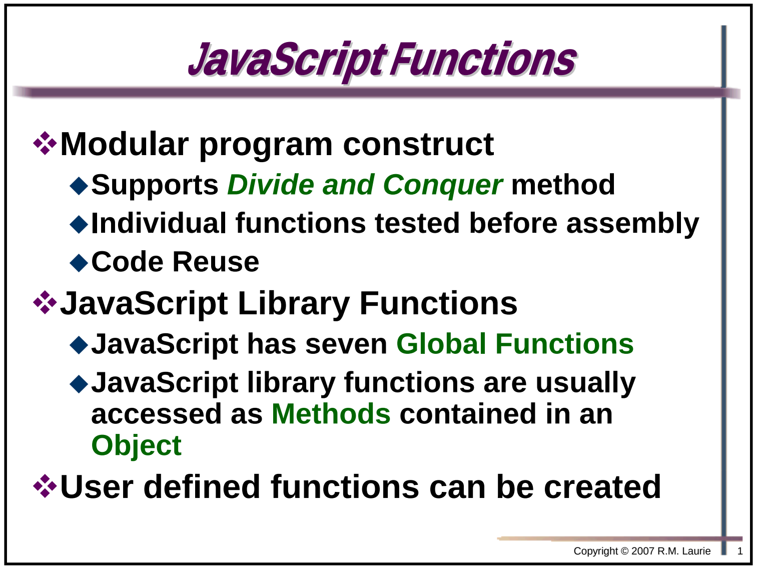
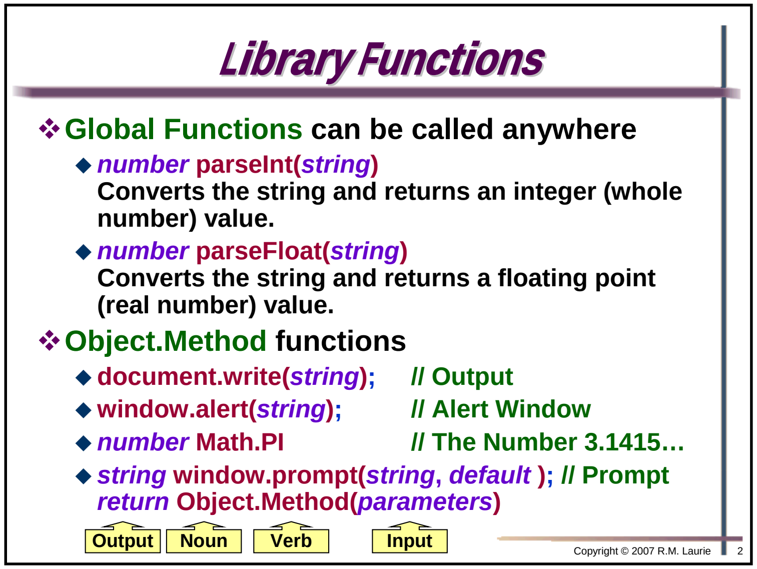


JavaScript Functions

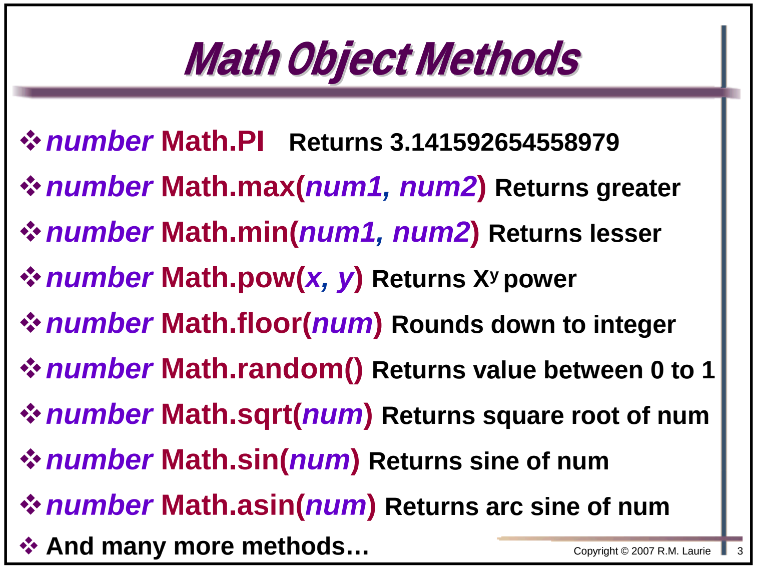
- ❖Modular program construct
  - ◆Supports *Divide and Conquer* method
  - ◆Individual functions tested before assembly
  - ◆Code Reuse
- ❖JavaScript Library Functions
  - ◆JavaScript has seven Global Functions
  - ◆JavaScript library functions are usually accessed as Methods contained in an Object
-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



