

Grow with Google: Lesson 13

Loops

1. Intro to Loops

- Permits us to run blocks of code multiple times.

2. While Loops

- Loops let you iterate over values and repeatedly run a block of code.
- Kind of surprised they started with `while` loops.

3. Part of a While Loop

- All loops repeat an action some number of times.

```
var start = 0; // when to start
while (start < 10) { // when to stop
  console.log(start);
  start = start + 2; // how to get to the next item
}
```

- If the stop or increment condition is not included, the code will run indefinitely, usually resulting in a crash.

4. Quiz: JuliaJames (4-1)

- FizzBuzz
 - Loop through the numbers 1 to 100.
 - If the number is divisible by 3, print "Fizz".
 - If the number is divisible by 5, print "Buzz".
 - If the number is divisible by both 3 and 5, print "FizzBuzz".
 - If the number is not divisible by 3 or 5, print the number.
- JuliaJames
 - Loop through the numbers 1 to 20.
 - If the number is divisible by 3, print "Julia".
 - If the number is divisible by 5, print "James".
 - If the number is divisible by 3 and 5, print "JuliaJames".
 - If the number is not divisible by 3 or 5, print the number.
- Conditional Statement Solution

```

var x = 1;

if (x % 3 === 0 && x % 5 === 0) {    // if x is divisible by 3 and divisible by
5
    console.log("JuliaJames");        // output "JuliaJames"
} else if (x % 3 === 0) {            // else if x is only divisible by 3
    console.log("Julia");            // output "Julia"
} else if (x % 5 === 0) {            // else if x is only divisible by 5
    console.log("James");            // output "James"
} else {                             // else if it is indivisible by both
    console.log(x);                  // output x
} x++;                               // increment x

```

- Ternary Expression Solution (with indentation).

```

var x = 1;
var output;

while (x <= 20) {

    output = x % 3 === 0 ?            // if x is divisible by 3
        (x % 5 === 0 ?                // and if x is divisible by 5
            "JuliaJames" :            // output "JuliaJames"
            "Julia") :                // else output "Julia"
        (x % 5 === 0 ?                // if x is divisible by 5
            "James" :                  // output "James"
            x);                       // else output = x

    console.log(output);              // output x
    x++;                             // increment x
}

```

- Output

```

1
2
Julia
4
James

```

```
Julia
7
8
Julia
James
11
Julia
13
14
JuliaJames
16
17
Julia
19
James
```

5. Quiz: 99 Bottles of Juice (4-2)

- I really struggled with this one because any of the previous methods, whether conditionals, ternary operators, or a switch, would have worked here.
- Ultimately I went with what I thought was the most lightweight and readable version.

```
var num = 99;
var plural;
var singular;

while (num > 0) {
  plural = num === 1 ? "bottle" : "bottles";
  singular = num === 2 ? "bottle" : "bottles";
  console.log(num + " " + plural + " of juice on the wall! " + num + " " +
plural + " of juice! Take one down, pass it around... " + (num-1) + " " +
singular + " of juice on the wall!");
  num--;
}
```

6. Quiz: Countdown, Liftoff! (4-3)

- A great countdown exercise with a twist.
- I finished this one in no time. It seemed to really lend itself to switches, given the context.

```

var countdown = 60;

while (countdown >= 0) {
  switch (countdown) {
    case 50:
      console.log("Orbiter transfers from ground to internal power");
      break;
    case 31:
      console.log("Ground launch sequencer is go for auto sequence start");
      break;
    case 16:
      console.log("Activate launch pad sound suppression system");
      break;
    case 10:
      console.log("Activate main engine hydrogen burnoff system");
      break;
    case 6:
      console.log("Main engine start");
      break;
    case 0:
      console.log("Solid rocket booster ignition and liftoff!");
      break;
    default:
      console.log("T-" + countdown + " seconds");
      break;
  }
  countdown--;
}

```

7. For Loops

- `while` loops break easily if a start condition, stop condition, or iterator is forgotten.
- `for` loops are recommended because they allow more control over the looping process.

8. Parts of a For Loop

- Like `while` loops, `for` loops each have a start point, stop point, and an iterator.
- Leaving out any part of these components will throw an error.
- The difference is, these conditions are defined up front.

```

for (var i = 1; // start at 1
     i <= 10; // stop when i is less than or equal to 10
     i++) { // iterate
    console.log(i); // print the value of i
} // Outputs the numbers 1 through 10.

```

9. Nested Loops

- Loops can be nested for additional complexity.
- For each value of x in the outer loop, the inner for loop executes completely. The outer loop starts with x = 0, and then the inner loop completes its cycle with all values of y.
- Once the inner loop is done iterating over y, then the outer loop continues to the next value, x = 1, and the whole process begins again.

```

for (var x = 0; x < 5; x = x + 1) {
    for (var y = 0; y < 3; y = y + 1) {
        console.log(x + ", " + y);
    }
}
/* Outputs
0, 0
0, 1
0, 2
1, 0
1, 1
1, 2
2, 0
2, 1
2, 2
3, 0
3, 1
3, 2
4, 0
4, 1
4, 2
*/

```

10. Increment and Decrement

- Most languages include an increment and decrement operator.

- `x = x+1` is the same as `x++` or `++x`.
- `x++` returns the original value of `x`, then increments it.
- `++x` increments `x`, then returns its new value.

- There are increment operators for all the operations.

- Addition

- `x = x + 1`
- `x++`
- `++x`
- `x += 1`

- Subtraction

- `x = x - 1`
- `x--`
- `--x`
- `x -= 1`

- Multiplication `x *= 1`

- Division `x /= 5`

11. Quiz: Changing the Loop (4-4)

- Convert a `while` loop into a `for` loop.

```
// While
var x = 9;
while (x >= 1) {
    console.log("hello " + x);
    x = x - 1;
}

// For
for (var x = 9; x >= 1; x--) {
    console.log("hello " + x)
}
```

12. Quiz: Fix the Error 1 (4-5)

- Don't forget any parts of the loop.

```
/* Before
for (x < 10; x++) {
    console.log(x);
}
```

```

}*/
// After
for (var x = 5; x < 10; x++) {
  console.log(x);
}

```

13. Quiz: Fix the Error 2 (4-6)

- Don't forget the semicolons!

```

for (var k = 0; k < 200; k++) {
  console.log(k);
}

```

14. Quiz: Factorials! (4-7)

- A factorial is calculated by multiplying a number by all the numbers below it.

```

0!=1
1!=1
2!=2*1=2
3!=3*2*1=6
4!=4*3*2*1=24
5!=5*4*3*2*1=120
n! = n*(n-1)*(n-2)...*1

```

- I really wanted to solve this one by decrementing the loop because the formula for factorials uses decrements. It took me a while.

```

var solution = 12;
for (var i = solution - 1; i > 1; i--) {
  solution *= i;
} console.log(solution);

```

- Udacity Forums [Beginner's Guide to Algorithms](#)
- 0! [What is Zero Factorial?](#)
- Numberphile [Zero Factorial](#)

15. Quiz: Find My Seat (4-8)

- I inverted this several times before I realized I had to swap the inner and outer loops.

```
for (var row = 0; row <= 25; row++) {  
    for (var seat = 0; seat <= 99; seat++) {  
        console.log(row + "-" + seat);  
    }  
}  
  
/* Outputs  
0-0  
0-1  
0-2  
...  
25-97  
25-98  
25-99  
*/
```

16. Lesson 4 Summary

- Functions up next.