CSCI 341 : HANDS ON

UNIT: Randomness

Activities

- Learn about and practice with **Objects**, Properties, and Methods
- Explore the **math object** and **random number generation**
- Read and complete the assignment
- Check your work!

Activity One

You can think of Objects in JavaScript as representing real life objects, with properties as characteristics of the object, and methods as actions that the object can perform. In real life, a die is an object. It has properties such as number of sides, facing value, and color. It has methods such has rolling, displaying a face value when you look at it, and being thrown at a craps board.

Object	Properties	Methods
	dice.sides = 6; dice.facingSide = 6;	dice.roll(); dice.displayFace();

Below is an example of an object with two properties (intSides and intFacingSide).

```
var dice = {
  //Properties
  intSides : 6,
  intFacingSide : 6
};
```

Below is an example of an object with two properties (intSides and intFacingSide) and two functions (roll and displayFace).

```
var dice = {
  //Properties
  intSides : 6,
  intFacingSide : 6,

  /********
  *Description: This function rolls the die.
  *Parameters: none
  *Return: none
  ********/
  roll : function() {
   alert("Rolling dice.");
   this.intFacingSide = Math.floor(this.intSides * Math.random()) + 1;
```

```
  }, // end roll

  /********
  *Description: This function writes the number of upward facing dots on the die to the page.
  *Parameters: none
  *Return: none
  ********/
  displayFace : function() {
   alert("Displaying Face.");
   //write your own code here
  } // end displayFace
}; // end dice object
```

Outside of an object, you can call its methods and set its properties by typing in the object's name then a dot followed by the method/property as such seen below:

```
dice.roll();   //This calls the roll method.
alert(dice.facingSide); //This alert is displaying the value of the facing side property.
```

Inside of an object's methods, you can refer to the object by using the keyword **this** instead of the object's name. Below is an example of how **this** can be used inside of an object's method.

```
this.facingSide = 6;
```

For more object examples, see pages 100 - 116 of the textbook.

Activity Two

Because users like variety in their software applications, all programming languages support the introduction of variability. Typically, this takes the form of a random number generator... which produces something pretty close to a random number.

A produced number is close to random, but in a format that is not ready for prime time - it's usually a real number between 0 and 1. To make the number presentable, we'll issue several built in commands.

First, Math.random() generates a real number between 0 and 1.
Second, the generated number is multiplied by 6 to constrain the final number to be between 1 and 6.
Third, Math.floor() would round down the produced number to the nearest integer.

Here's sample Javascript code for a dice rolling game:

```
intFacingSide = Math.floor( Math.random() * 6 ) + 1;
```

For more function examples, see pages 134 - 135 of the textbook.

Assignment

The captain has a bottle o' grog to award to the crew mate who can guess his secret number. So, your job is to use constructor notation to create a Pirate object with the properties of rank, favorite number, and html tag id. The object should also have methods that allow the Pirate to choose a random number and speak a phrase. Here, "speaking" means to send the phrase to that pirate's output area. Just to give the crew a fighting chance, the captain should choose an integer between 1 and 10.

Use your Pirate object to instantiate a captain and at least three ship mates. Hmmm... This sounds like a good time to use **constructor notation**!

1. The captain should use his speak() method to announce the guessing game to the crew.
2. Next, the captian will choose a secret number but not tell anyone!
3. Each crew mate will choose a number and then use his speak() method to output his name, rank, and numeric guess to his own html element. For example, "Sailor Mast Hugger chooses 4. Argh."
4. After the crew members tell their guesses, the captain uses his speak() method to tell his number.
5. Any crew member who guesses the captain's secret number wins the prize! If nobody guesses the secret number, the captain gets to keep his bottle o' grog for himself.
6. Bonus points will be awarded for the captain to announcing the winner of the contest.

- Helpful Hint #1: Consider allowing the .speak() method to accept a phrase as a parameter so each pirate can say more than one phrase.
- Helpful Hint #2: Choosing a number is an essential function of each pirate, so the pirate's favorite number should NOT be passed as a parameter into the constructor function. A method should choose the number and assign it to the secretNumber property.

Important Procedures for All Labs

Here are some general notes for perfection that you should follow for every assignment:

1. Please produce all web content to HTML5 standards.
2. Please **validate** all your files.
3. Be sure to update the header block comments for each file.
4. Be sure to check your browser's console / developer tools for error free code.
5. Test your code in Chrome and Edge at a minimum.
6. Use only your own original code for all labs.
7. Be sure to put your CSS and JavaScript in a separate files from your html.
8. Be sure to read through the lab rubric in Canvas.
9. Submit your lab in Canvas for grading.

Holler if you have any questions!

Mission Accomplished!

Argh, great job. Your dice be roll'n like the sea! Now pack it up old salt. Another assignment is coming in with the tide.