Functions

Returning Values

Chapter 9, Examples 9-1, 9-2, 9-8
Chapter 5, Example 5-14
landscape2 - multiple trees

...  
// use the tree function to  
// create a forest  
tree(40, 90);  
tree(100, 90);  
tree(160, 90);  
}  

// draw tree  
// with bottom of trunk at x, y  
function tree(x, y) {  
  ...  
}  

Starter: https://editor.p5js.org/sanghosuh/sketches/IKcseWrg  
https://editor.p5js.org/cs105/sketches/KRLY8p0mS
vehicle

// draw a vehicle
// x, y position of centre bottom
// hue colour hue to fill the car body
// (0 to 360 degrees)
function vehicle(x, y, hue) {

  ...

}

https://editor.p5js.org/cs105/sketches/ylnubok|s
// draw trees behind with a for loop

// draw the car

// draw trees in front with a while loop
// draw trees behind with a for loop
for (let x = 50; x < width; x += 100) {
  tree(x, 90);
}

// draw the car
vehicle(vehicleX, 92, vehicleHue);

// draw trees in front with a while loop
let x = 100;
while (x < width) {
  tree(x, 95);
  x = x + 100;
}

Starter: https://editor.p5js.org/sanghosuh/sketches/RhdjdCtK
https://editor.p5js.org/sanghosuh/sketches/gOlv92iR
Analogy

- Function is like ...
  - requesting an action

createCanvas(100, 100)

background(220)

What are we drawing today?

Bot Ross

"Get me canvas of size 100 x 100"

"Colour the background with grayscale of 220"
Analogy

- Function is like ...
  - requesting an action
  - asking a question
Analogy

- Function is like...
  - requesting an action
  - asking a question

when you ask a question, you get some form of response in return
(e.g., 3, 10, yes (true), no (false), ...)

AT A CONCERT...

SHOULD I JUST GO HOME AND WATCH A MOVIE?!

HMM... LET ME FIRST FIND OUT HOW MANY ARE IN FRONT OF ME

THREE PEOPLE!
THANKS!
Functions that Return Values

- We use functions that return values all the time:

```javascript
let x = random(1, 100);
let a = map(x, 1, 200, 20, 30);
fill(random(0, 255));
```
A Function with a *Return Value*

```javascript
function sum (a, b, c) {
    ...
    return ;
}
```

this is where the value is returned
Example

// calculate the sum of three numbers
function sum(a, b, c) {
    let total = a + b + c;
    return total;
}

required return statement
Function Calling Trace Through

```javascript
function setup() {
    let answer = sum(1, 2, 3);
}

function sum(a, b, c) {
    let total = a + b + c;
    return total;
}
```
function setup() {
    let answer = sum(1, 2, 3);
}

function sum(a, b, c) {
    let total = a + b + c;
    return total;
}
function setup() {
    let answer = sum(1, 2, 3);
}

function sum(a, b, c) {
    let total = a + b + c;
    return total;
}
function setup() {
    let answer = sum(1, 2, 3);
}

function sum(a, b, c) {
    let total = a + b + c;
    return total;
}
function setup() {
    let answer = sum(1, 2, 3);
}

function sum(a, b, c) {
    let total = a + b + c;
    return total;
}
function setup() {
    let answer = sum(1, 2, 3);
}

function sum(a, b, c) {
    let total = a + b + c;
    return total;
}
function setup() {
    let answer = sum(1, 2, 3);
}

function sum(a, b, c) {
    let total = 1 + 2 + 3;
    return total;
}
function setup() {
    let answer = sum(1, 2, 3);
}

function sum(a, b, c) {
    let total = 6;
    return total;
}
function setup() {
    let answer = sum(1, 2, 3);
}

function sum(a, b, c) {
    let total = 6;
    return total;
}
function setup() {
    let answer = sum(1, 2, 3);
}

function sum(a, b, c) {
    let total = 6;
    return 6;
}
function setup() {
    let answer = sum(1, 2, 3);
}

function sum(a, b, c) {
    let total = 6;
    return 6;
}
function setup() {
    let answer = 6;
}

function sum(a, b, c) {
    let total = a + b + c;
    return total;
}
Analogy

setup() \rightarrow \text{sum(\ldots)}
Analogy

setup()  

sum(...)

