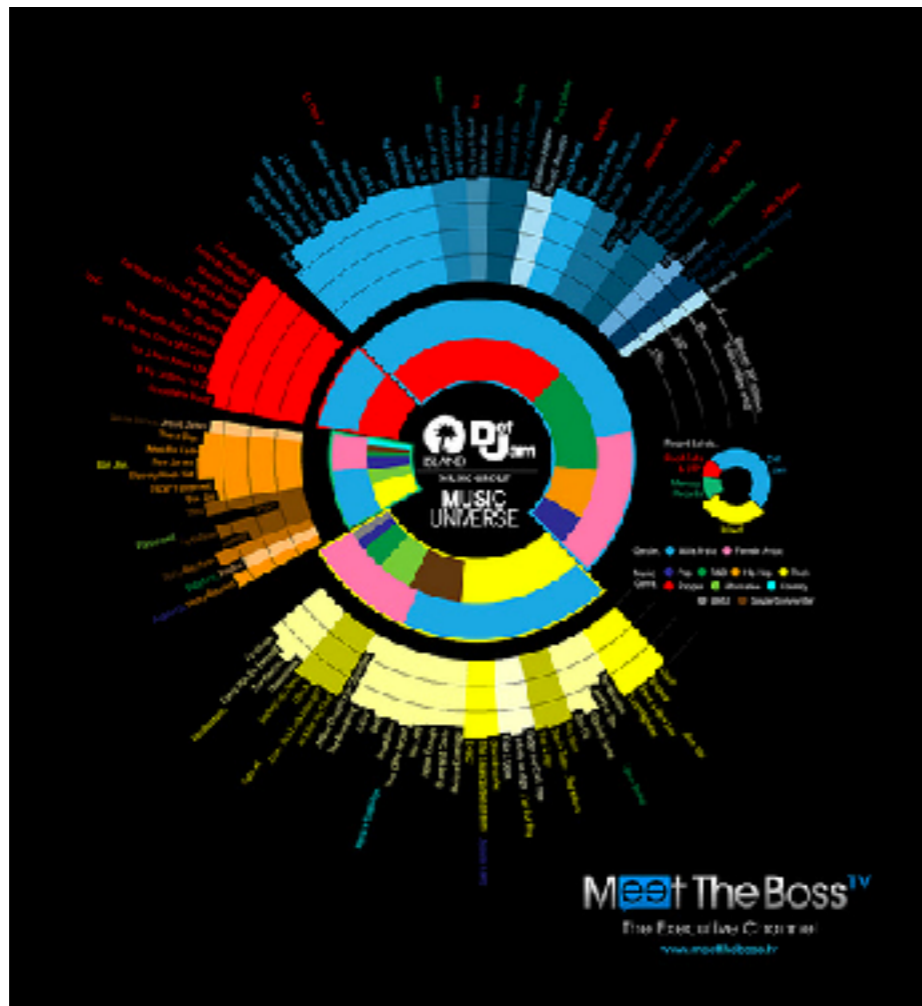




Module 03

Input / Output



Data visualization



Procedural content

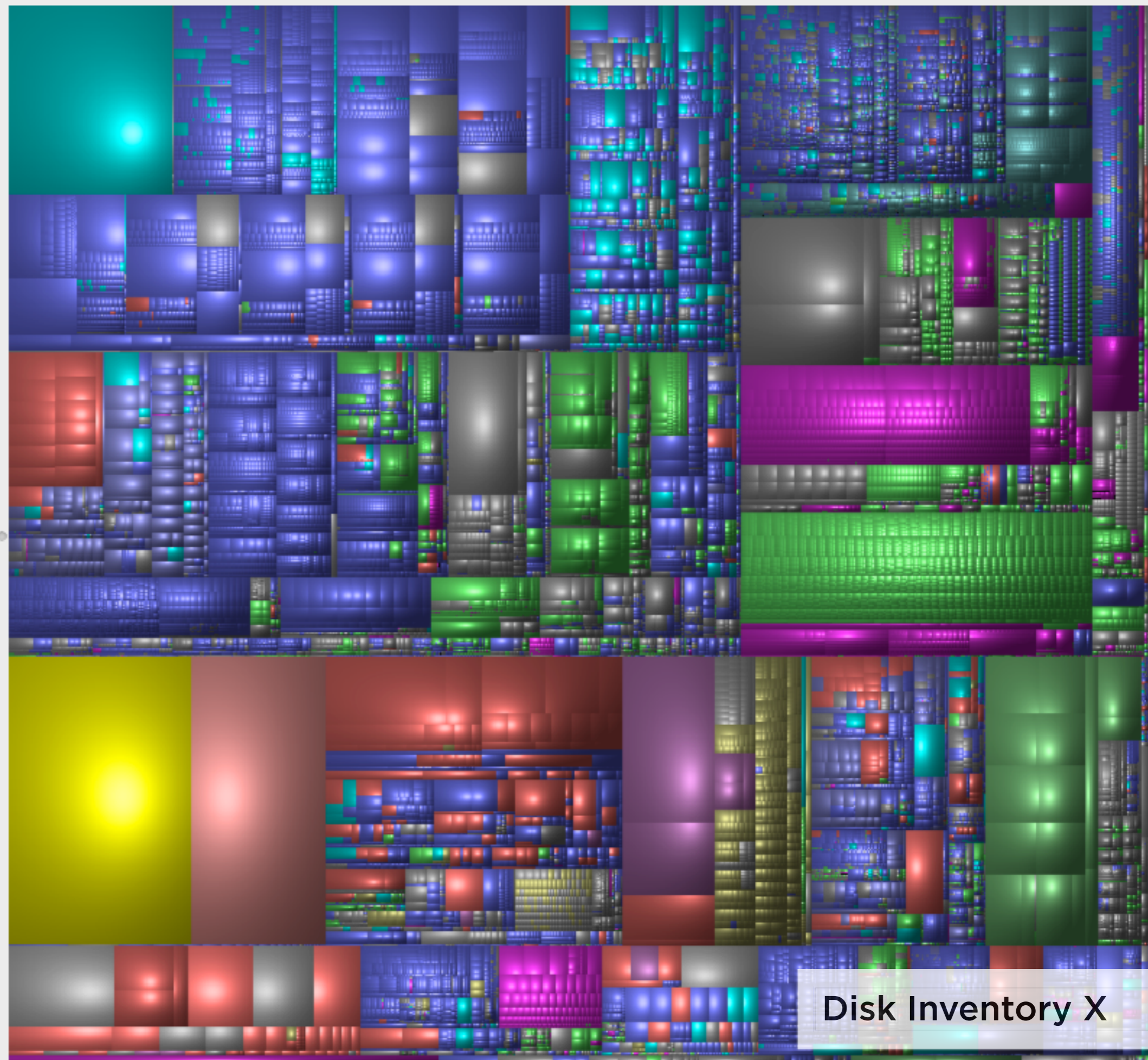


