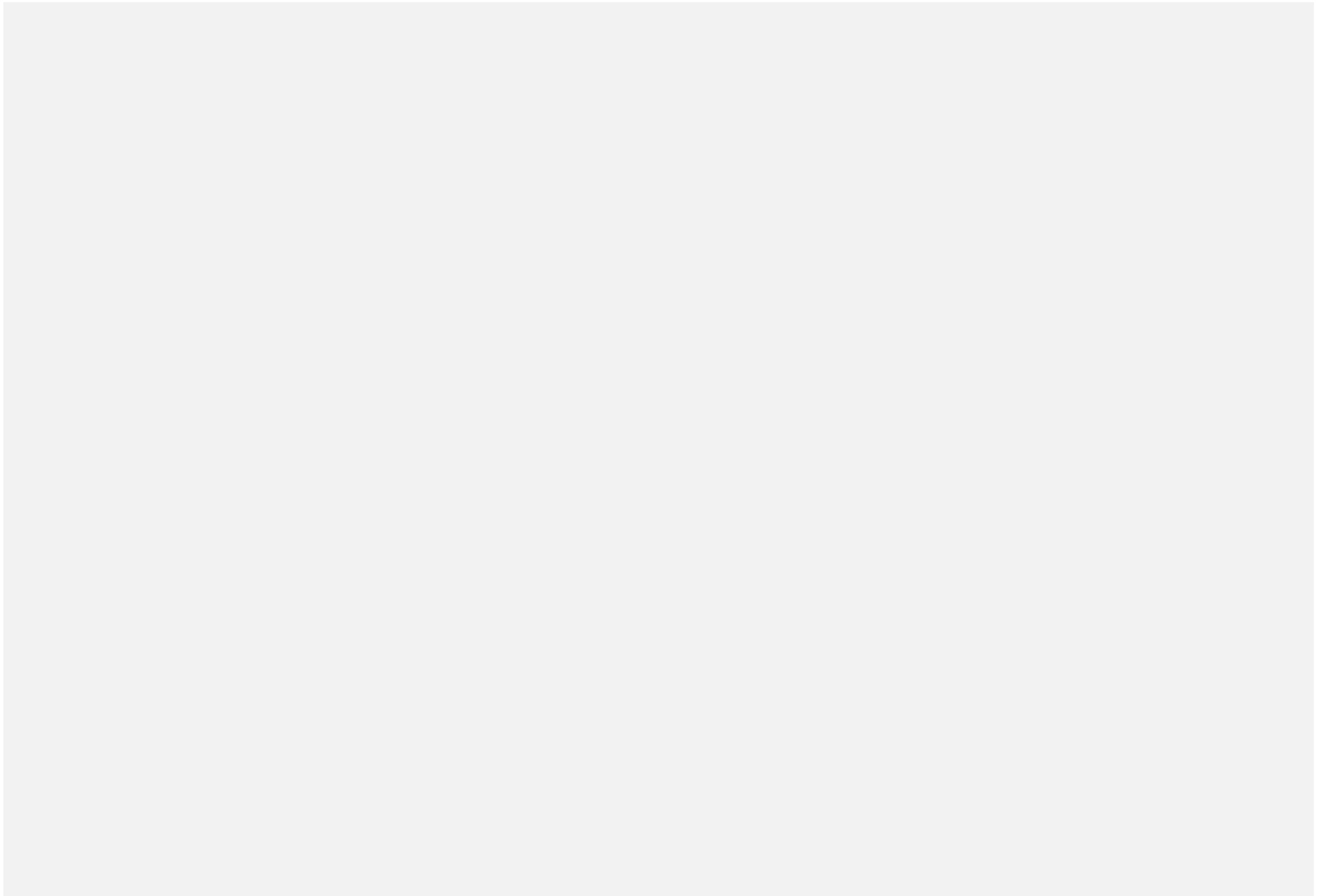




Module 03

Input / Output

