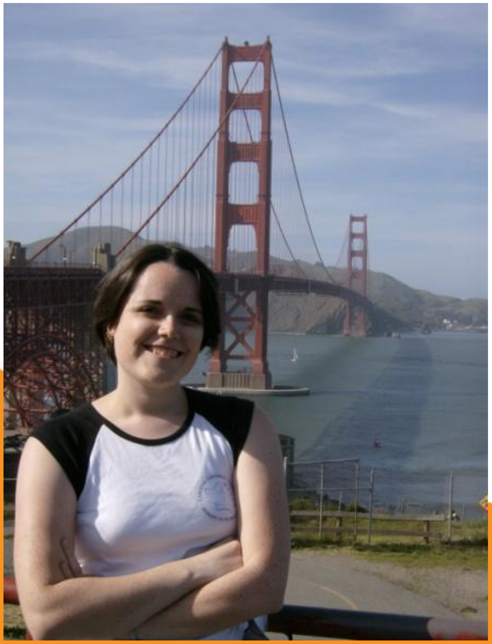
Welcome!

GO
CODE
GIRL
IMAGINE. DESIGN. CREATE.

ABOUT ME



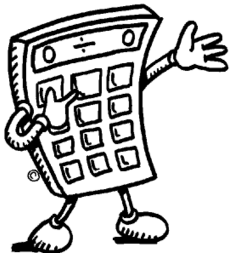
GO
CODE
GIRL
IMAGINE. DESIGN. CREATE.

Who Are You?

- What school do you go to?
- What grade are you in?
- What made you come to the workshop?
- What’s something interesting we can’t tell by looking at you?

GO CODE GIRL
IMAGINE. DESIGN. CREATE.

What is Computer Science?



Solving Problems!

GO
CODE
GIRL
IMAGINE. DESIGN. CREATE.

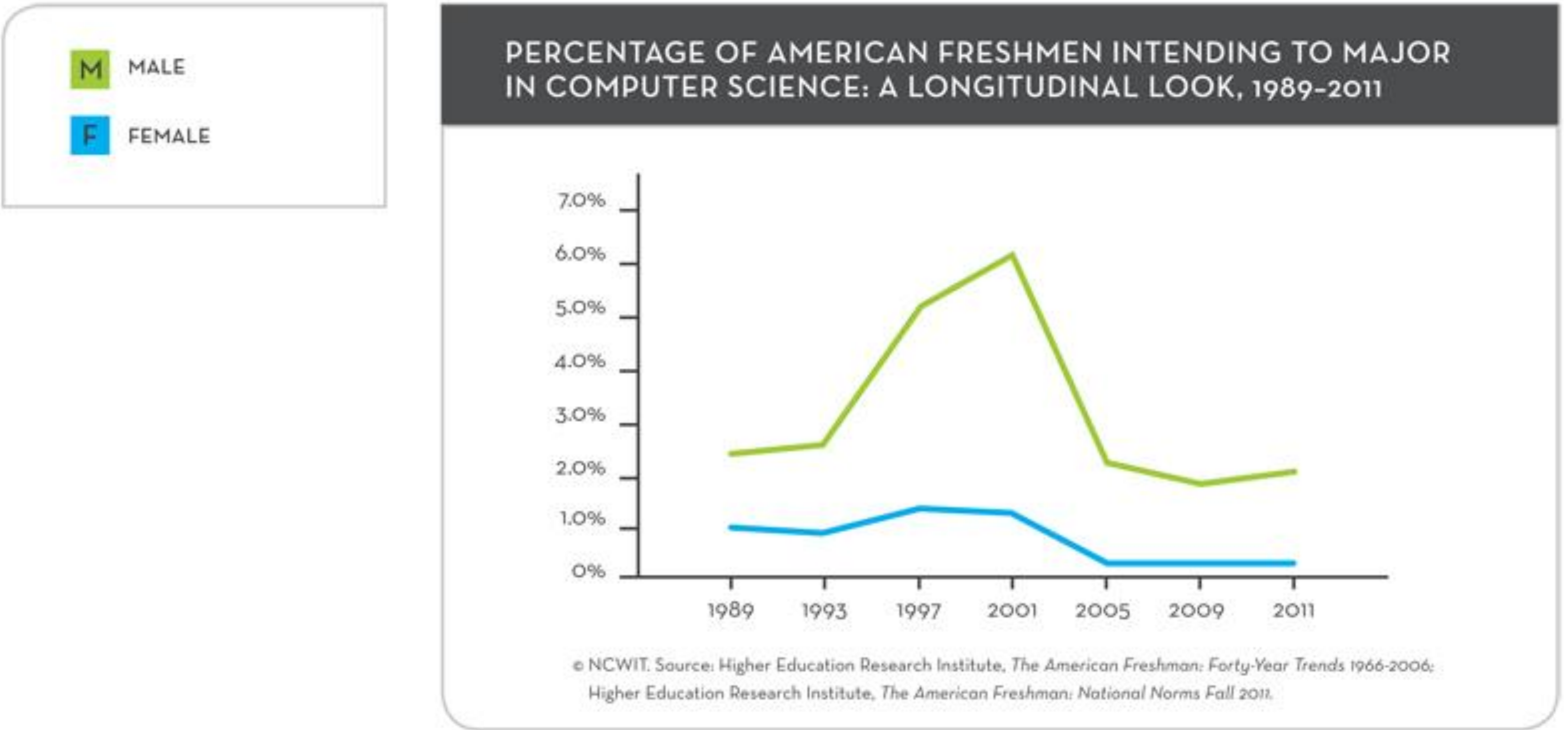
Pathways in Computer Science

http://www.youtube.com/watch?v=jq_EcstLlfE

What About Women in CS?

GO CODE GIRL

IMAGINE. DESIGN. CREATE.