Workflow
integration

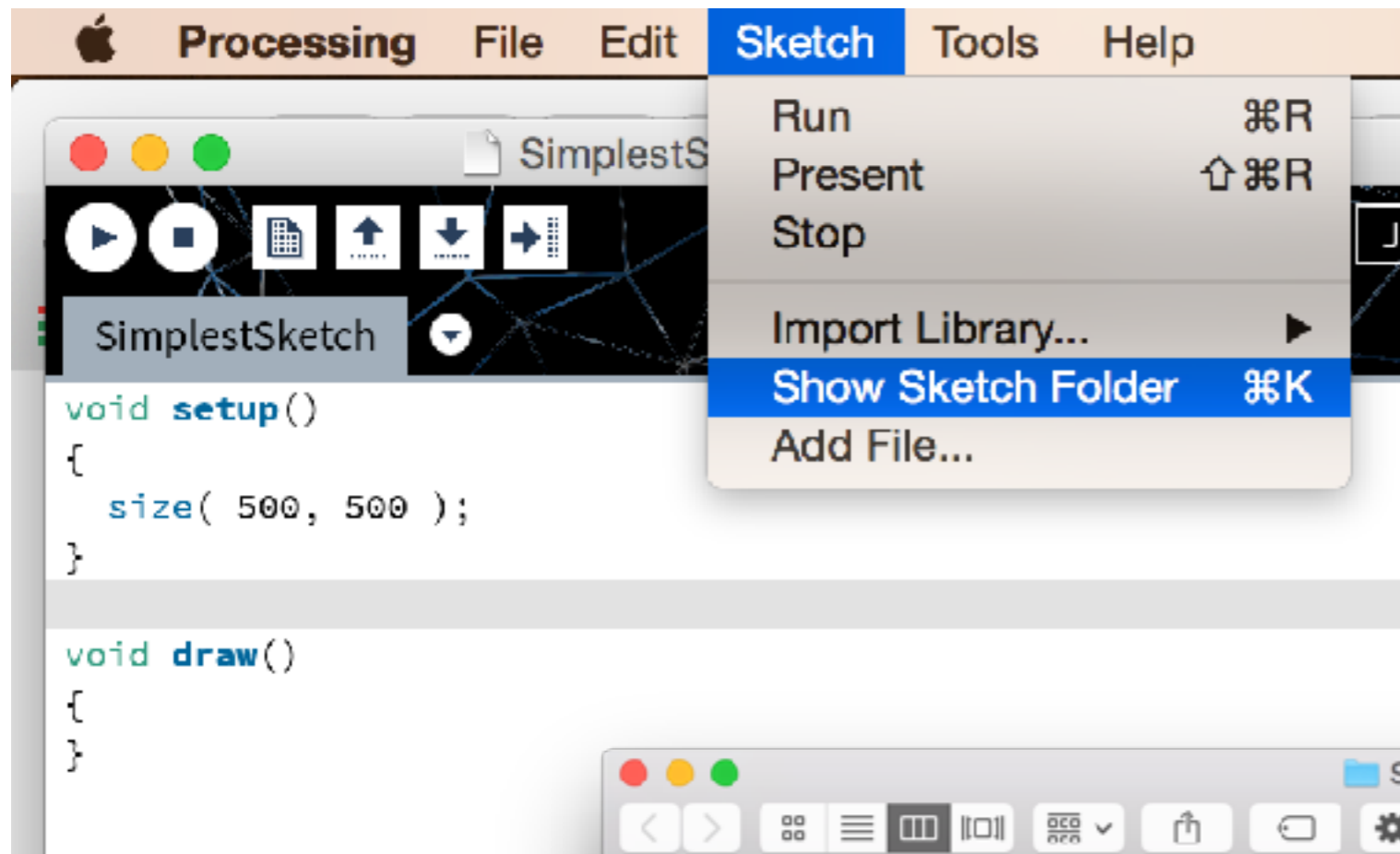
We can write more interesting programs when we can exchange information with the outside world.