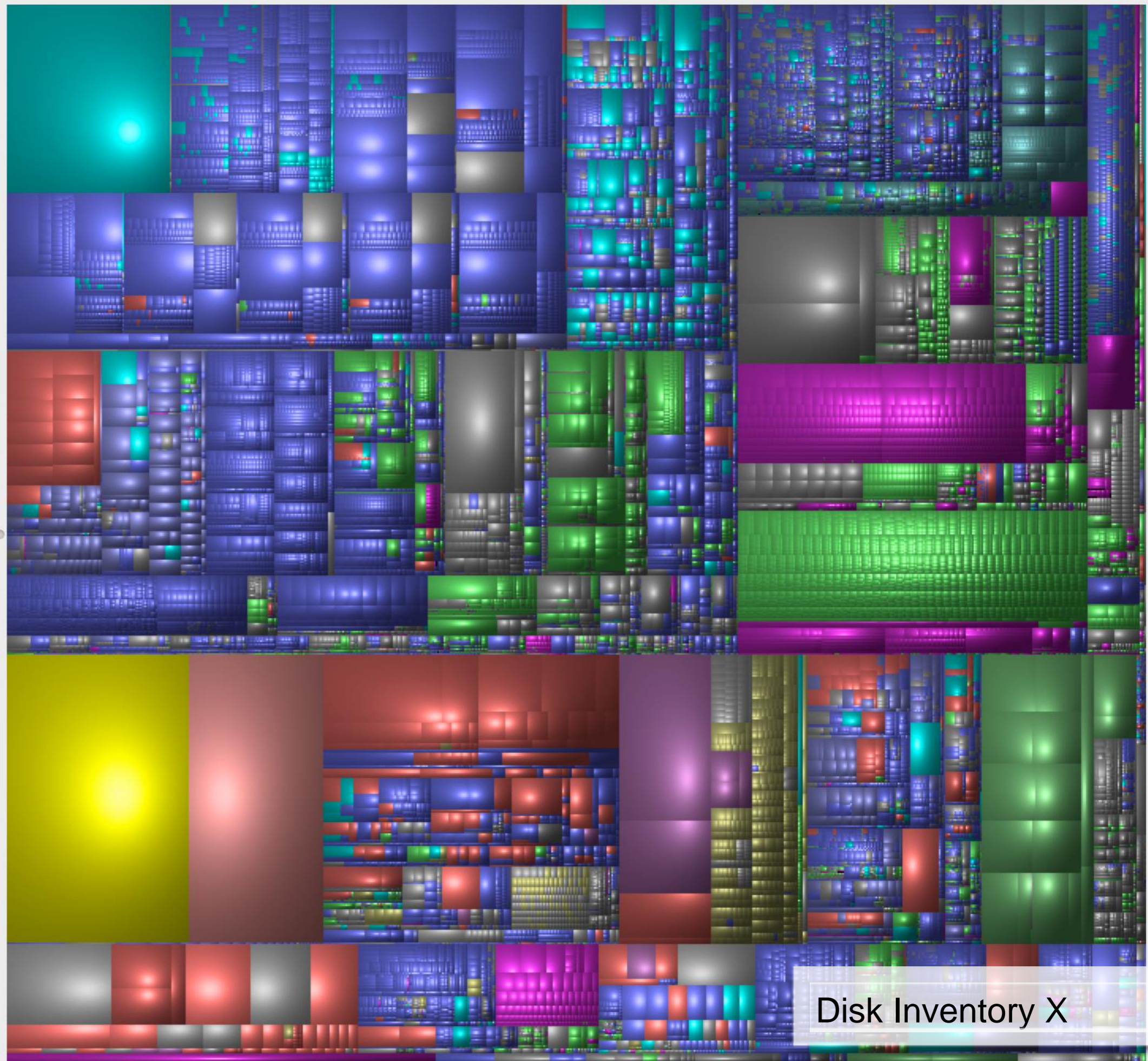
Which of these functions completes the code below to print hooray?