\( a = 1 \)  
\( b = 2 \)  
\( c = 3 \)
Analogy

setup()  \[\text{sum(\ldots)}\]

\[a + b + c\]
Analogy

setup()  sum(...)
Analogy

setup() → sum(...)

6
So how are they used? and how can we use them?
So how are they used? and how can we use them?

- Functions return values that have one of the **types**: number, boolean, string, ...
So how are they used? and how can we use them?

- Functions return values that have one of the types: number, boolean, string, ...
- Idea: we can use them anywhere we use the same type of value
Example

- Since $\text{sum}(a, b, c)$ returns number, let’s see how we can represent numbers below with $\text{sum}(a, b, c)$

```javascript
let a = 12

function setup() {
  b = 12 * 100
  c = 2 + 4 + 12
  point(12, 10)
  if (12 > 3) { ...
```
Example

- **Idea**: we can use them *anywhere* we use the same *type* of value

```javascript
let a = 12
let a = sum(2, 4, 6);

function setup() {
    b = 12 * 100
    b = sum(2, 4, 6) * 100;
    c = 2 + 4 + 12
    c = sum(2, 4, sum(2, 4, 6));

    point(12, 10)
    point(sum(2, 4, 6), 10);

    if (12 > 3) { ... }  
    if (sum(2, 4, 6) > 3) { ... }
```
function setup() {
  let x = sum(2, 4, 6);
  let y = sum(23, 45, 71) / 2;
  let z = sum(x, y, 10);
  line(0, 0, width, sum(x, y, z));
}

// calculate the sum of three numbers
function sum(a, b, c) {
  let total = a + b + c;
  return total;
}
create a function that returns boolean value (true or false)

// if mouse on left side, return true
// otherwise, return false
function mouseOnLeftHalf() {
  ...
}

Starter: https://editor.p5js.org/sanghosuh/sketches/nbINwyyZ
https://editor.p5js.org/sanghosuh/sketches/CkrPcv-4
Possible Logic Error: Forgetting to Return a Value

- If you don’t return a value, the function returns “undefined”

```javascript
function setup() {
    let a = thrice(1);
}

function thrice(a) {
    let r = a + a + a;
}
```

- a will be “undefined”

- no return statement
Possible Logic Error: Forgetting to Assign

- You usually want to use a function return value
  - it may be a logic error if you don’t assign it to a variable

```javascript
function setup() {
  sum(3, 6, 9); // nothing assigned
  random(1, 1000); // nothing assigned
}
```

there is no one to receive it
myMax

// returns the largest number (a or b)
function myMax(a, b) {
    let largest;
    if (a > b) {
        largest = a;
    } else {
        largest = b;
    }
    return largest;
}
Conducting Unit Tests to Ensure Your Function Works

- Unit test refers to **testing a single function** with a number of cases to ensure that it **works as expected for any case**
myMax (unit tests)

function setup() {

    // unit tests
    print(myMax(5, 10), 10); // answer is 10
    print(myMax(15, 10), 15); // answer is 15
    print(myMax(-500, 10), 10); // answer is 10
    print(myMax(0.5, 0.1), 0.5); // answer is 0.5
    print(myMax(10, 10), 10); // answer is 10
}

myMax (early return)

// returns the largest number (a or b)
function myMax2(a, b) {
    if (a > b) {
        return a;
    } else {
        return b;
    }
}

https://editor.p5js.org/sanghosuh/sketches/_HFh56PO
P5 has built-in functions: max, min

- return the largest of two numbers
  \[ \text{max}(a, b) \]

- return smallest of two numbers
  \[ \text{min}(a, b) \]
**myConstrain**

Constrain a value to be inside a certain range.

If value is in between low and high, return the value.

If value is less than low, return low

If value is greater than high, return high.

// constrain n to be between low and high value
function myConstrain(n, low, high) {
    ...

}

https://editor.p5js.org/cs105/sketches/EraxO6Fuz
P5 has a built-in function: constrain

- constrains n to be between low and high
  constrain(n, low, high)