Module 03

Input / Output
We can write more interesting programs when we can exchange information with the outside world.
Problem with the outside world: there’s a lot of it.
<table>
<thead>
<tr>
<th>Disk Inventory X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk Inventory X</td>
</tr>
<tr>
<td>Size</td>
</tr>
<tr>
<td>------------------</td>
</tr>
<tr>
<td>File</td>
</tr>
<tr>
<td>98.2 GB</td>
</tr>
<tr>
<td>54.7 GB</td>
</tr>
<tr>
<td>19.0 GB</td>
</tr>
<tr>
<td>17.7 GB</td>
</tr>
<tr>
<td>11.3 GB</td>
</tr>
<tr>
<td>10.4 GB</td>
</tr>
<tr>
<td>9.8 GB</td>
</tr>
<tr>
<td>7.9 GB</td>
</tr>
<tr>
<td>5.4 GB</td>
</tr>
<tr>
<td>3.5 GB</td>
</tr>
<tr>
<td>3.5 GB</td>
</tr>
<tr>
<td>2.9 GB</td>
</tr>
<tr>
<td>2.4 GB</td>
</tr>
<tr>
<td>10,032 MB</td>
</tr>
<tr>
<td>9,637 MB</td>
</tr>
<tr>
<td>4,004 MB</td>
</tr>
<tr>
<td>1,246 MB</td>
</tr>
<tr>
<td>954 MB</td>
</tr>
<tr>
<td>45.1 MB</td>
</tr>
<tr>
<td>23.5 MB</td>
</tr>
<tr>
<td>11.8 MB</td>
</tr>
<tr>
<td>8.0 MB</td>
</tr>
<tr>
<td>7.9 MB</td>
</tr>
<tr>
<td>6.7 MB</td>
</tr>
<tr>
<td>5.9 MB</td>
</tr>
<tr>
<td>5.3 MB</td>
</tr>
<tr>
<td>1.7 MB</td>
</tr>
<tr>
<td>1.3 MB</td>
</tr>
<tr>
<td>1.2 MB</td>
</tr>
<tr>
<td>1.1 MB</td>
</tr>
<tr>
<td>879 KB</td>
</tr>
<tr>
<td>694 KB</td>
</tr>
<tr>
<td>514 KB</td>
</tr>
<tr>
<td>493 KB</td>
</tr>
</tbody>
</table>
Use the Sketch Folder as a gateway to the outside world.
Use Sketch → Add File... to make a file available to your sketch, or drop the file into the sketch folder directly.

Any files created by the sketch will be left in the sketch folder.
1. Reading and writing images
A built-in function that takes the name of a file as a String parameter, finds that file in your sketch folder, and tries to import it as an image. Returns an object of type PImage.
High-level PImage operations

PImage img;

void setup()
{
  size( 800, 800 );
  img = loadImage( "some_image.jpg" );
}

void draw()
{
  background( 255 );

  imageMode( CORNER );
  noTint();
  image( img, 0, 0 );
  image( img, width - img.width, height - img.height );

  tint( 255, 120, 120 );
  imageMode( CENTER );
  image( img, width/2, height/2, 250, 250 );
}
High-level PImage operations

PImage img;

void setup()
{
    size( 800, 800 );
    img = loadImage( "some_image.jpg" );
}

void draw()
{
    background( 255 );

    imageMode( CORNER );
    noTint();
    image( img, 0, 0 );
    image( img, width - img.width, height - img.height );

    tint( 255, 120, 120 );
    imageMode( CENTER );
    image( img, width/2, height/2, 250, 250 );
}
High-level PImage operations

PImage img;

void setup()
{
  size( 800, 800 );
  img = loadImage( “some_image.jpg” );
}

void draw()
{
  background( 255 );
  imageMode( CORNER );
  noTint();
  image( img, 0, 0 );
  image( img, width - img.width, height - img.height );
  tint( 255, 120, 120 );
  imageMode( CENTER );
  image( img, width/2, height/2, 250, 250 );
}

Draw the image at the given coordinates, at its natural size.
High-level PImage operations

PImage img;

void setup()
{
  size( 800, 800 );
  img = loadImage( "some_image.jpg" );
}

void draw()
{
  background( 255 );

  imageMode( CORNER );
  noTint();
  image( img, 0, 0 );
  image( img, width - img.width, height - img.height );

  tint( 255, 120, 120 );
  imageMode( CENTER );
  image( img, width/2, height/2, 250, 250 );
}

