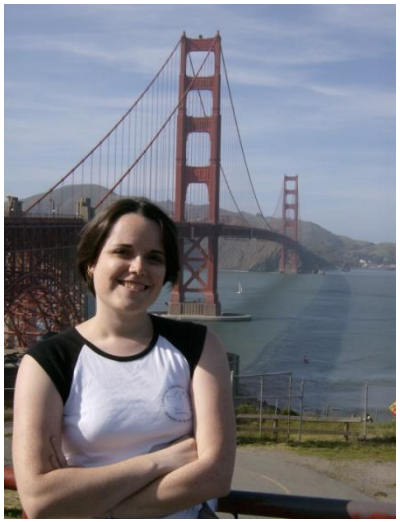
Create Interactive Art and More!

By Gail Carmichael

About Me



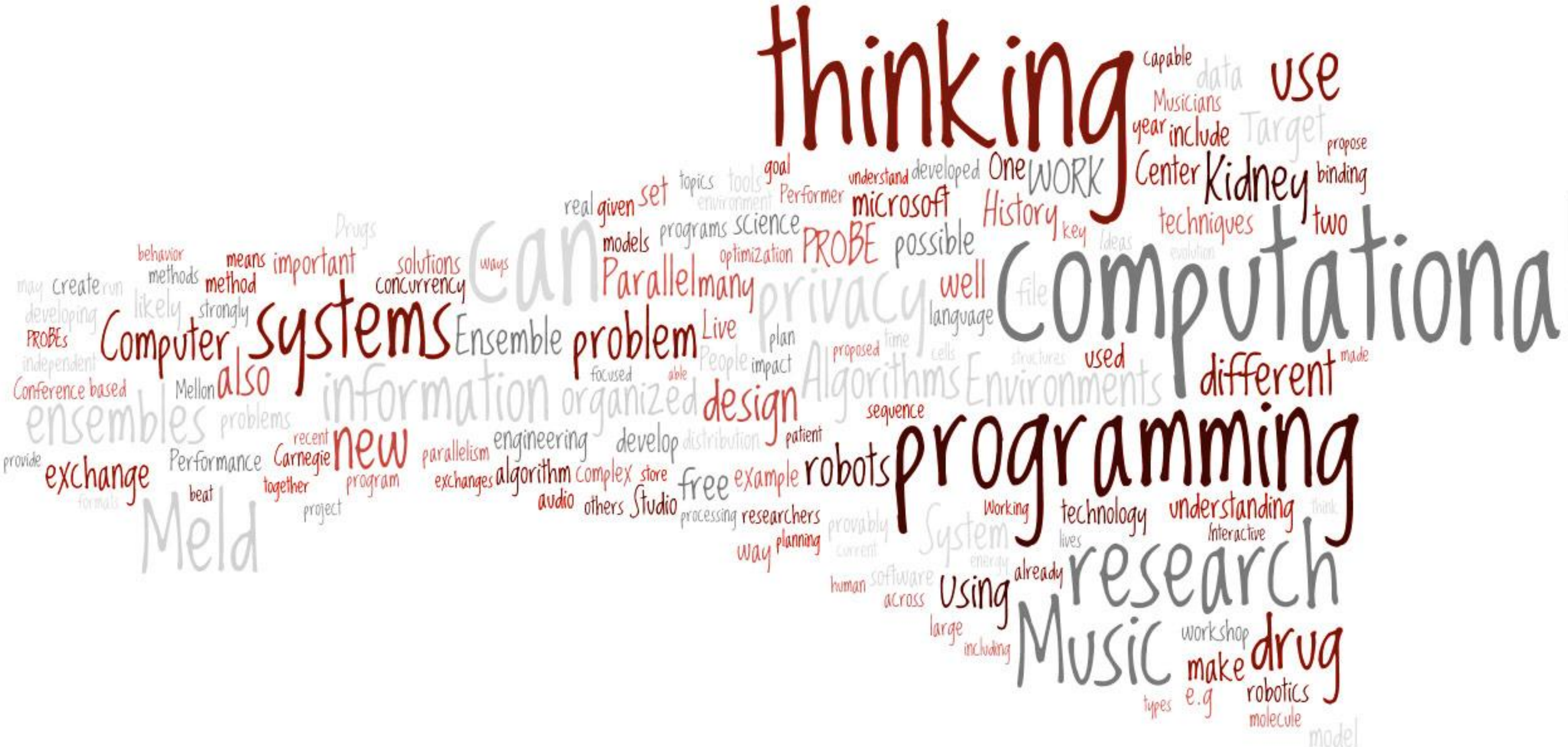
About Me

Bachelor of Computer Science (2002-2007)
Masters of Computer Science (2007-2009)
PhD Computer Science (2009 - now)

