JavaScript Functions and Objects

Phil Colbert
University of Oregon
Functions

• In Mathematics, a function is an expression which takes one or more arguments as input and outputs exactly one result:
  – \( f(x) = 2 \times x \)
  – \( g(y, z) = y + z + 2 \)

• In Computer Science, a function is an expression which takes zero or more arguments as input and outputs zero or one result.
  – Functions which do not output a result are sometimes called procedures.
  – So far, we’ve been using procedures to handle events.
• function dogsAge(age) {
    return age*7;
}
alert(dogsAge(2));
Functions

• function dogsAge(age) {
  return age*7;
}

alert(dogsAge(2));
  – Prints 14

  • The variable age (the parameter) is automatically assigned to the value 2 (the argument).
  • The value 2*7 is returned from the function, so dogsAge(2) results in 14, which is printed.
Functions

• function rectangleArea(width, height) {
   var area = width*height;
   return area;
}

alert(rectangleArea(10, 11));
Functions

• function rectangleArea(width, height) {
  var area = width*height;
  return area;
}

alert(rectangleArea(10, 11));
  – 10*11 is 110, so this prints 110.
Functions

• function addUp(numArray) {
    var total = 0;
    for (var i = 0; i < numArray.length; i++) {
        total += numArray[i];
    }
    return total;
}

alert(addUp([1, 2, 3]));
Functions

- function addUp(numArray) {
  var total = 0;
  for (var i = 0; i < numArray.length; i++) {
    total += numArray[i];
  }
  return total;
}

alert(addUp([1, 2, 3]));
- $1 + 2 + 3 = 6$, so this prints 6.
Functions

• function getAvatar(points) {
    var avatar;
    if (points < 100) {
        avatar = "Mouse";
    }
    else if (points > 100 && points < 1000) {
        avatar = "Cat";
    }
    else {
        avatar = "Ape";
    }
    return avatar;
}

alert(getAvatar(500));
Functions

• function getAvatar(points) {
    var avatar;
    if (points < 100) {
        avatar = "Mouse";
    }
    else if (points > 100 && points < 1000) {
        avatar = "Cat";
    }
    else {
        avatar = "Ape";
    }
    return avatar;
}

alert(getAvatar(500));

- 500 is > 100 and 500 is < 1000, so this prints “Cat”.
Random Revisited

<!doctype html>
<html lang="en">
<head>
  <title>Random Song</title>
  <meta charset="utf-8">
  <script>
    var songs = ["Jeremy", "Even Flow", "Once"];

    function suggestSong() {
      var index = Math.floor(Math.random()*songs.length);
      var suggestion = songs[index];
      return suggestion;
    }

    var song = suggestSong();
    alert(song);
  </script>
</head>
<body>
</body>
</html>
Random Revisited

<script>
    var songs = ['Jeremy', 'Even Flow', 'Once'];

    function suggestSong() {
        var index = Math.floor(Math.random() * songs.length);
        var suggestion = songs[index];
        return suggestion;
    }

    var song = suggestSong();
    alert(song);
</script>
Random Revisited

```html
<script>
    var songs = ["Jeremy", "Even Flow", "Once"];

    function suggestSong() {
        var index = Math.floor(Math.random() * songs.length);
        var suggestion = songs[index];
        return suggestion;
    }

    var song = suggestSong();
    alert(song);
</script>
```
Random Revisited

• var songs = ["Jeremy", "Even Flow", "Once"]; 
  – Create a variable “songs” assigned to an array containing three song names.
Random Revisited

<script>
    var songs = ["Jeremy", "Even Flow", "Once"];

    function suggestSong() {
        var index = Math.floor(Math.random() * songs.length);
        var suggestion = songs[index];
        return suggestion;
    }

    var song = suggestSong();
    alert(song);
</script>
Random Revisited

- function suggestSong() {
  var index = Math.floor(Math.random()*songs.length);
  var suggestion = songs[index];
  return suggestion;
}

- Defined a function “suggestSong” which takes no arguments as input and returns a random song from the “songs” array.
Random Revisited

