FLOW CONTROL & OBJECTS

With Arrays

Flow Control

- conditional statements
- loops

What's So Great About Flow Control?

What's So Great About Flow Control?

- Making Decisions!

Truthy vs. Falsey

- Falsey things:
  - false, 0,'', undefined, null
- Truthy things:
  - everything else

If Statements

```
var truthy = true;

if (truthy) {
    alert("definitely not false!");
}
```

if...else

```
var questionable = false;

if (questionable) {
    alert("I always tell the truth");
} else {
     alert("I lie!");
}
```

Alien Invasion: if...else if...else

```
var numberAliens = 13;

if (numberAliens < 4) {
    alert("We can take them");
} else if (numberAliens < 14) {
    alert("You know karate, right?");
} else {
    alert("RUN!");
}
```

Equality Operators

- Test for equality: ==
  - `if (4 == 4) { //do something }`
- Test for inequality: !=
  - `if (4 != 5) { //do something }`
- Test for greater than: >=
  - `if (5 >= 4) { //do something }`
- Test for less than: <=
  - `if (4 <= 5) { //do something }`

Logical Operators

- AND: &&
- Or: ||

```
if (3 == '3' || 14 == 4) {
    alert('sloths rule!');
} else {
    alert('sloths drool!');
}
```

Exercise Prep: Convert String to Number

```
var five = Number("5")
var six = Number("6")

console.log(five + six)
```

Exercise: Aliens & Sea Monsters

- Use the file materials/flow_control/exercises/aliens_sea_monsters.html
- Write a program in an html file that asks the hero (user) how much of the following they have:
  - ounces of alien poison
  - sea monster slaying knives
  - pounds of body armor
- Also ask their name
- If their name is Lancelot or Odysseus, then they will need half as many weapons & armor to survive. Otherwise, they will need at least 104 ounces of alien poison, 28 knives and 56 pounds of armor.
- Tell the hero whether they survive or not using jQuery to insert their survival status into the div#survival-status.
- Hints:
  - add text to the div like this `$('#survival-status').text("hi")`
  - convert strings to numbers like this: `var number = Number("5")`

Switch Statements

```
var vacationLocation = "Fiji";

switch (vacationLocation) {
    case "Fiji" :
        alert("I'm going to Fiji, baby!");
        break;
    case "Cancun" :
        alert("I'm going to Cancun, baby!");
        break;
}
```

While Loops

```
var purplePeopleEater = 4;

while (purplePeopleEater >= 0) {
    alert("CHomP, ChOMP, nomnom.");
    --purplePeopleEater;
}
```

Exercise: Duck, Duck, Moose!

- Use the file materials/flow_control/exercises/duck.html
- (People used to be surprised by Geese, but in the modern age we are jaded and need the extra pizazz of a moose to get us to run around in a circle.)
- Write a program that sets a variable to a random number between 1 and 10. Make a while loop that alerts "duck" and decrements the random number with each loop. After the while loop, alert "MOOSE!".
- Hint: var randNum = Math.floor(Math.random()*11)

do...while

- Similar to while loops but they ensure that the block of code gets executed at least once even if the conditional expression evaluates to false immediately.

```
var purplePeopleEater = -1;

do {
    alert("CHomP, ChOMP, nomnom.");
    --purplePeopleEater;
} while (purplePeopleEater >= 0)
```

For Loops

```
for (initial value; condition; update) {
  ...
}
```

Example

```
for (var i = 0; i < 3; i++) {
    alert("Pizza!");
}
```

Exercise: Conditionals & Loops

- Use the file materials/flow_control/exercises/sum_2_3.html
- Write a program that sums together all the numbers from 1 to 1000 that are multiples of 2 and 3 (both). Print the final result into the div#two-and-three.
  - Hint: use &&
- Write a program that sums together all the numbers from 1 to 1000 that are multiples of 2 or 3 (either). Print the final result into the div#two-or-three.
  - Hint: use ||
- One more hint: use modulus: %

```
console.log(4 % 1) // 0
console.log(4 % 2) // 0
console.log(4 % 3) // 1
console.log(4 % 4) // 0
```

Objects

What is an Object?

What is an Object?

- Objects are like nouns, people, places, things
  - Objects have methods which are like verbs
    - find, update, readFile, setDate
  - Objects have properties which are like adjectives
    - color = green, price = $5, startDate = 1/22/2016

Javascript Built in Objects

- Boolean
- Number
- String
- Math
- Regex
- Date
- Array

Javascript String Objects

- When we call String object methods on a string primitive, Javascript temporarily converts the primitive into a String object to perform the method call, then discards the temporary String object.

```
var stringPrimitive = 'hello';
var stringObject = new String('hello');
alert(stringPrimitive.length);
alert(stringObject.length);
```

- What are some issues with this?

