CSCE 120: Learning To Code

Processing Data I
Hacktivity 8.1

Introduction

Prior to engaging in this hacktivity, you should have completed all of the pre-class activities as outlined in this module. At the start of class, you will be randomly assigned a partner to work with on the entirety of this hacktivity as a *peer programming* activity. Your instructor will inform you of your partner for this class.

One of you will be the driver and the other will take on the navigator role. Recall that a driver is in charge of the keyboard and computer while the navigator is in charge of the handout and directing the activity. However, you are *both* responsible for contributing and discussing solutions. If you were a driver/navigator in the prior activity, switch roles for this activity.

1 Knowledge Check

With your partner, discuss and answer each of the following questions. Write your answers down on a separate sheet of paper or type them up in a plain text file.

1. What are the three essential components of a loop control structure?
2. Describe what each of the following does to the variable `i`:
   a) `i++;`
   b) `i--;`
   c) `i += 10;`
   d) `i -= 3;`
3. For each of the following code snippets, indicate what they print to the console.

a)

```
for(var i=1; i<=10; i++) {
  console.log(i);
}
```

b)

```
for(var i=17; i>0; i-=3) {
  console.log(i);
}
```

c)

```
var i = 10;
while(i >= 0) {
  console.log(i);
  i--;
}
```

d)

```
var i = 3;
while(i < 11) {
  console.log(i);
  i += 5;
}
```

e)

```
for(var i=1; i<=5; i++) {
  for(var j=1; j<=5; j++) {
    console.log( i * j );
  }
}
```

4. Consider the following array of elements:
   `var arr = [8, 4, 2, 9, 3, 1, 5];`
   For each of the following code snippets, indicate what they print to the console or what effect(s) they have.

a)

```
arr.forEach(function(element, index) {
  console.log(element);
}
```

b)

```
arr.forEach(function(element, index, a) {
  a[index] = element + 10;
}
```

c)

```
$.each(arr, function(index, value) {
  console.log(index + value);
});
```

2 Warm-up Exercises

Download the code we've provided from GitHub using the URL, https://github.com/cbourke/LoopProject. Open the project in Light Table and open the file, `exercises/exercises.js`.

Complete the following exercises.

1. Rewrite the for loop in Question 3(a) above as an equivalent while loop. Then rewrite the while loop in Question 3(c) above as an equivalent for loop.
2. Write a loop that sums integers $1+2+3+\cdots+100$ and prints the answer. Generalize your loop so that it works for any sum $1+2+\cdots+n$
3. Write a loop that iterates over elements in an array and adds one to each value
4. A common interview question for programming is the FizzBuzz question. The task is to print out integers 1 through 100. However, for any number that is divisible by 3, output `Fizz` and for any number that is divisible by 5, output `Buzz`. For any number that is divisible by 3 and 5, output `FizzBuzz`. Solve the FizzBuzz problem using JavaScript. Recall that you can test if a number is divisible by another number using the mod operator: `(x % 2) === 0` for example is true if `x` is even.
5. Suppose we have an array of integers and an integer n. Write a snippet of code to determine all the integers 1 through n that are missing from the array. For example, if our array is `var arr = [8, 2, 9, 4, 3, 3, 4]` and $n = 10$, then we should output $1, 5, 6, 7, 10$.
6. Would you take a job if I paid you a dollar per week, but promised to double your pay each week? Figure out how much you would earn in your first six months (26 weeks) if you said yes by writing code to compute it.

3 Number Guessing Game

The Number Guessing Game is a game where one player (the computer) thinks of a number x between 1 and 100. Then the other player (the user) makes a guess. If the guess is correct, the game is over. Otherwise, the computer informs the user whether or not the number x is higher or lower than their guess as a hint. The player continues to guess until they guess correctly.

Implement this game in JavaScript. To do so, we've included a starter file, `exercises/guess.js` that generates a random integer between 1 and 100 for you. Use `prompt()` to read in guesses from the user and `alert()` to inform them of their win/number of guesses.