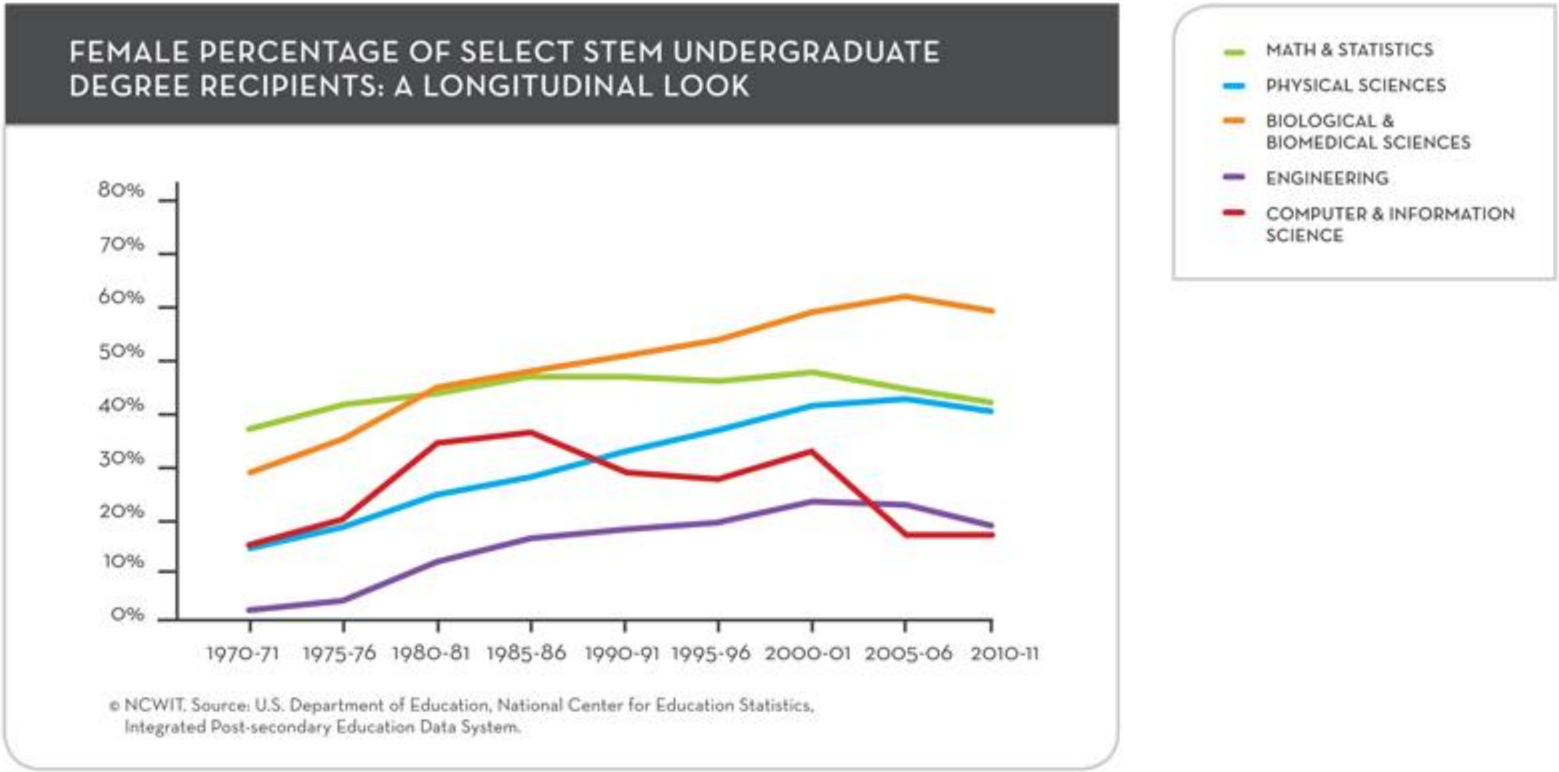


PERCENTAGE OF AMERICAN FRESHMEN INTENDING TO MAJOR IN COMPUTER SCIENCE: A LONGITUDINAL LOOK, 1989–2011

© NCWIT. Source: Higher Education Research Institute, *The American Freshman: Forty-Year Trends 1966-2006*; Higher Education Research Institute, *The American Freshman: National Norms Fall 2011*.

GO CODE GIRL

IMAGINE. DESIGN. CREATE.



FEMALE PERCENTAGE OF SELECT STEM UNDERGRADUATE DEGREE RECIPIENTS: A LONGITUDINAL LOOK

© NCWIT. Source: U.S. Department of Education, National Center for Education Statistics, Integrated Post-secondary Education Data System.

GO CODE GIRL

IMAGINE. DESIGN. CREATE.

Women Earn:

57% of all undergraduate degrees

42% of all undergraduate math and statistics degrees

40% of all undergraduate physical sciences degrees

but only …

18% of all undergraduate computer and information sciences degrees

http://www.ncwit.org/infographic/3435

Why Do We Have This Problem?

- Why don’t girls go into computer science?
- Why is this a bad thing?
- What kinds of things would make you interested in taking computer science in high school and university?

GO CODE GIRL

IMAGINE. DESIGN. CREATE.

Endless Possibilities of Computer Science

http://www.youtube.com/watch?v=DYBPotROKC8

Processing Demo!

GO CODE GIRL
IMAGINE. DESIGN. CREATE.

http://www.gailcarmichael.com/gocodegirl/demo.html

Variables

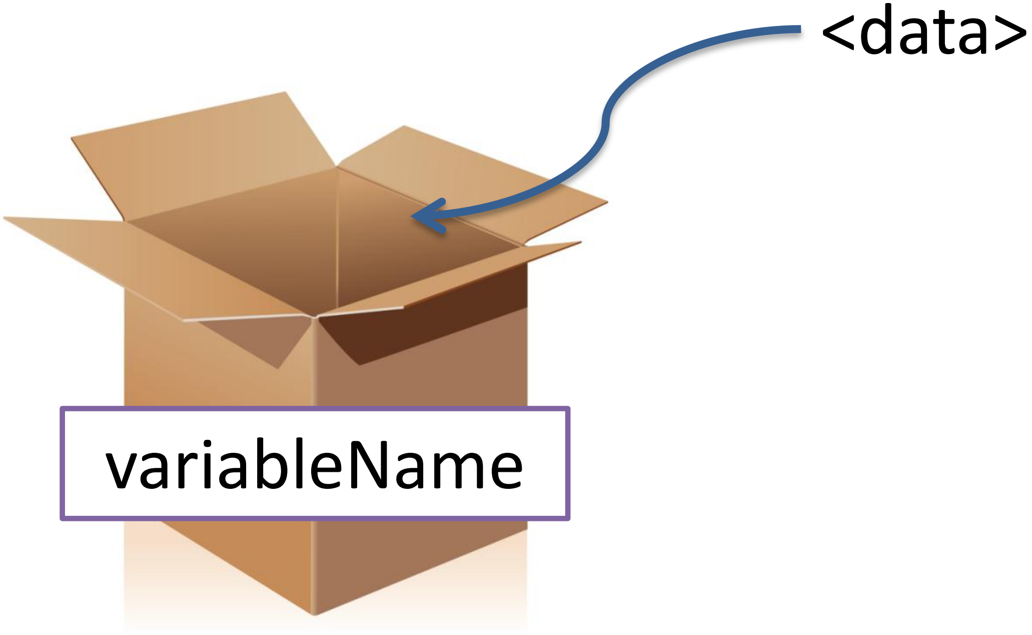
GO
CODE
GIRL
IMAGINE. DESIGN. CREATE.

Variables

variableName

```
int currentSequenceLength;
```

Variables

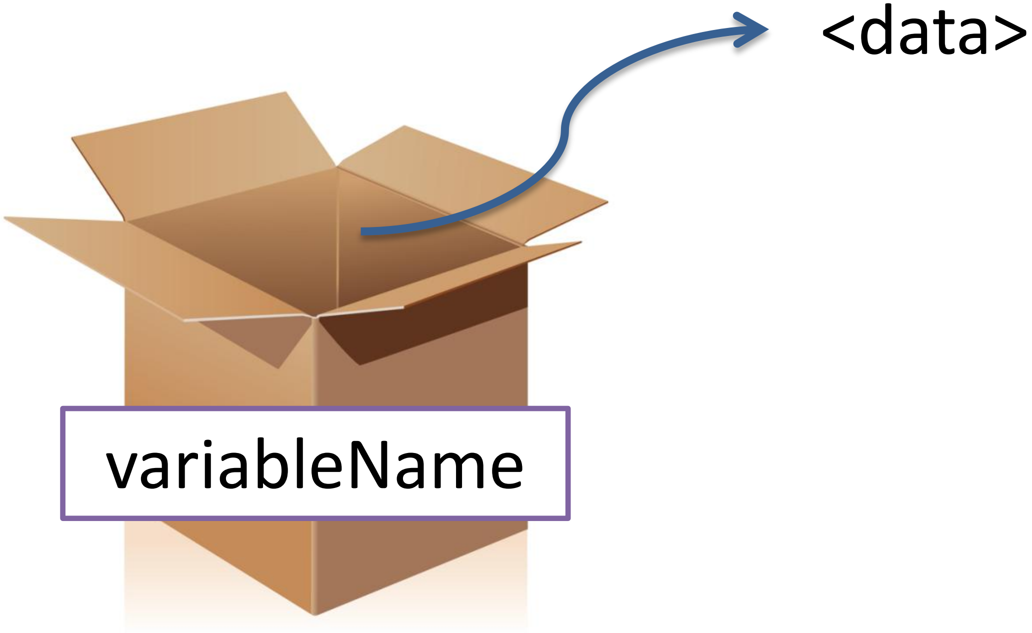


<data>

variableName

```
currentSequenceLength = 2;
```

Variables

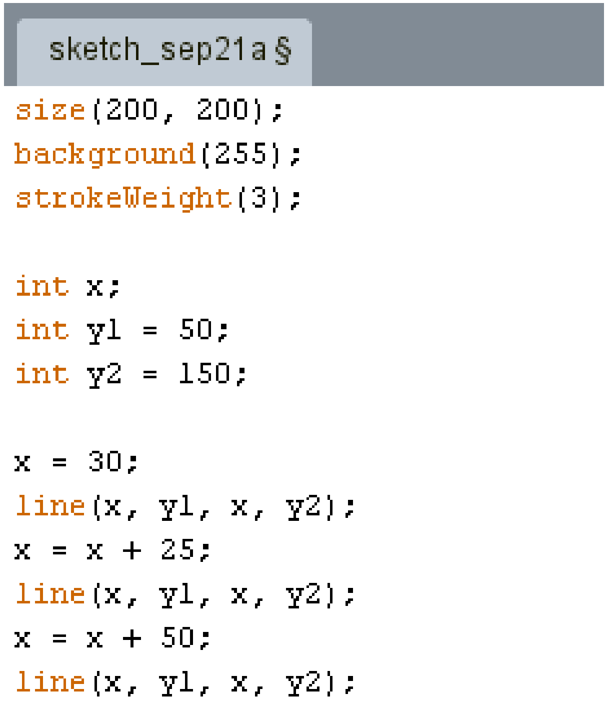


<data>

variableName

```
currentSequenceItemIndex == currentSequenceLength - 1;
```

```
Shape currentClickedItem = null; // for during user-click mode
Shape currentSequenceItem = null; // for showing-thesequence mode

PFont font;
String currentTextStatus;
final String memorizeText = "Memorize this...";
final String clickText = "Now you try.";
```



```
size(200, 200);
background(255);
strokeWeight(3);

int x;
int y1 = 50;
int y2 = 150;

x = 30;
line(x, y1, x, y2);
x = x + 25;
line(x, y1, x, y2);
x = x + 50;
line(x, y1, x, y2);
```

