Chapter 1

***Listing 1-1.*** Referencing an External Script

<!DOCTYPE html>

<html>

<head>

<title>JavaScript Primer</title>

</head>

<body>

<!-- reference the myScript.js script file -->

<script src="scripts/myScript.js"></script>

</body>

</html>

***Listing 1-2.*** Using an Inline Script

<!DOCTYPE html>

<html>

<head>

<title>JavaScript Primer</title>

</head>

<body>

<!-- an inline script -->

<script>console.log("Hello");</script>

</body>

</html>

***Listing 1-3.*** Statement Execution

<!DOCTYPE html>

<html>

<head >

<title>JavaScript Primer</title>

<script>

console.log("I am a statement"); console.log("I am also a statement");

</script>

</head>

<body>

<script>

console.log("Here is another statement"); console.log("Here is the last statement");

</script>

</body>

</html>

***Listing 1-4.*** No Semicolons—All Good

...

<script>

console.log("Here is a statement") console.log("Here is the last statement") </script> ...

***Listing 1-5.*** Both Statements on the Same Line—NOT Good

<script>

console.log("Here is a statement") console.log("Here is the last statement");

</script>

***Listing 1-6.*** Both Statements on the Same Line—All Good

<script>

console.log("Here is a statement"); console.log("Here is the last statement");

</script>

***Listing 1-7.*** Using Comments

<!DOCTYPE html>

<html>

<head >

<title>JavaScript Primer</title>

<script>

// The lines in this script block execute first

console.log("I am a statement"); console.log("I am also a statement");

</script>

</head>

<body>

<script>

***Listing 1-8.*** A Simple Function

<!DOCTYPE html>

<html>

<head>

<title>JavaScript Primer</title>

<script>

function mySimpleFunction() { console.log("Hello");

}

mySimpleFunction(); mySimpleFunction();

</script>

</head>

<body>

</body>

</html>

***Listing 1-9.*** A Function with Arguments and a Return Value

<!DOCTYPE html>

<html>

<head>

<title>JavaScript Primer</title>

<script>

function tripler(numberToTriple) {

return 3 \* numberToTriple;

}

console.log(tripler(150)); console.log(tripler(300));

</script>

</head>

<body>

</body>

</html>

***Listing 1-10.*** Declaring Multiple Variables at Once

<!DOCTYPE html>

<html>

<head>

<title>JavaScript Primer</title>

</head>

<body>

<script>

var color = "red";

console.log("The color is " + color);

</script>

</body>

</html>

***Listing 1-11.*** Declaring Multiple Variables at Once

<!DOCTYPE html>

<html>

<head>

<title>JavaScript Primer</title>

</head>

<body>

<script>

// declare some variables var color, size, shape;

// assign values to them color = 'blue'; size = 'large'; shape = 'circular';

console.log("Your widget is the color " + color + " and its size is " + size + ". It is " + shape + " in shape.");

</script>

</body>

</html>

***Listing 1-12.*** Numbers in JavaScript

<!DOCTYPE html>

<html>

<head>

<title>JavaScript Primer</title> </head>

<body>

<script>

var val1 = 22; var val2 = 23;

console.log(val1 + val2);

var val3= 22.5; var val4 = 23.5; console.log(val3 + val4);

var val5= 50; var val6 = .6;

console.log(val5 + val6);

// watch out! var val7= 25; var val8 = "25"; console.log(val7 + val8);

</script>

</body>

</html>

***Listing 1-13.*** Common Operators in Action

<!DOCTYPE html>

<html>

<head>

<title>JavaScript Primer</title>

</head>

<body>

<script>

console.log("Doing assignment");

var myName = "Catie"; console.log(myName);

console.log("Doing arithmetic"); console.log(5 + 5); // 10 console.log(5 - 5); // 0 console.log(5 \* 5); // 25 console.log(5 / 5); // 1 console.log(5 % 5); // 0 console.log(11 % 10); // 1

console.log("Doing comparisons"); console.log(11 > 10); // true console.log(11 < 10); // false console.log(10 >= 10); // true console.log(11 <= 10); // false

console.log("Doing string concatenation"); console.log(myName + " Grant"); // "Catie Grant"

console.log("Doing boolean logic"); console.log(true && true); // true console.log(true && false); // false console.log(true || true); // true console.log(true || false); // true

</script>

</body>

</html>

***Listing 1-14.*** Converting Types and Then Comparing

<!DOCTYPE html>

<html>

<head>

<title>JavaScript Primer</title>

</head>

<body>

<script>

var valueOne = 3; var valueTwo = "3"; if (valueOne == valueTwo) {

console.log("ValueOne and ValueTwo are the same");