www.gailcarmichael.com

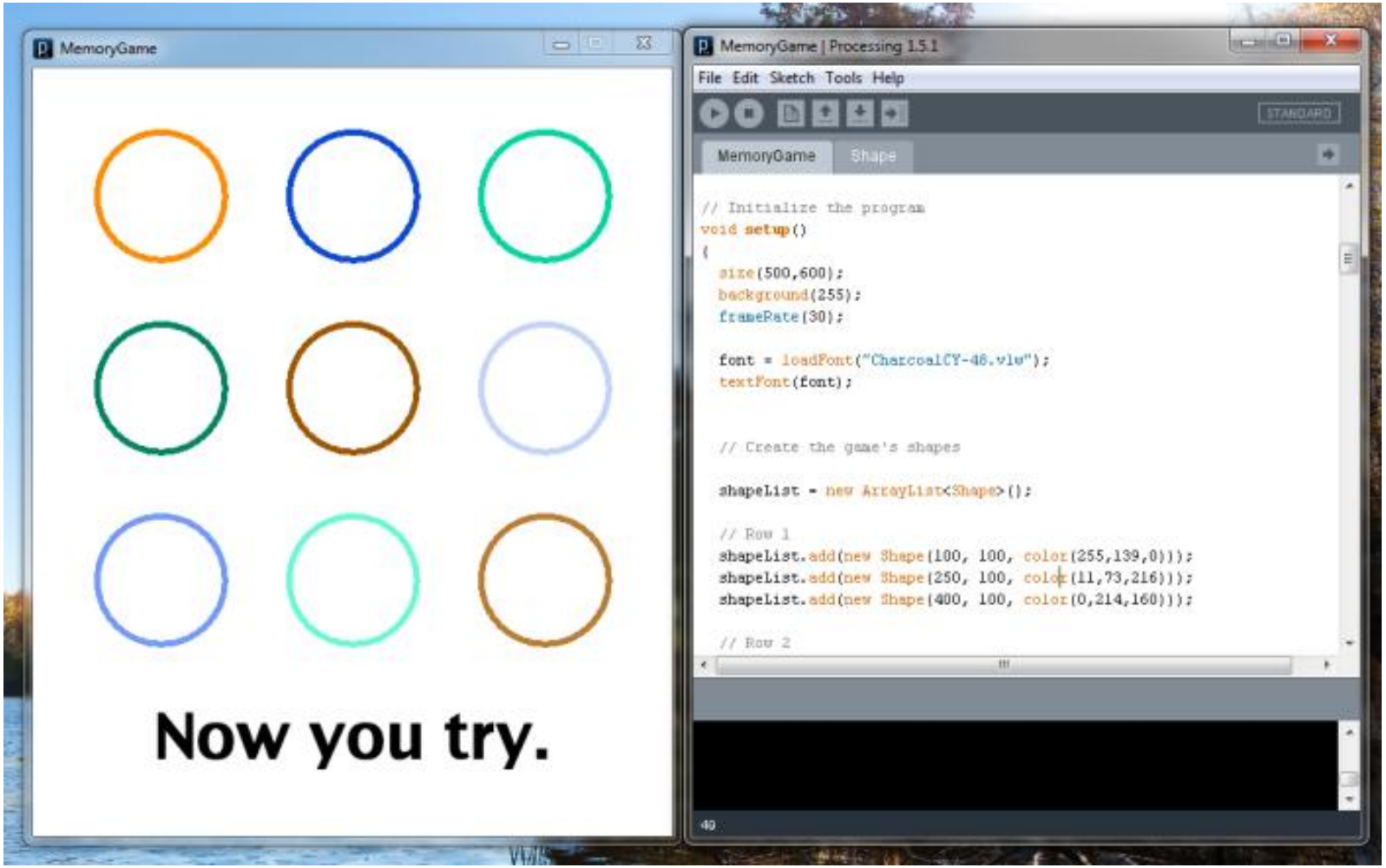
Who Are You?