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

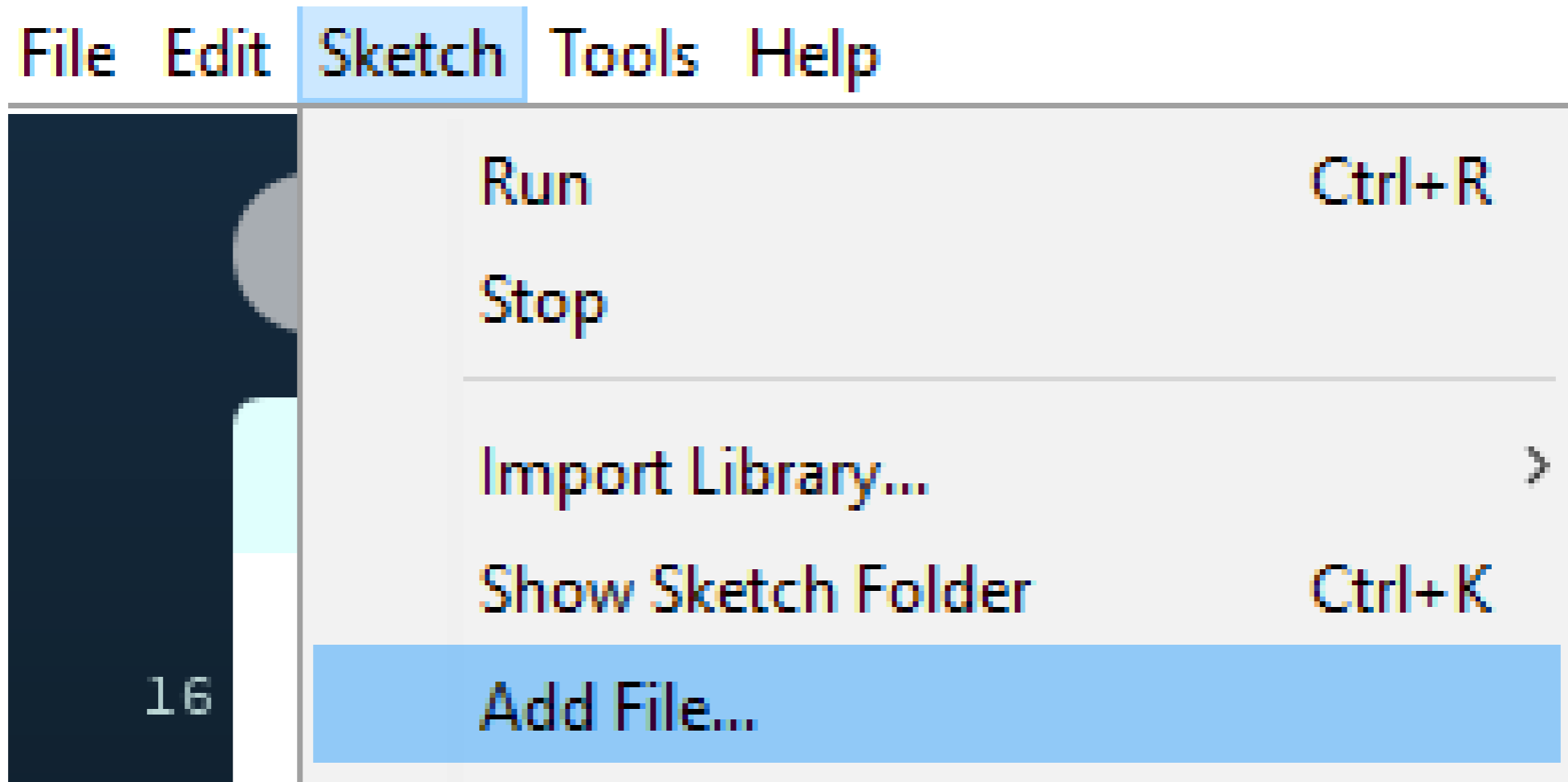
Problem with the outside world: there's a lot of it.

	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
nfl	9.8 GB
ies	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
stransporter	45.1 MB
2	23.5 MB
he	11.8 MB
n	8.0 MB
	7.9 MB
config	6.7 MB
_ext	5.9 MB
erProject	5.3 MB
mbnails	1.7 MB
andelbulber	1.3 MB
nc	1.2 MB
s.cache-1	1.1 MB
-applet	879 kB
ond-fonts.c...	694 kB
ications	514 kB
ative Cloud	422 kB



Disk Inventory X

Use the **Sketch Folder** as a gateway to the outside world.