True, False, and If

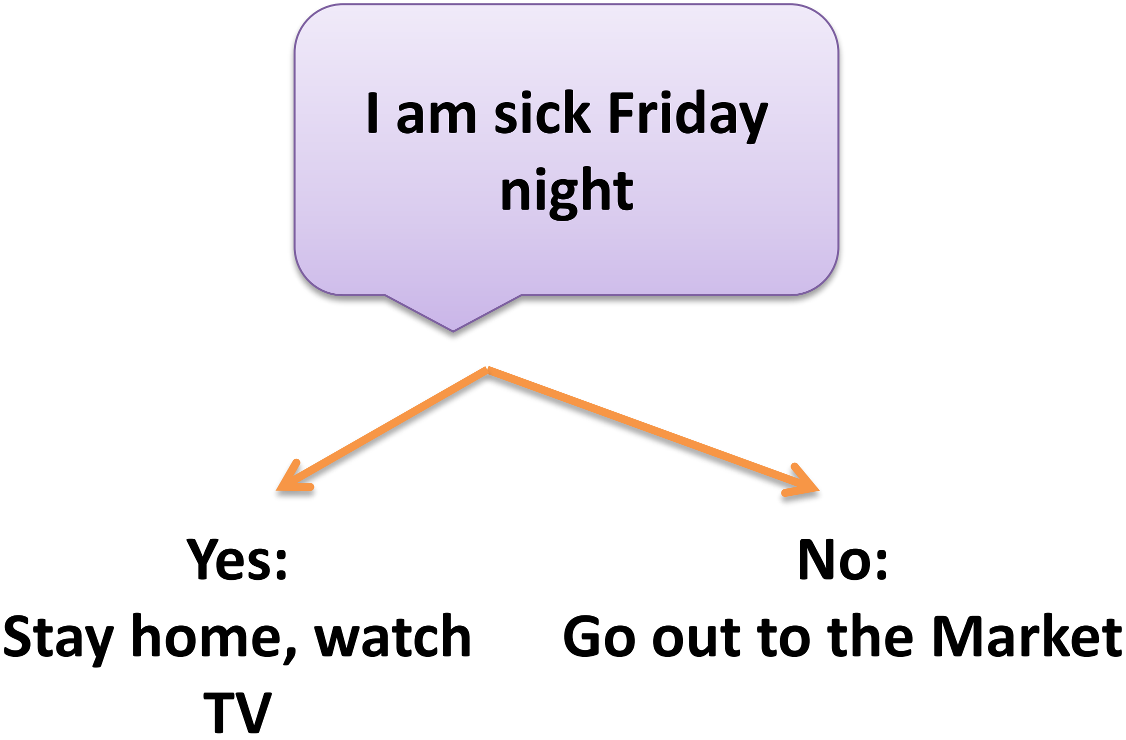
boolean

Yes/
True

or

No/
False

If/Else Statements

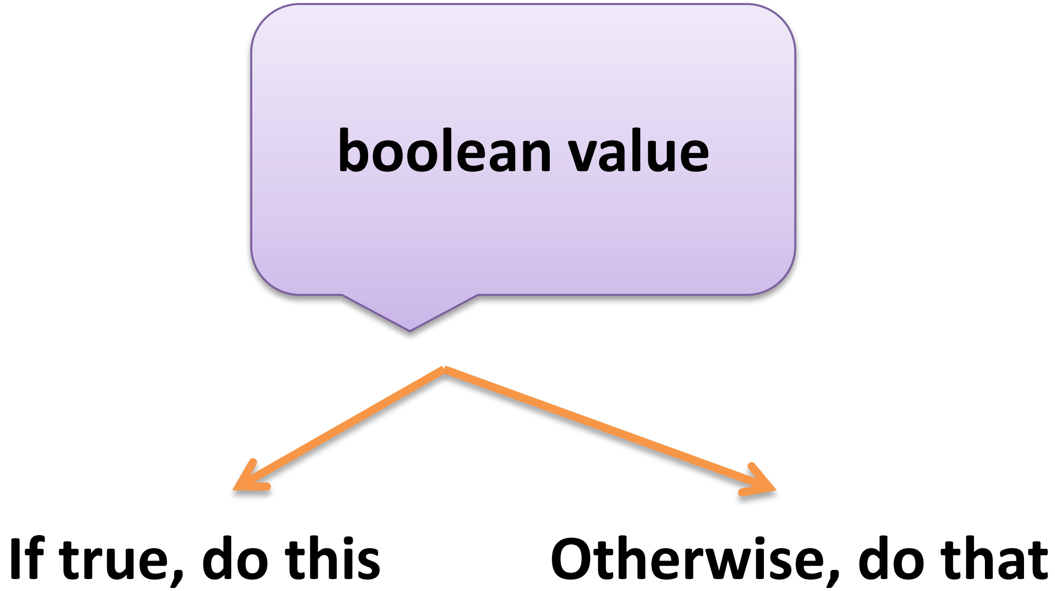


I am sick Friday night

Yes:
Stay home, watch TV

No:
Go out to the Market

If/Else Statements

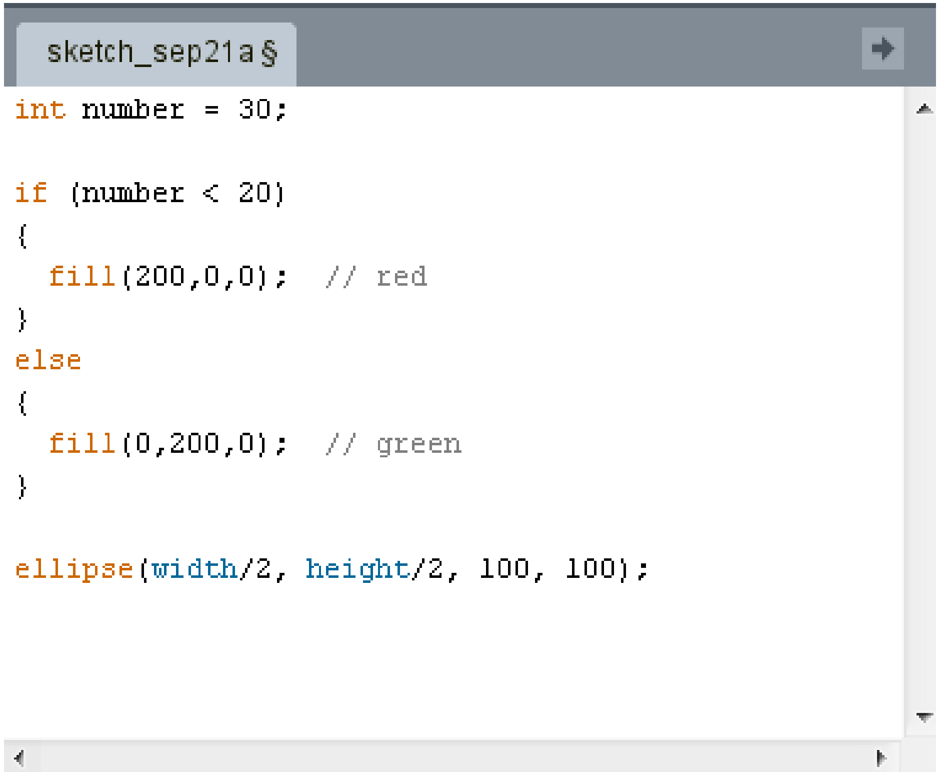


boolean value

If true, do this

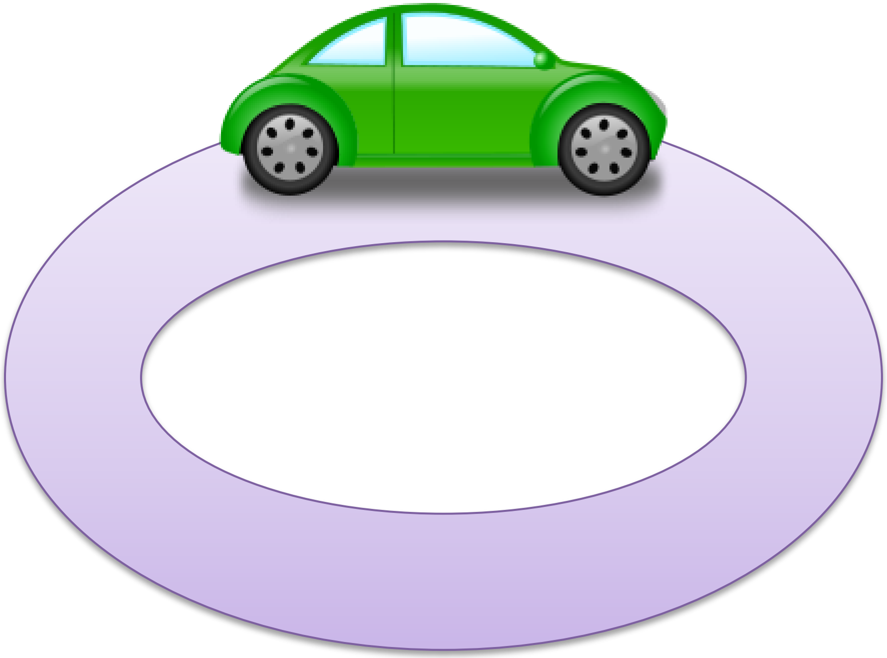
Otherwise, do that

```
if (atEndOfSequence())
{
  // Switch back to showing-sequence mode
  showingSequence = true;
  currentSequenceItemIndex = 0;
  currentSequenceItem = memorySequenceList.get(currentSequenceItemIndex);
  currentTextStatus = memorizeText;
  currentSequenceLength++;
  currentSequenceItem.startTimer();
}
else
{
  // Move to the next item
  currentSequenceItemIndex++;
  currentSequenceItem = memorySequenceList.get(currentSequenceItemIndex);