- Where do you work?
- What do you do?
- What do you love?



thinking
Computational
programming
research
Music
systems
Computer
information
ensembles
Meld
Can
privacy
Algorithms
Environments
problem
design
new
exchange
Ensemble
Parallel
many
organized
robots
Kidney
use
drug
different
Center
History
PROBE
microsoft
WORK
One
well
possible
techniques
two
used
understanding
technology
Using
make
robotics
workshop
already
also
important
solutions
concurrency
method
means
methods
create
strongly
likely
Mellon
Conference
based
PROBEs
Performance
Carnegie
program
together
beat
project
parallelism
engineering
develop
algorithm
exchanges
complex
audio
others
Studio
free
example
store
processing
researchers
way
planning
patient
sequence
proposed
language
developed
understand
goal
set
given
real
models
programs
science
topics
tools
Performer
optimization
Live
People
impact
plan
focused
include
year
Musicians
capable
data
Target
propose
binding
key
file
made
Working
Interactive
System
provably
human
software
across
large
including
types
e.g
molecule
model
provide
recent
problems
behavior

Marching Orders


MemoryGame
Now you try.
MemoryGame | Processing 1.5.1
File Edit Sketch Tools Help
STANDARD
MemoryGame
Shape
// Initialize the program
void setup()
{
size(500,600);
background(255);
frameRate(30);
font = loadFont("CharcoalCY-48.vlw");
textFont(font);
// Create the game's shapes
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
49

Programming Fundamentals

data types

int

a number without decimals

float

a number with decimals

String

textual data

color

red, green, blue

boolean

yes/no

statement

a single instruction

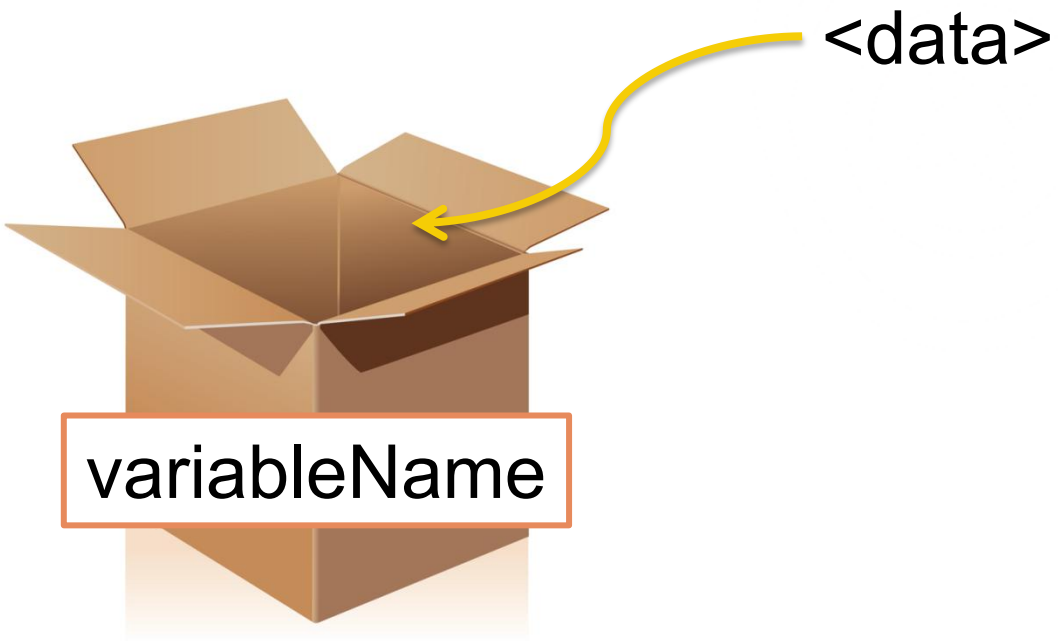
sketch_sep21a §

```
background(255);
size(250, 250);
fill(300, 25, 60);
strokeWeight(5);
ellipse(width/2, height/2, 100, 100);
```

