Week 2 Note:

Programming:

* A series of instructions that tell your computer how to perform a task
* Written in machine code and assembly language
* High Level Programming: such as C, C++, Java, scripting languages.

How is Javascript so important to programming?

* JavaScript program is made up of a series of statements. Each statement ends with a new line or semicolon.

Arrays, Logic, and Loops:

* Arrays: an ordered list of values
  + Const myArray = [];
  + << []
* Logic: logical conditions that allow you to control the flow of a program by running different blocks of code
  + If statement:
    - const age = 23;
    - if (age < 18) {
    - console.log('Sorry, you are not old enough to play this game');
    - }
* Loops: will repeat a piece of code over and over again according to certain conditions.
  + - let bottles = 10;
    - while (bottles > 0){
    - console.log(`There were ${bottles} green bottles, hanging on a wall. And if one green bottle should accidentally fall, there'd be ${bottles-1} green bottles hanging on the wall`);
    - bottles--;
    - }
  + Infinite Loops:
    - let n = 1;
    - while(n>0){
    - console.log('Hello');
    - n++;
    - }
  + For Loops:
    - for (let bottles = 10 ; bottles > 0 ; bottles--) {
    - console.log(`There were ${bottles} green bottles, hanging on a wall. And if one green bottle should accidentally fall, there'd be ${bottles-1} green bottles hanging on the wall`);
    - }
* Functions: is a chunk of code that can be referenced by a name, and is almost like a small, self-contained mini program. Functions can help reduce repetition and make code easier to follow.
  + Declarations:
    - function hello(){
    - console.log('Hello World!');
    - }

Code Sandbox:

Chapter 2. Program Structure, 2.1 Looping a triangle

for (let line = "#"; line.length < 8; line += "#")

console.log(line);

Chapter 3. Functions, 3.2 Recursion:

function isEven(n) {

if (n == 0) return true;

else if (n == 1) return false;

else if (n < 0) return isEven(-n);

else return isEven(n - 2);

}

console.log(isEven(50));

// → true

console.log(isEven(75));

// → false

console.log(isEven(-1));

// → false

Chapter 4. Object and Arrays, 4.1 The sum of a range

function range(start, end, step = start < end ? 1 : -1) {

let array = [];

if (step > 0) {

for (let i = start; i <= end; i += step) array.push(i);

} else {

for (let i = start; i >= end; i += step) array.push(i);

}

return array;

}

function sum(array) {

let total = 0;

for (let value of array) {

total += value;

}

return total;

}

console.log(range(1, 10))

// → [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

console.log(range(5, 2, -1));

// → [5, 4, 3, 2]

console.log(sum(range(1, 10)));

// → 55