Functions in Processing

CSE 120 Winter 2020

Instructor: Sam Wolfson

Teaching Assistants: Yae Kubota, Eunia Lee, Erika Wolfe

Apple dropped plan for encrypting backups after FBI complained

"Apple Inc dropped plans to let iPhone users fully encrypt backups of their devices in the company's iCloud service after the FBI complained that the move would harm investigations, six sources familiar with the matter told Reuters.

The tech giant's reversal, about two years ago, has not previously been reported. It shows how much Apple has been willing to help U.S. law enforcement and intelligence agencies, despite taking a harder line in high-profile legal disputes with the government and casting itself as a defender of its customers' information."

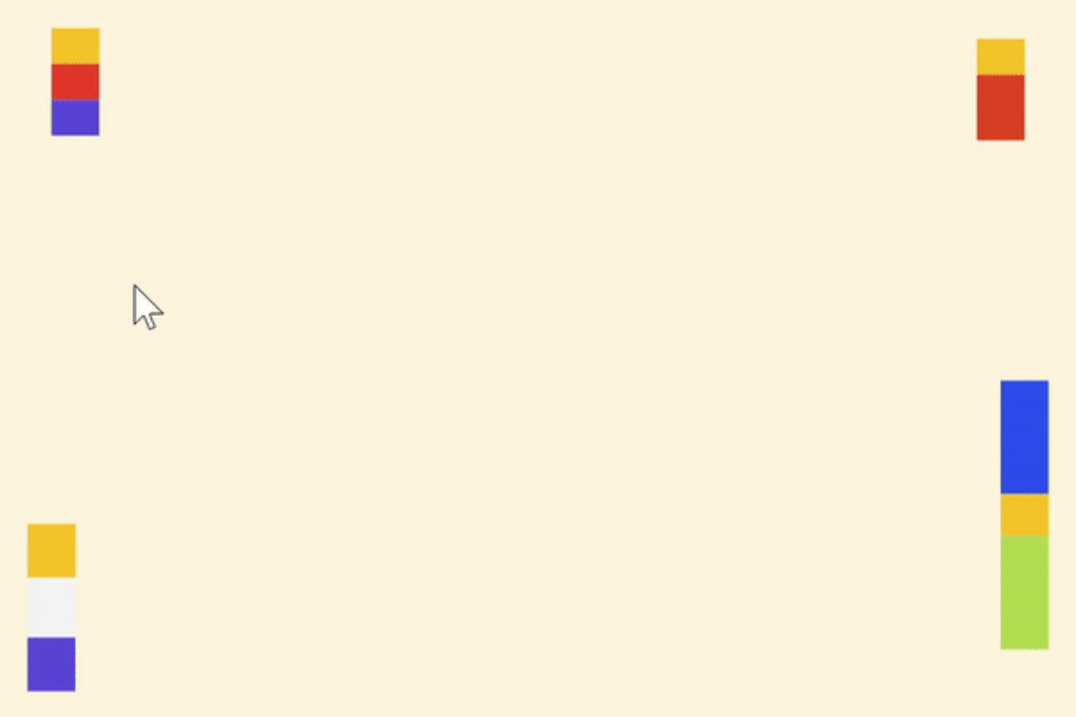
- https://www.reuters.com/article/us-apple-fbi-icloud-exclusive/exclusive-apple-dropped-plan-for-encrypting-backups-after-fbi-complained-sources-idUSKBN1ZK1CT

Administrivia

- Assignments:
  - Website Setup [checkoff] due tomorrow (1/23)
  - Reading Check 3 due tomorrow @ 3:30 pm (1/23)
  - Lego Family [submit] due Friday (1/24)
  - Animal Functions [submit] due next Tuesday (1/28)
- Editing your portfolio from home
  - Download and install Cyberduck & VS Code
  - Re-do Steps 3 & 4 from the website setup
- Quiz grades released – regrade requests open
- Make sure to take advantage of office hours and Piazza!