variables

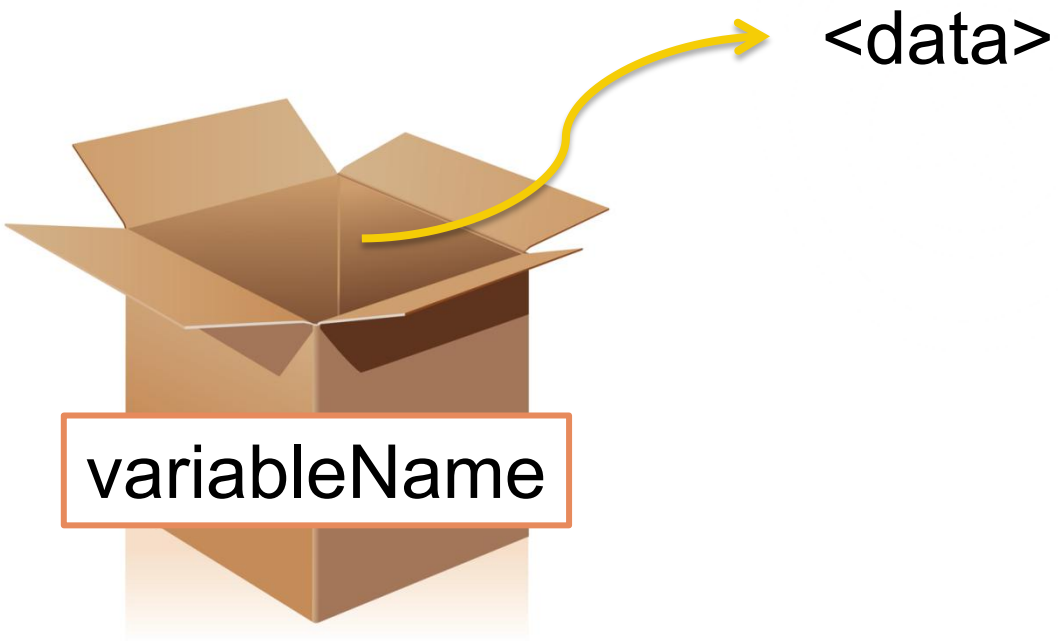

```
int circleRadius;
```

variables




```
circleRadius = 50;
```

variables




```
ellipse(10, 10, circleRadius, circleRadius);
```

variables

sketch_sep21a §

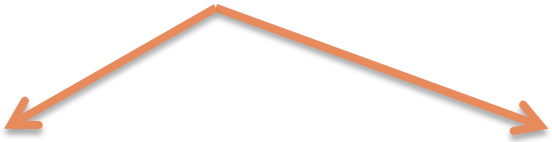
```
size(200, 200);
background(255);
strokeWeight(3);

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);
```