Draw the image at the given coordinates, scaled.
High-level PImage operations

PImage img;

void setup()
{
    size( 800, 800 );
    img = loadImage( "some_image.jpg" );
}

void draw()
{
    background( 255 );
    imageMode( CORNER );
    noTint();
    image( img, 0, 0 );
    image( img, width - img.width, height - img.height );
    tint( 255, 120, 120 );
    imageMode( CENTER );
    image( img, width/2, height/2, 250, 250 );
}
High-level PImage operations

PImage img;

void setup()
{
  size(800, 800);
  img = loadImage(“some_image.jpg”);
}

void draw()
{
  background(255);

  imageMode(CORNER);
  noTint();
  image(img, 0, 0);
  image(img, width - img.width, height - img.height);

  tint(255, 120, 120);
  imageMode(CENTER);
  image(img, width/2, height/2, 250, 250);
}
High-level PImage operations

PImage img;

void setup()
{
  size( 800, 800 );
  img = loadImage( "some_image.jpg" );
}

void draw()
{
  background( 255 );
  imageMode( CORNER );
  noTint();
  image( img, 0, 0 );
  image( img, width - img.width, height - img.height );
  tint( 255, 120, 120 );
  imageMode( CENTER );
  image( img, width/2, height/2, 250, 250 );
}

Change the anchor point of the image.
Image no-nos

```java
PImage img = loadImage( "some_image.jpg" );
```

Don’t try to load the image in the global variable declaration. This will usually fail.
void draw()
{
    PImage img = loadImage( "some_image.jpg" );
    image( img, 0, 0 );
}

Don’t load images in `draw()`. This won’t break the program, but it will work much harder than necessary. Load the image once in `setup()`.
Standard image idiom

PImage img;

void setup()
{
    img = loadImage("some_image.jpg");
}

void draw()
{
    image( img, 0, 0 );
    image( img, width - img.width, height - img.height );
    image( img, width/2, height/2, 250, 250 );
}
Standard image idiom

PImage img;  // Global variable to hold image.

void setup()
{
  img = loadImage("some_image.jpg");
}

void draw()
{
  image(img, 0, 0);
  image(img, width - img.width, height - img.height);
  image(img, width/2, height/2, 250, 250);
}
Standard image idiom

PImage img;  // Global variable to hold image.

void setup()
{
  img = loadImage( "some_image.jpg" );
}

Load image in setup().

void draw()
{
  image( img, 0, 0);
  image( img, width - img.width, height - img.height );
  image( img, width/2, height/2, 250, 250 );
}
Standard image idiom

```
PIimage img;  // Global variable to hold image.

void setup()
{
    img = loadImage("some_image.jpg");
}

void draw()
{
    // Load image in setup().
    image(img, 0, 0);
    image(img, width - img.width, height - img.height);
    image(img, width/2, height/2, 250, 250);
}
```
Writing images

Several ways to do this. Easiest is to take a screenshot.

```java
void save(String filename) { ... }
```

Save the contents of the sketch window to an image with the given file name.

```java
void saveFrame() { ... }
void saveFrame(String name_template) { ... }
```

Same as above, but include a counter in the saved file name. Useful for animations.
void keyPressed()
{
    if (key == 's') {
        save("screen.png");
    }
}
2. Reading and writing illustrations
Raster image: represented using a grid of pixels.

Vector illustration: represented using geometric paths.
Images

loadImage()  
PImage
image()

Illustrations

loadShape()  
PShape
shape()
PShape tiger;

void setup()
{
    size( 500, 500 );
    tiger = loadShape( "tiger.svg" );
}

void draw()
{
    shape( tiger, 0, 0 );
}
The **PShape** class has a `disableStyle()` method that forces the SVG to be drawn with the current fill and stroke settings.

```java
void draw()
{
    background( 255 );
    if( keyPressed ) {
        tiger.disableStyle();
        fill( 255, 0, 0 );
        noStroke();
    } else {
        tiger.enableStyle();
    }
    shape( tiger, 0, 0 );
}
```
Writing illustrations

Processing can export any drawing to PDF or SVG (PDF is nicer). But the functionality isn’t built-in—you need to request it.

```java
import processing.pdf.*;
```

“Import directive”: make all the functionality in the named library available in this sketch.
Use `beginRecord()` and `endRecord()` to copy all drawing commands into an external file.

```java
import processing.pdf.*;

void setup()
{
    beginRecord( PDF, "output.pdf" );
    // Draw something here
    endRecord();
}
```

Image: alexpoole.info
boolean recording = false;

void draw()
{
    if (recording) {
        beginRecord(PDF, "output.pdf");
    }

    // Draw as usual

    if (recording) {
        endRecord();
        recording = false;
    }
}

void keyPressed()
{
    if (key == 's') {
        recording = true;
    }
}
3. Reading and writing text
Marley was dead: to begin with. There is no doubt whatever about that. The register of his burial was signed by the clergyman, the clerk, the undertaker, and the chief mourner. Scrooge signed it: and Scrooge's name was good upon 'Change, for anything he chose to put his hand to. Old Marley was as dead as a door-nail.