Library Functions

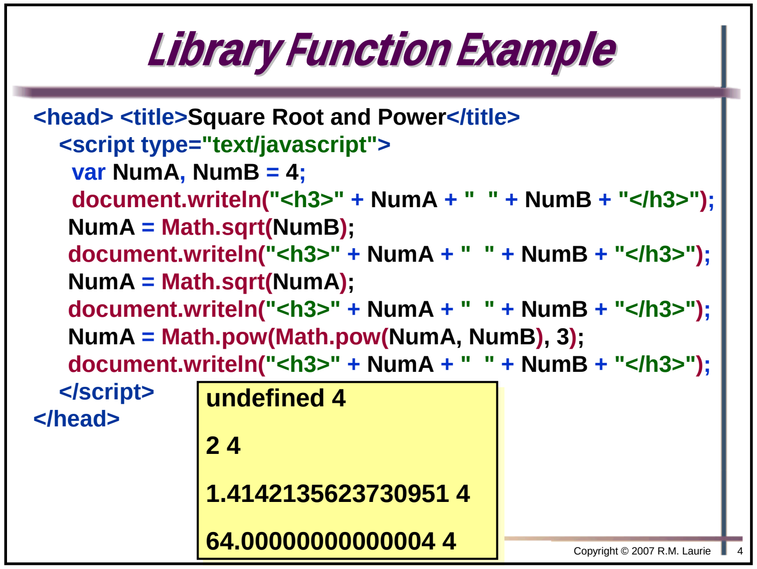
- ❖Global Functions can be called anywhere
  - ◆*number* parseInt(*string*)
Converts the string and returns an integer (whole number) value.
  - ◆*number* parseFloat(*string*)
Converts the string and returns a floating point (real number) value.
- ❖Object.Method functions
  - ◆document.write(*string*); // Output
  - ◆window.alert(*string*); // Alert Window
  - ◆*number* Math.PI // The Number 3.1415…
  - ◆*string* window.prompt(*string*, *default*); // Prompt
return Object.Method(*parameters*)

Output Noun Verb Input



Math Object Methods

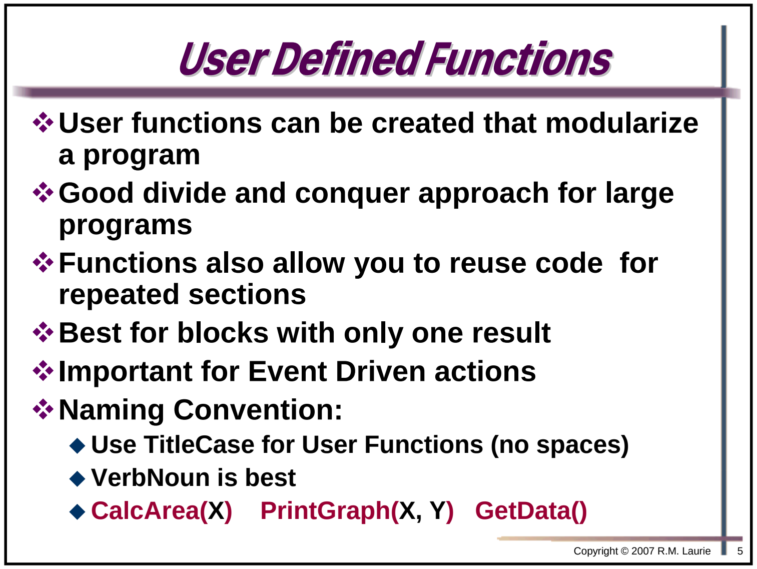
- ❖*number* Math.PI Returns 3.141592654558979
- ❖*number* Math.max(*num1*, *num2*) Returns greater
- ❖*number* Math.min(*num1*, *num2*) Returns lesser
- ❖*number* Math.pow(*x*, *y*) Returns X^y power
- ❖*number* Math.floor(*num*) Rounds down to integer
- ❖*number* Math.random() Returns value between 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…