Problem with the outside world: there's a lot of it.

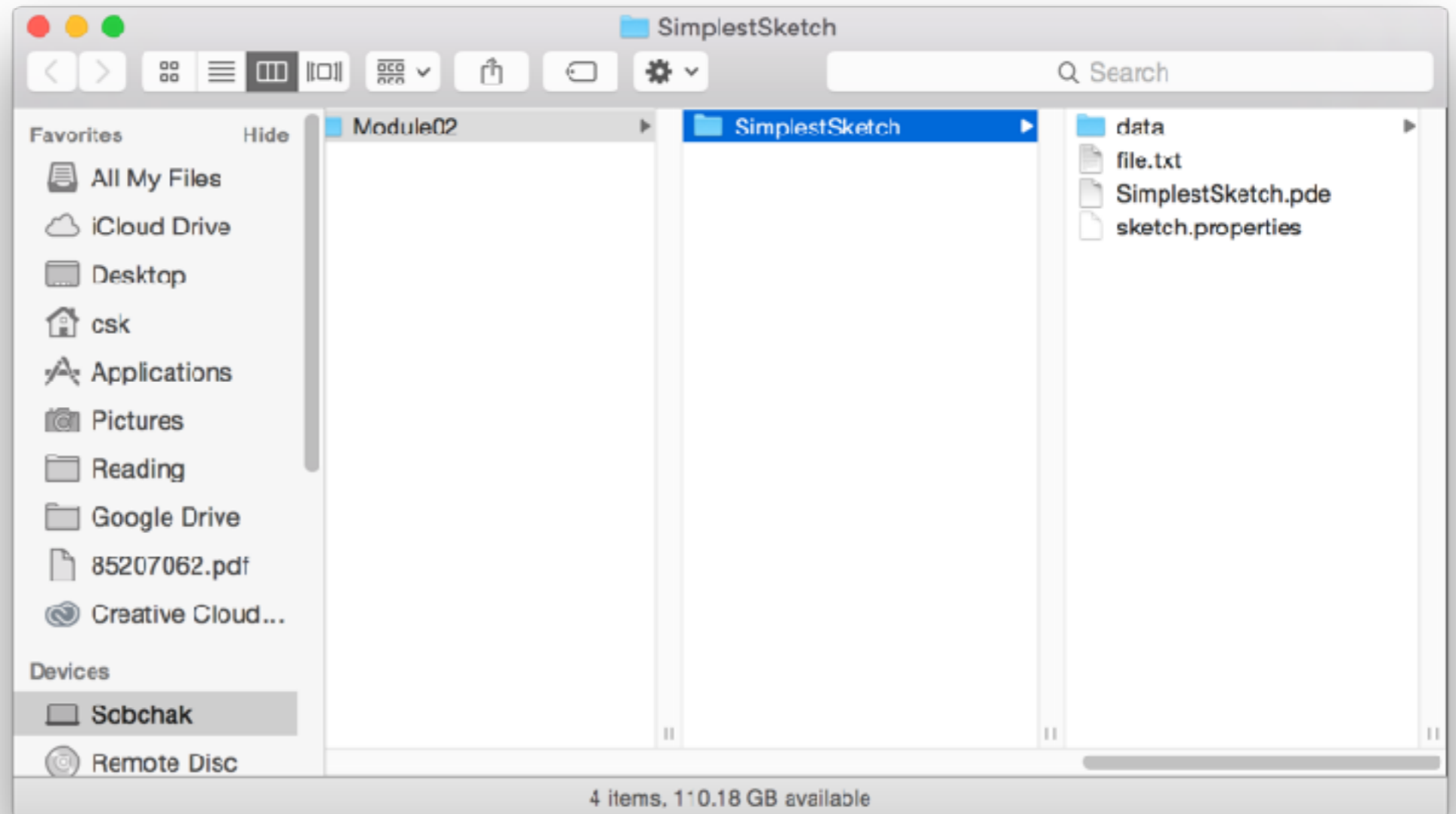
	Size
k	98.2 GB
ary	54.7 GB
ures	19.0 GB
ktop	17.7 GB
ic	11.3 GB
ching	10.4 GB
hfl	9.8 GB
es	7.9 GB
Backup	5.4 GB
nloads	3.5 GB
uments	3.5 GB
ice	2.9 GB
gle Drive	2.4 GB
-archive	10,03.2 MB
	9,63.7 MB
roid	4,00.4 MB
.tar.gz	1,24.6 MB
ederalBackup	95.4 MB
transporter	45.1 MB
2	23.5 MB
ne	11.8 MB
	8.0 MB
	7.9 MB
config	6.7 MB
_ext	5.9 MB
erProject	5.3 MB
mbnails	1.7 MB
adelbulber	1.3 MB
ic	1.2 MB
s.cache-1	1.1 MB
-applet	879 kB
ond-fonts.c...	694 kB
ications	514 kB
tive Cloud	432 kB