<script>
  var songs = ["Jeremy", "Even Flow", "Once"];  

  function suggestSong() {
    var index = Math.floor(Math.random() * songs.length);
    var suggestion = songs[index];
    return suggestion;
  }

  var song = suggestSong();
  alert(song);
</script>
Random Revisited

- var song = suggestSong();
  alert(song);
  
  - Create a variable “song” which is assigned to the random song name returned by suggestSong() and then print the value of the “song” variable.
Random Revisited

<script>
    var songs = ["Jeremy", "Even Flow", "Once"];

    function suggestSong() {
        var index = Math.floor(Math.random()*songs.length);
        var suggestion = songs[index];
        return suggestion;
    }

    var song = suggestSong();
    alert(song);
</script>
Objects

• Programming languages provide arrays because it is useful (and sometimes necessary) to assign multiple items to a single variable.

• Objects allow us to assign multiple named items to a single variable:
  – var myObject = {apple:"good", orange:"bad"};
    alert(myObject.apple);
  • Prints “good”.

• HTML elements are represented as objects in JavaScript, which is why we modify their properties similarly.
Objects

• Can also create an empty object and then add properties:
  – var myObject = {};
    myObject.apple = "good";
    myObject.orange = "bad";
    alert(myObject.apple);
  • Prints “good”.
<!doctype html>
<html lang="en">
<head>
<title>Random Song</title>
<meta charset="utf-8">
<script>
var songs = new Array();
songs.push({title:"Jeremy", artist:"Pearl Jam");
songs.push({title:"Even Flow", artist:"Pearl Jam");
songs.push({title:"Pompeii", artist:"Bastille");

function suggestSong() {
    var index = Math.floor(Math.random() * songs.length);
    var suggestion = songs[index];
    return suggestion;
}

var song = suggestSong();
alert("song: " + song.title + " artist: " + song.artist);

</script>
</head>
<body>
</body>
</html>
Random Objects

<script>
    var songs = new Array();
songs.push({title:"Jeremy", artist:"Pearl Jam"});
songs.push({title:"Even Flow", artist:"Pearl Jam"});
songs.push({title:"Pompeii", artist:"Bastille"});

    function suggestSong() {
        var index = Math.floor(Math.random()*songs.length);
        var suggestion = songs[index];
        return suggestion;
    }

    var song = suggestSong();
    alert("song: " + song.title + " artist: " + song.artist);
</script>
Random Objects

<script>

var songs = new Array();
songs.push({title:"Jeremy", artist:"Pearl Jam");
songs.push({title:"Even Flow", artist:"Pearl Jam");
songs.push({title:"Pompeii", artist:"Bastille");

function suggestSong() {
    var index = Math.floor(Math.random()*songs.length);
    var suggestion = songs[index];
    return suggestion;
}

var song = suggestSong();
alert("song: " + song.title + " artist: " + song.artist);

</script>
Random Objects

• var songs = new Array();
songs.push({title:"Jeremy", artist:"Pearl Jam"});
songs.push({title:"Even Flow", artist:"Pearl Jam"});
songs.push({title:"Pompeii", artist:"Bastille"});
  – Create a new empty array and assign it to the variable “songs”.
    • This is equivalent to: var songs = [];
  – Add three items to the array, all objects with “title” and “artist” properties and various values for those properties.
Random Objects

<script>
  var songs = new Array();
  songs.push({title:"Jeremy", artist:"Pearl Jam"});
  songs.push({title:"Even Flow", artist:"Pearl Jam"});
  songs.push({title:"Pompeii", artist:"Bastille"});

  function suggestSong() {
    var index = Math.floor(Math.random()*songs.length);
    var suggestion = songs[index];
    return suggestion;
  }

  var song = suggestSong();
  alert("song: " + song.title + " artist: " + song.artist);
</script>
Random Objects

• var song = suggestSong();
  alert("song: " + song.title + " artist: " + song.artist);
  – Prints the randomly-selected song’s title and artist properties along with some literal text to identify those properties when they’re printed.
Random Objects

<script>
    var songs = new Array();
    songs.push({title:"Jeremy", artist:"Pearl Jam");
    songs.push({title:"Even Flow", artist:"Pearl Jam");
    songs.push({title:"Pompeii", artist:"Bastille");

    function suggestSong() {
        var index = Math.floor(Math.random()*songs.length);
        var suggestion = songs[index];
        return suggestion;
    }

    var song = suggestSong();
    alert("song: " + song.title + " artist: " + song.artist);
</script>
Classes

• Classes provide templates for objects so that you don’t need to specify the property names repeatedly.