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. 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 iron ever

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

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

Scrooge never painted but Mr. Marley's name. There sat two chairs and a table, above the door of the counting-house. Scrooge and Marley's name was known as Scrooge and Marley. Sometimes people
Oh! But how was it that he had it? Scrooge was a miser, avaricious, grasping, swartling, and tight-fisted old fellow, of the kind that has had horses flint from which no steel had ever
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
Nobody ever saw him in the street to-day, with his usual look. "My dear Scrooge, how are you? When will you dine to-day?" In beggars implored him to bestow a trifle, no children asked

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 "n

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

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

"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 ([129.97.149.101]) 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/+DUxNKRRICRRKOZfc2CQLKSng==

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

//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)"

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

CMU Pronunciation Dictionary

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

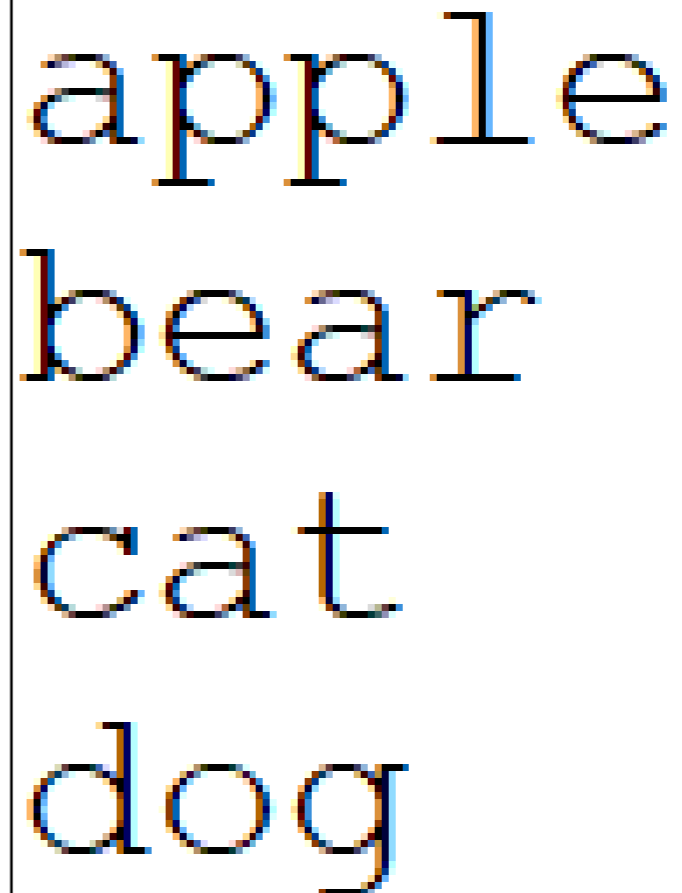
Reading text

Reading text from a file can be quite painful in many programming languages. JavaScript p5 keeps it simple:

```
myArr = loadStrings(filename);
```

Load a text file from the data folder. Break it up into lines and **return an array of Strings**, one per line.


```
let lines = [];  
function preload() {  
  lines = loadStrings( "data/mywords.txt" );  
}  
  
function setup() {  
  createCanvas( 600, 600 );  
  textSize(24);  
}  
  
function draw() {  
  background(220);  
  for (let i = 0; i < lines.length; i++) {  
    text( lines[i], 10, 30 + (i * 30));  
  }  
}
```



apple
bear
cat
dog

Shopping List

```
function preload() {  
  lines = loadStrings("data/shoppinglist.txt");  
}  
  
function setup() {  
  createCanvas(600, 600);  
  textSize(24);  
}  
  
function draw() {  
  background(220);  
  for (let i = 0; i < lines.length; i++) {  
    text(lines[i], 10, 30 + (i * 30));  
  }  
}
```

apple	2	kg
banana	5	kg
potato	4	kg
onion	2	kg

Breaking up 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. Two techniques: (see example next slide)

```
.split()  
splitTokens()
```

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.)