Disk Inventory X



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

```
PImage loadImage( String filename ) { ... }
```

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" );
```

```
}
```

Read the image into the running sketch.

```
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 );
```

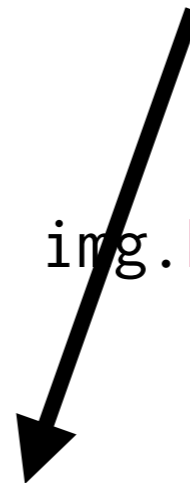
```
  image( img, width - img.width, height - img.height );
```

```
  tint( 255, 120, 120 );
```

```
  imageMode( CENTER );
```

```
  image( img, width/2, height/2, 250, 250 );
```

Draw the image at the given coordinates, scaled.



```
}
```

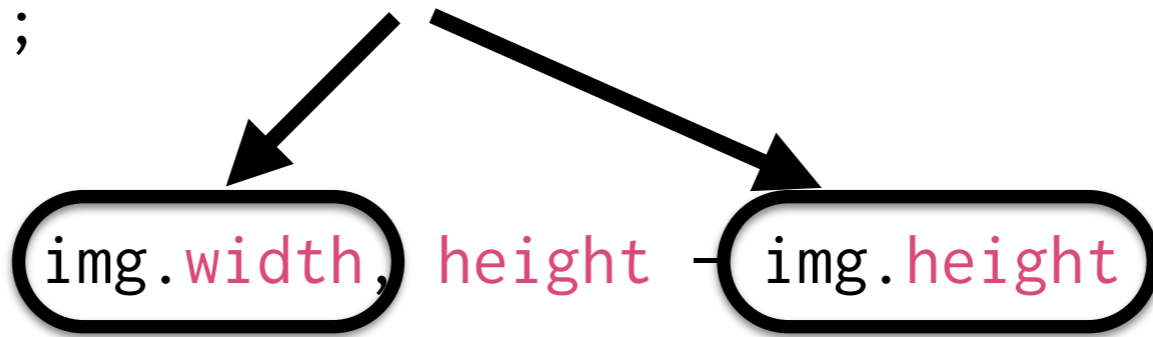

High-level PImage operations

```
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 );  
}
```

Ask the image for its dimensions.



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 );
```

```
}
```

Apply a colour wash to all images.



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

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



```
void setup()  
{  
  size( 800, 800 );  
  ...  
}
```

Don't try to load the image in the global variable declaration. This will usually fail.

