Making an example of Pacman

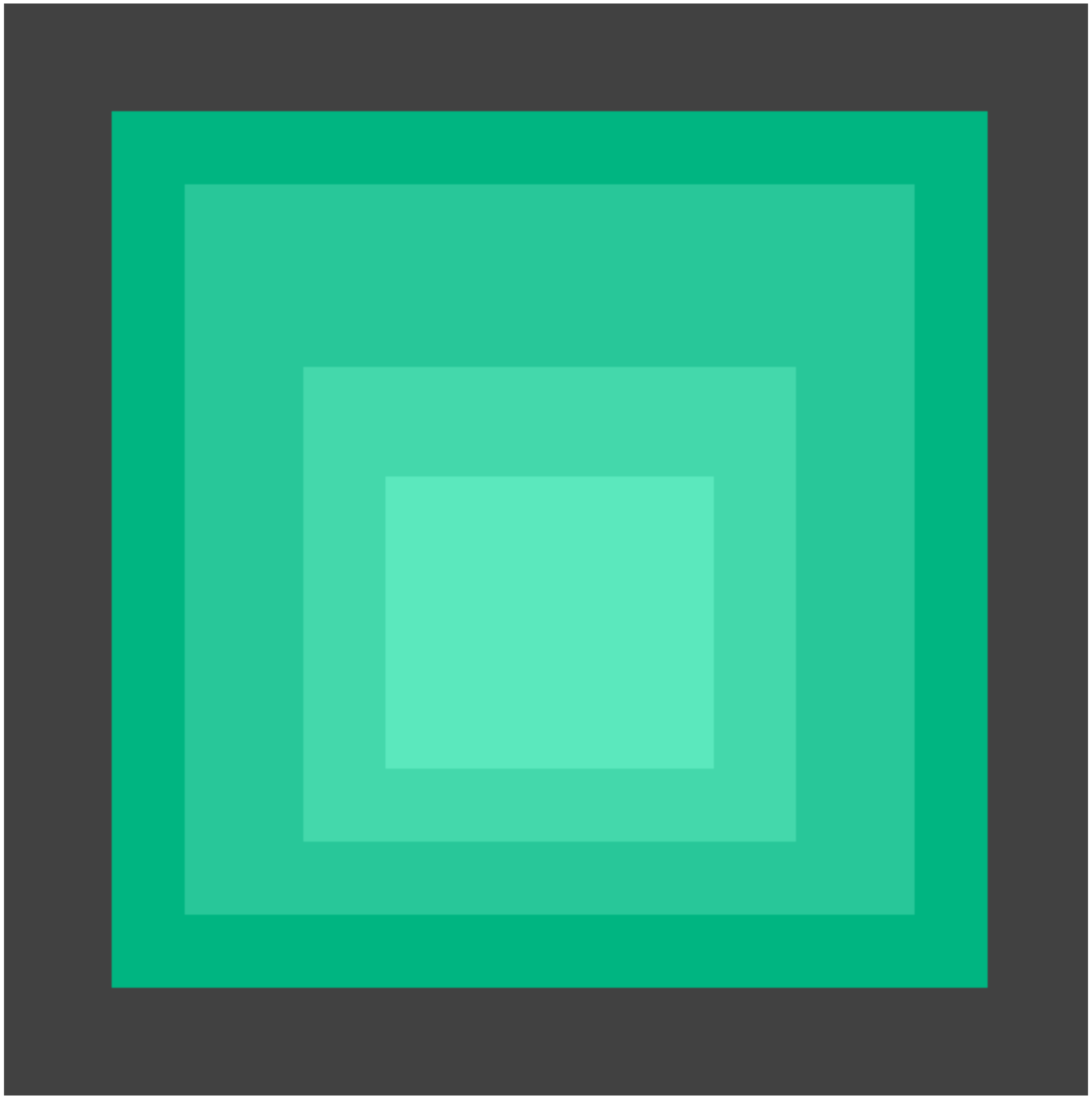
# Getting Into Processing

*Lawrence Snyder*
*University of Washington, Seattle*

# Announcements

- How's It Going?

Just Do It!



# Review

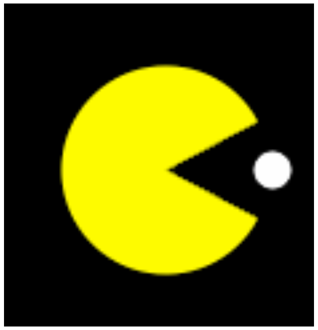
- Last time, we discussed communicating by blinking an eye, how an algorithm that orders the letters by frequency is better, how ASCII encodes North American and Western European letters, how UTF-8 covers all the rest of the symbols a computer should know about (100,000 at the moment), and that meta data is essential to say how information is used.
- Can you think of any application you use that uses meta data?

