Did you do this?

- Go the CS Computer Lab (Room 231 PSB)
- Log in
- Start the Processing application (Make sure it is Version 2.x)
- In a web browser, go to the Tutorials section of processing.org http://www.processing.org/tutorials/gettingstarted/
- Read the Getting Started tutorial (by Casey Reas & Ben Fry) and try out the two examples of simple Processing programs presented there
- If you’d like, install Processing 2.x on your own computer
- Read Chapter 1 (Read pages 1-12, skim 12-32)

Drawing Basics

- Canvas — computer screen
  `size(width, height);`
- Drawing Tools — shape commands
- Colors — grayscale or RGB
  `background(125);`

Drawing Tools - Basic Shapes

- Point
  `point(x, y);`
- Line
  `line(x1, y1, x2, y2);`
- Triangle
  `triangle(x1, y1, x2, y2, x3, y3);`
- Rectangle
  `rect(x, y, width, height);`
- Ellipse
  `ellipse(x, y, width, height);`
- Arc
  `arc();`
- Quad
  `quad();`
- Triangle
  `triangle();`
- Polygon
  `polygon();`
- Curve
  `curve();`

Drawing & Shape Attributes

- Anti-aliasing
  - smooth();
  - noSmooth();
- Stroke
  - noStroke();
  - strokeWidth(<pixel width>);
  - stroke(<stroke color>);
- Fill
  - noFill();
  - fill(<fill color>);
Drawing Tools - Basic Shapes

- Point
- Line
- Triangle
- Rectangle
- Ellipse

Basic Shapes: Arcs

- What is an arc?

Basic Shapes: Arcs

```javascript
arc(x, y, width, height, startAngle, endAngle);
```

Basic Shapes: Arcs

```javascript
noFill();
stroke(255, 0, 0);
arc(200, 200, 150, 150, 0, TWO_PI);
```

Basic Shapes: Arcs

```javascript
fill(255, 0, 0);
stroke(255, 0, 0);
arc(200, 200, 150, 150, 0, PI);
```

Basic Shapes: Arcs

```javascript
noStroke();
fill(12, 37, 80);
quad(100, 50, 150, 100, 100, 150, 50, 100);
```

Basic Shapes: Arcs

```javascript
fill(163, 208, 193);
quad(100, 50, 150, 100, 100, 150, 250, 100);
```

Basic Shapes: Arcs

```javascript
fill(240, 127, 71);
quad(100, 50, 200, 50, 250, 100, 50, 100);
```
Basic Shapes: Polygons

```pseudocode
beginShape();
vertex(x1, y1);
vertex(xN, yN);
endShape(CLOSE);
```

More Complex Curves

```pseudocode
beginShape();
curveVertex(x1, y1);
curveVertex(xN, yN);
endShape(CLOSE);
```

Example: A Penguin

```pseudocode
beginShape();
curveVertex(x1, y1);
curveVertex(xN, yN);
endShape();
```

Review: Drawing Basics

- **Canvas**
  ```
  size(width, height);
  ```

- **Drawing Tools**
  ```
  point(x, y);
  line(x1, y1, x2, y2);
  ```

- **Colors**
  ```
  color(red, blue, green);
  ```

Simple Program Structure

```pseudocode
// Create and set canvas
size(width, height);
smooth();
background(color);

// Draw something
...

// Draw something else
...

// etc.
```
Simple Program Structure

// Draw a simple house
// Create and set canvas
size(300, 300);
smooth();
background(187, 193, 127);
// wall
fill(206, 224, 14);
rect(50, 150, 200, 100);
// Draw Door
fill(72, 26, 2);
rect(125, 200, 50, 50);
// Draw roof
fill(224, 14, 14);
triangle(50, 150, 150, 50, 250, 150);

Variables: Naming Values

- **Values**
  - Values: 42, 3.14159, 2013, "Hi, my name is Joe!", true, false, etc.
  - Numbers
    - Integers
      ```
      int meaningOfLife = 42;
      int year = 2013;
      ```
    - Floating point numbers
      ```
      float pi = 3.14159;
      ```
  - Strings
    ```
    String greeting = "Hi, my name is Joe!";
    ```
  - Boolean
    ```
    boolean keyPressed = true;
    ```