Image no-nos

```
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

`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 );
```

Use image in draw().

```
image( img, width/2, height/2, 250, 250 );
```

```
}
```

You can also copy a *region* out of a source image, and scale it to any rectangle in the sketch window.

```
copy( img, sx, sy, sw, sh, dx, dy, dw, dh );
```

You can also copy a *region* out of a source image, and scale it to any rectangle in the sketch window.

```
copy( img, sx, sy, sw, sh, dx, dy, dw, dh );
```

**The source image to copy
pixels from**

You can also copy a *region* out of a source image, and scale it to any rectangle in the sketch window.

```
copy( img, sx, sy, sw, sh, dx, dy, dw, dh );
```

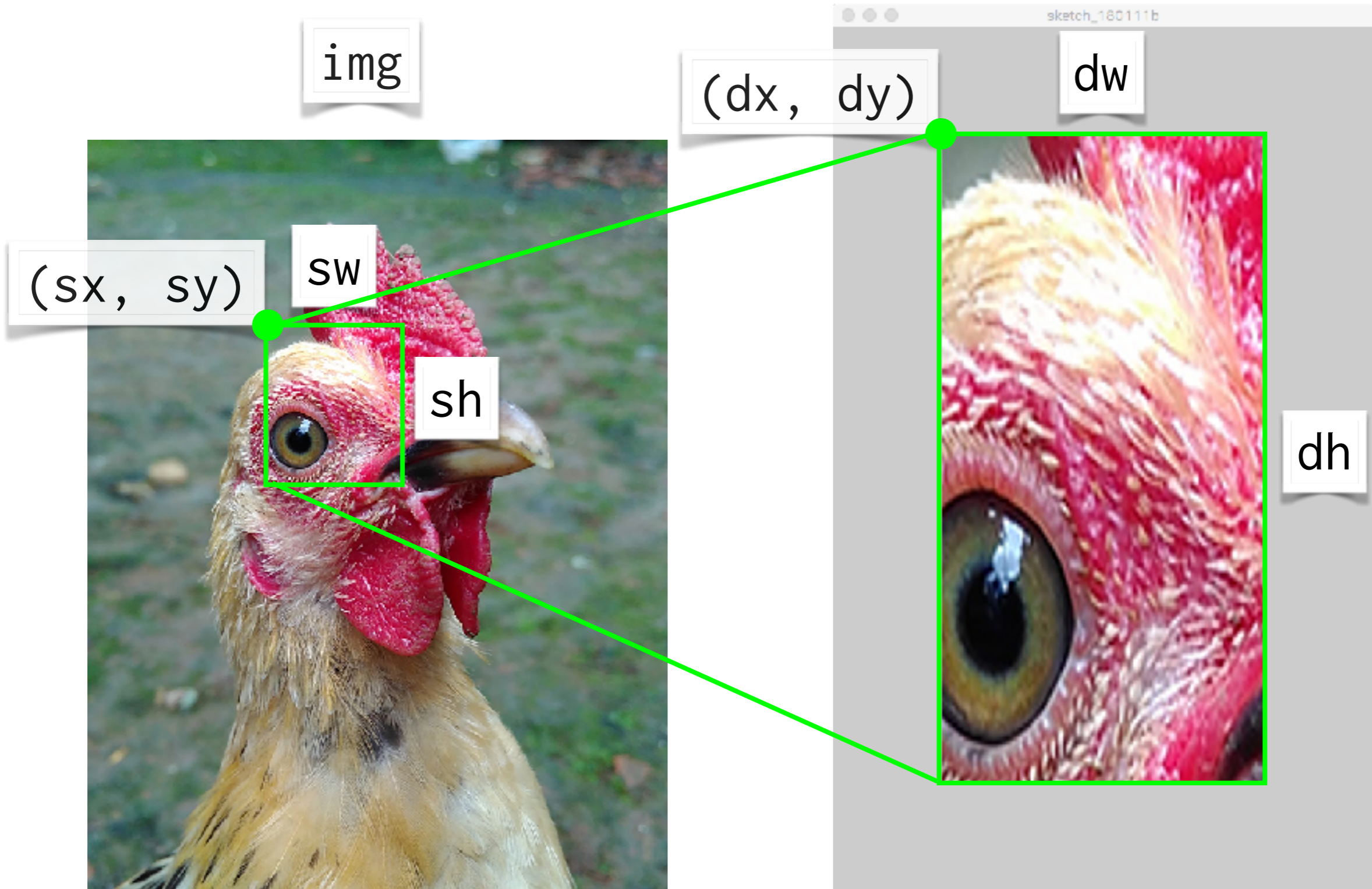
A rectangle of pixels in the source image. Just like the arguments in a call to `rect()`

You can also copy a *region* out of a source image, and scale it to any rectangle in the sketch window.

```
copy( img, sx, sy, sw, sh, dx, dy, dw, dh );
```

A rectangle of pixels in the sketch window. Again, just like a call to `rect()`

```
copy( img, sx, sy, sw, sh, dx, dy, dw, dh );
```



Writing images

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

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

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

```
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.



Raster image: represented using a grid of pixels.
JPG, PNG, GIF, BMP, TIFF, ...



Vector illustration: represented using geometric paths.
PDF, EPS, AI, SVG, ...



Images

```
loadImage()  
  PImage  
  image()
```

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.