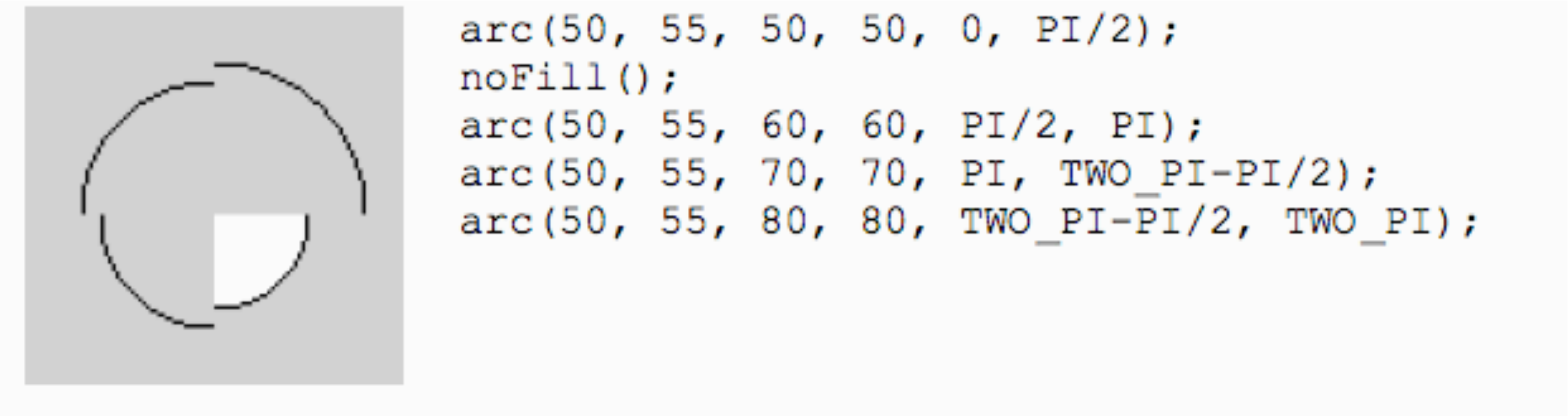
# Being Successful with Computers

- Two habits of computer people … they would be good ones for everyone
  - #1 When things go wrong, figure out what's wrong yourself … only ask when all else fails
    - Why do they behave like this? Stuff goes wrong all the time … you think you have problems with computers; try being an expert!
  - #2 Persistence … stay with it until you get it; everyone in class showed persistence with Lightbot 2.0 … we will need that a lot more
    - Why are they persistent? Because many things just take effort … and when you get it, it's satisfying!

# Check Out GSwP, 20-21

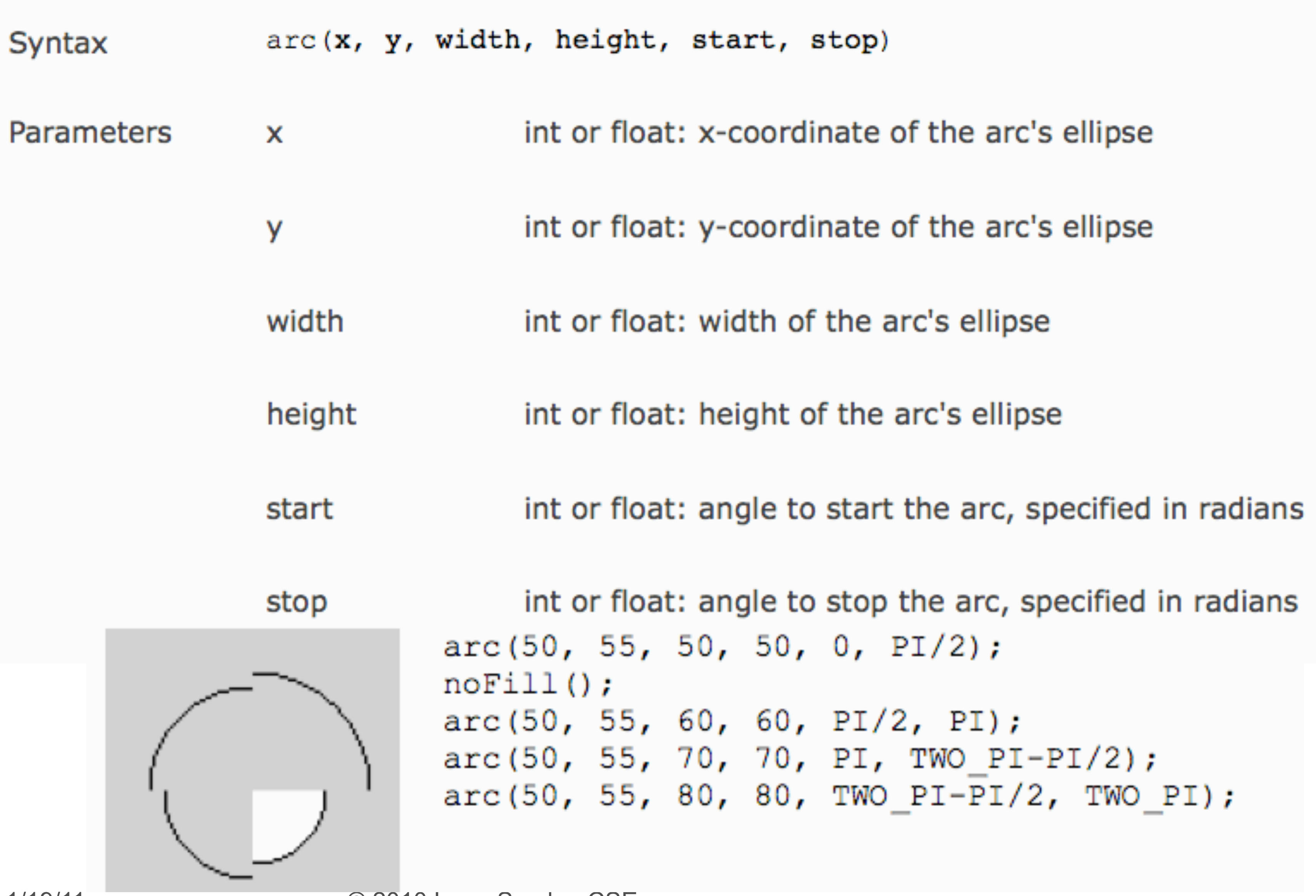
- Drawing arcs … a less intuitive operation as an example of working out the details
- Can we draw a figure that looks like Pacman?
- What does the tutorial say?





```
arc(50, 55, 50, 50, 0, PI/2);
noFill();
arc(50, 55, 60, 60, PI/2, PI);
arc(50, 55, 70, 70, PI, TWO_PI-PI/2);
arc(50, 55, 80, 80, TWO_PI-PI/2, TWO_PI);
```

