

15-104 Introduction to Computing for Creative Practice

Fall 2020

10 Functions

Instructor: Tom Cortina, tcortina@cs.cmu.edu, GHC 4117, 412-268-3514

1



Functions

- We've used functions that are predefined.
(e.g. `random`, `ellipse`, `rectMode`, etc.)
- We pass arguments to these functions (a function call).
(e.g. `ellipse(50, 60, 100, 100);`)
- Each function assigns these arguments to a set of parameters.
(e.g. `ellipse(x, y, w, h)`)
- When the function completes its computation, it can return a result.
(e.g. `var y = floor(random(1,10));`)
- Computation continues where we left off after the function call.

2

Programmer-defined functions

- We can define our own functions that can be called from `draw` (or from each other).
- General format:

```
function name ( parameterlist ) {
    function body
}
```
- When you call a function, you should supply the same number of arguments as it has parameters.
- Your function can call other functions as well. (...including themselves!)

3

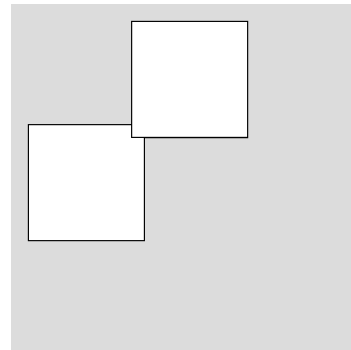
Basic Example

```
function setup() {
  createCanvas(300, 300);
  background(220);
}

function draw() {
  a(15);
}

function a(x) {
  b(x, 104);
  b(104, x);
}

function b(x, y) {
  rect(x, y, 100, 100);
}
```

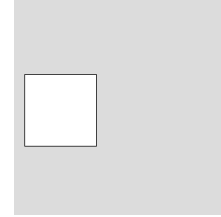


4

Odd Behavior

```
function a(value) {
  b(15, 104, value);
}

function b(x, y) {
  rect(x, y, 100, 100);
}
```



```
function a(value) {
  b(value);
}

function b(x, y) {
  rect(x, y, 100, 100);
}
```

ERROR: rect() was expecting Number for parameter #1 (zero-based index), received an empty variable instead.

LESSON: Match the number of arguments to the number of parameters.

5

arc

```
arc(x, y, w, h, start, stop, [mode])
```

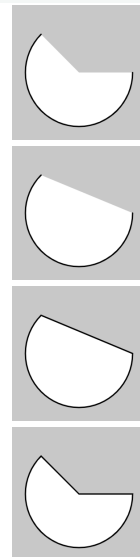
Draws an arc based on an ellipse with center **x**, **y** and width **w** and height **h**, starting at angle **start** and ending at angle **stop**. Angles are in radians.

Optional mode values used to fill the arc:


OPEN	open semi-circle
CHORD	closed semi-circle
PIE	close pie segment

Examples (shown at right, in order) – from p5.js reference:


```
arc(50, 50, 80, 80, 0, PI + QUARTER_PI);
arc(50, 50, 80, 80, 0, PI + QUARTER_PI, OPEN);
arc(50, 50, 80, 80, 0, PI + QUARTER_PI, CHORD);
arc(50, 50, 80, 80, 0, PI + QUARTER_PI, PIE);
```



6



Owls (McCarthy, Reas, Fry)




```


function setup() {
  createCanvas(480, 120);
}
function draw() {
  background(204);
  // Left owl
  translate(110, 110);
  stroke(0);
  strokeWeight(70);
  // Body:
  line(0, -35, 0, -65);
  noStroke();
  fill(255);
  // Left eye dome:
  ellipse(-17.5, -65, 35, 35);
  // Right eye dome:
  ellipse(17.5, -65, 35, 35);
  // Chin:
  arc(0, -65, 70, 70, 0, PI);
  fill(0);
  // Left eye:
  ellipse(-14, -65, 8, 8);
  // Right eye:
  ellipse(14, -65, 8, 8);
  // Beak:
  quad(0, -58, 4, -51, 0, -44,
    -4, -51);
  // cont'd on next slide