```
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.

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

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.

```
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.

```
import processing.pdf.*;

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

```
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;  
    }  
}
```

```
}
```

Idiom for PDF recording

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 ironmongery 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! squeezing, wrenching, grasping, scraping, clutching, covetous old sinner! Hard and sharp as flint, from which no steel had ever struck out brilliant fire: self-interest and solitude reigned in his soul, and he did not know how to give pleasure to his fellow-creatures. He carried his own low temperature always about with him; he iced his office in the dog-days; 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, and no pouring rain more obdurate, than Scrooge. He knew the secret of these things. He had a little fire in the grate; and a dog, lying on a rug before it, would boast that the atmosphere was superior to any other in only one respect. They often came down handsomely, and Scrooge never did.

Nobody ever stopped him on the street, but he would stop others 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.

Received: from CONNMBX02.connect.uwaterloo.ca ([129.97.149.109]) by
connhub1.connect.uwaterloo.ca ([129.97.149.101]) with mapi id 14.03.0319.002;
Tue, 17 Jan 2017 15:57:38 -0500

From: Rishabh Moudgil <rishabh.moudgil@uwaterloo.ca>

To: Craig Kaplan <csk@uwaterloo.ca>

CC: 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/+DUxNKRRlCRRK0Zfc2CQLKSng==

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_748888CA42FDF349AF07A8978DDED060281C9EC0connmbx02_"

MIME-Version: 1.0

--_000_748888CA42FDF349AF07A8978DDED060281C9EC0connmbx02_

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

Content-Transfer-Encoding: quoted-printable

Hi Craig.

108.62.132.133 - - [17/Jan/2017:00:00:14 -0500] "GET /bridges-galleries/art-exhibits/ HTTP/1.1" 200 10943 "http://gallery.bridgesmathart.org/exhibitions/2017-joint-mathematics-meetings" "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:50.0) Gecko/20100101 Firefox/50.0"

108.62.132.133 - - [17/Jan/2017:00:00:15 -0500] "GET /tmp/cache/images/cms/arrow-right.gif HTTP/1.1" 404 195 "http://bridgesmathart.org/tmp/cache/stylesheet_combined_6fa5fb1be8f2682b13e4cf7292f5937a.css" "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:50.0) Gecko/20100101 Firefox/50.0"

108.62.132.133 - - [17/Jan/2017:00:00:16 -0500] "GET /bridges-galleries/conference-photos/ HTTP/1.1" 200 14016 "http://bridgesmathart.org/bridges-galleries/art-exhibits/" "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:50.0) Gecko/20100101 Firefox/50.0"

73.64.123.57 - - [17/Jan/2017:00:01:24 -0500] "GET /2014/bridges2014-235.pdf HTTP/1.1" 200 948062 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.95 Safari/537.36"

58.10.140.128 - - [17/Jan/2017:00:01:25 -0500] "GET /wp-login.php HTTP/1.1" 404 195 "-" "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1"

58.10.140.128 - - [17/Jan/2017:00:01:26 -0500] "GET / HTTP/1.1" 200 12340 "-" "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1"

64.126.161.169 - - [17/Jan/2017:00:01:28 -0500] "GET /2012/cdrom/proceedings/92/paper_92.pdf HTTP/1.1" 200 218338 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/602.3.12 (KHTML, like Gecko)"

64.126.161.169 - - [17/Jan/2017:00:01:29 -0500] "GET /apple-touch-icon-precomposed.png HTTP/1.1" 404 195 "-" "Safari/10602.3.12.0.1 CFNetwork/720.5.7 Darwin/14.5.0 (x86_64)"

64.126.161.169 - - [17/Jan/2017:00:01:29 -0500] "GET /apple-touch-icon.png HTTP/1.1" 404 195 "-" "Safari/10602.3.12.0.1 CFNetwork/720.5.7 Darwin/14.5.0 (x86_64)"

64.126.161.169 - - [17/Jan/2017:00:01:29 -0500] "GET /favicon.ico HTTP/1.1" 404 195 "-" "Safari/10602.3.12.0.1 CFNetwork/720.5.7 Darwin/14.5.0 (x86_64)"