Breaking up lines

```
function setup() {  
  let s = "hello out there";  
  print(s); // hello out there  
  
  let myArr1 = [];  
  myArr1 = s.split(" ");  
  print(myArr1); // myarr1 is of length 3  
                  // ["hello", "out", "there"]  
  
  let myArr2 = [];  
  myArr2 = splitTokens(s, " ");  
  print(myArr2); // myarr2 is of length 3  
                  // ["hello", "out", "there"]  
}
```

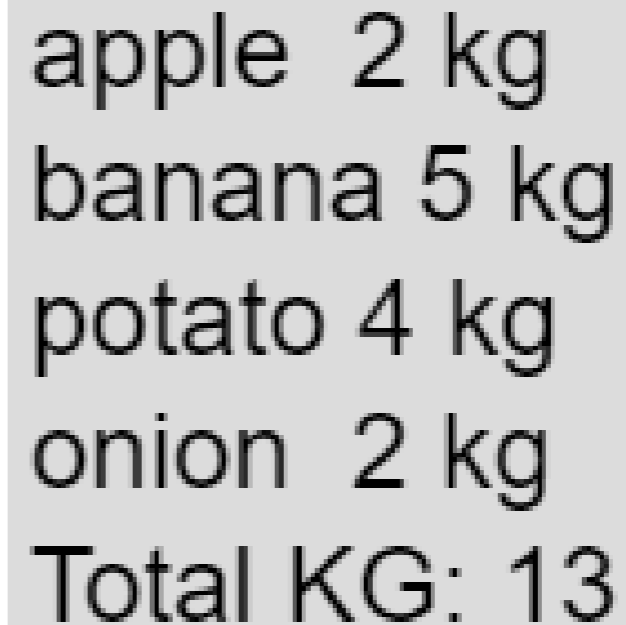

Sometimes we need strings converted to numbers

Assume we have a shopping list and we want to calculate the total weight of the items in the list.

apple	2 kg
banana	5 kg
potato	4 kg
onion	2 kg

Total KG

```
let lines = [];  
let words = [];  
let nextI;  
function preload() {  
  lines = loadStrings("data/shoppinglist.txt");  
}  
function setup() {  
  createCanvas(600, 600);  
  textSize(24);  
}  
function draw() {  
  background(220);  
  let totalKG = 0;  
  for (let i = 0; i < lines.length; i++) {  
    words = splitTokens(lines[i], " ");  
    text(lines[i], 10, 30 + (i * 30));  
    totalKG = totalKG + int(words[1]);  
    nextI = i + 1;  
  }  
  text("Total KG: " + totalKG, 10, 30+(nextI * 30));  
}
```



apple 2 kg
banana 5 kg
potato 4 kg
onion 2 kg
Total KG: 13

SpeedReader Example

- Read in a text file.
- Make one big long list (array) of “words”
 - Words may contain punctuation in this example
- Display one word at a time

SpeedReader

```
let lines = [];  
let words = [];  
let index = 0;  
function preload() {  
  lines = loadStrings("data/marley.txt");  
}  
function setup() {  
  createCanvas(400, 200);  
  textSize(50);  
  textAlign(CENTER);  
  fill(255);  
  let allLines = join( lines, " ");  
  words = splitTokens(allLines);  
  frameRate(1);  
}  
function draw() {  
  background( 80 );  
  text(words[index], width/2, height/2);  
  index = (index + 1) % words.length;  
}
```


marley.txt

- 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,

Question about marley.txt

- How many times does the word “the” occur?
 - Get rid of or ignore punctuation

```
words = splitTokens(allLines, “ .,:;<>?!@#$$%^&*()”);
```

- Capitalization does matter (“the” and “The” both count)

```
let wordInUpperCase = words[i].toUpperCase();
```

Add to Setup

```
let count = 0;
for (let i = 0; i < words.length; i++) {
  let wordInUpperCase = words[i].toUpperCase();
  if (wordInUpperCase === "THE") {
    count = count + 1;
  }
}
print("The count: ", count);
```

Writing text to a File

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

A similar mechanism puts text into the file.

```
saveStrings(list, textFilename);
```

Writing to a Text File

```
let words = 'apple bear cat dog';
let list = [];
function setup() {
  createCanvas(100, 100);
  background(200);
  text('click here to save', 10, 10, 70, 80);
  let list = split(words, ' ');
}
function mousePressed() {
  if (mouseX > 0 && mouseX < width &&
      mouseY > 0 && mouseY < height) {
    saveStrings(list, 'nouns.txt');
  }
}
```

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.

