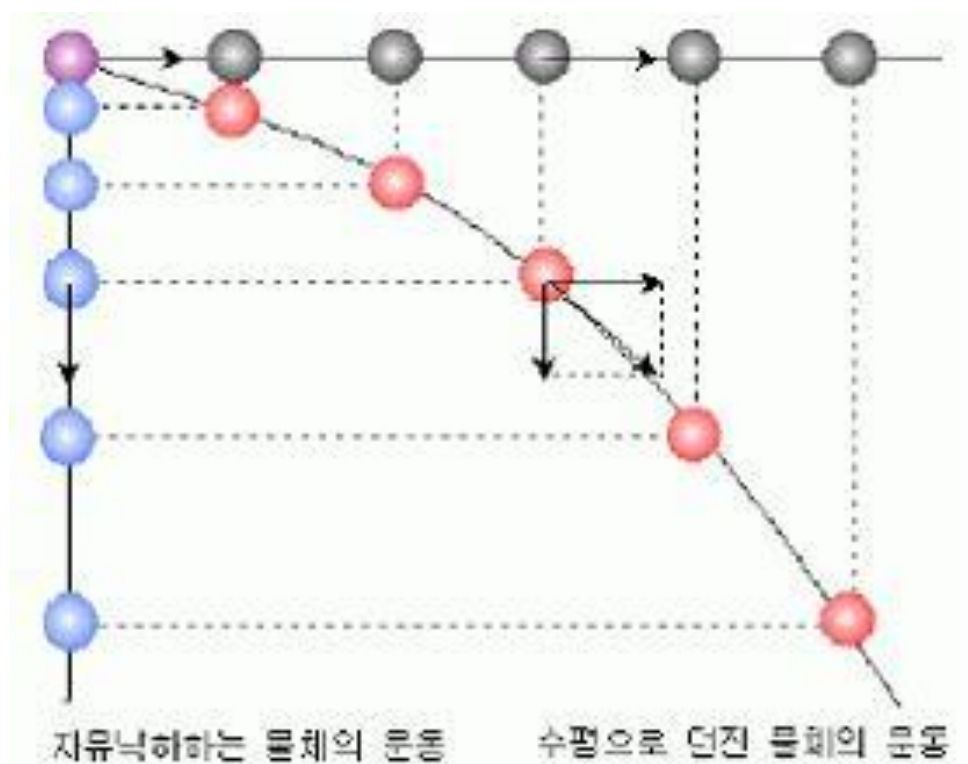


1. Vectors

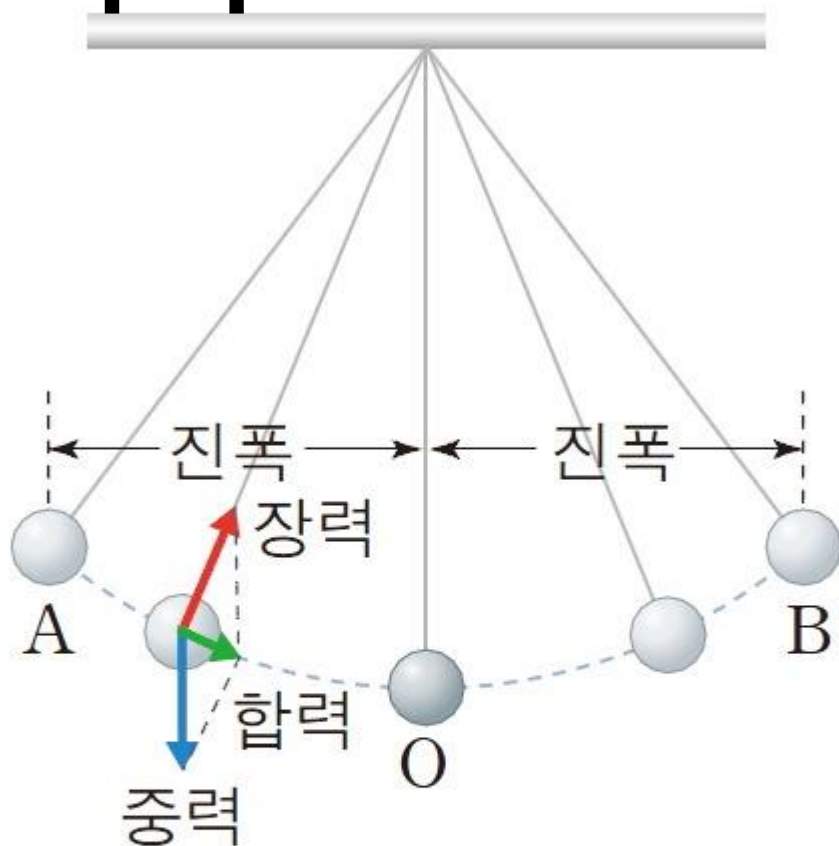
박종화
suakii@gmail.com

목적



공의 낙하 운동