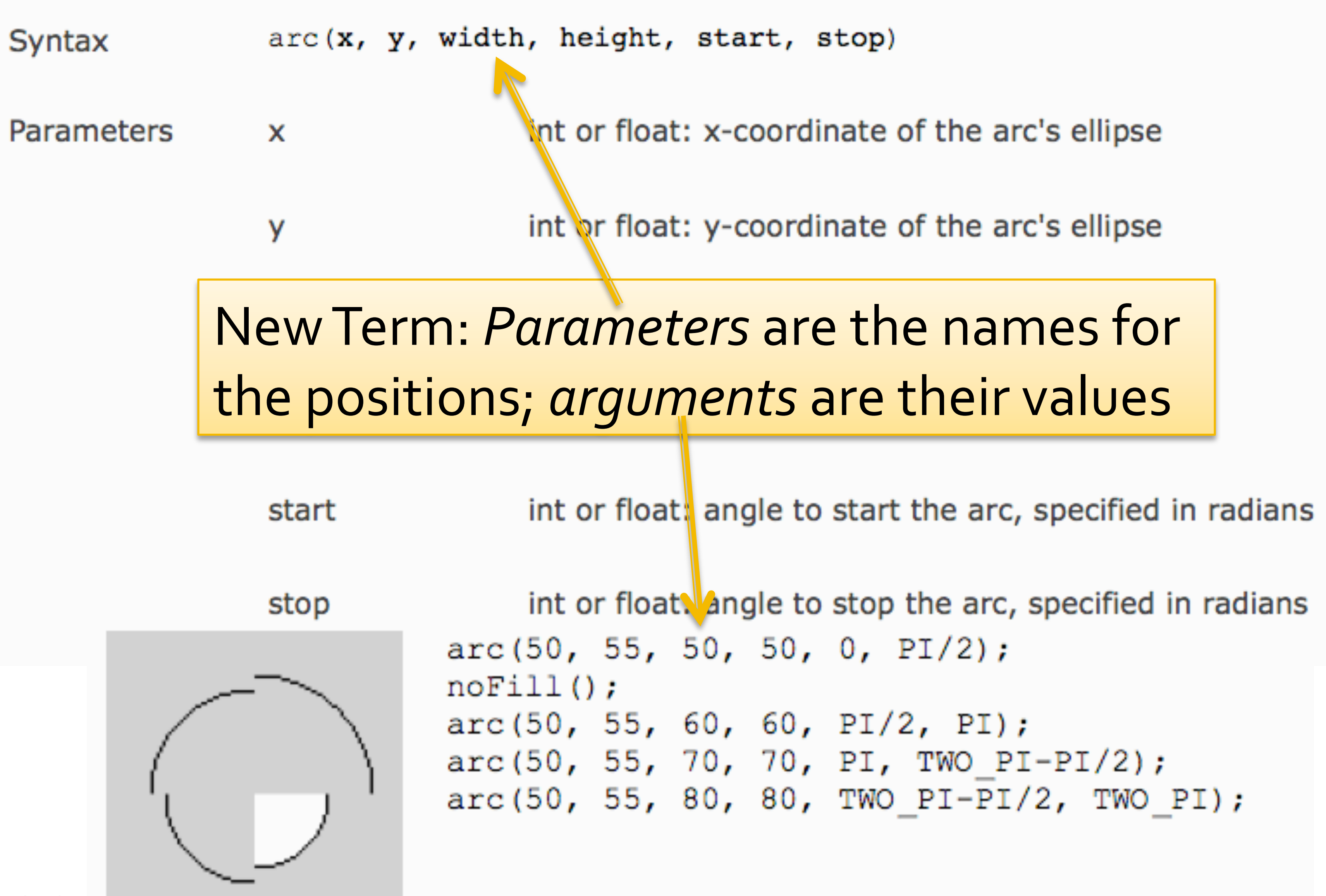
# What The Parameters Mean …



| | | |
|---|---|---|
| Syntax | `arc(x, y, width, height, start, stop)` | |
| Parameters | x | int or float: x-coordinate of the arc's ellipse |
| | y | int or float: y-coordinate of the arc's ellipse |
| | width | int or float: width of the arc's ellipse |
| | height | int or float: height of the arc's ellipse |
| | start | int or float: angle to start the arc, specified in radians |
| | stop | int or float: angle to stop the arc, specified in radians |

```
arc(50, 55, 50, 50, 0, PI/2);
noFill();
arc(50, 55, 60, 60, PI/2, PI);
arc(50, 55, 70, 70, PI, TWO_PI-PI/2);
arc(50, 55, 80, 80, TWO_PI-PI/2, TWO_PI);
```

# Two Related Terms …



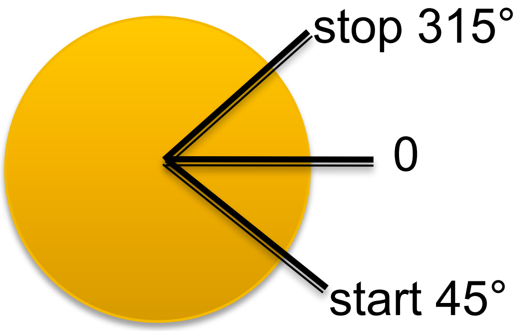
| | | |
|---|---|---|
| Syntax | `arc(x, y, width, height, start, stop)` | |
| Parameters | x | int or float: x-coordinate of the arc's ellipse |
| | y | int or float: y-coordinate of the arc's ellipse |
| | start | int or float: angle to start the arc, specified in radians |
| | stop | int or float: angle to stop the arc, specified in radians |

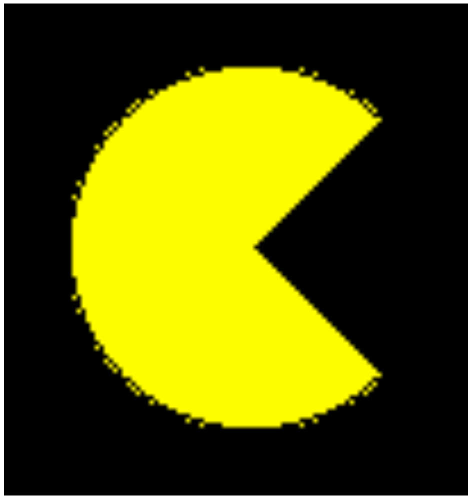
```
arc(50, 55, 50, 50, 0, PI/2);
noFill();
arc(50, 55, 60, 60, PI/2, PI);
arc(50, 55, 70, 70, PI, TWO_PI-PI/2);
arc(50, 55, 80, 80, TWO_PI-PI/2, TWO_PI);
```

New Term: *Parameters* are the names for the positions; *arguments* are their values

# Making Things Work For Ourselves

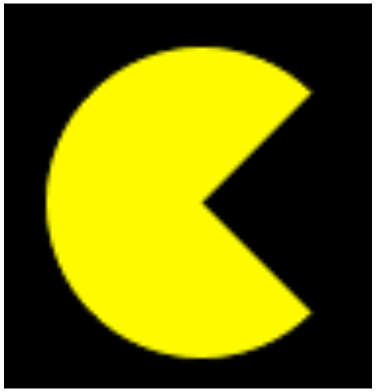
- What do we need to make Pacman?