boolean

Yes/ True *or* **No/ False**

if/else statement

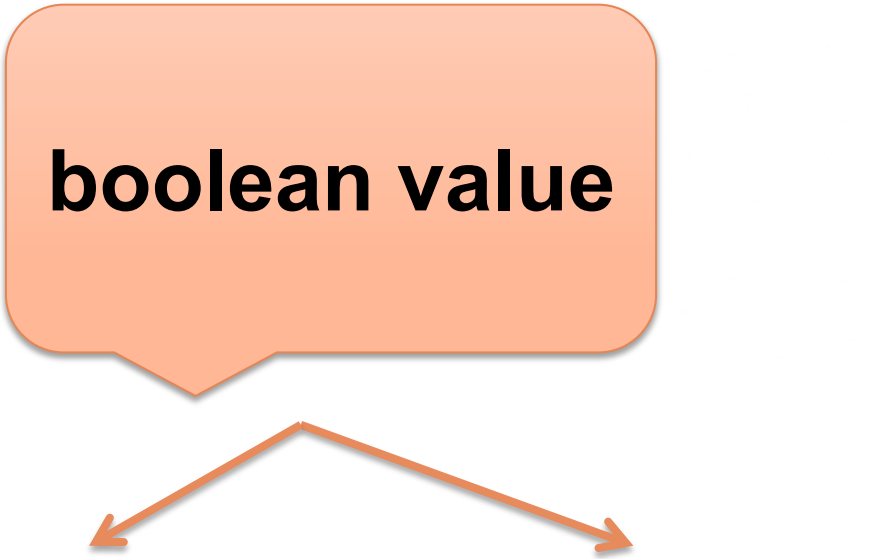
I am sick Friday night



Yes:
Stay home, watch TV

No:
Go out to the Market

if/else statement

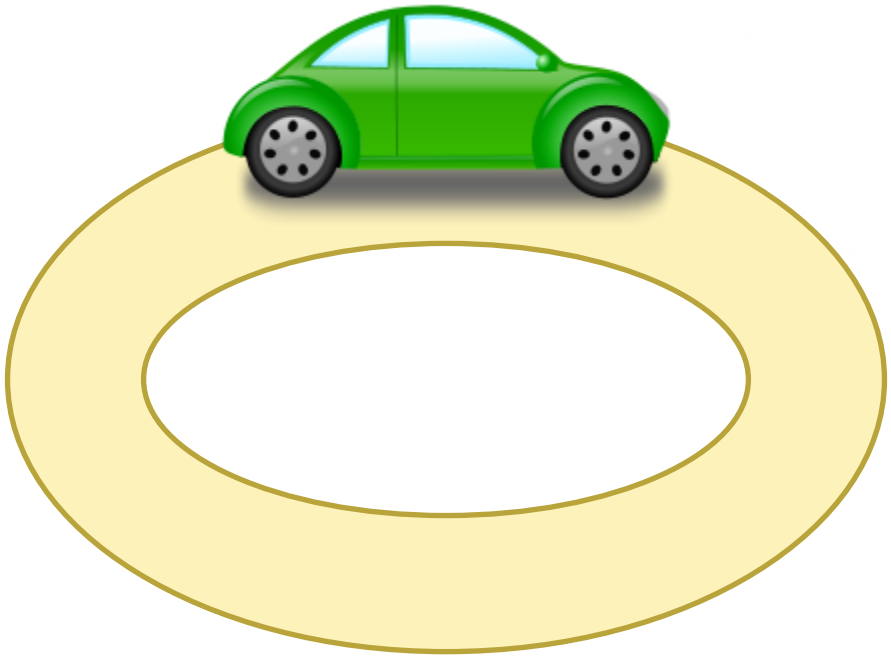


if/else statement

sketch_sep21a §

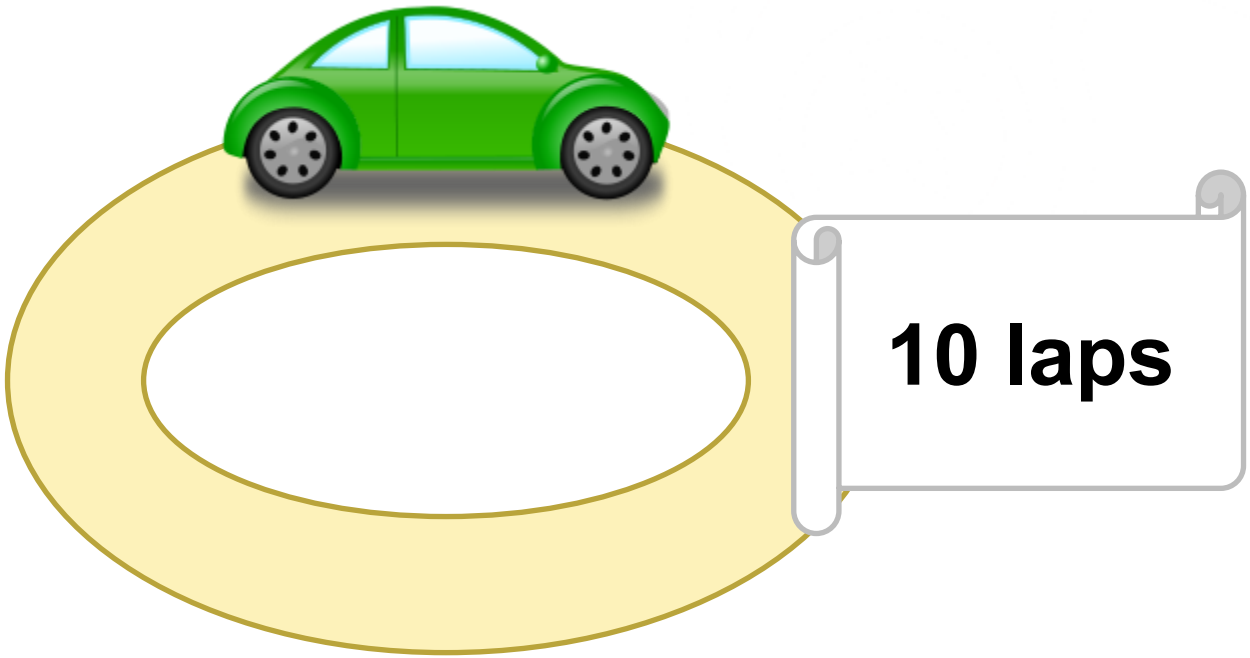
```
int number = 30;

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 track exactly ten times

for loop

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

Drive the track exactly ten times

for loop

sketch_sep21a §

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

What about three rows of circles?

for loop

sketch_sep21a §

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

while loop

Drive the track while the car still has gas

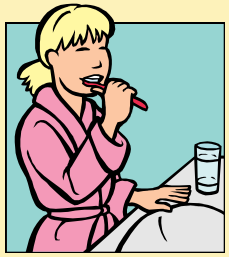
while loop

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

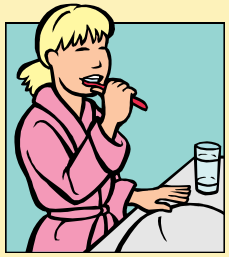
Drive the track while the car still has gas

methods

Morning Routine

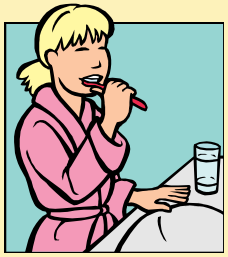


Bedtime Routine



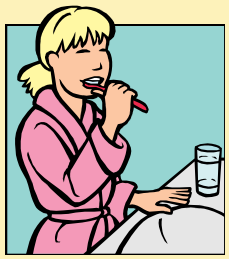
methods

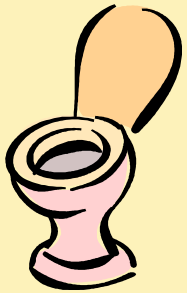
Routine



methods

Routine(doThisFirst)





customize

methods

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

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

methods

sketch_sep21a §

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

void drawACircle(int radius)
{
  ellipse(width/2, height/2, 100, 100);
}
```