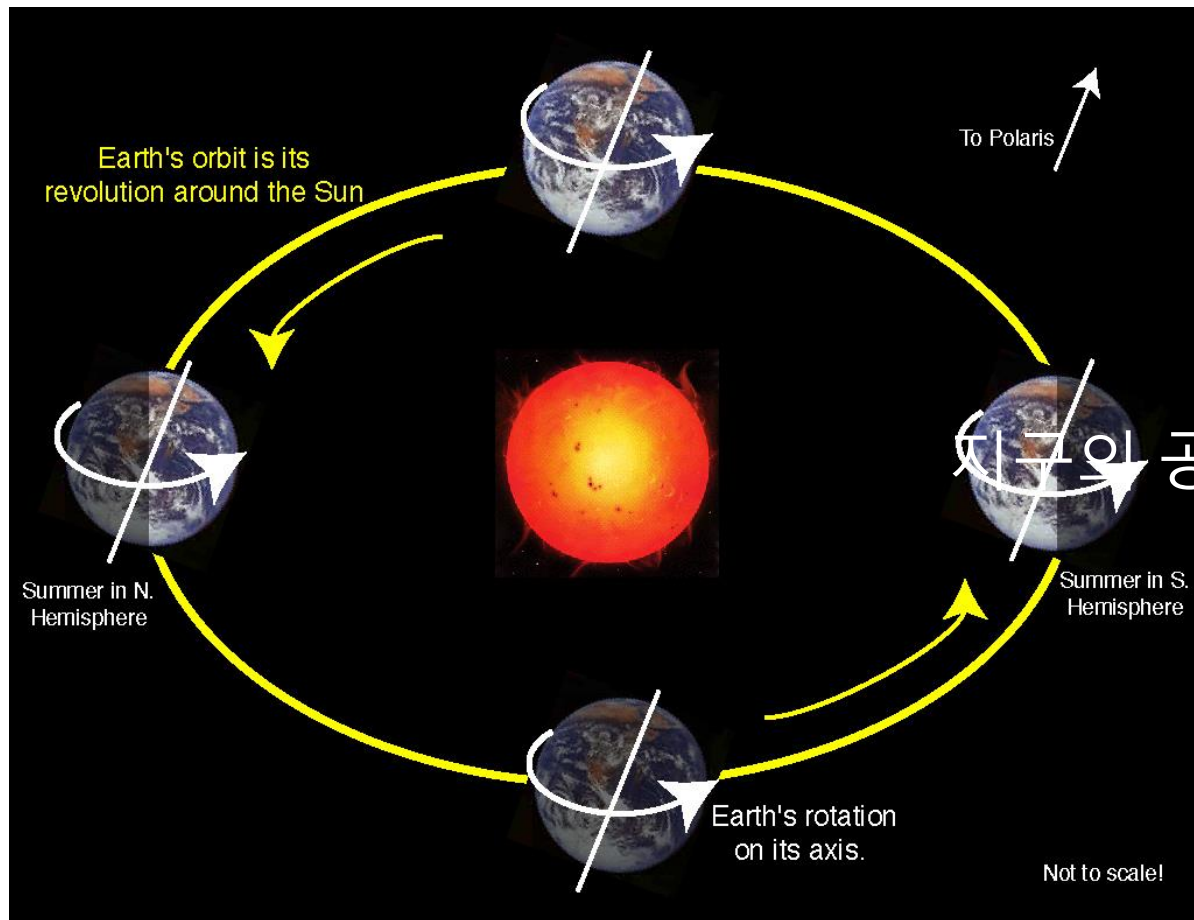
목적



진자의 운동

A	속력 증가	O	속력 감소	B
속력 0	<div> <div></div> <div></div> </div>	속력 최대	<div> <div></div> <div></div> </div>	속력 0

