

CSE 503

Introduction to Computer Science for Non-Majors

Dr. Eric Mikida

epmikida@buffalo.edu

208 Capen Hall

Day 09

Arrays and Loops in JavaScript

Announcements

- Autolab for PA #1 will be up by tonight...try submitting sooner rather than later.
- Solutions for Exercises from Lecture 08 will be posted by tonight

Recap

- Looked at selection statements in JavaScript
 - `if` and `else` keywords are the same as in python
 - No `elif` keyword...instead use `else if`
 - Boolean conditions delimited by ()
 - Bodies delimited by { }
- Two exercises in JavaScript

Exercise #1 from Lecture 08

Define a function in JavaScript named `totalCost`. This function takes two numerical arguments, a price and a quantity.

The price is the price of a single item, and quantity are the number of items being purchased. The function should return the total cost of these items.

Write some tests before defining your function.

To assert something in JavaScript, call the `console.assert(...)` function

Exercise #1 Solution

```
1. function totalCost(c, q) {  
2.   return c * q;  
3. }
```

Exercise #2 from Lecture 08

Now we want to extend our previous example to be able to handle if an item is on sale! Define a function named `saleCost` that takes 3 arguments: the price of an item, the quantity being purchased, and a Boolean value indicating if there is a sale.

If the third argument is `false`, then there is no sale, and you can return the cost just like we did in Exercise #1.

If the third argument is `true` if there is a "buy one get one free" sale, so you only have to pay for half of the items.

Write some tests...think about the different cases that could arise and what the answer should be before you code!

Exercise #2 Solution

```
1.  function saleCost(c, q, sale) {
2.      if (sale && q % 2 == 0) {
3.          return totalCost(c, q / 2);
4.      } else if (sale && q % 2 == 1) {
5.          return totalCost(c, (q + 1) / 2);
6.      } else {
7.          return totalCost(c, q);
8.      }
9.  }
```

Review

We've seen all of these constructs in Python and JavaScript, make sure you understand them.

Literal Values

Expressions

Operators (arithmetic, relational, boolean)

Variables

Assignment Statements

Return Statements

Function Definitions

Function Calls

Environments

Parameters

Arguments

Comments

Assert Statements/Function

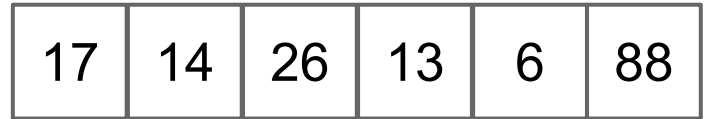
Selection Statements

Arrays

- Variables can only hold a single value
- Arrays are objects that can hold a list of many values



Conceptually, a variable can be thought of as a box that can hold one value.



An array can be thought of as many boxes in a row, where each one can hold a value.

Arrays in JavaScript

- One way to create an array is to simply list its variables
- In JavaScript, arrays are delimited with []

```
["Alicia", "Ariel", "Zach", "Cory", "Kate"]
```

Arrays in JavaScript

- How can we use an array? Just like with values, we can assign an array to a variable that we can use later

```
let names = ["Alicia", "Ariel", "Zach", "Cory", "Kate"];
```

Arrays in JavaScript

- How can we use an array? Just like with values, we can assign an array to a variable that we can use later

```
let names = ["Alicia", "Ariel", "Zach", "Cory", "Kate"];
```

...but wait...I thought variables could only hold one value

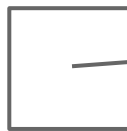
Arrays in JavaScript

- An Array is an *object* in JavaScript
- Objects have their own internal structure, as well as an address in memory

length	5
0	"Alicia"
1	"Ariel"
2	"Zach"
3	"Cory"
4	"Kate"

Arrays in JavaScript

- An Array is an *object* in JavaScript
- Objects have their own internal structure, as well as an address in memory
- What we actually assign to the variable is the address of the object.
 - So the variable is holding one value...an address!



names



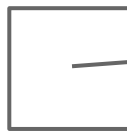
length	5
0	"Alicia"
1	"Ariel"
2	"Zach"
3	"Cory"
4	"Kate"

Arrays in JavaScript

- So *now* what can we do with an array?
 - Access individual values in an array by using []
 - Indexing in JavaScript starts at 0
 - Access the length of an array with .length

```
1. console.log(names[0]);  
2. console.log(names[3]);  
3. console.log(names.length);
```

What does each line print?



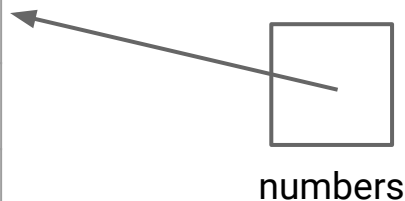
names

length	5
0	"Alicia"
1	"Ariel"
2	"Zach"
3	"Cory"
4	"Kate"

Arrays in JavaScript

- Each array is a separate object, with its own environment

length	6
0	17
1	14
2	26
3	13
4	6
5	88



length	5
0	"Alicia"
1	"Ariel"
2	"Zach"
3	"Cory"
4	"Kate"

Control Flow: Repetition

- What if we want to do something to every element of an array
 - ie: print them out, add them up, update their value, etc.
- We could write the same statement multiple times...but this is both tedious, and inflexible. What if the size of the array changes? What if we don't know the size?
- We need a way to repeat statements...

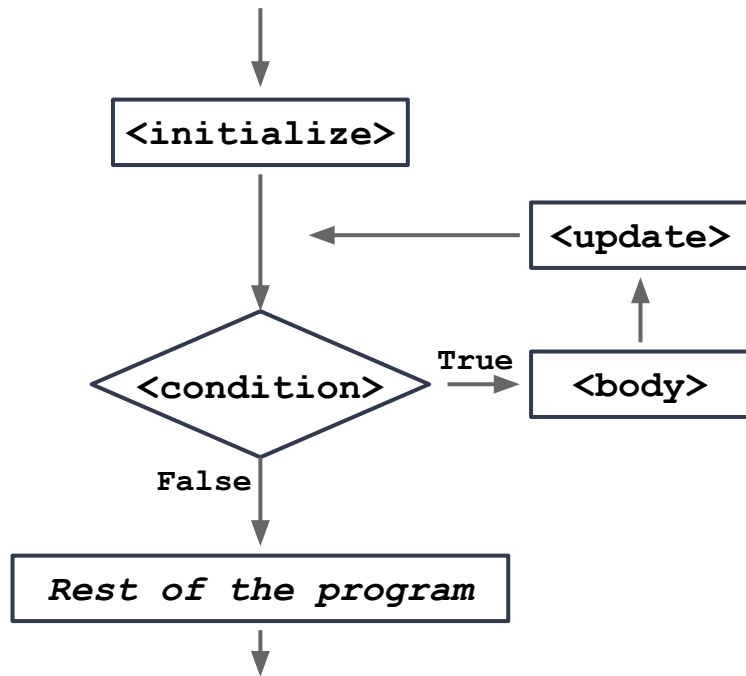
For Loops in JavaScript

- A for loop is a type of *loop* (a way to repeat some piece of a program multiple times)

```
for (<initialize>; <condition>; <update>) {  
    <body>  
}
```

For Loop Control Flow

```
for (<initialize>; <condition>; <update>) {  
    <body>  
}
```



For Loop Example

```
for (let i = 0; i < names.length; i = i + 1) {  
  console.log("Hello " + names[i]);  
}
```