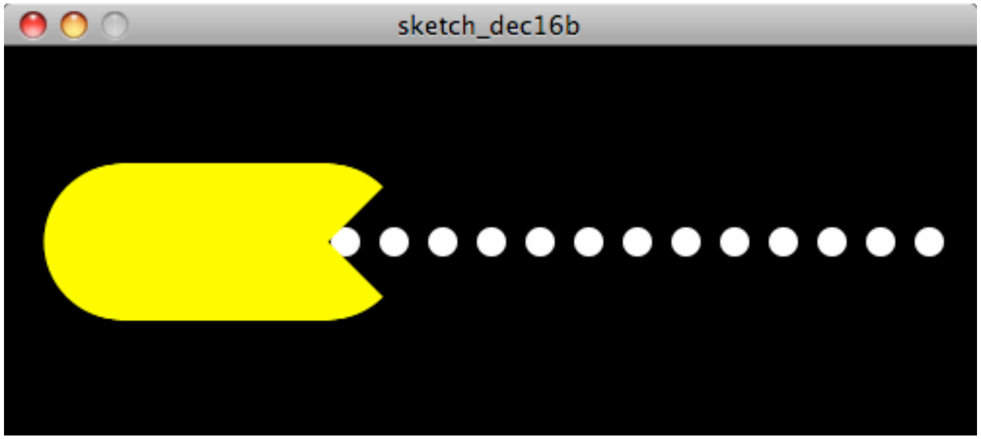
- arc( 100, 100, 80, 80, radians(45), radians(315));

smooth();



# The Basic Part is Defined …

- Now it's time to make it move!
  - For that, define the program so it is active



```
int mv = 60;

void setup( ) {
  size(500,200);
  background(0);
  noStroke();
  smooth();
  fill(255);
  for (int i=0; i< 16; i++) {
    ellipse(100+25*i, 100, 15, 15);
  }
}

void draw( ) {
  fill(255, 255, 0);
  arc(mv, 100, 80, 80, radians(45), radians(315));
  mv = mv + 1;
}
```

Add one to mv after each time the image is drawn

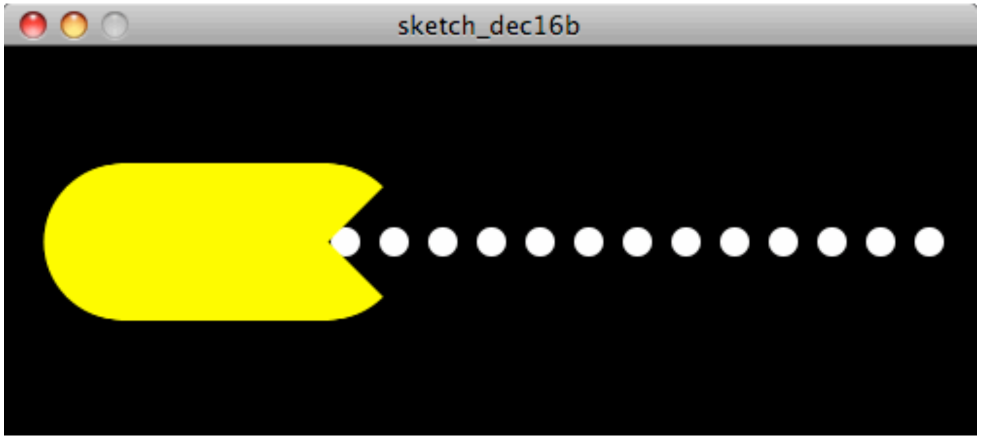
# The Basic Part is Defined …

- Now it’s time to make it move!
  - For that, define the program so it is active

```
int mv = 60;

void setup( ) {
  size(500,200);
  background(0);
  noStroke();
  smooth();
  fill(255);
  for (int i=0; i< 16; i++) {
    ellipse(100+25*i, 100, 15, 15);
  }
}

void draw( ) {
  fill(255, 255, 0);
  arc(mv, 100, 80, 80, radians(45), radians(315));
  mv = mv + 1;
}
```



Houston, we have a problem

Add one to mv after each time the image is drawn

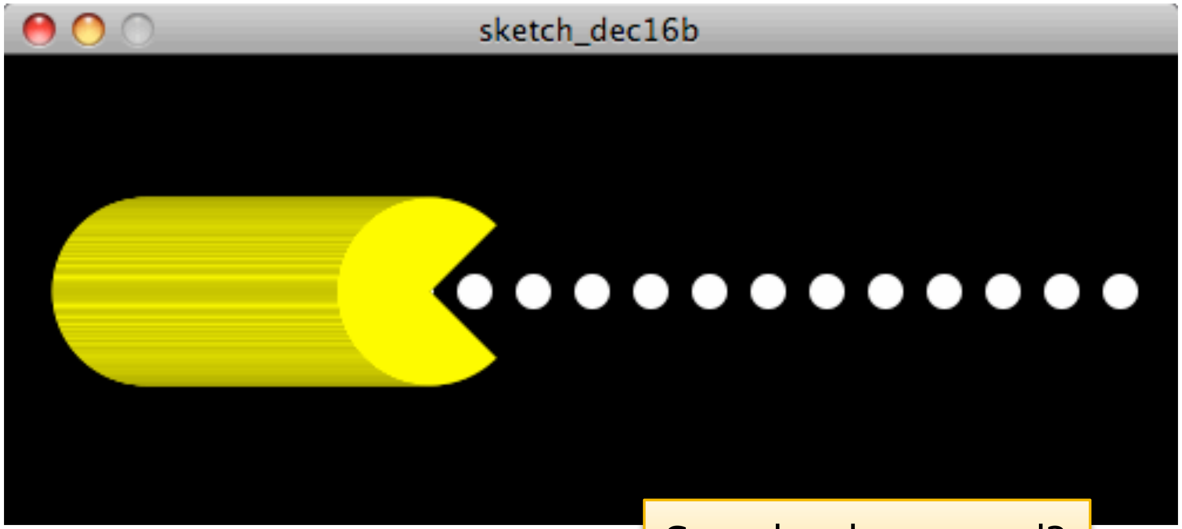
# Analysis of What's Happening

- We repeatedly redraw Pacman one position to the right, but he needs to be erased first.
  - A common trick – we've used it before – is to redraw the (black) background which rids us of Pacman, but it also rids us of the "pills"
  - We could redraw both, but then we need to figure out how many he's eaten so we draw only those ahead of him
  - Or, we could leave the pills there, and just erase him