} else {

console.log("ValueOne and ValueTwo are NOT the same");

}

</script>

</body>

</html>

***Listing 1-15.*** Type Conversion Examples

<!DOCTYPE html>

<html>

<head>

<title>JavaScript Primer</title>

</head>

<body>

<script>

// create a string variable var myStringType = "22";

// use the handy typeof operator to

// confirm the type is indeed string console.log(typeof myStringType );

// create a number variable var myNumberType = 45;

// use the handy typeof operator to

// confirm the type is indeed number console.log(typeof myNumberType );

// Now let's convert myStringType to

// a number type using the parseInt()

// method

var myStringType = parseInt(myStringType);

// confirm the type is indeed number console.log(typeof myStringType );

// finally, let's convert the myNumberType

// to a string

var myNumberType = myNumberType.toString();

// confirm the type is indeed string console.log(typeof myNumberType );

</script>

</body>

</html>

***Listing 1-16.*** Pre- vs. Post-Increment Behavior

<!DOCTYPE html>

<html>

<head >

<title>JavaScript Primer</title>

</head>

<body>

<script>

// Pre-increment var myNumber = 1; myNumber = myNumber + 1; myNumber = ++myNumber;

console.log("Pre-increment result is " + myNumber);

// Post-increment var myOtherNumber = 1; myOtherNumber = myOtherNumber + 1; myOtherNumber = myOtherNumber++;

console.log("Post increment result is " + myOtherNumber);

</script>

</body>

</html>

***Listing 1-17.*** Creating Objects

<!DOCTYPE html>

<html>

<head>

<title>JavaScript Primer</title>

</head>

<body>

<script>

// Example 1 var myFirstObject = {}; myFirstObject.firstName = "Andrew"; myFirstObject.lastName = "Grant"; console.log(myFirstObject.firstName);

// Example 2 var mySecondObject = { firstName: "Andrew", lastName: "Grant"

};

console.log(mySecondObject.firstName);

// Example 3

var myThirdObject = new Object(); myThirdObject.firstName = "Andrew"; myThirdObject.lastName = "Grant"; console.log(myThirdObject.firstName);

</script>

***Listing 1-18.*** Accessing and Changing Object Values

<!DOCTYPE html>

<html>

<head>

<title>JavaScript Primer</title>

</head>

<body>

<script>

var myFirstObject = {}; myFirstObject.firstName = "Andrew"; console.log(myFirstObject.firstName);

myFirstObject.firstName = "Monica"; console.log(myFirstObject.firstName);

myFirstObject["firstName"] = "Catie"; console.log(myFirstObject["firstName"]);

</script>

</body>

</html>

***Listing 1-19.*** Associative Array

<!DOCTYPE html>

<html>

<head>

<title>JavaScript Primer</title>

</head>

<body>

<script>

var myFirstObject = {};

myFirstObject["firstName"] = "Catie";

console.log(myFirstObject["firstName"]);

// Here we use a variable to determine which

// property we are accessing var propertyName = "firstName";

myFirstObject[propertyName] = "Christopher"; console.log(myFirstObject["firstName"]);

</script>

</body>

</html>

***Listing 1-20.*** An Object with a Method

<!DOCTYPE html>

<html>

<head>

<title>JavaScript Primer</title>

</head>

<body>

<script>

var myCleverObject = {

firstName: "Andrew",

age: 21,

myInfo: function () {

console.log("My name is " + this.firstName + ". "); console.log("My age is " + this.age + ".");

}

};

myCleverObject.myInfo();

</script>

</body>

</html>

***Listing 1-21.*** The for in Loop

<!DOCTYPE html>

<html>

<head>

<title>JavaScript Primer</title>

</head>

<body>

<script>

var myObject = { firstname: "Andrew", surname:"Grant", age: 21

};

for (var prop in myObject) { console.log(myObject[prop]);

}

</script>

</body>

</html>

***Listing 1-22.*** The for Loop in Action

<!DOCTYPE html>

<html>

<head>

<title>JavaScript Primer</title>

</head>

<body>

<script>

console.log("Looping started");

// set up the for loop here for (i = 0; i < 5; i++) {

console.log("The current value of i is " + i + ". We will loop again because " + i + "

is less than 5");

}

console.log("Looping finished");

</script>

</body>

</html>

***Listing 1-23.*** The while Loop in Action

<!DOCTYPE html>

<html>

<head>

<title>JavaScript Primer</title>

</head>

<body>

<script>

var i = 0; while (i < 10) {

console.log("The value of i is " + i);

i++;