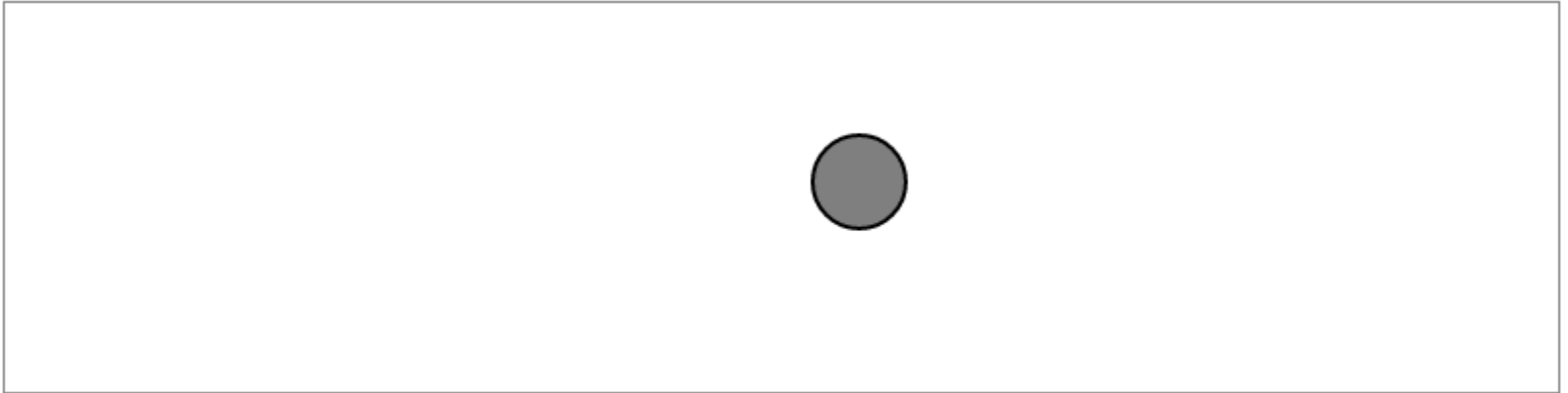
목적



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);
  strokeWeight(2);
  fill(127);
  ellipse(x, y, 48, 48);
}
```

코드에 등장하는 변수들

x, y : 위치

$xSpeed, ySpeed$: 속도

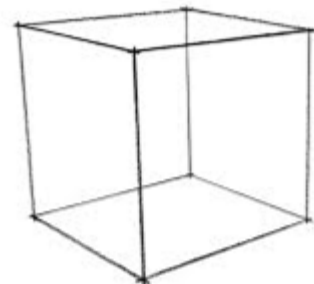
+) 힘, 바람, 마찰

x, y, z

·
·
·



현실

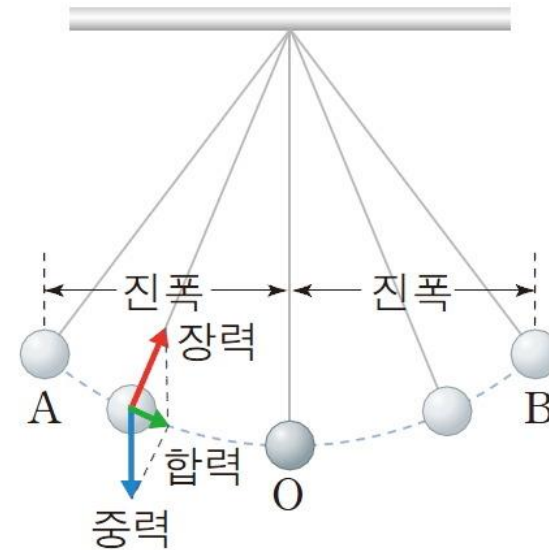
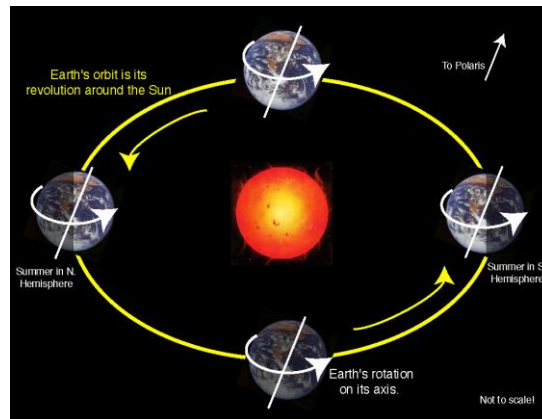
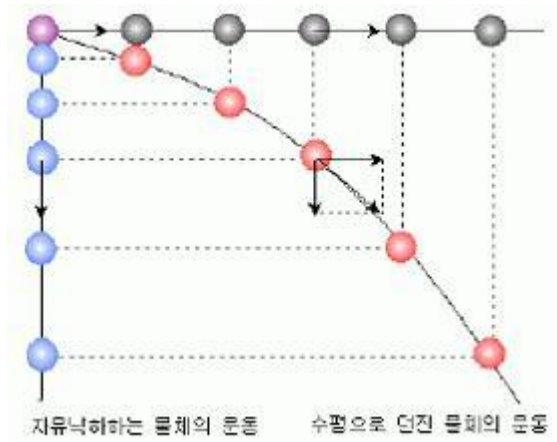


3
D

변수 너무 많다

- Now Change to Vectors

벡터란 무엇인가



A	속력 증가	O	속력 감소	B
속력 0	←→	속력 최대	←→	속력 0
	속력 감소		속력 증가	

벡터란 무엇인가

벡터 (물리)