# Following Tactic #3

- We first draw Pacman with black fill to obliterate him…

```
void draw( ) {
  fill(0);
  arc(mv, 100, 80, 80, radians(45), radians(315));
  fill(255, 255, 0);
  arc(mv, 100, 80, 80, radians(45), radians(315));
  mv = mv + 1;
}
```

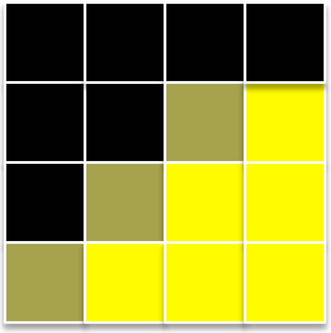


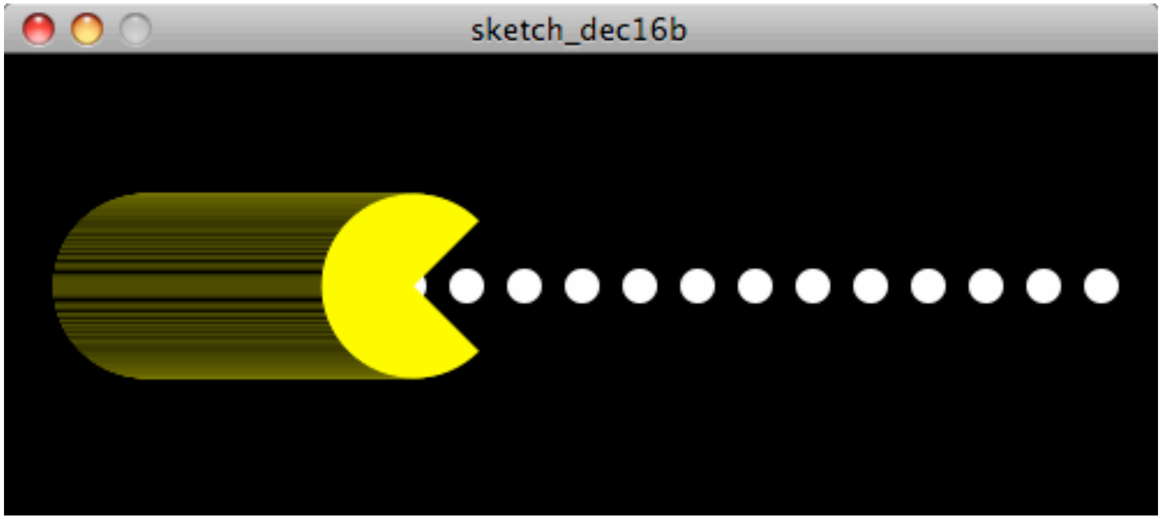
So, what happened?

# Moving mv=mv+1 Helps … Some

- The improved code rids us of nearly everything

```
void draw( ) {
  fill(0);
  arc(mv, 100, 80, 80, radians(45), radians(315));
  mv = mv + 1;
  fill(255, 255, 0);
  arc(mv, 100, 80, 80, radians(45), radians(315));
}
```