Lego Family

1) Create an abstracted family/group of characters
   - Can't use the Simpsons or Teenage Mutant Ninja Turtles
   - Be careful with level of detail vs. time you want to spend
2) Add motion to get them to come together

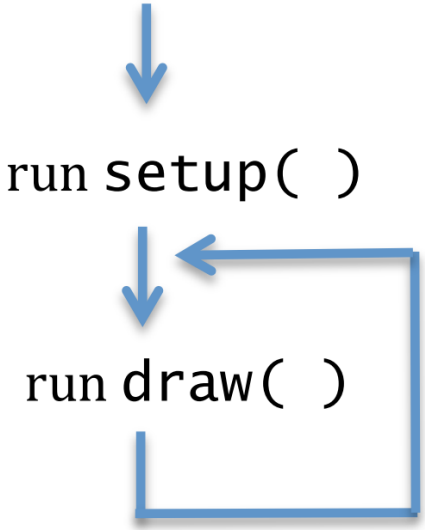


Functions (So Far)

- Used for **abstraction**
  - *Detail Removal:* subtasks with intuitive names
  - *Generalization:* don't repeat code

Lightbot:

Processing:



- `line(), rect(), …`
- `min(), max()`

Program Execution with Functions

- Functions "break" the normal sequential execution model
  - When function is **called**, begin execution of function code
  - When end of function is reached, jump back to where function was called from
    - The keyword associated with this is `return`

- **Analogy:** Song lyrics with a repeated chorus

Rick Astley could've used some functions…

We're no strangers to love
You know the rules and so do I
A full commitment's what I'm thinking of
You wouldn't get this from any other guy
I just wanna tell you how I'm feeling
Gotta make you understand

Verse 1

Never gonna give you up
Never gonna let you down
Never gonna run around and desert you
Never gonna make you cry
Never gonna say goodbye
Never gonna tell a lie and hurt you

Chorus

We've known each other for so long
Your heart's been aching but you're too shy to say it
Inside we both know what's been going on
We know the game and we're gonna play it
And if you ask me how I'm feeling
Don't tell me you're too blind to see

Verse 2

Never gonna give you up
Never gonna let you down
Never gonna run around and desert you
Never gonna make you cry
Never gonna say goodbye
Never gonna tell a lie and hurt you

Chorus

Never gonna give you up
Never gonna let you down
Never gonna run around and desert you
Never gonna make you cry
Never gonna say goodbye
Never gonna tell a lie and hurt you

Chorus

We've known each other for so long
Your heart's been aching but you're too shy to say it
Inside we both know what's been going on

Verse 3

Verse 1
Chorus
Verse 2
Chorus
Chorus
Verse 3
Chorus
Chorus

Rick Astley could've used some functions…

We're no strangers to love
You know the rules and so do I
A full commitment's what I'm thinking of
You wouldn't get this from any other guy
I just wanna tell you how I'm feeling
Gotta make you understand

Verse 1

Never gonna give you up
Never gonna let you down
Never gonna run around and desert you
Never gonna make you cry
Never gonna say goodbye
Never gonna tell a lie and hurt you

Chorus

We've known each other for so long
Your heart's been aching but you're too shy to say it
Inside we both know what's been going on
We know the game and we're gonna play it
And if you ask me how I'm feeling
Don't tell me you're too blind to see

Verse 2

Never gonna give you up
Never gonna let you down
Never gonna run around and desert you
Never gonna make you cry
Never gonna say goodbye
Never gonna tell a lie and hurt you

Chorus

Never gonna give you up
Never gonna let you down
Never gonna run around and desert you
Never gonna make you cry
Never gonna say goodbye
Never gonna tell a lie and hurt you

Chorus

We've known each other for so long
Your heart's been aching but you're too shy to say it
Inside we both know what's been going on

Verse 3

Verse 1
Chorus
Verse 2
Chorus
Chorus
Verse 3
Chorus
Chorus

Rick Astley could've used some functions…

Verse 1

We're no strangers to love
You know the rules and so do I
A full commitment's what I'm thinking of
You wouldn't get this from any other guy
I just wanna tell you how I'm feeling
Gotta make you understand

Chorus

Never gonna give you up
Never gonna let you down
Never gonna run around and desert you
Never gonna make you cry
Never gonna say goodbye
Never gonna tell a lie and hurt you

Verse 2

We've known each other for so long
Your heart's been aching but you're too shy to say it
Inside we both know what's been going on
We know the game and we're gonna play it
And if you ask me how I'm feeling
Don't tell me you're too blind to see