64.126.161.169 - - [17/Jan/2017:00:01:30 -0500] "GET /apple-touch-icon-precomposed.png HTTP/1.1" 404 195 "-" "Safari/10602.3.12.0.1 CFNetwork/720.5.7 Darwin/14.5.0 (x86_64)"

64.126.161.169 - - [17/Jan/2017:00:01:30 -0500] "GET /apple-touch-icon.png HTTP/1.1" 404 195 "-" "Safari/10602.3.12.0.1 CFNetwork/720.5.7 Darwin/14.5.0 (x86_64)"

64.126.161.169 - - [17/Jan/2017:00:01:30 -0500] "GET /favicon.ico HTTP/1.1" 404 195 "-" "Safari/10602.3.12.0.1 CFNetwork/720.5.7 Darwin/14.5.0 (x86_64)"

64.126.161.169 - - [17/Jan/2017:00:01:31 -0500] "GET /apple-touch-icon-precomposed.png HTTP/1.1" 404 195 "-" "Safari/10602.3.12.0.1 CFNetwork/720.5.7 Darwin/14.5.0 (x86_64)"

64.126.161.169 - - [17/Jan/2017:00:01:31 -0500] "GET /apple-touch-icon.png HTTP/1.1" 404 195 "-" "Safari/10602.3.12.0.1 CFNetwork/720.5.7 Darwin/14.5.0 (x86_64)"

64.126.161.169 - - [17/Jan/2017:00:01:32 -0500] "GET /favicon.ico HTTP/1.1" 404 195 "-" "Safari/10602.3.12.0.1 CFNetwork/720.5.7 Darwin/14.5.0 (x86_64)"

64.126.161.169 - - [17/Jan/2017:00:01:32 -0500] "GET /apple-touch-icon-precomposed.png HTTP/1.1" 404 195 "-" "Safari/10602.3.12.0.1 CFNetwork/720.5.7 Darwin/14.5.0 (x86_64)"

64.126.161.169 - - [17/Jan/2017:00:01:32 -0500] "GET /apple-touch-icon.png HTTP/1.1" 404 195 "-" "Safari/10602.3.12.0.1 CFNetwork/720.5.7 Darwin/14.5.0 (x86_64)"

PROCESSING P R AA1 S EH0 S IH0 NG
PROCESSION P R AH0 S EH1 SH AH0 N
PROCESSION(1) P R OW0 S EH1 SH AH0 N
PROCESSIONAL P R AH0 S EH1 SH AH0 N AH0 L
PROCESSIONAL(1) P R OW0 S EH1 SH AH0 N AH0 L
PROCESSIONS P R OW0 S EH1 SH AH0 N Z
PROCESSOR P R AA1 S EH2 S ER0
PROCESSOR'S P R AA1 S EH2 S ER0 Z
PROCESSORS P R AA1 S EH2 S ER0 Z
PROCH P R AA1 K
PROCHASKA P R AH0 HH AA1 S K AH0
PROCHAZKA P R AH0 HH AA1 Z K AH0
PROCHNOW P R AA1 N AW0
PROCIDA P R OW0 CH IY1 D AH0
PROCK P R AA1 K
PROCKTER P R AA1 K T ER0
PROCLAIM P R OW0 K L EY1 M
PROCLAIMED P R OW0 K L EY1 M D
PROCLAIMING P R OW0 K L EY1 M IH0 NG
PROCLAIMS P R OW0 K L EY1 M Z
PROCLAMATION P R AA2 K L AH0 M EY1 SH AH0 N
PROCLAMATIONS P R AA2 K L AH0 M EY1 SH AH0 N Z
PROCLIVITIES P R OW0 K L IH1 V AH0 T IY0 Z
PROCLIVITY P R OW0 K L IH1 V AH0 T IY0
PROCONSUL P R OW0 K AA1 N S AH0 L
PROCOPTIO P R OW0 K OW1 P IY0 OW0