• In JavaScript, classes are specified by defining a function which adds properties to itself:
  – function MyObject/appleFeelings, orangeFeelings) {
      this.apple = appleFeelings;
      this.orange = orangeFeelings;
    }
    var myObject = new MyObject ("good", "bad");
    alert(myObject.apple);
  • Prints “good”.

• In JavaScript, classes are specified by defining a function which adds properties to itself:
function Song(songTitle, songArtist) {
    this.title = songTitle;
    this.artist = songArtist;
}

var songs = new Array();
songs.push(new Song("Jeremy", "Pearl Jam"));
songs.push(new Song("Even Flow", "Pearl Jam"));
songs.push(new Song("Pompeii", "Bastille"));

function suggestSong() {
    var index = Math.floor(Math.random() * songs.length);
    var suggestion = songs[index];
    return suggestion;
}

var song = suggestSong();
alert("song: " + song.title + " artist: " + song.artist);
function Song(songTitle, songArtist) {
    this.title = songTitle;
    this.artist = songArtist;
}

var songs = new Array();
songs.push(new Song("Jeremy", "Pearl Jam");
songs.push(new Song("Even Flow", "Pearl Jam");
songs.push(new Song("Pompeii", "Bastille");

function suggestSong() {
    var index = Math.floor(Math.random() * songs.length);
    var suggestion = songs[index];
    return suggestion;
}

var song = suggestSong();
alert("song: " + song.title + " artist: " + song.artist);
</script>
function Song(songTitle, songArtist) {
    this.title = songTitle;
    this.artist = songArtist;
}

var songs = new Array();
songs.push(new Song("Jeremy", "Pearl Jam").
songs.push(new Song("Even Flow", "Pearl Jam");
songs.push(new Song("Pompeii", "Bastille");

function suggestSong() {
    var index = Math.floor(Math.random() * songs.length);
    var suggestion = songs[index];
    return suggestion;
}

var song = suggestSong();
alert("song: " + song.title + " artist: " + song.artist);
Classes

• function Song(songTitle, songArtist) {
  this.title = songTitle;
  this.artist = songArtist;
}

  – Uses the “this” keyword to add “title” and “artist” properties to the Song function (which is an object itself).
Classes

<script>
function Song(songTitle, songArtist) {
    this.title = songTitle;
    this.artist = songArtist;
}

var songs = new Array();
songs.push(new Song("Jeremy", "Pearl Jam"));
songs.push(new Song("Even Flow", "Pearl Jam"));
songs.push(new Song("Pompeii", "Bastille"));

function suggestSong() {
    var index = Math.floor(Math.random() * songs.length);
    var suggestion = songs[index];
    return suggestion;
}

var song = suggestSong();
alert("song: " + song.title + " artist: " + song.artist);
</script>
Classes

• var songs = new Array();
songs.push(new Song("Jeremy", "Pearl Jam"));
songs.push(new Song("Even Flow", "Pearl Jam"));
songs.push(new Song("Pompeii", "Bastille"));
   - Creates a new “songs” array and adds three instances of the “Song” object with various arguments.
<script>
    function Song(songTitle, songArtist) {
        this.title  = songTitle;
        this.artist = songArtist;
    }

    var songs = new Array();
songs.push(new Song("Jeremy", "Pearl Jam"));
songs.push(new Song("Even Flow", "Pearl Jam"));
songs.push(new Song("Pompeii", "Bastille"));

    function suggestSong() {
        var index = Math.floor(Math.random()*songs.length);
        var suggestion = songs[index];
        return suggestion;
    }

    var song = suggestSong();
    alert("song: " + song.title + " artist: " + song.artist);
</script>
Methods