Verse 3

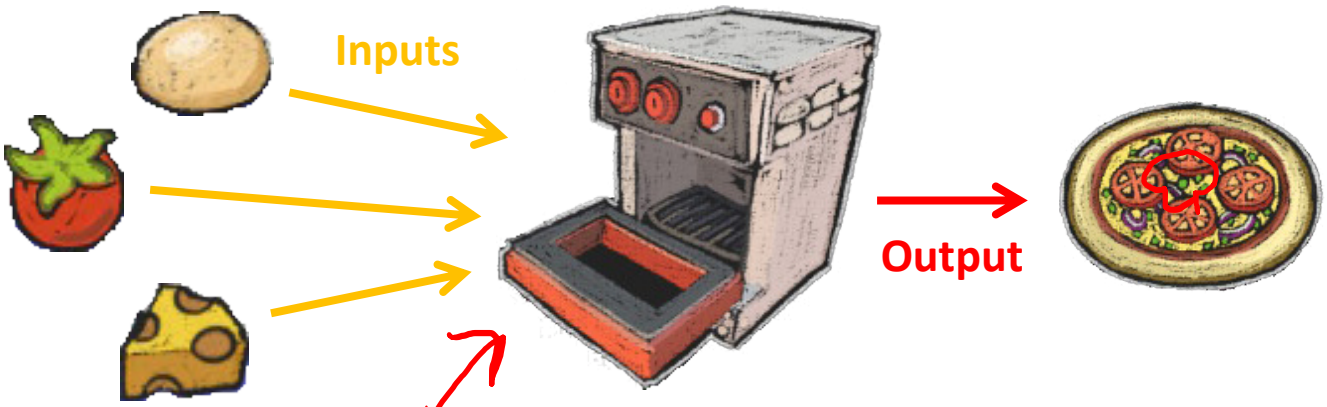
We've known each other for so long
Your heart's been aching but you're too shy to say it
Inside we both know what's been going on
We know the game and we're gonna play it
I just wanna tell you how I'm feeling
Gotta make you understand

Verse 1
Chorus
Verse 2
Chorus
Chorus
Verse 3
Chorus
Chorus

We can eliminate writing down all but 1 of the choruses!

Data Passing with Functions

- It takes in zero or more inputs, completes some task(s), and then returns a value
  - Functions can do more in Processing than in Lightbot!
- **Analogy:** An Oven



- **Analogy:** Song lyrics with that change *slightly*
  - *Parameterized* Example: Old MacDonald
    - Chorus(cow, moo), Chorus(pig, oink), Chorus(duck, quack)

Old MacDonald had a farm…

Old MacDonald had a farm, E-I-E-I-O
And on his farm he had a cow, E-I-E-I-O
With a moo-moo here and a moo-moo there
Here a moo there a moo
Everywhere a moo-moo
Old MacDonald had a farm, E-I-E-I-O

Old MacDonald had a farm, E-I-E-I-O
And on his farm he had a pig, E-I-E-I-O
With a (snort) here and a (snort) there
Here a (snort) there a (snort)
Everywhere a (snort-snort)
Old Macdonald had a farm, E-I-E-I-O

```
verse(animal, noise) {
  Old MacDonald had a farm, E-I-E-I-O
  And on his farm he had a {animal}, E-I-E-I-O
  With a {noise}-{noise} here and a {noise}-{noise} there
  Here a {noise} there a {noise}
  Everywhere a {noise}-{noise}
  Old Macdonald had a farm, E-I-E-I-O
}
```

Old MacDonald had a farm, E-I-E-I-O
And on his farm he had a horse, E-I-E-I-O
With a neigh, neigh here and a neigh, neigh there
Here a neigh there a neigh
Everywhere a neigh, neigh
Old Macdonald had a farm, E-I-E-I-O

```
verse("cow", "moo")
verse("pig", "(snort)")
verse("horse", "neigh")
```

