

## Wrap Up

### Topics

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

1

## Core vs. Object-Oriented

- The JavaScript language is divided in 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.

2

## 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** associate 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.

3

## Core vs. OO Example

```
<head><title> Example</title>
<script type="text/javascript">
  <!--
  var name = "Patti";
  function printName(num) {
    for (i = 0; i < num; i++) {
      document.write("<br>" + name);
    }
  }
  //-->
</script>
</head>
<body>
  <script type="text/javascript">
  <!--
  printName(5);
  //-->
</script>
</body>
```

Annotations:

- Core JavaScript (global variable @) points to `var name = "Patti";`
- Core JavaScript points to `function printName(num) {`
- Object-Oriented JavaScript -> write method of document object points to `document.write("<br>" + name);`
- Core JavaScript points to `printName(5);`

4

## Split Method for String Objects

```
*****
** split - divides the string object calling the method on a
delimitter
** Inputs: can vary, in this example parameter is delimiter used
to break up the string
** Outputs: an array of string objects
*****
var fullname = "John Edward Doe";
var names = fullname.split("d");
document.write(names[0] + "<br/>"); // prints "John E"
document.write(names[1] + "<br/>"); // prints "war"
document.write(names[2] + "<br/>"); // prints " Doe"
```

Annotations:

- Object-Oriented JavaScript => split method of string object returns array object points to `var names = fullname.split("d");`

5

## Length Property of Array Objects

```
<head><title> Example</title>
<script type="text/javascript">
  <!--
  function printArray(array) {
    for (i = 0; i < array.length; i++) {
      document.write("<br>" + array[i]);
    }
  }
  //-->
</script>
</head>
<body>
  <script type="text/javascript">
  <!--
  var fullname = prompt("Enter full name");
  names = fullname.split(" ");
  printArray(names);
  //-->
</script>
</body>
```

Annotations:

- Object-Oriented JavaScript => length property of array object points to `for (i = 0; i < array.length; i++) {`
- Core JavaScript points to `var fullname = prompt("Enter full name");`
- Core JavaScript points to `printArray(names);`

6

## onClick Event of Button Objects

```
<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>
```

Core JavaScript

Core JavaScript  
embedded in HTML tag

7

## GetValidNumberInput()

```
/**
 * 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;
}
```

8