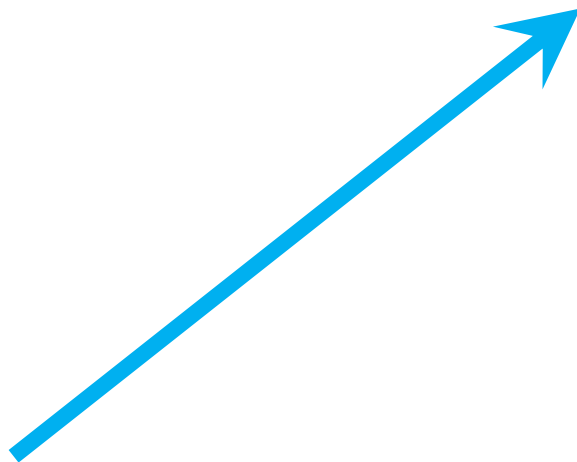
위키백과, 우리 모두의 백과사전.

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.

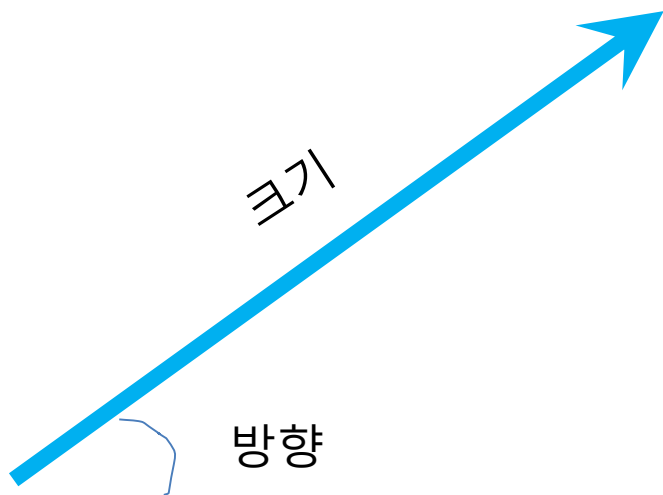
솔직히 다들 이미 다 알잖아?

벡터는 뭘까

I'm a Vector

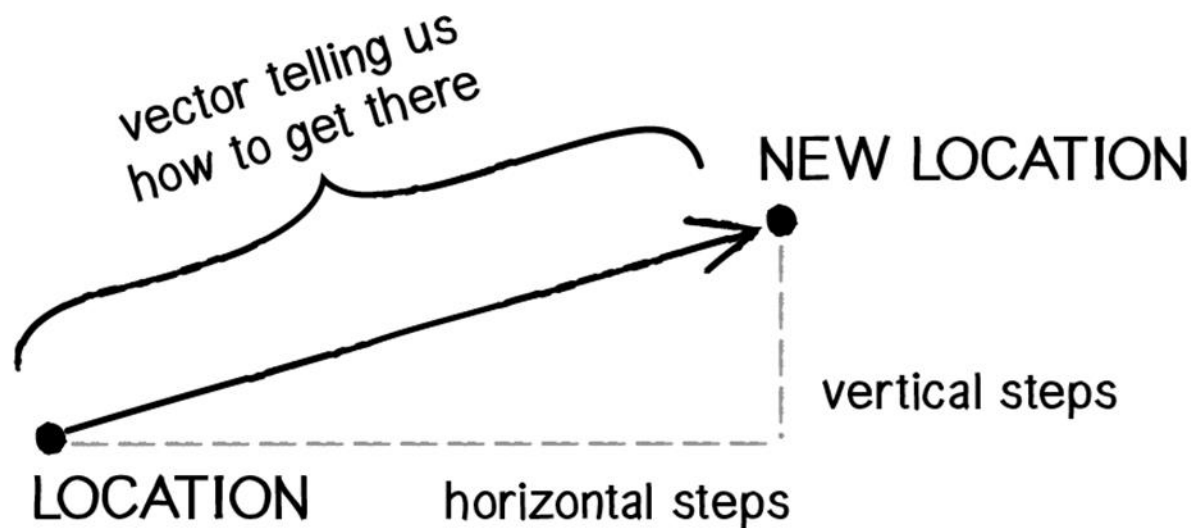


벡터는 뭘까



크기와 방향
동시에 갖는 것

벡터는 무엇일까?



벡터로 바꾸면 변수가 줄어

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



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

$$\begin{bmatrix} x \\ y \end{bmatrix}$$

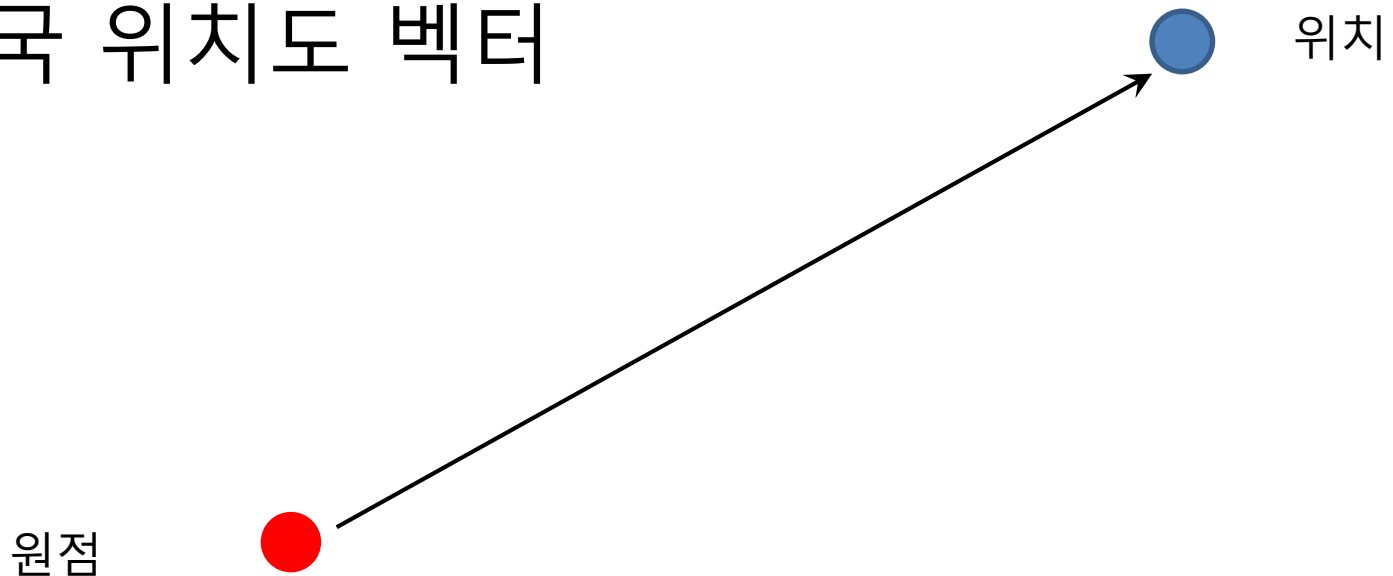

위치벡터

$$\begin{bmatrix} xspeed \\ yspeed \end{bmatrix}$$