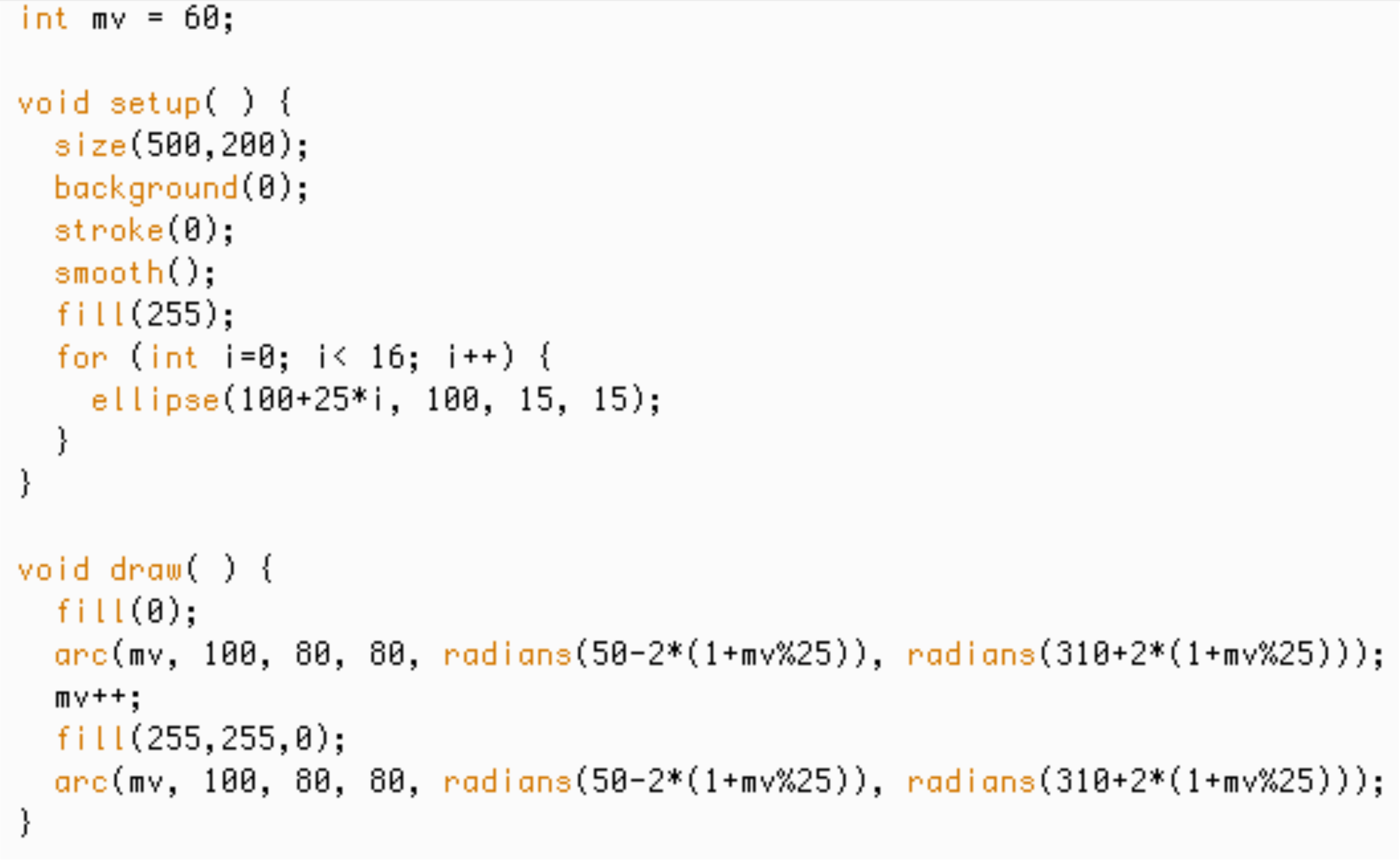


# Now What To Do?

- Ideas?

Just Do It!

# The Code To This Point … He Works!



```
int mv = 60;

void setup( ) {
  size(500,200);
  background(0);
  stroke(0);
  smooth();
  fill(255);
  for (int i=0; i< 16; i++) {
    ellipse(100+25*i, 100, 15, 15);
  }
}

void draw( ) {
  fill(0);
  arc(mv, 100, 80, 80, radians(50-2*(1+mv%25)), radians(310+2*(1+mv%25)));
  mv++;
  fill(255,255,0);
  arc(mv, 100, 80, 80, radians(50-2*(1+mv%25)), radians(310+2*(1+mv%25)));
}
```

# Planning the "Chomp"

- Clearly, to make Pacman chomp on the pills, we need to draw arcs that get closer and closer to being circles … so in the
  arc(mv,100,80,80,radians(45),radians(315))
  function, 45 → 0 and 315 → 360
- How much should we change them by?
  - 45 → 0 by at several rates
    - by 15 in 3 steps
    - by 9 in 5 steps
    - by 5 in 9 steps
    - by 3 in 15 steps

# Consider 9 Chomps of 5 Each

- We need to abstract
  - 45 – 5*0 = 45
  - 45 – 5*1 = 40
  - 45 – 5*2 = 35
  - 45 – 5*3 = 30
  - …
  - 45 – 5*9 = 0
- So, radians(45-5*<*number in 0 to 9*>)

# Mod – A Very Clever Operator

- Mod – short for modulo – is a programming operator like divide and is written with %
  - Mod gives the remainder from a division …
  - 4%4 is 0, because 4 divides into 4 evenly, i.e. 0 left
  - 5%4 is 1, because 4 goes into 5 once with 1 leftover
  - 6%4 is 2, because 4 goes into 6 once with 2 leftover
  - 7%4 is 3, because 4 goes into 7 once with 3 leftover
  - 8%4 is 0 again, because 4 divides 8 evenly
  - 9%4 is 1 again, because 4 goes into 9 twice w/ 1 left
  - …

mod is a very useful operator

# Using Mod with `mv`

- So, to get the numbers 0 through 9 for our equation, we need to write `mv%`<*what?*>

- We need to abstract
  - 45 – 5*0 = 45
  - 45 – 5*1 = 40
  - 45 – 5*2 = 35
  - 45 – 5*3 = 30
  - …
  - 45 – 5*9 = 0