Library Function Example

```
<head> <title>Square Root and Power</title>
  <script type="text/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>
```

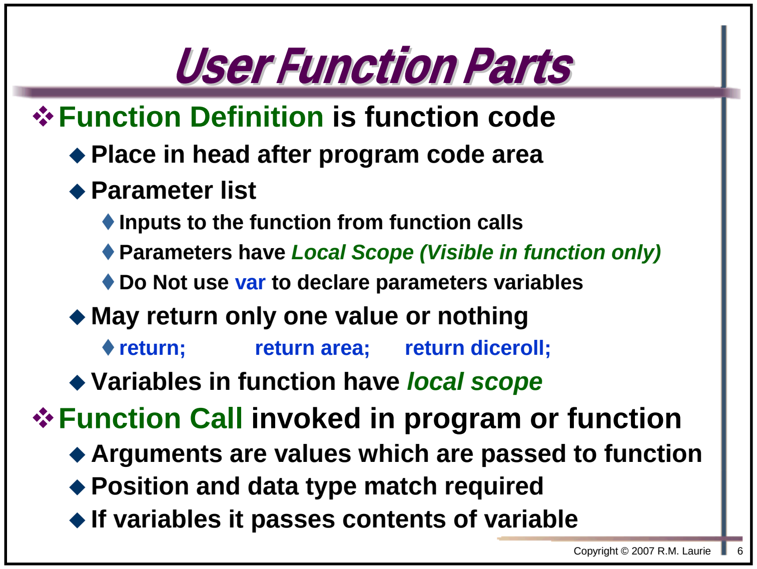
```
undefined 4

2 4

1.4142135623730951 4

64.00000000000004 4
```



User Defined Functions

- ❖ User functions can be created that modularize a program
- ❖ Good divide and conquer approach for 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()

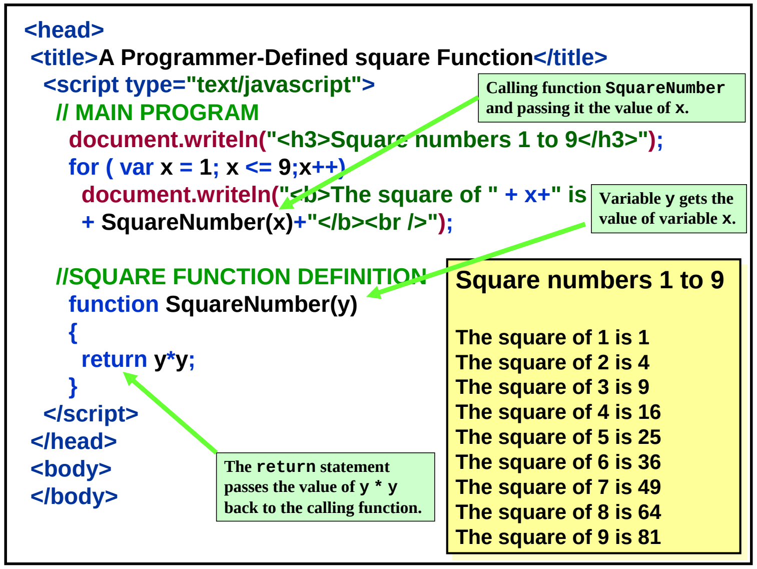
5



User Function Parts

- ❖ Function Definition is function code
  - ◆ Place in head after program code area
  - ◆ Parameter list
    - ♦ Inputs to the function from function calls
    - ♦ Parameters have *Local Scope (Visible in function only)*
    - ♦ Do Not use var to declare parameters variables
  - ◆ May return only one value or nothing
    - ♦ return; return area; return diceroll;
  - ◆ Variables in function have *local scope*
- ❖ Function Call invoked in program or function
  - ◆ Arguments are values which are passed to function
  - ◆ Position and data type match required
  - ◆ If variables it passes contents of variable