House-Drawing Function

```
void setup() {
  size(500, 500);
}

void draw() {
  background(255);
  drawHouse(70, 10);
  rect(0, 0, 10, 10);
}

void drawHouse(int x, int y) {
  triangle(x, y, x-40, y+40, x+40, y+40);  // roof
  rect(x-40, y+40, 80, 80);                // walls
  rect(x+10, y+80, 20, 40);                // door
  return;
}
```

x ← 70
y ← 10

Return Type

return type

```
void drawHouse(int x, int y) {
  triangle(x, y, x-40, y+40, x+40, y+40);  // roof
  rect(x-40, y+40, 80, 80);                // walls
  rect(x+10, y+80, 20, 40);                // door
  return;
}
```

- What the function sends back to whoever called it
  - Can be any of the datatypes: `int`, `float`, `color`, etc.
  - If not returning anything, then we use `void`

Function Name

function name

```
void drawHouse(int x, int y) {
  triangle(x, y, x-40, y+40, x+40, y+40);  // roof
  rect(x-40, y+40, 80, 80);                // walls
  rect(x+10, y+80, 20, 40);                // door
  return;
}
```

- Does not matter to computer, but does to humans
  - Should describe what the function does
- Subject to same naming constraints as variables
- No two functions (or variables) can have the same name

Parameters

parameters

```
void drawHouse(int x, int y) {
  triangle(x, y, x-40, y+40, x+40, y+40);  // roof
  rect(x-40, y+40, 80, 80);                // walls
  rect(x+10, y+80, 20, 40);                // door
  return;
}
```

- Required part of every function definition
  - Must be surrounded by parentheses
  - If no parameters, parentheses are left empty ( )
- Datatype and name for every parameter must be specified
  - Separate parameters with commas

Function Body

body

```
void drawHouse(int x, int y) {
  triangle(x, y, x-40, y+40, x+40, y+40);  // roof
  rect(x-40, y+40, 80, 80);                // walls
  rect(x+10, y+80, 20, 40);                // door
  return;
}
```

- Body is enclosed in curly braces { }
  - Parameters are variables that are used inside the body
  - As opposed to globals: defined outside a function
- Body of a function is indented for better readability
  - Processing uses two spaces by default
  - Can use Edit → "Auto Format" (Ctrl+T on Windows or Cmd+T on Mac) to clean yours up automatically ☺

Lightbot Functions

- Lightbot functions had a different syntax, but similar parts:

function name parameters body

```
F.turn_around() Right, Right.
```

Functions Worksheet

```
void setup() {
  size(500, 500);
}

void draw() {
  drawHouse(70, 10);
}

void drawHouse(int x, int y) {
  triangle(x, y, x-40, y+40, x+40, y+40);  // roof
  rect(x-40, y+40, 80, 80);                // walls
  rect(x+10, y+80, 20, 40);                // door
  return;
}
```

function call

function name

return type

parameters

body

- Make sure you *explain* why you see what you see!

Donatello as a Function

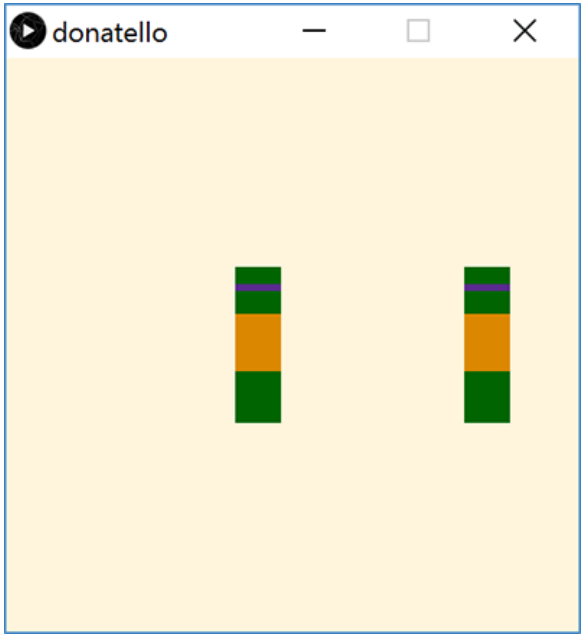
```
// draw Donatello
void drawDon() {
  fill(0, 100, 0);            // dark green
  rect(xPos, 182, 40, 15);    // top of head

  fill(88,44,141);            // purple
  rect(xPos, 197, 40, 6);     // bandana mask

  fill(0, 100, 0);            // dark green
  rect(xPos, 203, 40, 20);    // bottom of head

  fill(219, 136, 0);          // dark yellow
  rect(xPos, 223, 40, 50);    // shell

  fill(0, 100, 0);            // dark green
  rect(xPos, 273, 40, 45);    // lower body
}
```