}

</script>

</body>

</html>

***Listing 1-24.*** JavaScripts if/else in Action

<!DOCTYPE html>

<html>

<head>

<title>JavaScript Primer</title>

</head>

<body>

<script>

var userIsLoggedIn = false;

if(userIsLoggedIn){

console.log("Welcome back - sending you to some very private data");

}else{

console.log("Sorry - access denied");

}

</script>

</body>

</html>

***Listing 1-25.*** Nested Conditional Logic

<!DOCTYPE html>

<html>

<head>

<title>JavaScript Primer</title>

</head>

<body>

<script>

var userIsLoggedIn = false; var userIsVIP = true;

if(userIsLoggedIn){

console.log("Welcome back - showing you some very private data");

if(userIsVIP){

console.log("You are entitled to a 25% discount!");

}else{

console.log("You are entitled to a 10% discount!");

}

}

</script>

</body>

</html>

***Listing 1-26.*** Working with Arrays

<!DOCTYPE html>

<html>

<head>

<title>JavaScript Primer</title>

</head>

<body>

<script>

var myArray = []; myArray[0] = "Andrew"; myArray[1] = "Monica"; myArray[2] = "Catie"; myArray[3] = "Jenna"; myArray[4] = "Christopher";

console.log("Item at index 0: " + myArray[0]); console.log("Item at index 1: " + myArray[1]); console.log("Item at index 2: " + myArray[2]); console.log("Item at index 3: " + myArray[3]); console.log("Item at index 4: " + myArray[4]);

</script>

</body>

</html>

***Listing 1-27.*** Using the Length Property

...

var myArray = [];

myArray[myArray.length] = "Andrew"; myArray[myArray.length] = "Monica"; myArray[myArray.length] = "Catie"; myArray[myArray.length] = "Jenna"; myArray[myArray.length] = "Christopher"; // Display the first item

console.log("The first item is: " + myArray[0]);

// Dislay the last item

console.log("The last item is: " + myArray[myArray.length - 1]); ...

***Listing 1-28.*** Enumerating an Array var myArray = ["Andrew","Monica","Catie","Jenna","Christopher"];

for(var i = 0; i < myArray.length; i++) {

console.log(myArray[i]);

}

***Listing 1-29.*** Modifying Array Values

<!DOCTYPE html>

<html>

<head>

<title>JavaScript Primer</title>

</head>

<body>

<script>

var myArray = ["Andrew","Monica","Catie","Jenna","Christopher"];

console.log("Before: ", myArray);

for(var i = 0; i < myArray.length; i++) { myArray[i] = myArray[i] + " Grant";

}

console.log("After: ", myArray);

</script>

</body>

</html>

***Listing 1-30.*** Storing a Function Reference in a Variable: Part 1

<!DOCTYPE html>

<html>

<head>

<title>JavaScript Primer</title>

</head>

<body>

<script>

var myFunctionReference = function () { console.log('callbacks part 1')

}

myFunctionReference(); myFunctionReference; myFunctionReference();

</script>

</body>

</html>

***Listing 1-31.*** Storing a Function Reference in a Variable: Part 2

<!DOCTYPE html>

<html>

<head>

<title>JavaScript Primer</title>

</head>

<body>

<script>

function anotherFunctionReference() { console.log('callbacks part 2');

}

var x = anotherFunctionReference;

x();

anotherFunctionReference();

x();

anotherFunctionReference();

x();

</script>

</body>

</html>

***Listing 1-32.*** A Simple Callback in Action

<!DOCTYPE html>

<html>

<head>

<title>JavaScript Primer</title>

</head>

<body>

<script>

var actionsToTakeWhenServerHasResponded = function () {

console.log('The server just responded!')

}

function communicateWithServer(callback) {

callback();

}

communicateWithServer(actionsToTakeWhenServerHasResponded);

</script>

</body>

</html>

***Listing 1-33.*** Sample JSON Data

{

"firstName": "John",

"lastName": "Smith",

"address": {

"streetAddress": "21 2nd Street",

"city": "New York",

"state": "NY",

"postalCode": 10021

},

"phoneNumbers": [

"212 555-1234",

"646 555-4567"

]

}

***Listing 1-34.*** The JSON from Listing 1-32 Represented as XML

<?xml version="1.0" encoding="UTF-8" ?>

<contact>

<firstName>John</firstName>

<lastName>Smith</lastName>

<address>

<streetAddress>21 2nd Street</streetAddress>

<city>New York</city>

<state>NY</state>

<postalCode>10021</postalCode>

</address>

<phoneNumbers>

<phoneNumber>212 555-1234</phoneNumber>

<phoneNumber>646 555-4567</phoneNumber>

</phoneNumbers>

</contact>