# Using Mod with `mv`

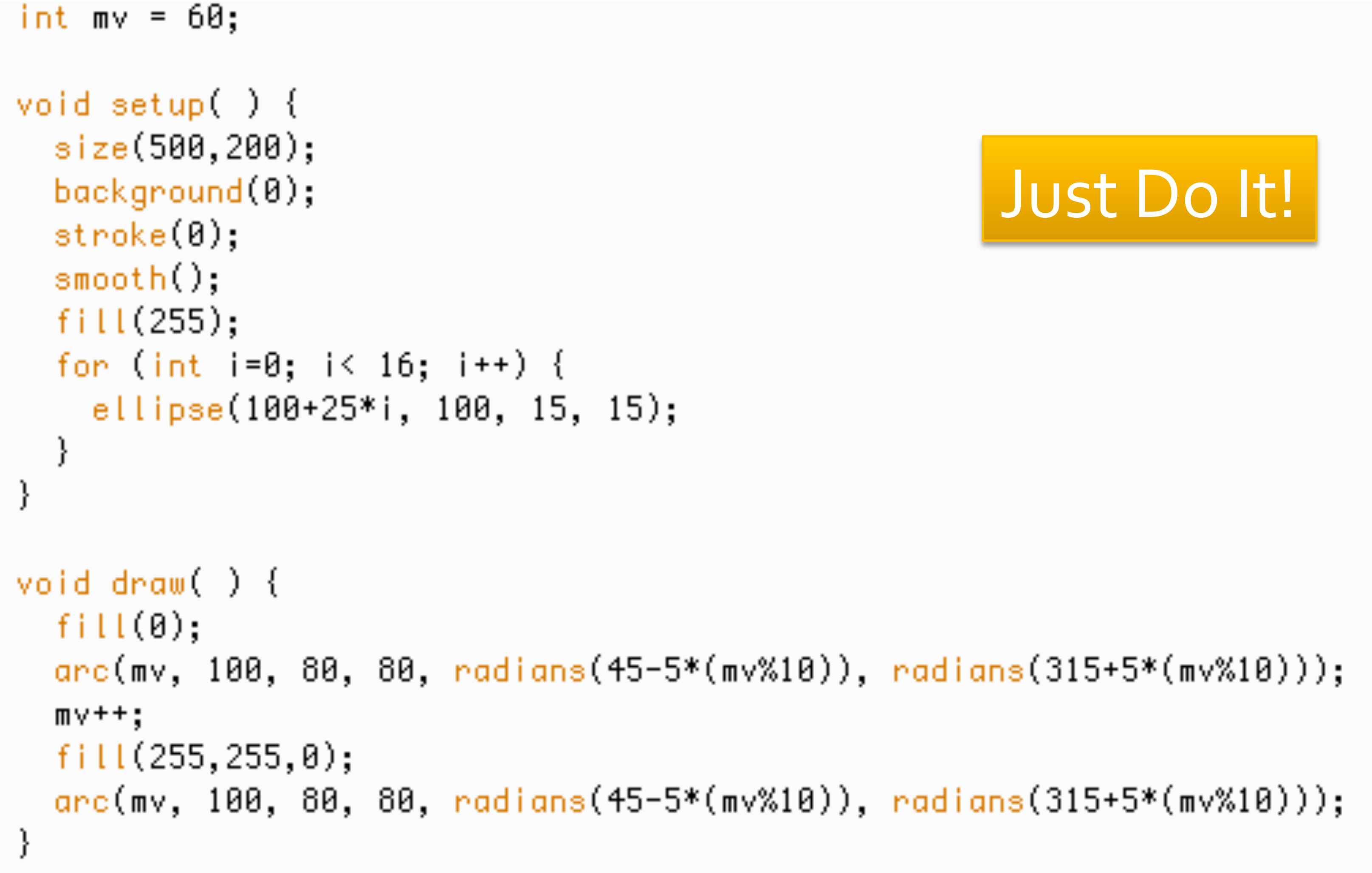
- So, to get the numbers 0 through 9 for our equation, we need to write `mv%`<*what?*>
- If you said "9" you were WRONG!
- `x%n` gives numbers 0 through n-1
- If you want numbers 0 through 9, you want 10 numbers or mv%10

- We need to abstract
  - 45 – 5*0 = 45
  - 45 – 5*1 = 40
  - 45 – 5*2 = 35
  - 45 – 5*3 = 30
  - …
  - 45 – 5*9 = 0

If you missed it, not to worry; EVERY programmer has, too

# Ready To Chomp

- The code …



```
int mv = 60;

void setup( ) {
  size(500,200);
  background(0);
  stroke(0);
  smooth();
  fill(255);
  for (int i=0; i< 16; i++) {
    ellipse(100+25*i, 100, 15, 15);
  }
}

void draw( ) {
  fill(0);
  arc(mv, 100, 80, 80, radians(45-5*(mv%10)), radians(315+5*(mv%10)));
  mv++;
  fill(255,255,0);
  arc(mv, 100, 80, 80, radians(45-5*(mv%10)), radians(315+5*(mv%10)));
}
```

Just Do It!

# Revise

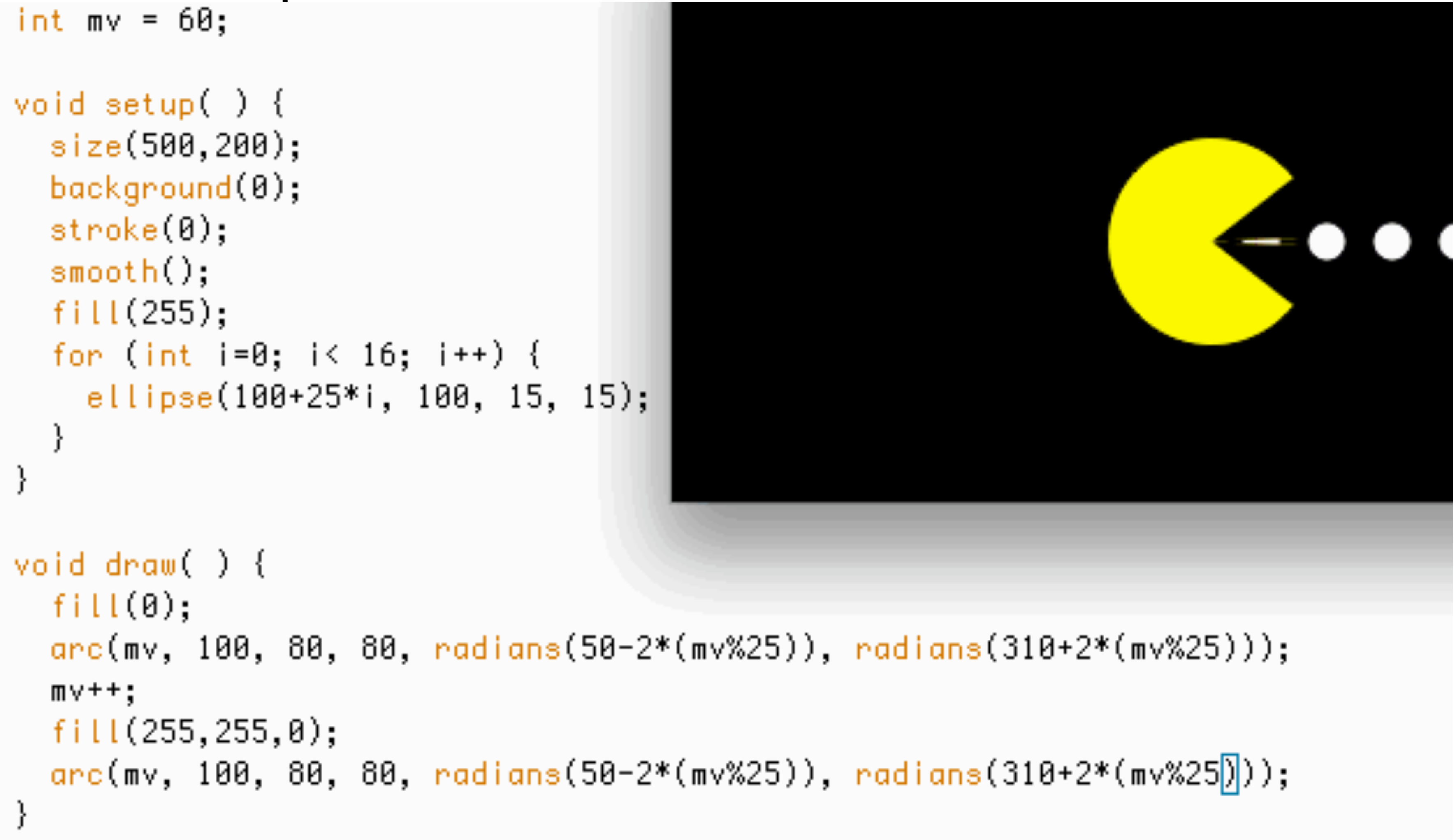
- Perhaps slower is better … need more frames, that is, repetitions of draw( ) to close mouth
- A rate of 3 in 15 steps would be somewhat better … meaning that the amount subtracted is 3*mv%<*what?*>
  … experimenting, I thought 25 frames was about right