Some String object methods:

- length()
- anchor()
- bold(), italics(), strike(), sub(), sup(), blink() (don't use these)
- charAt()
- indexOf()
- concat()
- split()
- slice()
- substring()
- toLowerCase(), toUpperCase()
- match(), replace(), search()

Javascript Strings are Immutable

- You can't change a string once it has been created. This is an important point to remember because when concatenating strings it requires Javascript to create entirely new objects.

Example:

```
var shoe = new String("Acme Cross Trainer - Men's");
alert(shoe.split(" ")[4]);
alert(shoe.substr(-5, 5));
alert(shoe.charAt(3));
alert(shoe.slice(3,4));
```

Javascript Boolean Objects

- While Strings and Numbers have their own methods & properties, Boolean objects inherit all of their methods and properties from the Object object and do not have any unique methods or properties.
- Boolean instance methods:
  - toString()
  - valueOf()

New Boolean Objects

- These evaluate to false:

```
alert(new Boolean());
alert(new Boolean('')); //empty string => false
alert(new Boolean(false));
alert(new Boolean(0));
```

- These evaluate to true:

```
alert(new Boolean(1));
alert(new Boolean('false')); //non-empty string => true
alert(new Boolean(true));
```

Javascript Number Objects

- Methods:
  - toString()
  - valueOf()
  - toExponential()
    - returns a string representing the number in exponential notation.
  - toFixed()
    - returns a string representing the number in fixed-point notation.
  - toPrecision()
    - returns a string representing the number using a specific precision.

Arrays

Making Arrays

```
var eggCarton = new Array('egg', 'egg', 'egg', null, null, null)
```

Accessing Arrays

```
eggCarton[0] // ‘egg’
eggCarton[1] // ‘egg’
eggCarton[2] // ‘egg’
eggCarton[3] // null
eggCarton[4] // null
eggCarton[5] // null
```

Array.length

```
>>> eggCarton.length
6
```

Array.push

```
>>> eggCarton.push('egg')
>>> eggCarton
["egg", "egg", "egg", null, null, null, "egg"]
```

Array.pop

```
>>> eggCarton.pop()
>>> eggCarton
["egg", "egg", "egg", null, null]
```

Array.unshift

```
>>> eggCarton.unshift(null)
>>> eggCarton
[null, "egg", "egg", "egg", null, null, null]
```

Array.shift

```
>>> eggCarton.shift()
>>> eggCarton
["egg", "egg", null, null, null]
```

Multidimensional Arrays

```
var multiply = new Array()
multiply[0] = new Array(0,0,0,0)
multiply[1] = new Array(0,1,2,3)
multiply[2] = new Array(0,2,4,6)
multiply[3] = new Array(0,3,6,9)

console.log(multiply[3][1])
console.log(multiply[3][2])
console.log(multiply[2][3])
console.log(multiply[3][3])
```

Exercise

- About Arrays in Javascript koans

Homework Exercise: Bubble Gum Game

- Write a game that determines a random number representing how many breaths it is going to take to make a bubble gum bubble pop (all bubbles should pop after 20 breaths). It should prompt the user for how many breaths they want to blow over and over again until the bubble pops. If the bubble pops with the exact number that it would have taken, then they win, if they go over, they lose. Determine how many points they get by adding the number of breaths they went over to the number of turns they had and subtracting it from 20.
- Use jQuery to keep the user updated about how many breaths they have used and once they win, congratulate them! Get creative with jQuery and make it as visually interesting as you can.
- Once the bubble pops, ask the user if they want to play another game. Keep the score for each game they play in an array. Once they have played more than one game, start displaying their average score on the page.
- Feel free to add any other fun features or change the features listed above to make it more interesting!

Extra Credit Homework:

- Make a scheduling system.
  - Create an array called schedule
  - Each index of schedule represents a day of the week. 0 is Sunday, 1 is Monday, etc. Indexes should point to an array containing the names of employees scheduled for that day.
  - Employees:
    - "Sally"
    - "Todd"
    - "Kim"
    - "Joe"
  - Joe and Kim work on weekends
  - Sally and Todd work on Tuesday and Thursday
  - Sally and Kim work on Monday, Wednesday and Friday

Homework

- Read Chapters:
  - 3 (Operators & Statements)
  - 4 (Objects)