Practice with Arrays and Loops

Lecture 8

Announcements

- Review Session - SN014 (Bigger Room!) - Tomorrow (Weds) 5pm
- Tutoring - Same room as review session (SN014) from 6-8pm follows review
- WS1 - Due Tonight
- Quiz 2 Thursday - Arrays, for loops, and environment diagrams with arrays
- PS2 - Implementations due Monday 2/18 at 11:59pm
  - Warning: Expect long queues and waits Sunday and Monday night in OH. Recommendation: Finish early and get extra credit!

Operation Assignment Operators

- Consider the following assignment statements:

```
i = i + 10;
s = s + "!";
```

- Increasing a variable, concatenating to a variable, and so on, are so common that there are built-in shorthand operators:

Operator	Syntax	Example	Equivalent To
Addition Assignment	`+=`	`i += 10;`	`i = i + 10;`
Subtraction Assignment	`-=`	`i -= 10;`	`i = i - 10;`
Multiplication Assignment	`*=`	`i *= 10;`	`i = i * 10;`
Division Assignment	`/=`	`i /= 10;`	`i = i / 10;`
Remainder Assignment	`%=`	`i %= 10;`	`i = i % 10;`
Concatenation Assignment	`+=`	`s += "!!!";`	`s = s + "!!!";`

Challenge Question #0 - pollev.com/compunc

- What is the result of calling: **`michaelJackson(3)`**

```
let michaelJackson = (force: number): string => {
    let s = "";
    for (let i = 1; i < force; i++) {
        s += "h";
        for (let h = 0; h < i; h += 1) {
            s += "e";
        }
    }
    return s;
};
```

Notes on Nested Loops

- **General Rule:** When the closing curly brace of a loop is encountered, the loop jumps back to the start of **its matching condition**.

- An inner loop will jump back up to the inner loop's condition and an outer loop will jump back up to the outer loop's condition.

- Thus, an inner loop must complete all of its **iterations** for *each* single iteration of an outer loop.

Challenge Question #1

In the main function, what is **printed**?

```
export let main = async () => {
    let input = [1, 2, 2, 3];
    let result = bar(input, 2);
    print(result);
};

let none = (a: number[], n: number): boolean => {
    for (let i = 0; i < a.length; i++) {
        if (a[i] === n) {
            return false;
        }
    }
    return true;
};
```

CQ2: Trace an Environment Diagram for the Program Listing Below

```
1  import { print } from "introcs";
2
3  export let main = async () => {
4      let zeros = fillZeros(2);
5      print(zeros);
6  };
7
8  let fillZeros = (n: number): number[] => {
9      let a = [];
10     for (let i = 0; i < n; i++) {
11         a[i] = 0;
12     }
13     return a;
14 };
15
16 main();
```

CQ3: Trace an Environment Diagram for the Program Listing Below

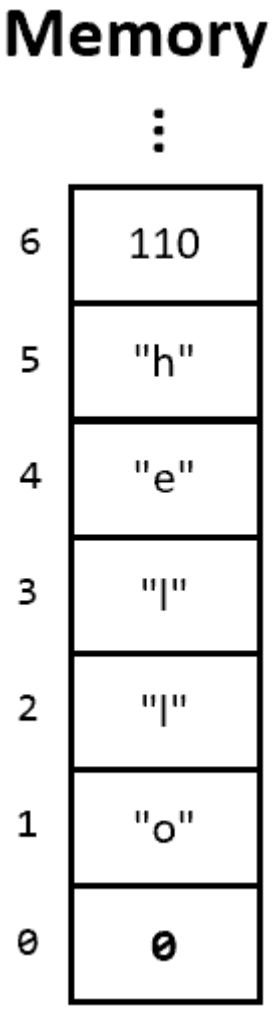
```
1  import { print } from "introcs";
2
3  export let main = async () => {
4      let tens = [10, 20, 30];
5      let b = clone(tens);
6      print(b);
7  };
8
9  let clone = (a: number[]): number[] => {
10     let c = [];
11     for (let i = 0; i < a.length; i++) {
12         c[i] = a[i];
13     }
14     return c;
15 };
16
17 main();
```

Challenge Question #4: What is printed?

```
let s = "abc";
print(s[1]);
print(s.length);
```

Strings are Arrays of Characters

- In the first video of the semester, your computer's memory was introduced with a diagram to the left.
- Notice cells 1-5 store individual characters... *not strings.*
- A string is an array of single characters underneath the hood.
  - We haven't needed to worry over this detail thanks to ***data abstraction!***
- We can "poke through" the abstraction!
  - Access individual characters with **`stringName[index]`**
  - Access the length of a string with **`stringName.length`**



string vs string[]

```
let a = ["1", "2", "3"];
let s = "123";
```

There is a very important difference between a string value and an array of single character strings:

A string's elements cannot be changed. An array's can be.

- You cannot reassign a character like `s[1] = "9";`
- You cannot append new characters to the end of it.
- More precisely, a string is an *immutable values* because its contents cannot change.
- When you concatenate two strings you are producing a third, new string value.

CQ5: Given the two code listings below, draw environment diagrams for each. Then respond on PollEv with whether b's value at line 7 is the same or different between examples. Finally, in the first example, what is b's value either way?

```
1  import { print } from "introcs";
2
3  export let main = async () => {
4      let a = ["hello"];
5      let b = a;
6      a = ["world"];
7      print(b);
8  };
9
10 main();
```

```
1  import { print } from "introcs";
2
3  export let main = async () => {
4      let a = ["hello"];
5      let b = a;
6      a[0] = "world";
7      print(b);
8  };
9
10 main();
```