CMU Pronunciation Dictionary

PROCESSING P R AA1 S EH0 S IH0 NG
PROCESSION P R AH0 S EH1 SH AH0 N
PROCESSION(1) P R OW0 S EH1 SH AH0 N
PROCESSIONAL P R AH0 S EH1 SH AH0 N AH0 L
PROCESSIONAL(1) P R OW0 S EH1 SH AH0 N AH0 L
PROCESSIONS P R OW0 S EH1 SH AH0 N Z
PROCESSOR P R AA1 S EH2 S ER0
PROCESSOR'S P R AA1 S EH2 S ER0 Z
PROCESSORS P R AA1 S EH2 S ER0 Z
PROCH P R AA1 K
PROCHASKA P R AH0 HH AA1 S K AH0
PROCHAZKA P R AH0 HH AA1 Z K AH0
PROCHNOW P R AA1 N AW0
PROCIDA P R OW0 CH IY1 D AH0
PROCK P R AA1 K
PROCKTER P R AA1 K T ER0
PROCLAIM P R OW0 K L EY1 M
PROCLAIMED P R OW0 K L EY1 M D
PROCLAIMING P R OW0 K L EY1 M IH0 NG
PROCLAIMS P R OW0 K L EY1 M Z
PROCLAMATION P R AA2 K L AH0 M EY1 SH AH0 N
PROCLAMATIONS P R AA2 K L AH0 M EY1 SH AH0 N Z
PROCLIVITIES P R OW0 K L IH1 V AH0 T IY0 Z
PROCLIVITY P R OW0 K L IH1 V AH0 T IY0
PROCONSUL P R OW0 K AA1 N S AH0 L
PROCOPTO P R OW0 K OW1 P TY0 OW0

Date,Low temperature,High temperature,Precipitation

01-Jan-14,-15.6,-8.9,0.1

02-Jan-14,-17.7,-15.1,0.1

03-Jan-14,-23.4,-13.1,0

04-Jan-14,-12.7,-2.5,0

05-Jan-14,-3.7,-1.2,19.1

06-Jan-14,-19.6,-2.1,7.7

07-Jan-14,-26.1,-18.7,1.5

08-Jan-14,-19.1,-11.1,0

09-Jan-14,-22.2,-8.3,0

10-Jan-14,-8.3,2.4,0

11-Jan-14,0.3,5.4,26.4

12-Jan-14,-0.8,1.3,0

13-Jan-14,0.4,5.8,0.2

14-Jan-14,-2.5,3.3,0

15-Jan-14,-8.5,-0.4,1.4

16-Jan-14,-8.7,-4,2.7

17-Jan-14,-8,-0.3,3.9

18-Jan-14,-10.1,-4.6,1.7

19-Jan-14,-10.7,-3.4,1.4

Reading text

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

```
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.

```
PROCESSING P R AA1 S EH0 S IH0 NG  
PROCESSION P R AH0 S EH1 SH AH0 N  
PROCESSION(1) P R OW0 S EH1 SH AH0 N  
PROCESSIONAL P R AH0 S EH1 SH AH0 N AH0 L  
PROCESSIONAL(1) P R OW0 S EH1 SH AH0 N AH0 L  
PROCESSIONS P R OW0 S EH1 SH AH0 N Z
```

dict.txt

```
PROCESSING P R AA1 S EH0 S IH0 NG
PROCESSION P R AH0 S EH1 SH AH0 N
PROCESSION(1) P R OW0 S EH1 SH AH0 N
PROCESSIONAL P R AH0 S EH1 SH AH0 N AH0 L
PROCESSIONAL(1) P R OW0 S EH1 SH AH0 N AH0 L
PROCESSIONS P R OW0 S EH1 SH AH0 N Z
```

dict.txt

```
void setup()
{
  String[] lines = loadStrings( "dict.txt" );
  printArray( lines );
}
```

```
PROCESSING P R AA1 S EH0 S IH0 NG
PROCESSION P R AH0 S EH1 SH AH0 N
PROCESSION(1) P R OW0 S EH1 SH AH0 N
PROCESSIONAL P R AH0 S EH1 SH AH0 N AH0 L
PROCESSIONAL(1) P R OW0 S EH1 SH AH0 N AH0 L
PROCESSIONS P R OW0 S EH1 SH AH0 N Z
```

dict.txt

```
void setup()
{
  String[] lines = loadStrings( "dict.txt" );
  printArray( lines );
}
```

```
[0] "PROCESSING P R AA1 S EH0 S IH0 NG"
[1] "PROCESSION P R AH0 S EH1 SH AH0 N"
[2] "PROCESSION(1) P R OW0 S EH1 SH AH0 N"
[3] "PROCESSIONAL P R AH0 S EH1 SH AH0 N AH0 L"
[4] "PROCESSIONAL(1) P R OW0 S EH1 SH AH0 N AH0 L"
[5] "PROCESSIONS P R OW0 S EH1 SH AH0 N Z"
```

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).

(Note that `join()` can reassemble individual strings into a single result.)


```
String s = "    Marley was    dead: to begin with. ";  
String[] toks = splitTokens( s );  
printArray( toks );
```

```
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.

```
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

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

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

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



**Create an object
to write to.**

Idiom for writing text

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

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

**Send some text to
the writer object.**

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

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();
```

Send the data out to permanent storage 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.