Donatello Function *Parameterized*

- Can now call `drawDon()` function with different arguments (stored in parameter `xDon`):

```
// draw Donatello
void drawDon(int xDon) {
  fill(0, 100, 0);          // dark green
  rect(xDon,182,40,15);  // top of head
  ...
```

```
void draw() {
  background(255, 245, 220);
  drawDon(200);
  drawDon(400);
}
```



- We can also add parameter `color mask` to draw the other Teenage Mutant Ninja Turtles!

Parameters vs. Arguments

```
// draw TMNT with parameters
void draw() {
  background(255, 245, 220);
  drawTurtle(200, color(88, 44, 141)); // donatello
  drawTurtle(400, color(255, 0, 0));   // raphael
}

// parameterized ninja turtle drawing function
void drawTurtle(int x, color mask) {
  fill(0, 100, 0);        // dark green
  rect(x, 182, 40, 15);   // top of head

  fill(mask);             // apply mask color
  rect(x, 197, 40, 6);    // bandana mask
  ...
```

arguments

parameters

- Implicit parameter/variable initialization with argument values

Parameters vs. Arguments

- When you define a function, you specify parameters
  - Parameters are *internal* variables/boxes for functions
  - Use parameters for values that you want to be different on different calls to this function
- When you call a function, you pass arguments
  - The order of the arguments must match the order of the parameters
  - Inside of the function, the parameters take the value of the arguments
- We define a function once but can call it as many times as we want (and in different ways)!

Solving Problems

- Understand the problem
  - What is the problem description?
  - What is specified and what is *un*specified?
  - What has been given to you (*e.g.* starter code)?
- Break the task down into less complex subtasks

- Example: Make a function that draws a row of five mice with their ears touching/overlapping. The mice should all be the same color except for the middle one, which should be red.

Parameter Example

```
// draw mouse at position (x,y) in color c
void mouse() {
  noStroke();
  fill(color(255,0,255));    // magenta color
  ellipse(50, 50, 50, 50);  // head
  ellipse(25, 30, 30, 30);  // right ear (left on screen)
  ellipse(75, 30, 30, 30);  // left ear (right on screen)

  fill(0);                  // black color
  ellipse(40, 44, 10, 10);  // right eye (left on screen)
  ellipse(60, 44, 10, 10);  // left eye (right on screen)

  stroke(0);                // black color
  line(20, 50, 48, 60);     // upper-right whisker
  line(80, 50, 52, 60);     // upper-left whisker
  line(25, 70, 48, 60);     // lower-right whisker
  line(75, 70, 52, 60);     // lower-left whisker
}
```

Parameter Example

```
void draw() {
  mouse(0,   0, color(255, 0, 0));
  mouse(100, 0, color(0, 255, 0));
  mouse(200, 0, color(0, 0, 255));
}

// draw mouse at position (x,y) in color c
void mouse(int x, int y, color c) {
  noStroke();
  fill(c);                          // argument color
  ellipse(50+x, 50+y, 50, 50);      // head
  ellipse(25+x, 30+y, 30, 30);      // right ear (left on screen)
  ellipse(75+x, 30+y, 30, 30);      // left ear (right on screen)

  fill(0);                          // always black
  ellipse(40+x, 44+y, 10, 10);      // right eye (left on screen)
  ellipse(60+x, 44+y, 10, 10);      // left eye (right on screen)

  stroke(0);                        // always black
  line(20+x, 50+y, 48+x, 60+y);     // upper-right whisker
  line(80+x, 50+y, 52+x, 60+y);     // upper-left whisker
  line(25+x, 70+y, 48+x, 60+y);     // lower-right whisker
  line(75+x, 70+y, 52+x, 60+y);     // lower-left whisker
}
```

Looking Forward

- Portfolio
  - Don't forget to add Taijitu, Logo Design, and Lego Family (one you finish it)!
- Animal Functions
  - Start in lab on Thursday, due Tuesday (1/28)
  - Design your own animal (like the frog shown here)

Example from CSE 120 18wi student