```

7



Owls (McCarthy, Reas, Fry) - continued




```

// Right owl
translate(70, 0);
stroke(0);
strokeWeight(70);
// Body:
line(0, -35, 0, -65);
noStroke();
fill(255);
// Left eye dome:
ellipse(-17.5, -65, 35, 35);
// Right eye dome:
ellipse(17.5, -65, 35, 35);
// Chin:
arc(0, -65, 70, 70, 0, PI);
fill(0);
// Left eye:
ellipse(-14, -65, 8, 8);
// Right eye:
ellipse(14, -65, 8, 8);
// Beak:
quad(0, -58, 4, -51, 0, -44,
  -4, -51);
}

```

The only thing different between the two owls is the **translate** function.

8




Owls with a function (comments omitted)

```
function setup() {
  createCanvas(480, 120);
}
function draw() {
  background(204);
  owl(110, 110);
  owl(180, 110);
}
function owl(x, y) {
  push();
  translate(x, y);
  stroke(0);
  strokeWeight(70);
  line(0, -35, 0, -65);


  noStroke();
  fill(255);
  ellipse(-17.5, -65, 35, 35);
  ellipse(17.5, -65, 35, 35);
  arc(0, -65, 70, 70, 0, PI);
  fill(0);
  ellipse(-14, -65, 8, 8);
  ellipse(14, -65, 8, 8);
  quad(0, -58, 4, -51, 0, -44,
      -4, -51);
  pop();
}
```

9




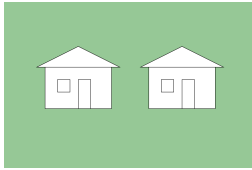
Lots of Owls!

```
function setup() {
  createCanvas(480, 120);
}
function draw() {
  background(204);
  for (var x = 35;
    x < width + 70; x += 70)
  {
    owl(x, 110);
  }
}
function owl(x, y) {
  // same as last slide
}
```



10

Functions calling Functions

```
function setup() {
  createCanvas(600, 400);
}

function draw() {
  background(150, 200, 150);
  house(mouseX + 25, mouseY);
  house(mouseX - 225, mouseY);
}

function house(x, y) {
  rect(x + 20, y, 160, 100);
  roof(x, y);
  door(x, y);
  windowpane(x, y);
}

function roof(x, y) {
  triangle(x, y, x + 100,
    y - 50, x + 200, y);
}


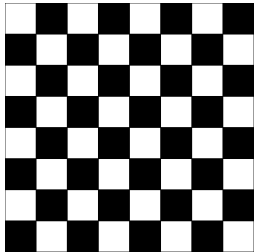
function door(x, y) {
  rect(x + 100, y + 30, 30, 70);
}

function windowpane(x, y) {
  rect(x + 50, y + 30, 30, 30);
}
```

We didn't call this function `window` since that is already a name of a function in p5.js!

11

Chess Board with a function

```
var grayvalue = 255;
function setup() { createCanvas(320, 320); }
function draw() {
  rectMode(CORNER);
  for (var row = 0; row < 8; row += 1) {
    drawRow(row);
  }
}

function drawRow(r) {
  for (var c = 0; c < 8; c += 1) { // r=row c=column
    fill( ((r+c) % 2) * 255 );
    rect(c*40, r*40, 40, 40);
  }
}
```

12



Color Bars with a function

```

var side = 40;
...
function draw() {
  for (var row = 0; row < 10; row += 1) {
    var limit = floor(random(0,11));
    drawRow(row, limit);
  }
  noLoop(); // stop animation
}
function drawRow(r, numSqrS) {
  for (var c = 0; c < numSqrS; c += 1) {
    fill(random(0,256), random(0,256), random(0,256));
    rect(c*side, r*side, side, side);
  }
}

```

