Variables, Classes, and Types

Recap

- Last week we discussed compared a variable to a bucket.

Variables

- Primitive types
  - int
  - float
  - string
  - long
  - double
  - boolean
  - color
  - byte

Variables

- Primitive types
  - **int**
  - **float**
  - long
  - double
  - **boolean**
  - **color**
  - byte

Variables

- Composite types
  - array
  - arrayList
  - HashMap
  - Object
  - String
  - XMLElement
  - Any Classes You make

Variables

- Composite types
  - **array**
  - **arrayList**
  - HashMap
  - **Object**
  - **String**
  - XMLElement
  - **Any Classes You make**

Type

- Every variable has a type
- Types can lead to frustrating errors
- Think of types as mathematical units like m, cm, s, m/s, etc
- 10m can't be turned into 10s, they are different units
- However, you could change m to cm
- Similarly, you can change an int to a float

Type

- The danger
- changing types can only increase precision
- int 2 can become float 2.0
- float 2.1 can not become int 2, because of the lose of precision

Classes

- Classes are Composite types
- The type is the name given to the class.

```
class Particle
{
   ...
}
```

Classes

```
Particle p = new Particle();

p.draw()
```

A Simple Class

```
class Particle
{
  int x, y, r;
  float velocityY;
  int life;

  Particle(int positionX, int positionY, int radius)
  {
    x = positionX;
    y = positionY;
    r = radius;
    velocityY = 0.0;
    life = 255;
  }
}
```

A Simple Class Function

```
void draw()
{
  ellipse(x,y,r,r);
}
```

Using a Class

```
Particle p;

void setup() {
   size(500,500);
   noStroke();
   fill(32,200,205);
   background(0,0,0);
   frameRate(30);
}
void draw() {
  background(0,0,0);
  if(p!=null)
  {
    p.draw();
  }
}
void mousePressed() {
  p = new Particle(mouseX, mouseY, 20);
}
```

Assignment

```
class Number {
      int i;
      Number()
      {
          i = 0;
      }
}

Number n1 = new Number();  --> n1 = 0;
Number n2 = new Number();  --> n2 = 0;

n1.i = 2; -->n1 = 2;
n2.i = 3; -->n2 = 3;

n1 = n2;  -->n1 and n2 = 3

n2.i = 4  -->n1 and n2 = 4... why?
```

Exercise

- Add a bounce function to the particle class
  - You will need to increase the life of the particle (with out changing it's value from 255)
  - You will need to check for when the ball reaches the bottom of the screen (height = 500 at the BOTTOM of the screen)
- Add multiple particles, slightly offset, the the mouse is pressed
  - You will need more Particle variables
  - You will need to change the mouseX and MouseY values slightly
  - You will need to call draw() for each Particle variable you make