6



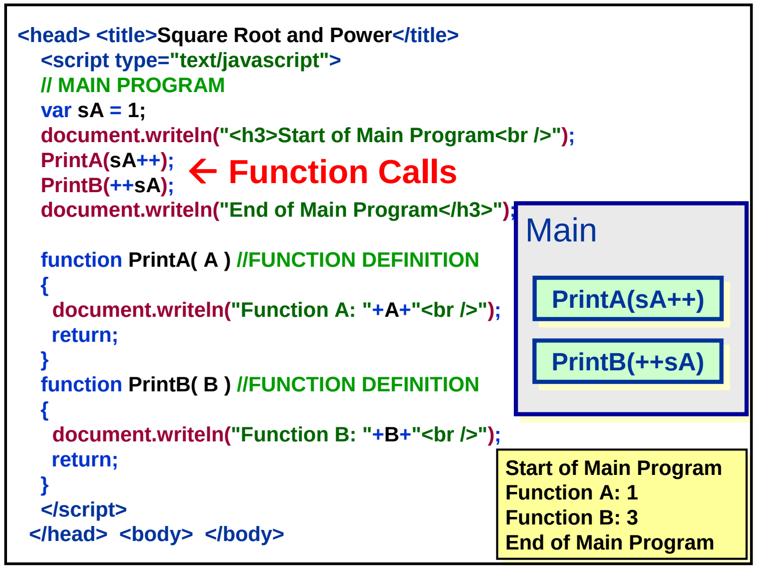
```
<head>
 <title>A Programmer-Defined square Function</title>
  <script type="text/javascript">
   // MAIN PROGRAM
     document.writeln("<h3>Square numbers 1 to 9</h3>");
     for ( var x = 1; x <= 9;x++)
       document.writeln("<b>The square of " + x+" is
       + SquareNumber(x)+"</b><br />");

   //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`.

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

```
Square numbers 1 to 9

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
```

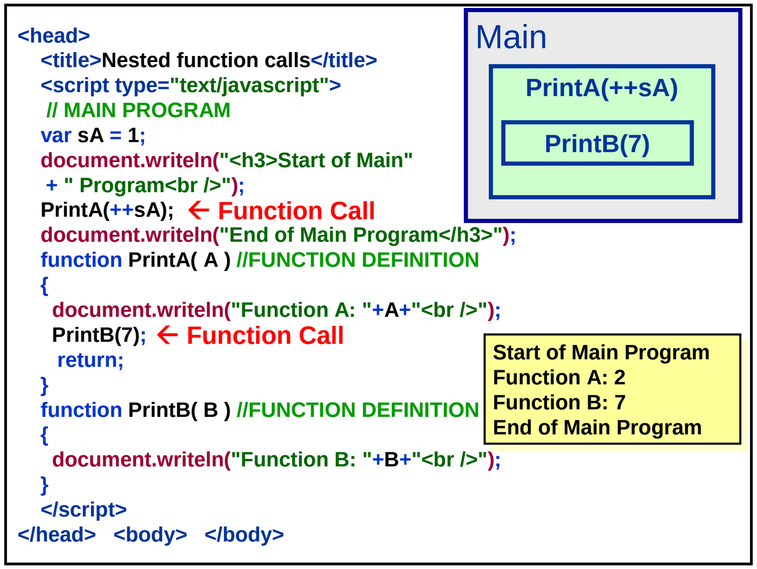


```
<head> <title>Square Root and Power</title>
   <script type="text/javascript">
   // MAIN PROGRAM
   var sA = 1;
   document.writeln("<h3>Start of Main Program<br />");
   PrintA(sA++);
   PrintB(++sA);
   document.writeln("End of Main Program</h3>");