속도벡터

벡터는 무엇일까?

- 속도: 한 점을 어느 방향/크기로 움직여야 하는지 결국 속도는 벡터
- 위치도 벡터: 원점에서 한 위치까지 보면 결국 위치도 벡터



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
x_, y_의 두 값을 저장

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 저장

벡터 덧셈

$x = x + \text{xspeed}$
 $y = y + \text{yspeed}$

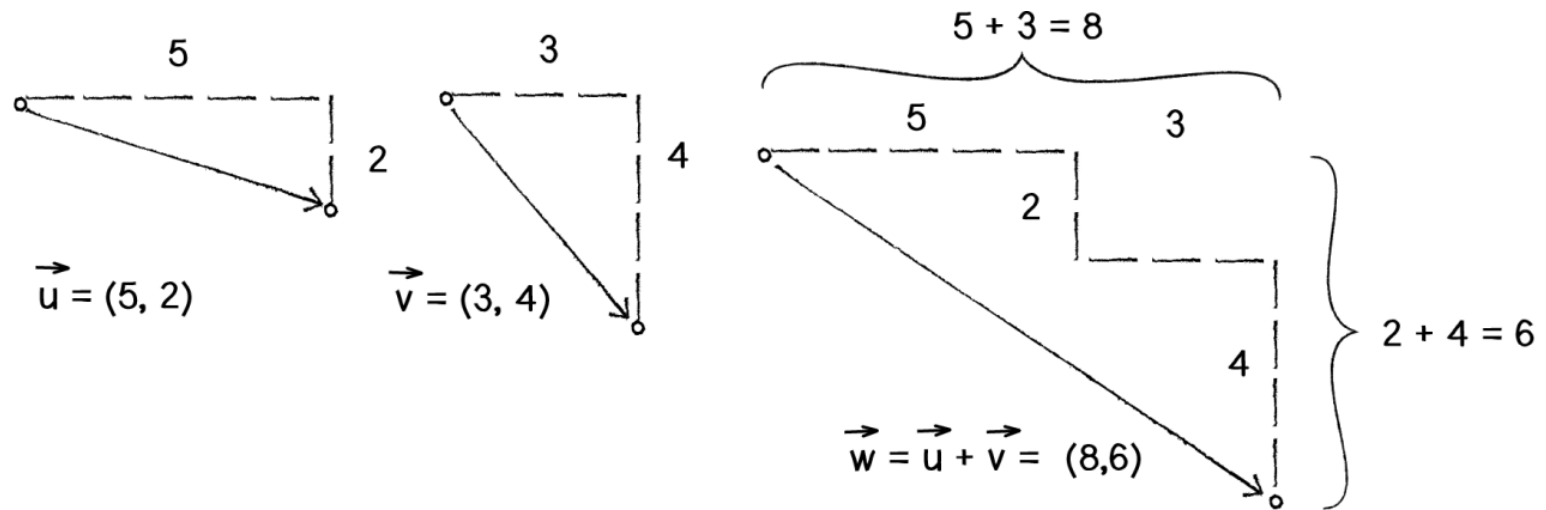


~~$\text{location} = \text{location} + \text{velocity} (?)$~~

벡터 덧셈

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

벡터 덧셈



x 성분은 x 성분끼리
y 성분은 y 성분끼리

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
```