Reading and writing images

Loading an Image

```
Let img;
```

```
function preload() {  
  img = loadImage( "data/boromir.jpg" );  
}
```


Display the Image (1 of 2 slides)

```
let img;

function preload() {
  img = loadImage( "data/boromir.jpg" );
}

function setup() {
  createCanvas( 800, 400 );
}
```

Display the Image (2 of 2 slides)

```
function 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);  
}
```

Image Loading Idiom

```
let img;  
function preload() {  
    img = loadImage("some_image.jpg");  
}  
function setup() {  
    ...  
}  
function draw() {  
    image(img, 0, 0, width, height);  
    ...  
}
```

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()`

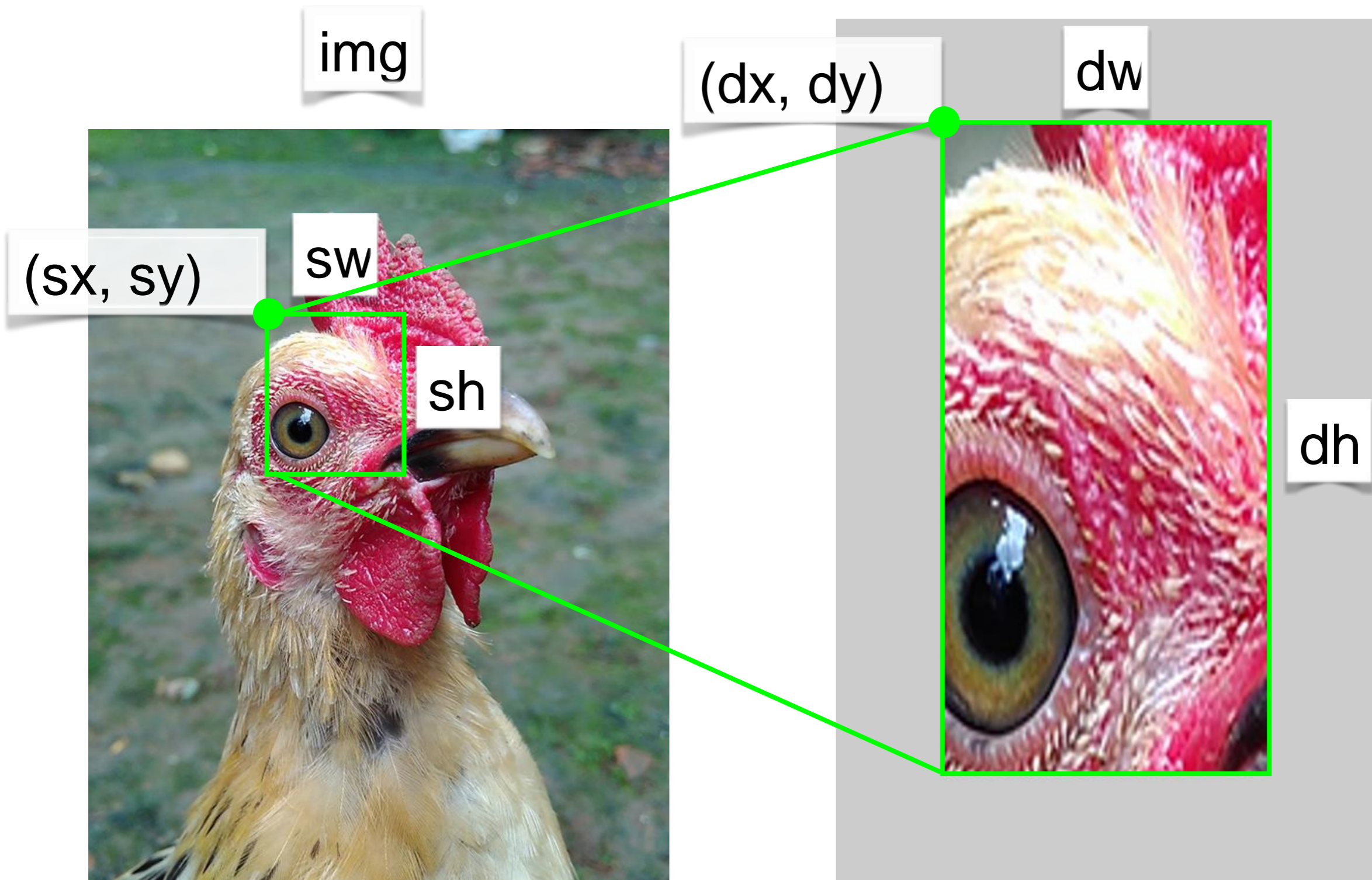
You can also copy a *region* out of a source image, and scale it to any rectangle in the sketch window.

```
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

Take a screenshot.