methods

Special Processing Methods

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

(etc…)

objects

objects

objects

Class Definition

Object Instance

objects

Class Definition

Object Instance

Object Instance

objects

Class Definition	Object Instance
Variables Methods	Variables Methods

objects

ObjectTest§ | Eye

```
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();
}
```

ObjectTest | Eye

```
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);
  }
}
```

arrays

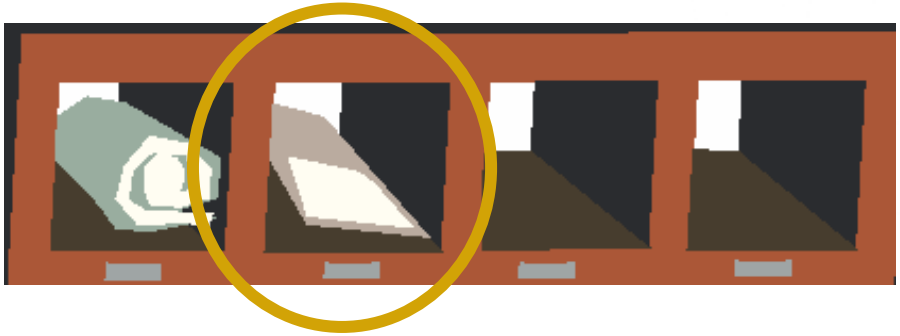
arrays

arrayName

0 1 2 3

arrays

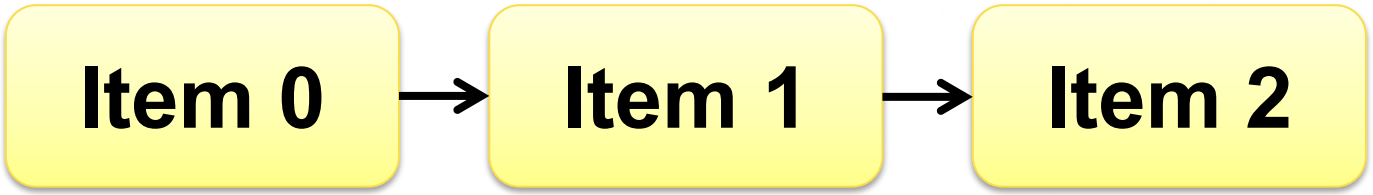
arrayName



0 1 2 3

arrayName[1]

Array Lists




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

Array Lists


```
eyeList.get(1);
```

Array Lists


```
eyeList.remove(1);
```

Other Stuff

Comments
Coordinates
Animation / frames