Mind! I don't mean to say that I know, of my own knowledge, what there is particularly dead about a door-nail. I might have been inclined, myself, to regard a coffin-nail as the deadest piece of ironmong ery in the trade. But the wisdom of our ancestors is in the simile; and my unhallowed hands shall not disturb it, or the Country's done for. You will therefore permit me to repeat, emphatically, that Marley was as dead as a door-nail.

Scrooge knew he was dead? Of course he did. How could it be otherwise? Scrooge and he were partners for I don't know how many years. Scrooge was his sole executor, his sole administrator, his sole assign, his sole residuary legatee, his sole friend and sole mourner. And even Scrooge was not so dreadfully cut up by the sad event, but that he was an excellent man of business on the very day of the funeral, and solemnised it with an undoubted bargain.

The mention of Marley's funeral brings me back to the point I started from. There is no doubt that Marley was dead. This must be distinctly understood, or nothing wonderful can come of the story I am going to relate. If we were not perfectly convinced that Hamlet's Father died before the play began, there would be nothing more remarkable in his taking a stroll at night, in an easterly wind, upon his own ramparts, than there would be in any other middle-aged gentleman rashly turning out after dark in a breezy spot--say Saint Paul's Churchyard for instance--literally to astonish his son's weak mind.

Scrooge never painted out Old Marley's name. There it stood, years afterwards, above the warehouse door: Scrooge and Marley. The firm was known as Scrooge and Marley. Sometimes people new to the business called Scrooge Scrooge, and sometimes Marley, but he answered to both names: it was all the same to him.

Oh! But he was a tight-fisted hand at the grind-stone, Scrooge! a squeezing, wrenching, grasping, clutching, covetous, old sinner! Hard and sharp as flint, from which no steel had ever struck out generous fire; secret, and self-contained, and solitary as an oyster. The cold within him froze his old features, nipped his pointed nose, shrivelled his cheek, stiffened his gait; made his eyes red, his thin lips blue and spoke out shrewdly in his grating voice. A frosty rime was on his head, and on his eyebrows, and his wiry chin. He carried his own low temperature always about with him; he iced his office in the dogdays; and didn't thaw it one degree at Christmas.

External heat and cold had little influence on Scrooge. No warmth could warm, no wintry weather chill him. No wind that blew was bitterer than he, no falling snow was more intent upon its purpose; no drizzle poured with less intent upon its destination than that which flew off him in every direction like shot. Though he lived alone, he had a great many people about him. He had two or three house servants, and a great many more of the village kind, to which ranks he gave a collective name of his own invention, and never could bring himself to recognize in full, though he heard it, as he heard other people's names, every day of his life.

Nobody ever stopped him in the street to say, with gladsome looks, "My dear Scrooge, how are you? When will you come to see me?" No beggars implored him to bestow a trifle, no children asked him what it was o'clock, no man or woman ever once in all his life inquired the way to such and such a place, of Scrooge. Even the blind men's dogs appeared to know him; and when they saw him coming on, would tug their owners into doorways and up courts; and then would wag their tails as though they said, "No eye at all is better than an evil eye, dark master!"

But what did Scrooge care? It was the very thing he liked. To edge his way along the crowded paths of life, warning all human sympathy to keep its distance, was what the knowing ones call "nuts" to Scrooge.

Once upon a time--of all the good days in the year, on Christmas Eve--old Scrooge sat busy in his counting-house. It was cold, bleak, biting weather: foggy withal: and he could hear the people in the court outside go wheezing up and down, beating their hands upon their breasts, and stamping their feet upon the pavement stones to warm them. The city clocks had only just gone three, but it was quite dark already--it had not been light all day: and candles were flaring in the windows of the neighbouring offices, like ruddy smears upon the palpable brown air. The fog came pouring in at every chink and keyhole, and was so dense without, that although the court was of the narrowest, the houses opposite were mere phantoms. To see the dingy cloud come drooping down, obscuring everything, one might have thought that Nature lived hard by, and was brewing on a large scale.