}
```



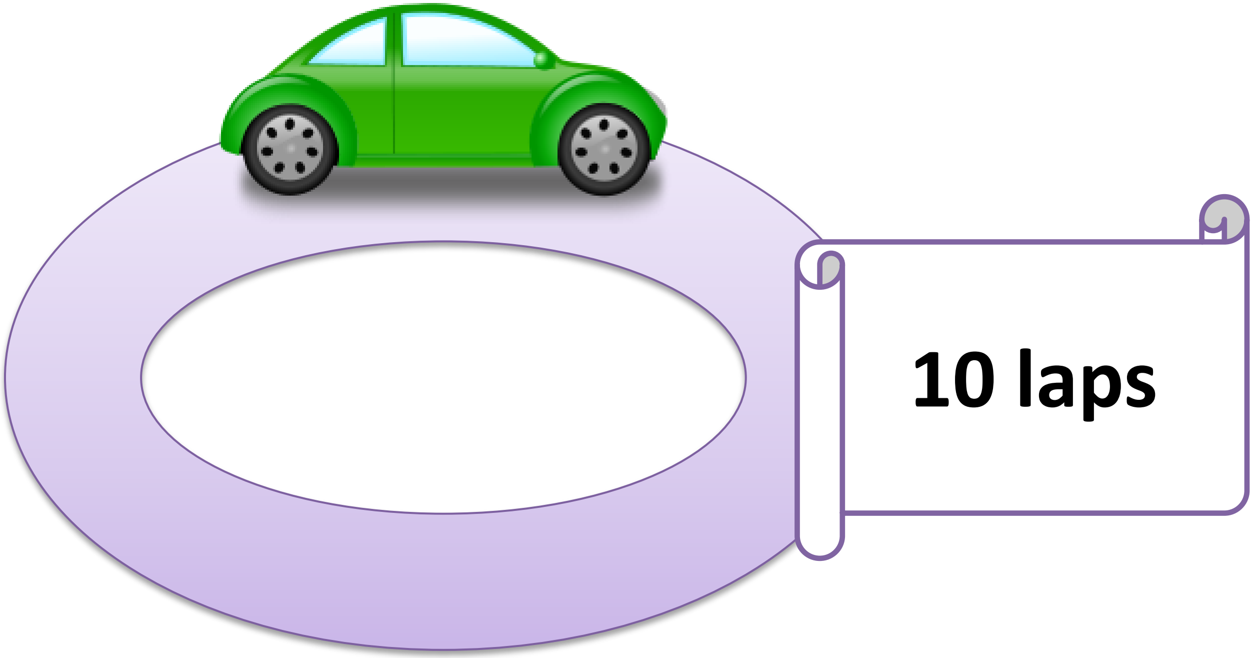
```
int number = 30;

if (number < 20)
{
  fill(200,0,0);  // red
}
else
{
  fill(0,200,0);  // green
}

ellipse(width/2, height/2, 100, 100);
```

Loops



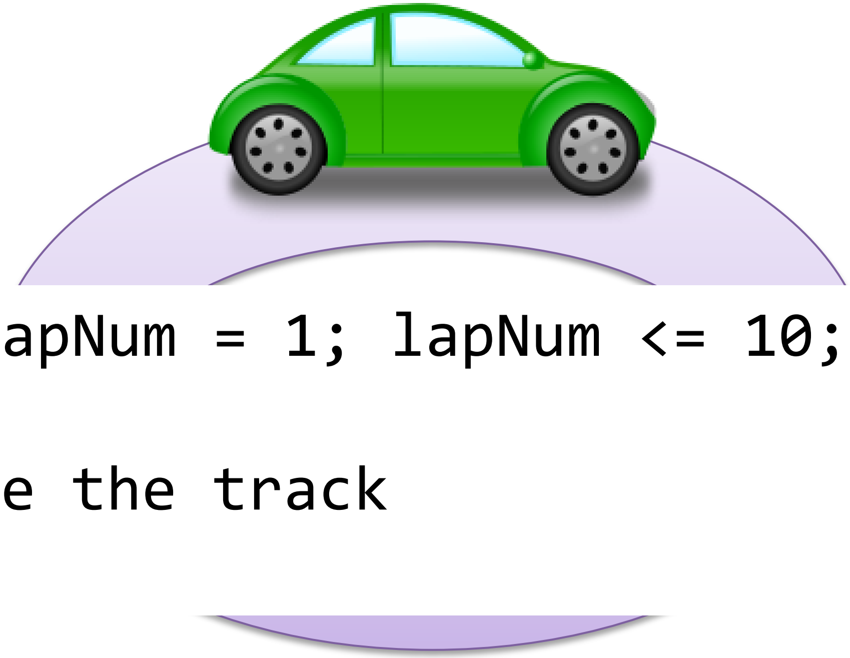
Drive the same track multiple times

for loop



Drive the same track exactly ten times

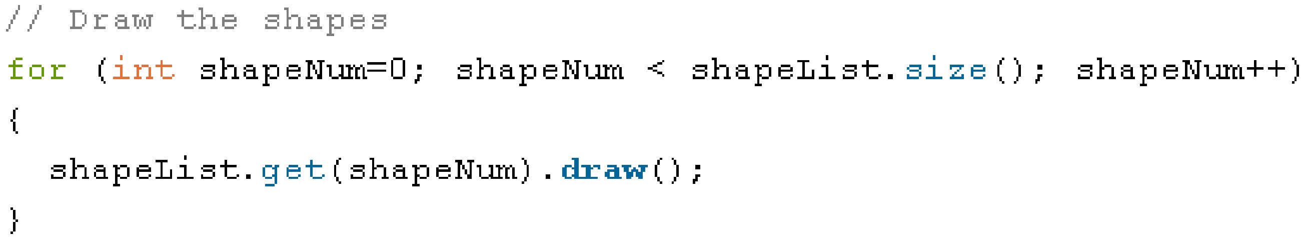
for loop



```
for (int lapNum = 1; lapNum <= 10; lapNum++)
{
   // drive the track
}
```

Drive the same track exactly ten times

for loop



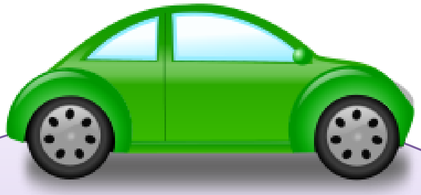
```
// Draw the shapes
for (int shapeNum=0; shapeNum < shapeList.size(); shapeNum++)
{
  shapeList.get(shapeNum).draw();
}
```

while loop

Drive the track while the car still has gas

GO CODE GIRL
IMAGINE. DESIGN. CREATE.