• We can also add functions to classes, which are then known as methods:
  – function MyObject(appleFeelings, orangeFeelings) {
    this.apple  = appleFeelings;
    this.orange = orangeFeelings;

    this.getAppleText = function() {
      return "apple: " + this.apple;
    }
  }
  
  var myObject = new MyObject ("good", "bad");
  alert(myObject.getAppleText());
  • Prints “apple: good”.
<html lang="en">
<head>
    <title>Random Song</title>
    <meta charset="utf-8">
    <script>
        function Song(songTitle, songArtist) {
            this.title = songTitle;
            this.artist = songArtist;

            this.getText = function() { return "song: " + this.title + " artist: " + this.artist; }
        }

        var songs = new Array();
        songs.push(new Song("Jeremy", "Pearl Jam"));
        songs.push(new Song("Even Flow", "Pearl Jam"));
        songs.push(new Song("Pompeii", "Bastille"));

        function suggestSong() {
            var index = Math.floor(Math.random()*songs.length);
            var suggestion = songs[index];
            return suggestion;
        }

        var song = suggestSong();
        alert(song.getText());
    </script>
</head>
<body>
</body>
</html>
Methods

<script>
  function Song(songTitle, songArtist) {
    this.title = songTitle;
    this.artist = songArtist;

    this.getText = function() { return "song: " + this.title + " artist: " + this.artist; }
  }

  var songs = new Array();
  songs.push(new Song("Jeremy", "Pearl Jam"));
  songs.push(new Song("Even Flow", "Pearl Jam"));
  songs.push(new Song("Pompeii", "Bastille"));

  function suggestSong() {
    var index = Math.floor(Math.random()*songs.length);
    var suggestion = songs[index];
    return suggestion;
  }

  var song = suggestSong();
  alert(song.getText());
</script>
Methods

<script>
    function Song(songTitle, songArtist) {
        this.title = songTitle;
        this.artist = songArtist;

        this.getText = function() { return "song: " + this.title + " artist: " + this.artist; }
    }

    var songs = new Array();
    songs.push(new Song("Jeremy", "Pearl Jam"));
    songs.push(new Song("Even Flow", "Pearl Jam"));
    songs.push(new Song("Pompeii", "Bastille"));

    function suggestSong() {
        var index = Math.floor(Math.random()*songs.length);
        var suggestion = songs[index];
        return suggestion;
    }

    var song = suggestSong();
    alert(song.getText());
</script>
• function Song(songTitle, songArtist) {
    this.title = songTitle;
    this.artist = songArtist;

    this.getText = function() {
        return "song: " + this.title + " artist: " + this.artist;
    }
}

– Adds a getText() method to the class which returns the title and artist of the Song instance along with some literal text for formatting.
Methods

<script>

function Song(songTitle, songArtist) {
    this.title = songTitle;
    this.artist = songArtist;

    this.getText = function() { return "song: " + this.title + " artist: " + this.artist; }
}

var songs = new Array();
songs.push(new Song("Jeremy", "Pearl Jam"));
songs.push(new Song("Even Flow", "Pearl Jam"));
songs.push(new Song("Pompeii", "Bastille"));

function suggestSong() {
    var index = Math.floor(Math.random() * songs.length);
    var suggestion = songs[index];
    return suggestion;
}

var song = suggestSong();
alert(song.getText());

</script>
Methods

• var song = suggestSong();
  alert(song.getText());
  – Calls the getText() method of the Song instance returned by suggestSong() to print the song information.
Methods

<script>
    function Song(songTitle, songArtist) {
        this.title = songTitle;
        this.artist = songArtist;

        this.getText = function() { return "song: " + this.title + " artist: " + this.artist; }
    }

    var songs = new Array();
    songs.push(new Song("Jeremy", "Pearl Jam");
    songs.push(new Song("Even Flow", "Pearl Jam");
    songs.push(new Song("Pompeii", "Bastille");

    function suggestSong() {
        var index = Math.floor(Math.random() * songs.length);
        var suggestion = songs[index];
        return suggestion;
    }

    var song = suggestSong();
    alert(song.getText());
</script>