

JavaScript Functions

- **Modular program construction**
 - Supports *Divide and Conquer* method
 - Individual functions tested before assembly
 - Code Reuse
- **JavaScript Library Functions**
 - JavaScript has seven **Global Functions**
 - JavaScript library functions are usually accessed as **Methods** contained in an **Object**
- **User defined functions can be created**

1

Copyright © 2002 R.M. Laurie

Library Functions

- **Global Functions** can be called anywhere
 - `int parseInt(string)`
Converts the *string* and returns an integer (whole number) value.
 - `float parseFloat(string)`
Converts the *string* and returns a floating point (real number) value.
- **Object.Method** functions
 - `document.write(string);` // Output
 - `window.alert(string);` // Alert Window
 - `int window.prompt(string, default);` // Prompt Window
 - `return Object.Method(parameters)`

2

Output Noun Verb Input

Copyright © 2002 R.M. Laurie

Math Object Methods

- `number Math.PI` Returns 3.141592654558979
- `number Math.max(num1, num2)` Returns greater
- `number Math.min(num1, num2)` Returns lesser
- `number Math.pow(x, y)` Returns X^y power
- `number Math.min(num1, num2)` Returns lesser
- `number Math.random()` Returns value between 0 to 1
- `number Math.sqrt(num)` Returns square root of *num*
- `number Math.sin(num)` Returns sine of *num*
- `number Math.asin(num)` Returns arc sine of *num*
- And many more methods...

3

Copyright © 2002 R.M. Laurie

Other Popular Objects

- **String Object**
 - `concat(string)` // same as `s1 + s2`
 - `split(delimiter)` // split string at *delimiter*
 - `StringVariable.toLowerCase()`
 - `StringVariable.toUpperCase()`
 - `StringVariable.blink()` // wraps the source string in
- **Date Object**
 - `getDate()`
 - `getTime()`
 - `toString()`
 - `valueOf()`
- **Number Object**
 - `toString(radix)`
 - `valueOf()`

4

Copyright © 2002 R.M. Laurie

Library Function Exa

```
undefined 4
2 4
1.4142135623730951 4
64.00000000000004 4
```

```
<HTML><HEAD>
<TITLE>Square Root and Power</TITLE>
<SCRIPT LANGUAGE = "JavaScript">
  var NumA, NumB=4;
  document.writeln("<H3>"+NumA+" "+NumB+"</H3>");
  NumA = Math.sqrt(NumB);
  document.writeln("<H3>"+NumA+" "+NumB+"</H3>");
  NumA = Math.sqrt(NumA);
  document.writeln("<H3>"+NumA+" "+NumB+"</H3>");
  NumA = Math.pow(Math.pow(NumA, NumB), 3);
  document.writeln("<H3>"+NumA+" "+NumB+"</H3>");
</SCRIPT></HEAD><BODY></BODY></HTML>
```

User Defined Functions

- User functions can be created that modularize a program
- Good for divide and conquer approach for tackle large programs
- Functions also allow you to reuse code for repeated sections
- Best for blocks with only one result
- Important for Event Driven actions
- Naming Convention:
 - Use TitleCase for User Functions (no spaces)
 - VerbNoun is best
 - CalcArea(X) PrintGraph(X, Y) GetData()

6

Copyright © 2002 R.M. Laurie

User Function Parts

- **Function Definition** contains function code
 - Usually placed at end of program
 - May have parameter list = required values
 - Can return only one value
- **Function Call** is function is invoked in a program or another function
 - Arguments are values which are passed to function
 - Position and data type match required
 - If variables it passes contents of variable

7

Copyright © 2002 R.M. Laurie

```
<HTML><HEAD>
<TITLE>A Programmer-Defined square Function</TITLE>
<SCRIPT LANGUAGE = "JavaScript">
  // MAIN PROGRAM
  document.writeln("<H3>Square numbers 1 to 8</H3>");
  for ( var x = 1; x <= 9; x++ )
    document.writeln("<B>The square of " + x +
      " is " + SquareNumber( x ) + "</B><BR>");
  // SQUARE FUNCTION DEFINITION
  function SquareNumber( y )
  {
    return y * y;
  }
</SCRIPT></HEAD><BODY></BODY></>
```

Calling function SquareNumber and passing it the value of x.