   function PrintA( A ) //FUNCTION DEFINITION
   {
    document.writeln("Function A: "+A+"<br />");
    return;
   }
   function PrintB( B ) //FUNCTION DEFINITION
   {
    document.writeln("Function B: "+B+"<br />");
    return;
   }
   </script>
  </head>  <body>  </body>
```

← Function Calls

Main
PrintA(sA++)
PrintB(++sA)

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



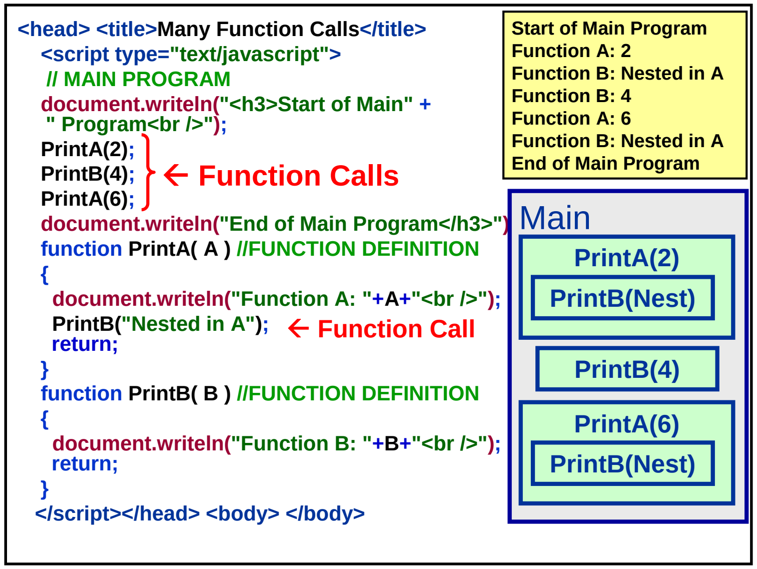
```
<head>
   <title>Nested function calls</title>
   <script type="text/javascript">
    // MAIN PROGRAM
   var sA = 1;
   document.writeln("<h3>Start of Main"
    + " Program<br />");
   PrintA(++sA);  ← Function Call
   document.writeln("End of Main Program</h3>");
   function PrintA( A ) //FUNCTION DEFINITION
   {
     document.writeln("Function A: "+A+"<br />");
     PrintB(7); ← Function Call
      return;
   }
   function PrintB( B ) //FUNCTION DEFINITION
   {
     document.writeln("Function B: "+B+"<br />");
   }
   </script>
</head>   <body>   </body>
```

Main

PrintA(++sA)

PrintB(7)

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



```
<head> <title>Many Function Calls</title>
   <script type="text/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
     return;
   }
   function PrintB( B ) //FUNCTION DEFINITION
   {
     document.writeln("Function B: "+B+"<br />");
     return;
   }
  </script></head> <body> </body>
```

```
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
```

Main

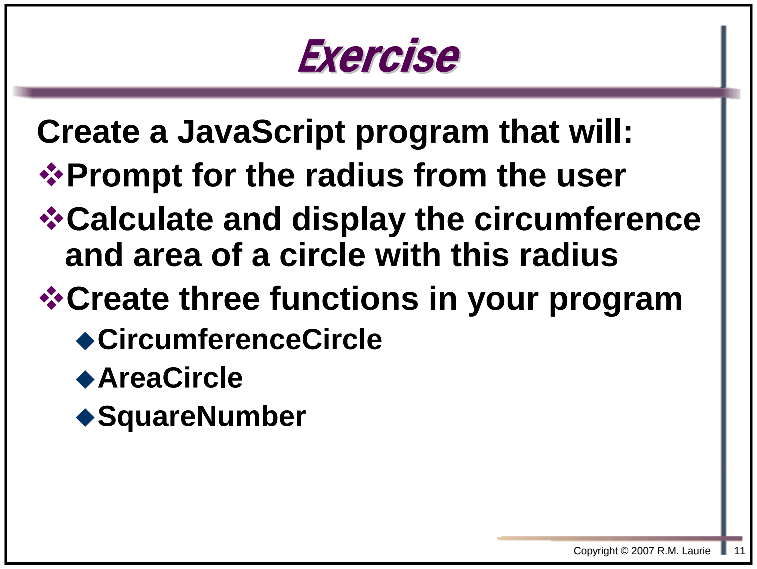
PrintA(2)

PrintB(Nest)

PrintB(4)

PrintA(6)

PrintB(Nest)



Exercise

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**

11