The door of Scrooge's counting-house was open that he might keep his eye upon his clerk, who in a dismal little cell beyond, a sort of tank, was copying letters. Scrooge had a very small fire, but the clerk's fire was so very much smaller that it looked like one coal. But he couldn't replenish it, for Scrooge kept the coal-box in his own room; and so surely as the clerk came in with the shovel, the master predicted that it would be necessary for them to part. Wherefore the clerk put on his white comforter, and tried to warm himself at the candle; in which effort, not being a man of a strong imagination, he failed.

"A merry Christmas, uncle! God save you!" cried a cheerful voice. It was the voice of Scrooge's nephew, who came upon him so quickly that this was the first intimation he had of his approach.
Hi Craig,

---

Rishabh Moudgil
rishabh.moudgil@uwaterloo.ca

Craig Kaplan
csk@uwaterloo.ca

Kevin Harrigan
kevinh@uwaterloo.ca

Kristina Bayda
kbayda@uwaterloo.ca

Travis Bartlett
travis.bartlett@uwaterloo.ca

Subject: A01 Marking Scheme

Thread-Topic: A01 Marking Scheme

Thread-Index: AdJw/+DUxNKRR1CRRKOZfc2CQLKSng==

Date: Tue, 17 Jan 2017 20:57:36 +0000

Message-ID: <748888CA42FDF349AF07A8978DDED060281C9EC0@connmbx02>

Accept-Language: en-CA, en-US

Content-Language: en-CA

X-MS-Exchange-Organization-AuthAs: Internal

X-MS-Exchange-Organization-AuthMechanism: 04

X-MS-Exchange-Organization-AuthSource: connhub1.connect.uwaterloo.ca

X-MS-Has-Attach:

X-MS-Exchange-Organization-SCL: -1

X-MS-TNEF-Correlator:

Content-Type: multipart/alternative;

boundary="_000_748888CA42FDF349AF07A8978DDED060281C9EC0@connmbx02"

MIME-Version: 1.0

--_000_748888CA42FDF349AF07A8978DDED060281C9EC0@connmbx02_

Content-Type: text/plain; charset="Windows-1252"

Content-Transfer-Encoding: quoted-printable

---

Hi Craig,
PROCESSING  PR AA1 S EH0 S IH0 NG
PROCESSION  PR AH0 S EH1 SH AH0 N
PROCESSION(1)  PR OW0 S EH1 SH AH0 N
PROCESSIONAL  PR AH0 S EH1 SH AH0 N AH0 L
PROCESSIONAL(1)  PR OW0 S EH1 SH AH0 N AH0 L
PROCESSIONS  PR OW0 S EH1 SH AH0 N AH0 L
PROCESSOR  PR AA1 S EH2 S ER0
PROCESSOR'S  PR AA1 S EH2 S ER0 Z
PROCESSORS  PR AA1 S EH2 S ER0 Z
PROCH  PR AA1 K
PROCHASKA  PR AH0 HH AA1 S K AH0
PROCHAZKA  PR AH0 HH AA1 Z K AH0
PROCHNOW  PR AA1 N AW0
PROCIDA  PR OW0 CH IY1 D AH0
PROCK  PR AA1 K
PROCKTER  PR AA1 K T ER0
PROCLAIM  PR OW0 K L EY1 M
PROCLAIMED  PR OW0 K L EY1 M D
PROCLAIMING  PR OW0 K L EY1 M IH0 NG
PROCLAIMS  PR OW0 K L EY1 M Z
PROCLAMATION  PR AA2 K L AH0 M EY1 SH AH0 N
PROCLAMATIONS  PR AA2 K L AH0 M EY1 SH AH0 N Z
PROCLIVITIES  PR OW0 K L IH1 V AH0 T IY0 Z
PROCLIVITY  PR OW0 K L IH1 V AH0 T IY0
PROCONSUL  PR OW0 K AA1 N S AH0 L
PROCOPIO  PR OW0 K OW1 P IY0 OW0
<table>
<thead>
<tr>
<th>Date</th>
<th>Low temperature</th>
<th>High temperature</th>
<th>Precipitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-Jan-14</td>
<td>-15.6</td>
<td>-8.9</td>
<td>0.1</td>
</tr>
<tr>
<td>02-Jan-14</td>
<td>-17.7</td>
<td>-15.1</td>
<td>0.1</td>
</tr>
<tr>
<td>03-Jan-14</td>
<td>-23.4</td>
<td>-13.1</td>
<td>0</td>
</tr>
<tr>
<td>04-Jan-14</td>
<td>-12.7</td>
<td>-2.5</td>
<td>0</td>
</tr>
<tr>
<td>05-Jan-14</td>
<td>-3.7</td>
<td>-1.2</td>
<td>19.1</td>
</tr>
<tr>
<td>06-Jan-14</td>
<td>-19.6</td>
<td>-2.1</td>
<td>7.7</td>
</tr>
<tr>
<td>07-Jan-14</td>
<td>-26.1</td>
<td>-18.7</td>
<td>1.5</td>
</tr>
<tr>
<td>08-Jan-14</td>
<td>-19.1</td>
<td>-11.1</td>
<td>0</td>
</tr>
<tr>
<td>09-Jan-14</td>
<td>-22.2</td>
<td>-8.3</td>
<td>0</td>
</tr>
<tr>
<td>10-Jan-14</td>
<td>-8.3</td>
<td>2.4</td>
<td>0</td>
</tr>
<tr>
<td>11-Jan-14</td>
<td>0.3</td>
<td>5.4</td>
<td>26.4</td>
</tr>
<tr>
<td>12-Jan-14</td>
<td>-0.8</td>
<td>1.3</td>
<td>0</td>
</tr>
<tr>
<td>13-Jan-14</td>
<td>0.4</td>
<td>5.8</td>
<td>0.2</td>
</tr>
<tr>
<td>14-Jan-14</td>
<td>-2.5</td>
<td>3.3</td>
<td>0</td>
</tr>
<tr>
<td>15-Jan-14</td>
<td>-8.5</td>
<td>-0.4</td>
<td>1.4</td>
</tr>
<tr>
<td>16-Jan-14</td>
<td>-8.7</td>
<td>-4</td>
<td>2.7</td>
</tr>
<tr>
<td>17-Jan-14</td>
<td>-8</td>
<td>-0.3</td>
<td>3.9</td>
</tr>
<tr>
<td>18-Jan-14</td>
<td>-10.1</td>
<td>-4.6</td>
<td>1.7</td>
</tr>
<tr>
<td>19-Jan-14</td>
<td>-10.7</td>
<td>-3.4</td>
<td>1.4</td>
</tr>
</tbody>
</table>
Reading text