Variable y gets the value of variable x.

Square numbers 1 to 8

```
The square of 1 is 1
The square of 2 is 4
The square of 3 is 9
The square of 4 is 16
The square of 5 is 25
The square of 6 is 36
The square of 7 is 49
The square of 8 is 64
The square of 9 is 81
```

The return statement passes the value of y * y back to the calling function.

```

<HTML><HEAD>
<TITLE>Square Root and Power</TITLE>
<SCRIPT LANGUAGE = "JavaScript">
  // MAIN PROGRAM
  var sA = 1;
  document.writeln("<H3>Start of Main
  Program<BR>");
  PrintA(sA++); ← Function Calls
  PrintB(++sA);
  document.writeln("End of Main Program</H3>");

  function PrintA( A ) //FUNCTION DEFINITION
  {
    document.writeln("Function A: "+A+"<BR>");
  }
  function PrintB( B ) //FUNCTION DEFINITION
  {
    document.writeln("Function B: "+B+"<BR>");
  }
</SCRIPT></HEAD><BODY></BODY></HTML>

```

Main

PrintA(sA++)

PrintB(++sA)

Start of Main Program
Function A: 1
Function B: 3
End of Main Program

```

<HTML><HEAD>
<TITLE>Square Root and Power</TITLE>
<SCRIPT LANGUAGE = "JavaScript">
  // MAIN PROGRAM
  var sA = 1;
  document.writeln("<H3>Start of Main
  Program<BR>");
  PrintA(++sA); ← Function Call
  document.writeln("End of Main
  Program</H3>");

  function PrintA( A ) //FUNCTION DEFINITION
  {
    document.writeln("Function A: "+A+"<BR>");
    PrintB(7); ← Function Call
  }
  function PrintB( B ) //FUNCTION DEFINITION
  {
    document.writeln("Function B: "+B+"<BR>");
  }
</SCRIPT></HEAD><BODY></BODY></HTML>

```

Main

PrintA(++sA)

PrintB(7)

Start of Main Program
Function A: 2
Function B: 7
End of Main Program

```

<HTML><HEAD>
<TITLE>Square Root and Power</TITLE>
<SCRIPT LANGUAGE = "JavaScript">
  // MAIN PROGRAM
  document.writeln("<H3>Start of Main
  Program<BR>");
  PrintA(2);
  PrintB(4); ← Function Calls
  PrintA(6);
  document.writeln("End of Main
  Program</H3>");

  function PrintA( A ) //FUNCTION DEFINITION
  {
    document.writeln("Function A: "+A+"<BR>");
    PrintB("Nested in A"); ← Function Call
  }
  function PrintB( B ) //FUNCTION DEFINITION
  {
    document.writeln("Function B: "+B+"<BR>");
  }
</SCRIPT></HEAD><BODY></BODY></HTML>

```

Main

PrintA(2)

PrintB(Nest)

PrintB(4)

PrintA(6)

PrintB(Nest)

Start of Main Program
Function A: 2
Function B: Nested in A
Function B: 4
Function A: 6
Function B: Nested in A
End of Main Program

Project 2: Part D

- Create a JavaScript program that will
 - Prompt for the radius from the user
 - Calculate and display the circumference and area of a circle with this radius
 - Create three functions in your program
 - CircumferenceCircle, AreaCircle, SquareNumber
- Display in the browser window:
 - The radius, circumference, and area.
 - Format the output as you wish
 - Display your name in the browser window
- Print code and browser display

Copyright © 2002 R.M. Laurie