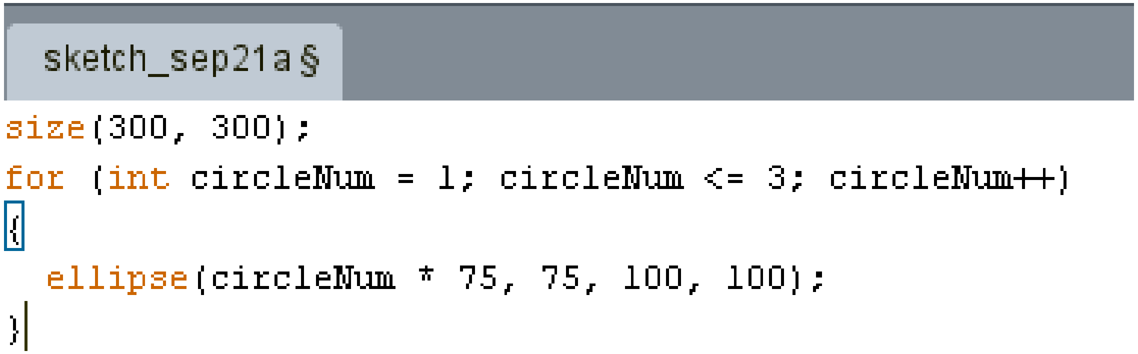
while loop



```
int gas = 100;
while (gas > 0)
{
  gas = gas - 5;
  // drive the track
}
```

Drive the track while the car still has gas

for loop



```
size(300, 300);
for (int circleNum = 1; circleNum <= 3; circleNum++)
{
  ellipse(circleNum * 75, 75, 100, 100);
}
```

What about three rows of circles?

Arrays and Lists

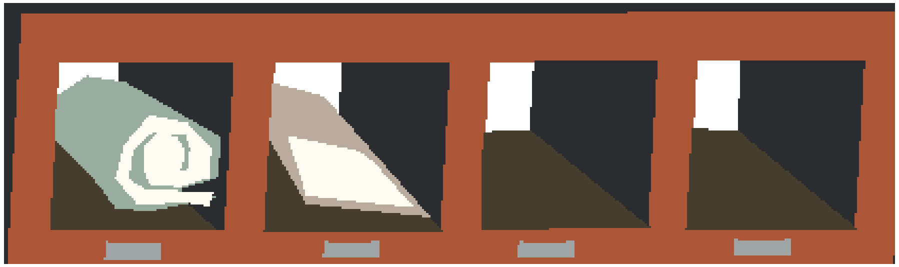
arrays

GO CODE GIRL

IMAGINE. DESIGN. CREATE.

arrays

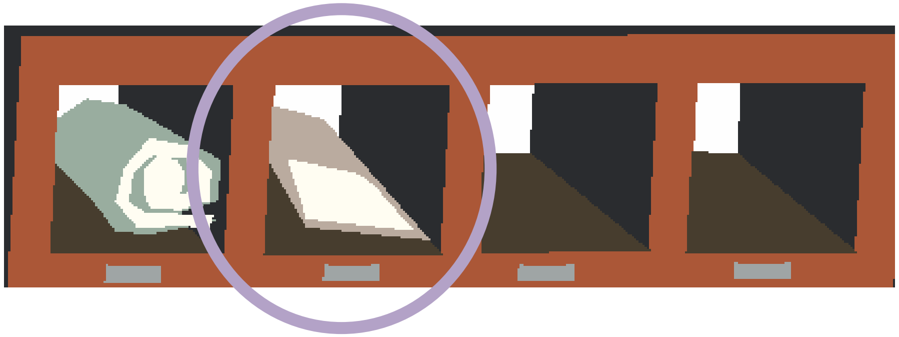
`arrayName`



0 1 2 3

arrays

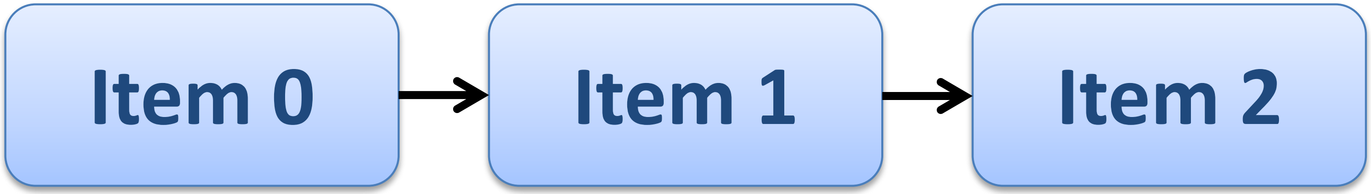
arrayName



0 1 2 3

arrayName[1]

Array List



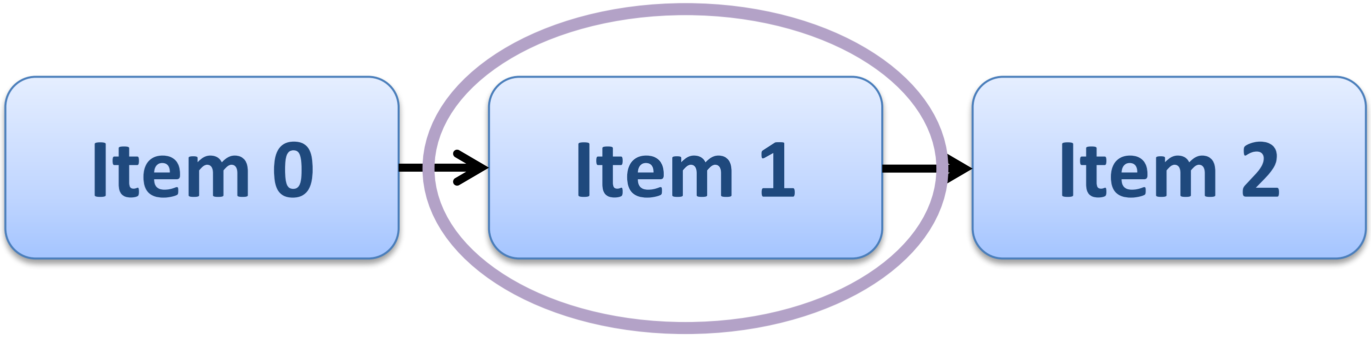
Item 0

Item 1

Item 2

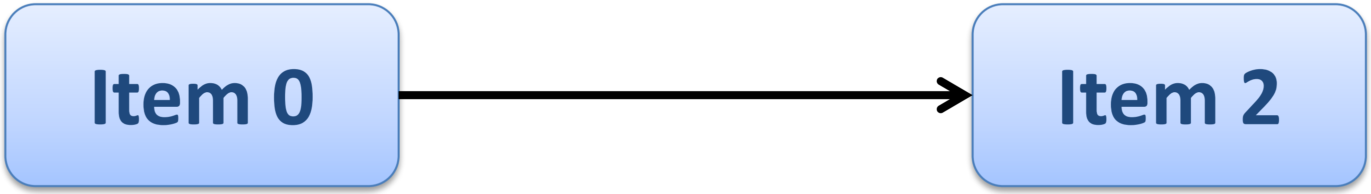
```
ArrayList<Shape> shapeList = new ArrayList<Shape>();
shapeList.add(new Shape()); // item 0
shapeList.add(new Shape()); // item 1
shapeList.add(new Shape()); // item 2
```