```
save("filename.png");
```

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

```
function keyPressed() {  
  if (key === 's') {  
    save("screen.png");  
  }  
}
```

Reading 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, ...



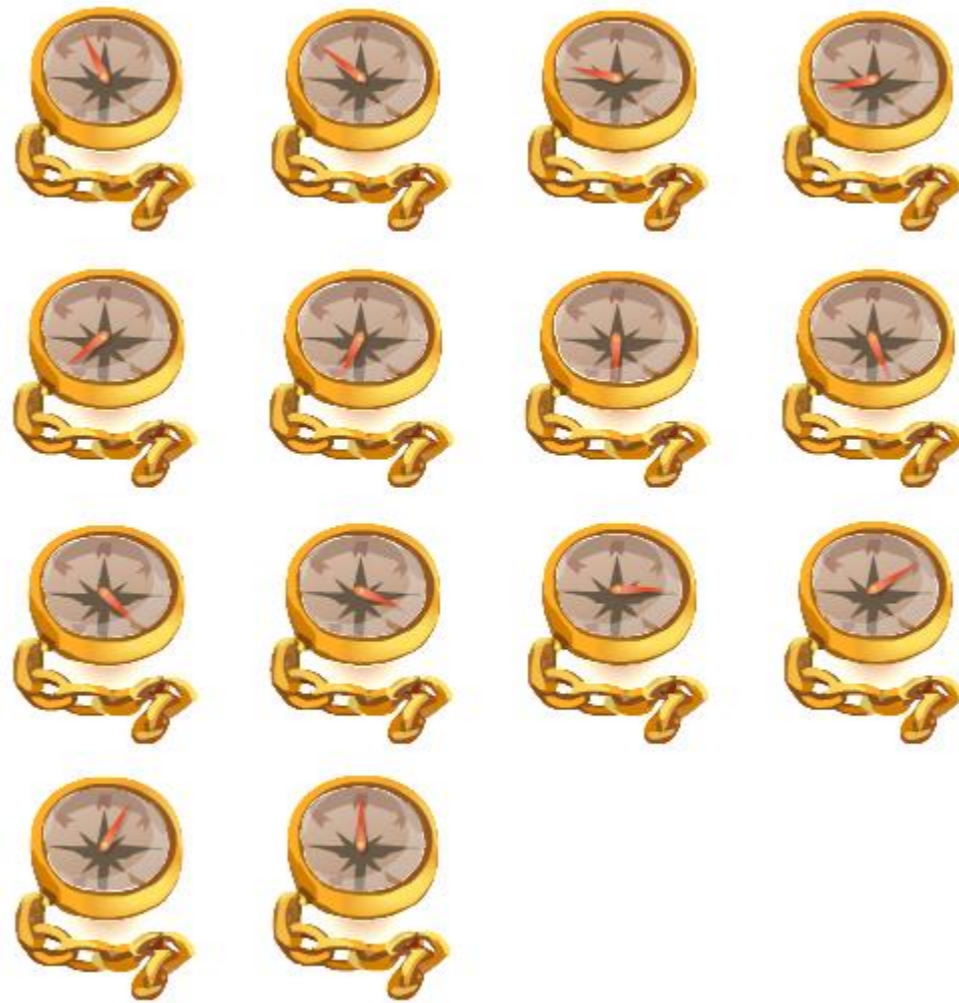
Vector illustration: represented using geometric paths.
PDF, EPS, AI, SVG, ...

Load a vector Image

```
let tiger;  
function preload() {  
  tiger = loadImage("data/tiger.svg");  
}  
function setup() {  
  createCanvas(600, 600);  
}  
function draw() {  
  background(220);  
  image(tiger, 0, 0);  
}
```

Using Sprite Files

Sprite: Compass



Sprite: Playing Cards

