

Objects

- ▶ Like many other programming languages, Processing is an object oriented language
- ▶ Objects allow us to encapsulate data and functions into one programming construct
- ▶ Objects provide a mechanism for simulating or modeling a particular entity.
- ▶ When we program in Processing, we create a class which contains a blueprint to create objects
- ▶ An object is an instance of a class

Example: a face

- ▶ Data (stored using class variables)
 - ▶ size
 - ▶ location
 - ▶ other attributes
- ▶ Functionality (provided with functions)
 - ▶ draw
 - ▶ move left
 - ▶ move right
 - ▶ blink
 - ▶ other functions

Anatomy of a class

- ▶ The definition of a class needs the following:
 - ▶ classname
 - ▶ body in curly braces
 - ▶ class variables (optional)
 - ▶ constructor function (optional)
 - ▶ other functions (optional)

A face class

```
class Face {  
  
    int x;  
    int y;  
    int size;  
  
    Face() { // constructor  
    }  
  
    // other functions  
    void draw() { }  
    void moveLeft() { }  
    void moveRight() { }  
    void blink() { }
```

Using objects

- ▶ We can create and use objects from defined classes

```
Face f;
void setup() {
  f = new Face();
}

void draw() {
  f.draw();
}

void keyPressed() {
  if (key == 'b') f.blink();
  if (keyCode == LEFT) f.moveLeft();
  if (keyCode == RIGHT) f.moveRight();
}
}
```

Anatomy Revisited

```
class NAME {  
  
    // variable declarations  
  
    NAME(...) { // constructor  
        // variable initializations  
    }  
  
    // support functions
```