Array List



```
shapeList.get(1);
```

Array List

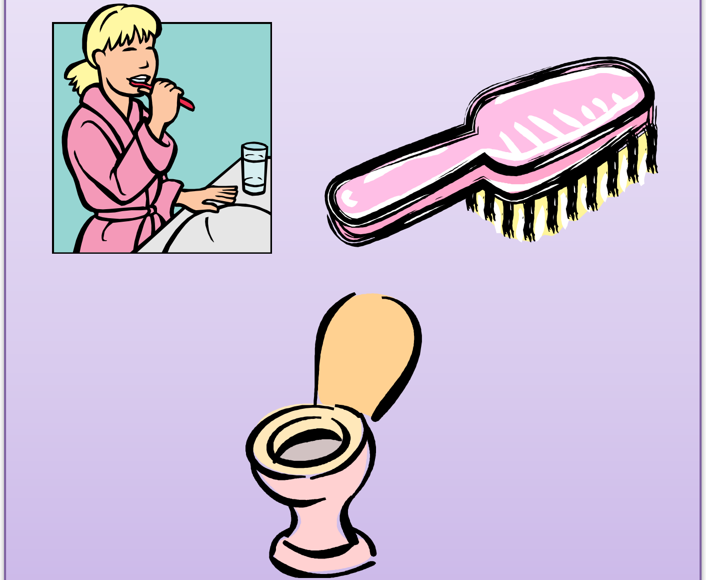


```
shapeList.remove(1);
```

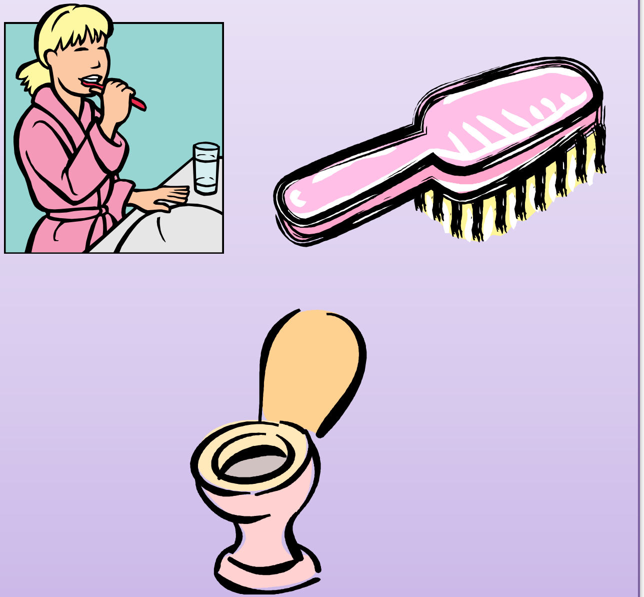
Methods

methods

Morning Routine

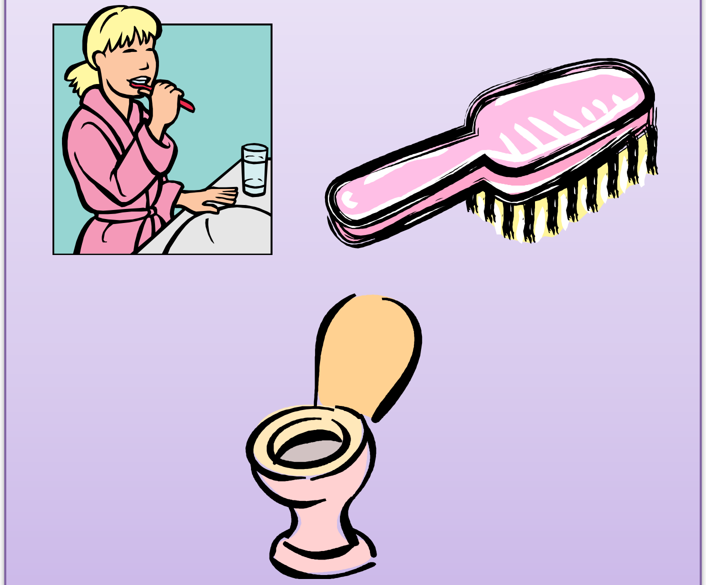


Bedtime Routine

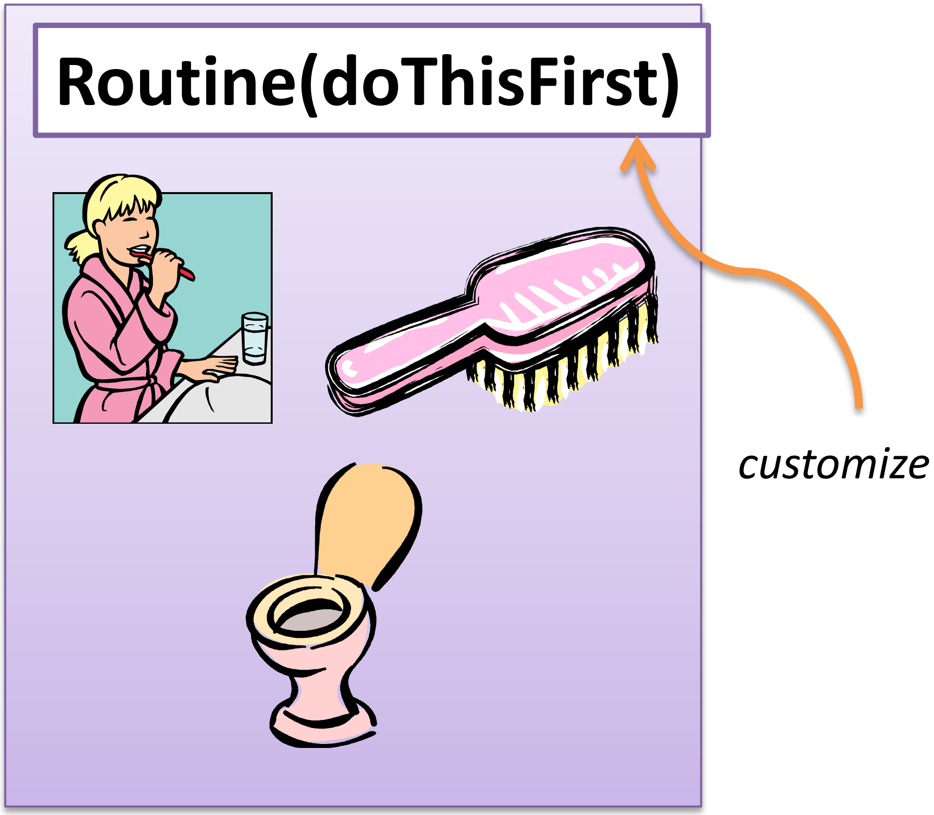


methods

Routine



methods



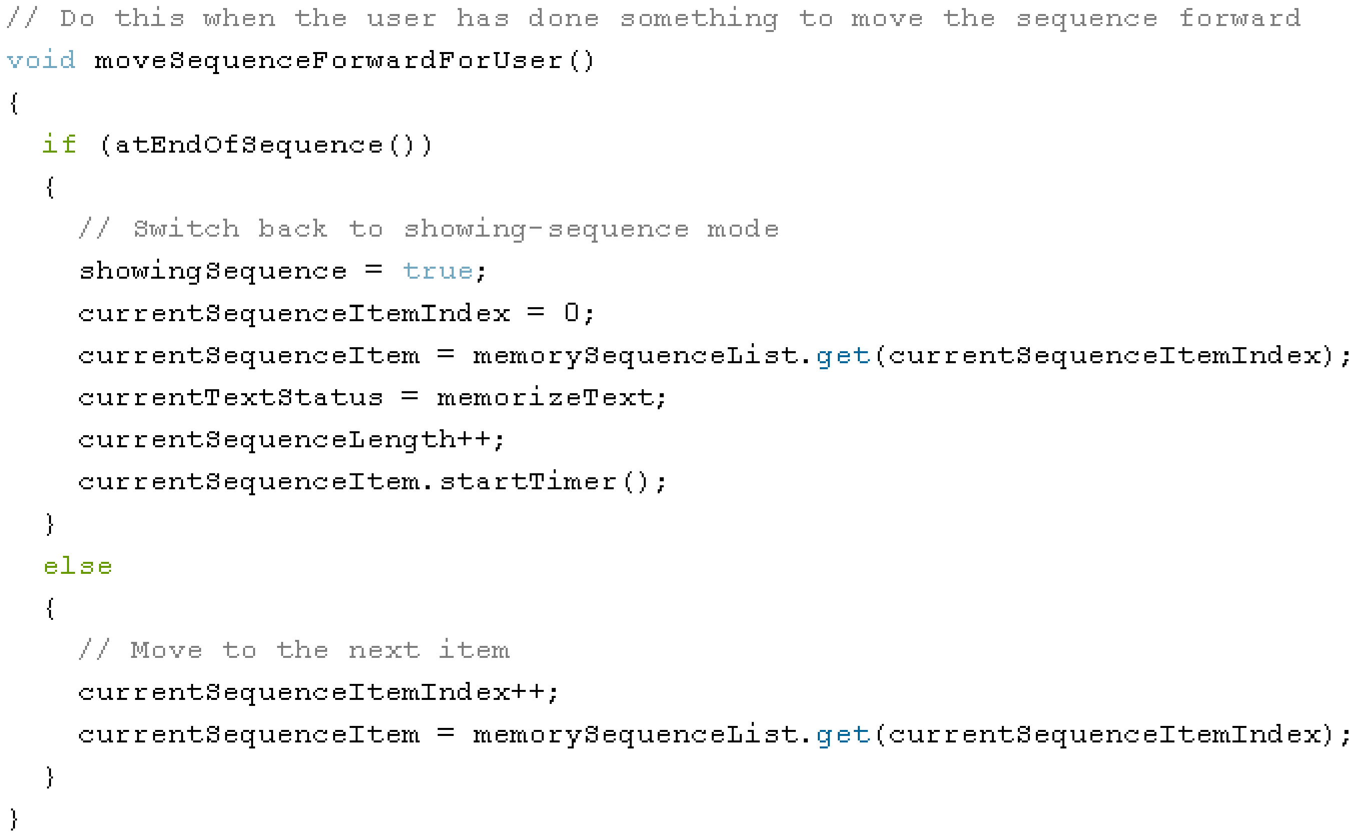
Routine(doThisFirst)

