Wrap Up

Topics
- Core vs. Object-Oriented JavaScript
- split() function
- GetValidNumberInput() function
- Exam Review
- SEQ’s – need a volunteer

Core vs. Object-Oriented

- The JavaScript language is divided into two basic components, the core and the object-oriented parts of the language.
- The core part of the language consists of things such as loops, control flow constructs, functions, variables and is derived from C.

Object-Oriented JavaScript

- The object-oriented part consists of complex objects such as windows, frames, documents and buttons, each of which has properties, methods, and events associated with them.
  - A property is a characteristic of an object.
  - A method is an action carried out by the object.
  - An event is an object generated as a result of user interaction with object.

Core vs. OO Example

```html
<head>
<title>Example</title>
<script type="text/javascript">
    var name = "Patti";
    function printName(num){
        for(i = 1; i < num; i++) {
            document.write("Hi");
        }
    }
</script>
</head>
<body>
<script type="text/javascript">
    printName(5);
</script>
</body>
```

Split Method for String Objects

```javascript
var name = "John Edward Doe";
var names = name.split(" ");
for(var i = 0; i < names.length; i++) {
    document.write(names[i] + "
");
}
```

Length Property of Array Objects

```javascript
var names = prompt("Enter full name");
var fullName = names.split(" ");
var namesLength = fullName.length;
```
onClick Event of Button Objects

```html
<!DOCTYPE html>
<html><head>
<title>Example</title>
<script type="text/javascript">
  // function pressed(){
  //   alert("You have pressed the button!");
  // }
  </script>
</head>
<body>
<button type="button" onclick="pressed()">Click me!</button>
</body>
</html>
```

GetValidNumberInput()

```javascript
/**
 * GetValidNumberInput – This function prompts user for an integer:
 * ** Inputs: promptString - message to be displayed to user
 *           lowerNum - lowest possible value for integer returned
 *           upperNum - highest possible value for integer returned
 * ** Output: an integer
 */
function GetValidNumberInput(promptString, lowerNum, upperNum) {
  var num = parseInt(prompt(promptString));
  while(!isNaN(num) && num > lowerNum && num < upperNum) {
    alert(num + " is not an integer between " + lowerNum + " and " + upperNum);
    num = parseInt(prompt(promptString));
  }
  return num;
}
```