Variables: Naming Rules & Conventions

- Names begin with a letter, an underscore (_), or a dollar sign ($)
  - Examples: `weight`, `_meaningOfLife`, `$value`
- Names may include numbers, but only after the initial character
  - Examples: `value1`, `score5`, `5bestFriends`
- No spaces are permitted in names
  - Examples: `value1`, `dollar_sign`
- Processing Conventions
  - Names begin with a lowercase letter
    - Example: `meaningOfLife`, `highestScore`
  - Constants are written in all caps
    - Example: `DAYS_IN_WEEK`, `PI`
The color type

- Processing has a type called color

```javascript
color firebrick = color(178, 34, 34);
color chartreuse = color(127, 255, 0);
color fuchsia = color(255, 0, 255);
```

```javascript
fill(firebrick);
rect(50, 100, 75, 125);
```

Expressions: Doing Arithmetic

- Assignment statement
  - `<variable> = <expression>`

```javascript
meaningOfLife = 42;
area = length * height;
perc = statePop / totalPop * 100.0;
```

- Operators
  - `+` (addition)
  - `-` (subtraction)
  - `*` (multiplication)
  - `/` (division)
  - `%` (modulus)

```javascript
mouth_x = ((leftIris_x + irisDiam) / 2 + eyeWidth) / 4;
```

Using Variables

```javascript
// Draw a simple house
// Create and set canvas
size(300, 300);
smooth();
background(187, 193, 127);
// wall
fill(206, 224, 14);
rect(houseX, houseY - wallHeight, houseWidth, wallHeight);
// Draw Door
fill(72, 26, 2);
rect(houseX + houseWidth / 2 - doorWidth / 2, houseY - doorHeight, doorWidth, doorHeight);
// Draw roof
fill(224, 14, 14);
triangle(houseX, houseY - wallHeight, houseX + houseWidth / 2, houseY - houseHeight, houseX + houseWidth, houseY - wallHeight);
```

A Better House Sketch

```javascript
// Draw a simple house
// Create and set canvas
size(300, 300);
smooth();
background(187, 193, 127);
// wall
fill(206, 224, 14);
rect(houseX, houseY - wallHeight, houseWidth, wallHeight);
// Draw Door
fill(72, 26, 2);
rect(houseX + houseWidth / 2 - doorWidth / 2, houseY - doorHeight, doorWidth, doorHeight);
// Draw roof
fill(224, 14, 14);
triangle(houseX, houseY - wallHeight, houseX + houseWidth / 2, houseY - houseHeight, houseX + houseWidth, houseY - wallHeight);
```

Arithmetic with int and float values

```javascript
int x = 42;
float x = 42.0;
```

```javascript
x = 7 / 2.0;
```
Arithmetic with int and float values

```
int x = 42;  // ok
float x = 42.0;  // same
float x = 7/2;  // error
float x = 7.0/2.0;  // 3.5
```

- Type of variable is important and determines the value that can be assigned to it.
- Result of division depends upon operands

```
type operation yields example
int/int  yields an integer result
float/int  yields a float result
int/float  yields a float result
float/float  yields a float result
```

Processing: Predefined Variables

- `width, height`
The width & height of the canvas used in the sketch
- `PI, HALF_PI, TWO_PI`
  For different values of \( \pi \). Note that

\[
\text{HALF_PI} = \pi / 2
\]

\[
\text{TWO_PI} = 2 \pi
\]

- `displayWidth, displayHeight`
The width and height of the monitor being used. This is useful in running full screen sketches using:

```
size(displayWidth, displayHeight);
```

- `mouseX, mouseY`
The current mouse location in sketch (...coming soon!)

Homework

- Read Chapter 2
- Read and do the Coordinate Systems & Shapes and Color tutorials on processing.org
- Review Processing commands:

```
size(), background(), 2D shapes: point(), line(), triangle(), rectangle(), quad(), ellipse(). Attributes and modes: stroke(), noStroke(), strokeWeight(), fill(), noFill(), rectMode(), ellipseMode().
```

- Color values (grayscale and RGB) and transparency.
- Understand the concept of an algorithm, pseudocode, syntax, and sequencing

**Extra: Drawing Text**

```
text(string, x, y);  
Draws string with bottom left corner at x, y

textSize(fontSize);
Can be used to specify font size

fill(); can be used to specify color
See Reference for using fonts and other options.
```