What do you think.

Just Do It!

# The Code

- Whoops!

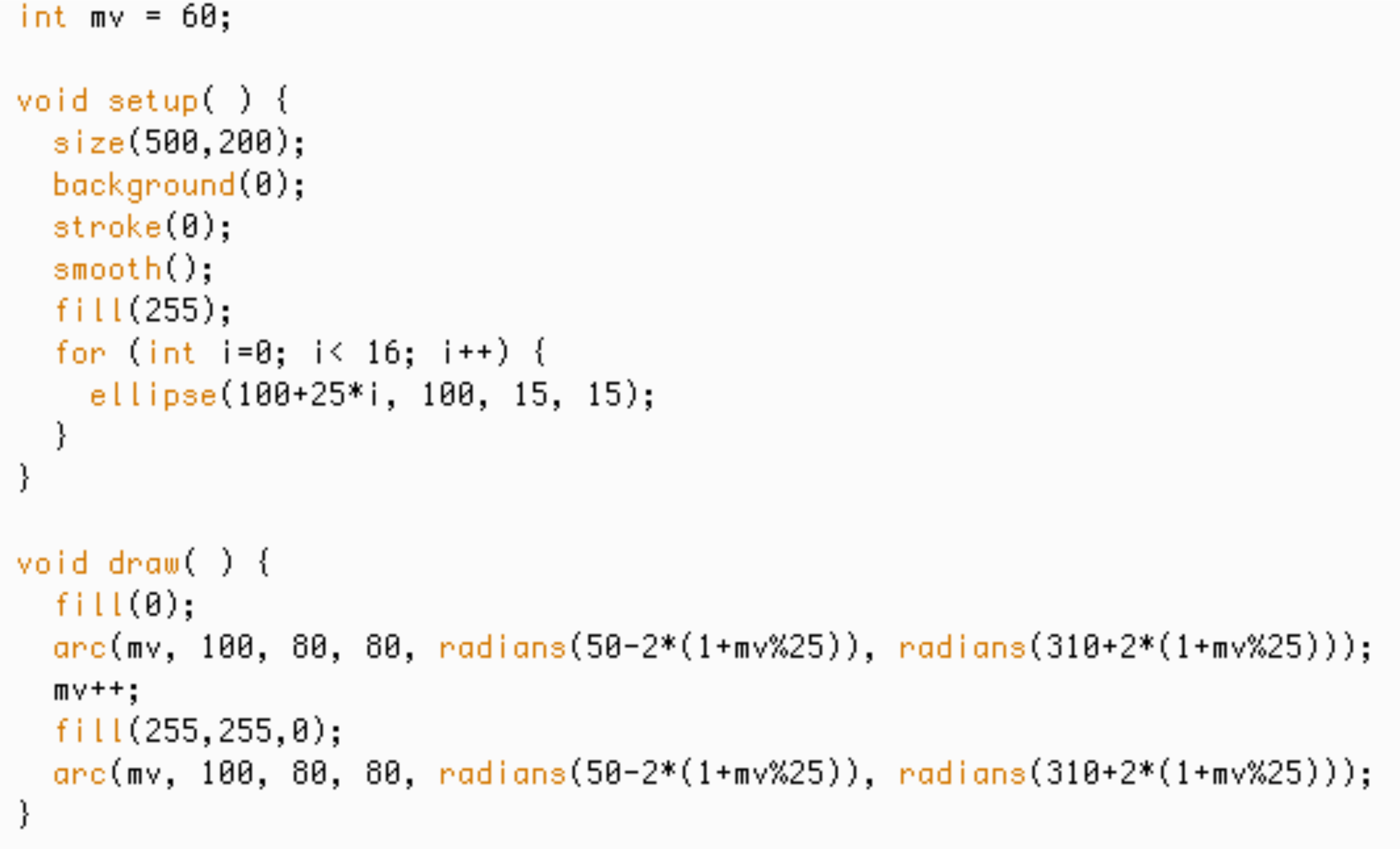


```
int mv = 60;

void setup( ) {
  size(500,200);
  background(0);
  stroke(0);
  smooth();
  fill(255);
  for (int i=0; i< 16; i++) {
    ellipse(100+25*i, 100, 15, 15);
  }
}

void draw( ) {
  fill(0);
  arc(mv, 100, 80, 80, radians(50-2*(mv%25)), radians(310+2*(mv%25)));
  mv++;
  fill(255,255,0);
  arc(mv, 100, 80, 80, radians(50-2*(mv%25)), radians(310+2*(mv%25)));
}
```

# The Code



```
int mv = 60;

void setup( ) {
  size(500,200);
  background(0);
  stroke(0);
  smooth();
  fill(255);
  for (int i=0; i< 16; i++) {
    ellipse(100+25*i, 100, 15, 15);
  }
}

void draw( ) {
  fill(0);
  arc(mv, 100, 80, 80, radians(50-2*(1+mv%25)), radians(310+2*(1+mv%25)));
  mv++;
  fill(255,255,0);
  arc(mv, 100, 80, 80, radians(50-2*(1+mv%25)), radians(310+2*(1+mv%25)));
}
```