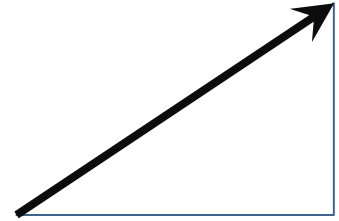
~~location = location + velocity (?)~~

location.add(velocity)

다른 함수들

- `add()` 더하기
- `sub()` 빼기
- `mult()` 벡터 크기를 실수배
- `div()` 벡터 크기를 나누기
- `dot()` 스칼라곱
- `cross()` 벡터곱 (3차원에서 지원)

다른 함수들



- `mag()` 벡터의 크기 구하기
- `normalize()` 단위벡터로 만들기
 - > 크기보단 **방향**이 중요할 때 주로 사용
- `limit()` 벡터의 크기 제한

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 더
한다

display 함수
위치 벡터가 가리키는 지점에 타원
을 그린다

checkEdges 함수
타원이 창을 벗어나면 창 안으로 되돌
린다

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

벡터의 쓰임

가속도

1. 일정한 가속
2. 랜덤 가속
3. 마우스를 따라 가속

벡터의 쓰임

1. 일정한 가속

```
class Mover {  
  
    PVector location;  
    PVector velocity;  
    PVector acceleration;  
}
```

가속도 벡터를 추가한다

벡터의 쓰임

1. 일정한 가속

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

update 함수

속도 벡터에 가속도 벡터를 더한다