customize

methods

```
void methodName(argumentType argument, …)
{
  // do stuff
}

returnType methodName(argumentType argument, …)
{
  // do stuff
  return <returnType>;
}
```

methods



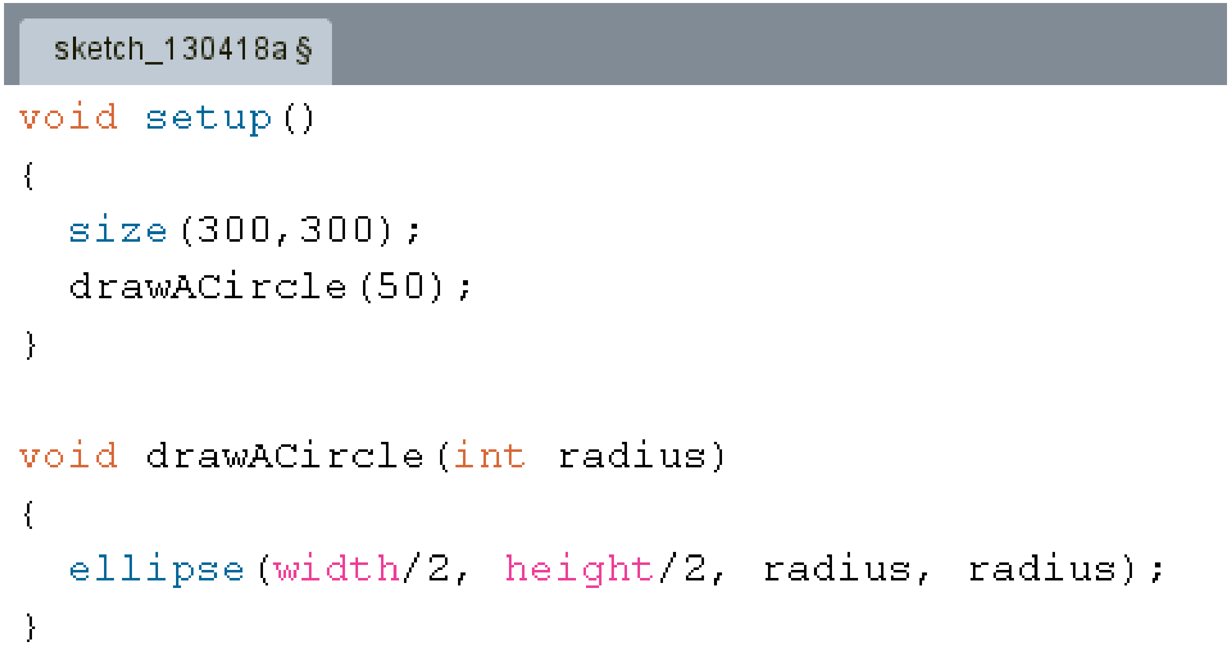
```
// Do this when the user has done something to move the sequence forward
void moveSequenceForwardForUser()
{
  if (atEndOfSequence())
  {
    // Switch back to showing-sequence mode
    showingSequence = true;
    currentSequenceItemIndex = 0;
    currentSequenceItem = memorySequenceList.get(currentSequenceItemIndex);
    currentTextStatus = memorizeText;
    currentSequenceLength++;
    currentSequenceItem.startTimer();
  }
  else
  {
    // Move to the next item
    currentSequenceItemIndex++;
    currentSequenceItem = memorySequenceList.get(currentSequenceItemIndex);
  }
}
```

methods

Special Processing Methods

```
void setup()
void draw()
void mouseClicked()
```

(etc…)



```
void setup()
{
  size(300,300);
  drawACircle(50);
}

void drawACircle(int radius)
{
  ellipse(width/2, height/2, radius, radius);
}
```

Objects

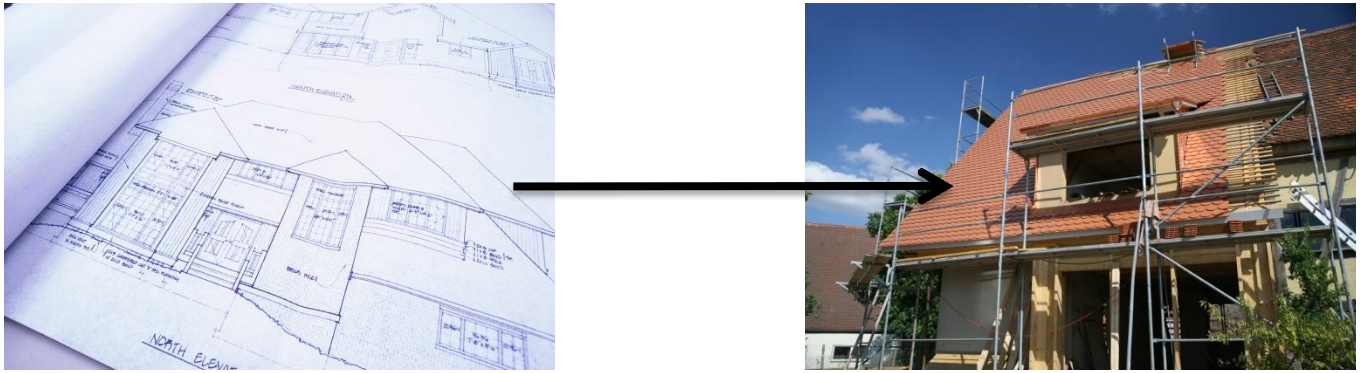
GO CODE GIRL

IMAGINE. DESIGN. CREATE.