Reading text from a file can be quite painful in many programming languages. Processing keeps it simple:

```java
String[] loadStrings(String filename) { ... }
```

Load a text file from the sketch folder. Break it up into lines and return an array of Strings, one per line.
void setup()
{
    String[] lines = loadStrings( "dict.txt" );
    printArray( lines );
}
Breaking up long lines

A line in a file may contain lots of individual chunks of data separated by whitespace. We’d like to break lines into words, just as we broke files into lines.

`String[] splitTokens(String line) { ... }`

Turn a line of text into an array of “words” (any non-whitespace characters separated by whitespace).
String s = "    Marley was       dead: to begin with. ";
String[] toks = splitTokens( s );
printArray( toks );

[0] "Marley"
[1] "was"
[2] "dead:"
[3] "to"
[4] "begin"
[5] "with."
Writing text

We know we can use `println()` to send any text to the console.

A similar mechanism allows us to create objects that stand in for text files. Sending those objects `println()` messages puts text into the file.

```java
PrintWriter createWriter(String filename) { ... }
```
Create an object that can output text to a file.
Idiom for writing text

PrintWriter pw = createWriter( "output.txt" );

pw.println( "Hello" );
pw.println( mouseX );
pw.println( PI );
pw.println( "THE END" );

pw.flush();
pw.close();
Idiom for writing text

```java
PrintWriter pw = createWriter("output.txt");
pw.println("Hello");
pw.println(mouseX);
pw.println(PI);
pw.println("THE END");
pw.flush();
pw.close();
```
Idiom for writing text

```java
PrintWriter pw = createWriter("output.txt");

pw.println("Hello");
pw.println(mouseX);
pw.println(PI);
pw.println("THE END");

pw.flush();
pw.close();
```

Send some text to the writer object.
Idiom for writing text

```java
PrintWriter pw = createWriter( "output.txt" );
pw.println( "Hello" );
pw.println( mouseX );
pw.println( PI );
pw.println( "THE END" );
pw.flush();
pw.close();
```

Send the data out to the hard drive and close the file.
Reasons to write text

Logging: Create a permanent record of the behaviour of the program to review later.

Persistence: Store information about the program’s state in an external file so that the sketch can restart with that state later.

Workflow: Create text output that can be read by another program for further processing.