Force Simulator

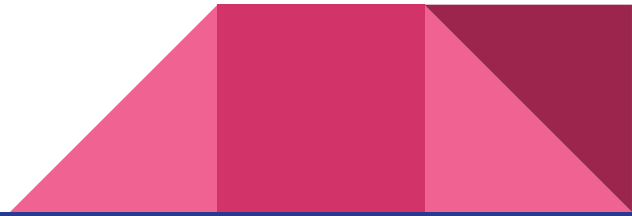
Processing

What we need to simulate an object in movement?

- Location x and y
- Velocity xVel and yVel

The velocity will make the object change its location:

```
x = x + xVel;
y = y + yVel;
```



How can we implement this in Processing?

Let's say the object we want to move is a ball, so let's create a class Ball:

```
class Ball{
  float x, y, xVel, yVel;

  Ball(float x, float y, xVel, yVel){
    this.x = x;
    this.x = y;
    this.xVel = xVel;
    this.yVel = yVel;
  }

  void move(){
    // change location based on velocity
  }
}
```

The setup() and draw() method will create objects Ball and simulate the movement.

Let's start coding

You will need to work with 2 files saved in the same processing folder

- ForceSmulator.pde
- Ball.pde

To add a second class to the project:

Click here and select "New Tab"

Vectors

A **vector** is an entity with magnitude and direction.

A **location** in terms of (x, y) in a Cartesian coordinate system refers to a vector that describes the displacement of a points from the origin of the coordinate system.

The x and y values are the horizontal and vertical components of a vector.

Velocity is the rate of change in the position of an object in a certain direction. Since it has both magnitude and direction, it is a vector.

v = d/t

v => velocity, d => displacement (change in position), t is time.

If an object has a velocity of 1m/s, it changes its position by 1 meter in 1 second.

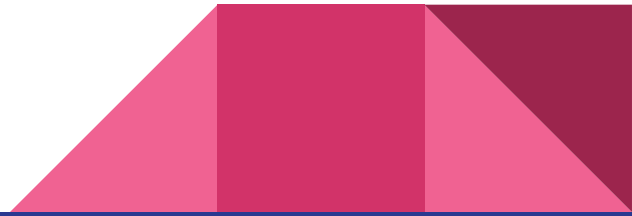
PVector

It is a class in Processing that stores the components of a vector (2 or 3 dimensional).

Fields: x, y, z

Methods: add(), sub(), mult(), div(), mag()

Documentation: https://processing.org/reference/PVector.html



Converting location and velocity to vectors

```
float x, y;
float xVel, yVel;
```

```
PVector location = new PVector(x, y);
PVector velocity = new PVector(xVel, yVel);
```

How should we update the location based on the velocity using vectors?

```
void move(){
  x = x + xVel;
  y = y + yVel;
}
```

```
void move(){
    location.add(velocity);
}
```

Which means add the velocity x component to the location x component, same for the y component.

Acceleration

While velocity changes location over time. **Acceleration changes velocity over time**. It could make velocity change magnitude and direction.

We can add acceleration to our programs and observe what happens:

We need a variable acceleration, which data type should it be?

```
PVector acceleration = new PVector(x, y);
```

How can we use this variable to Change velocity over time?

```
velocity.add(acceleration);
```

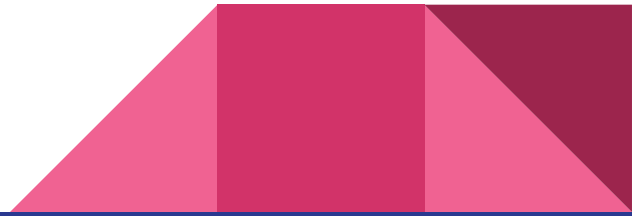
Let's update our code to use vectors and add acceleration.

You can add acceleration like this:

```
acceleration = new PVector(0.2, 0); // play with the numbers
```

OR

```
acceleration = PVector.random2D();
```



Force

Force is a vector that causes an object with mass to accelerate.

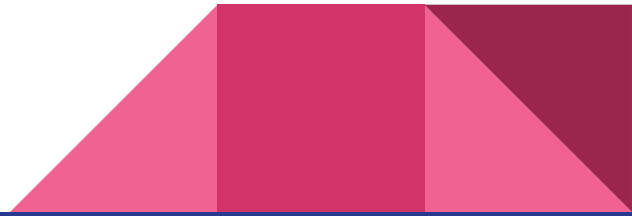
Examples: gravity, wind

"Every object will remain at rest or in uniform motion in a straight line unless compelled to change its state by the action of an external force," (Newton's first law)

F = mass * acceleration (Newton's second law)
acceleration = F / mass

Mass is the amount of matter in an object.

Bigger objects accelerate less and more force is needed.



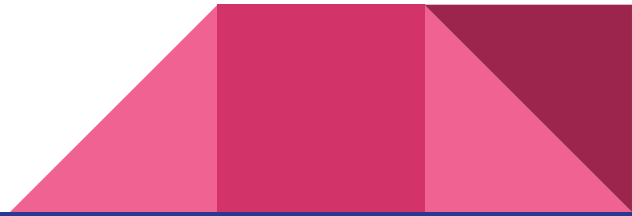
How are we going to apply a force?

```
PVector force = new PVector(0, 0.01);
ball.applyForce(force);
```

Where should the ball move after applying that force?

So, let's call that force gravity:

```
PVector gravity = new PVector(0, 0.01);
ball.applyForce(gravity);
```



Force of gravity

F = (m1 * m2 * G) / d^2

m1= Earth's mass (constant)

m2 = Our object mass (ball)

G = Universal gravitational constant

d = Distance between the centers of the 2 objects (assuming is constant)

F = C * mass of object

A= F / mass => A = C * mass/mass

A = C

The acceleration due to gravity is independent of the mass of the object.

Adding another force

If we add a new force, how can the ball use both forces?

We have to accumulate the forces in acceleration (not only 1, it will replace acceleration).

acceleration = SUM(forces) / mass