Objects

Objects

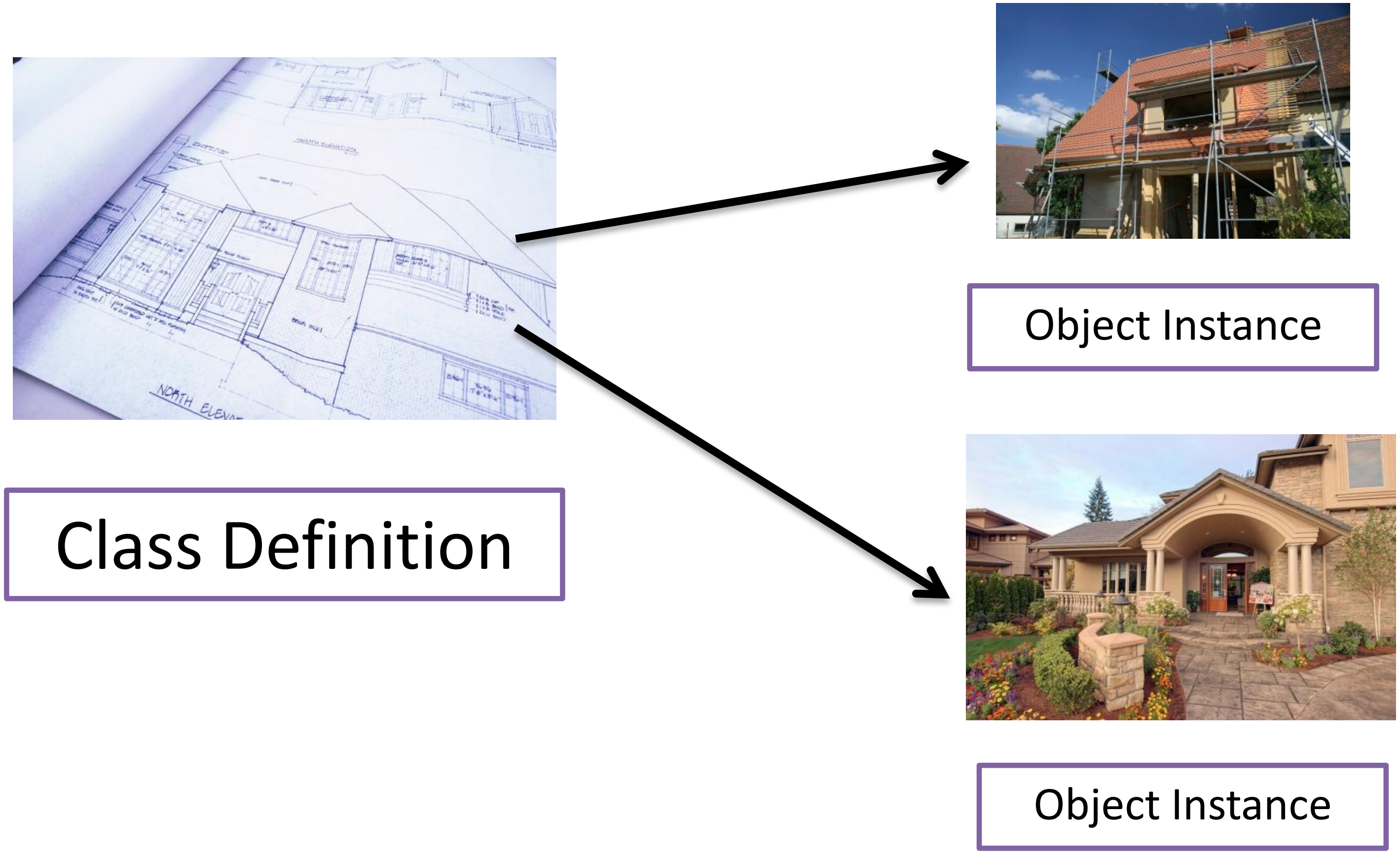
Objects



Class Definition

Object Instance

Objects

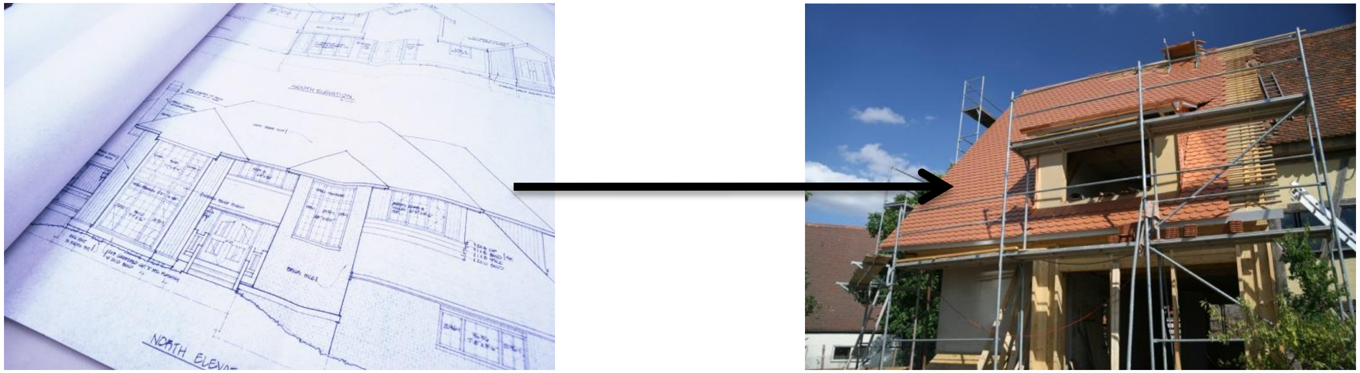


Class Definition

Object Instance

Object Instance

Objects



Class Definition

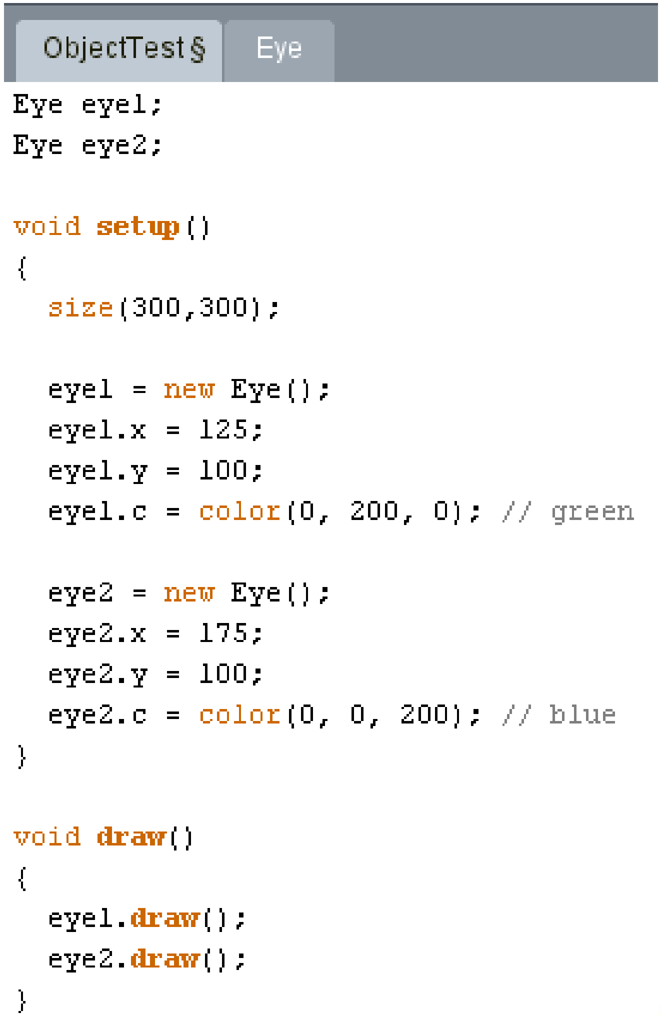
Variables
Methods

Object Instance

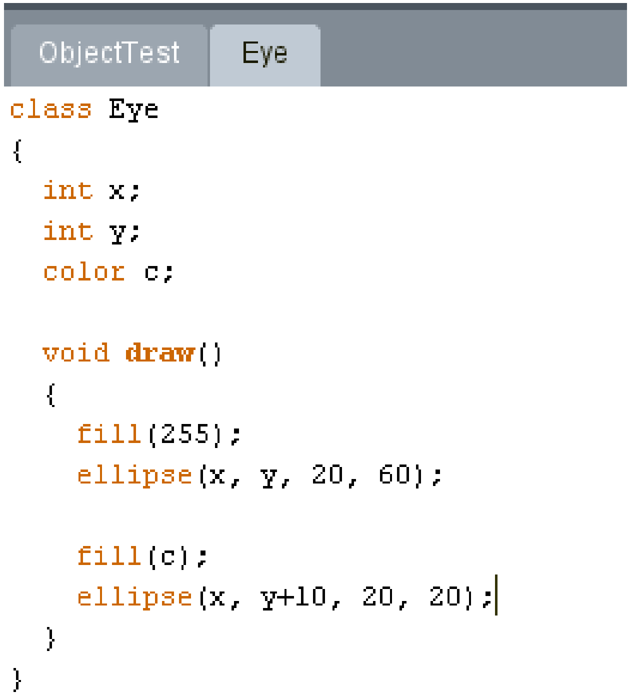
Specific Variables
Specific Methods

```
class Shape
{
 private int x;
 private int y;

 private color fillColor;

 private boolean litUp;
 private boolean pausing;
 private int timerCount;

 private int diameter = 100;

 final int numFramesBetweenShapes = 30 * 1;
 final int numFramesToLightShape = 30 * 1;


 // Constructor
 Shape(int newX, int newY, color c)
 {
   x = newX;
   y = newY;
   fillColor = c;
   litUp = false;
   timerCount = 0;
 }
```

```
shapeList = new ArrayList<Shape>();

// Row 1
shapeList.add(new Shape(100, 100, color(255,139,0)));
shapeList.add(new Shape(250, 100, color(11,73,216)));
shapeList.add(new Shape(400, 100, color(0,214,160)));

// Row 2
shapeList.add(new Shape(100, 250, color(0,132,99)));
shapeList.add(new Shape(250, 250, color(158,86,0)));
shapeList.add(new Shape(400, 250, color(195,212,249)));

// Row 3
shapeList.add(new Shape(100, 400, color(114,155,249)));
shapeList.add(new Shape(250, 400, color(107,249,213)));
shapeList.add(new Shape(400, 400, color(186,124,50)));
```



```
Eye eye1;
Eye eye2;

void setup()
{
  size(300,300);

  eye1 = new Eye();
  eye1.x = 125;
  eye1.y = 100;
  eye1.c = color(0, 200, 0); // green

  eye2 = new Eye();
  eye2.x = 175;
  eye2.y = 100;
  eye2.c = color(0, 0, 200); // blue
}

void draw()
{
  eye1.draw();
  eye2.draw();
}
```



```
class Eye
{
  int x;
  int y;
  color c;

  void draw()
  {
    fill(255);
    ellipse(x, y, 20, 60);

    fill(c);
    ellipse(x, y+10, 20, 20);
  }
}
```