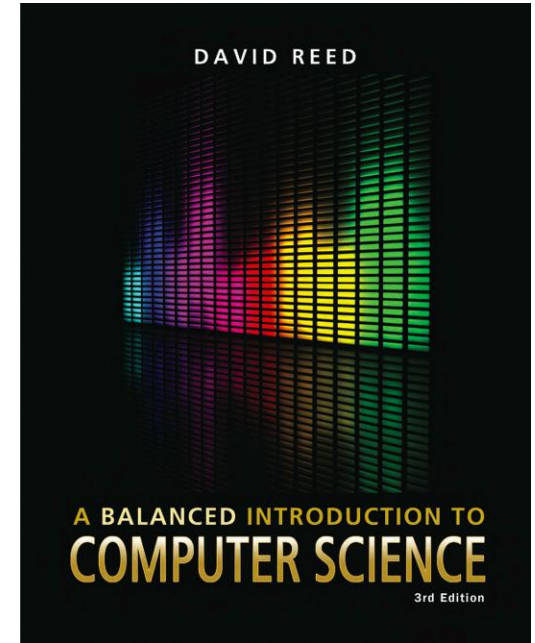


A Balanced Introduction to Computer Science, 3/E

David Reed, Creighton University

**©2011 Pearson Prentice Hall
ISBN 978-0-13-216675-1**



Chapter 7 Functions and Randomness

Predefined Functions



recall: in mathematics, a *function* is a mapping from inputs to a single output

- e.g., the absolute value function: $|-5| \rightarrow 5$, $|17.3| \rightarrow 17.3$

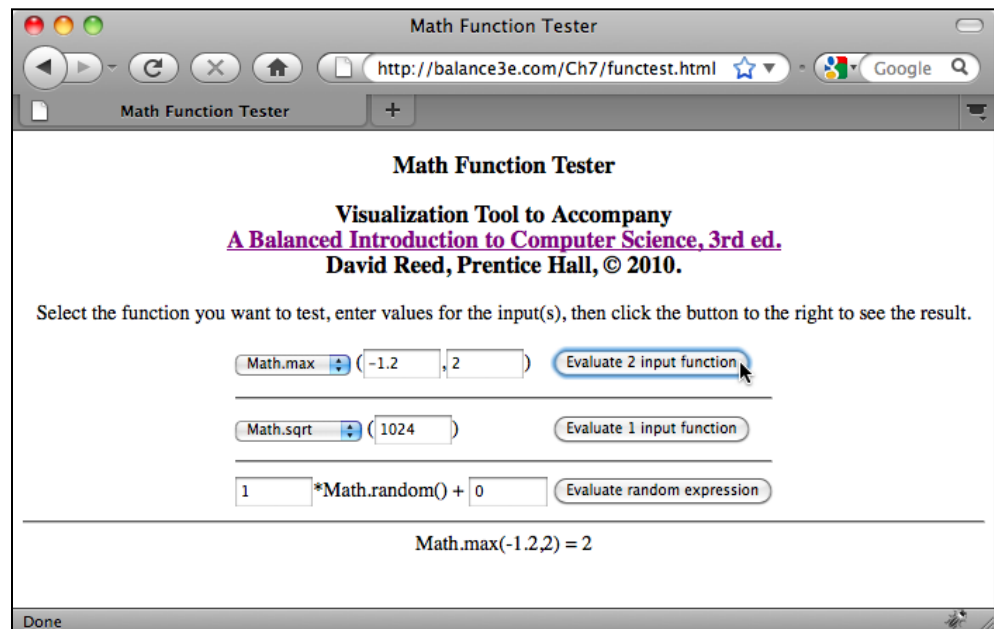
in JavaScript, a function is applied to inputs via a *function call*

- specify the function name, followed by inputs in parentheses

```
num = parseFloat(document.getElementById('numBox').value);
```

in addition to `parseFloat`,
JavaScript has
numerous predefined
mathematical functions

the `functest.html` page
allows you to explore
these



Math Functions



`Math.sqrt` determines the square root

`Math.sqrt(9)` $\rightarrow \sqrt{9} = 3$
`Math.sqrt(12.25)` $\rightarrow \sqrt{12.25} = 3.5$

`Math.max` determines the maximum of two values

`Math.max(12, 8.5)` $\rightarrow 12$
`Math.max(-3, -8)` $\rightarrow -3$

`Math.pow` raises a number to a power

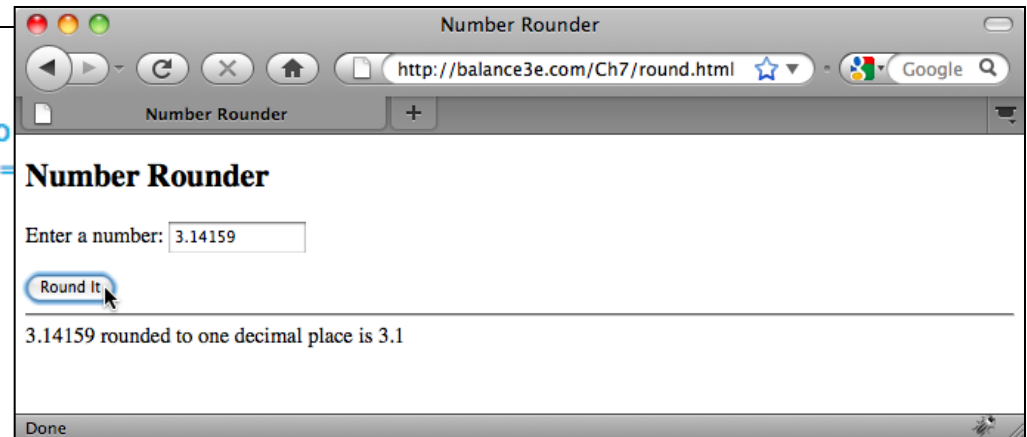
`Math.pow(2, 10)` $\rightarrow 2^{10} = 1024$
`Math.pow(2, -1)` $\rightarrow 2^{-1} = 0.5$
`Math.pow(9, 0.5)` $\rightarrow 9^{0.5} = 3$

`Math.min`, `Math.abs`, `Math.round`, `Math.ceil`, `Math.floor`, ...

Rounding Page



```
1. <!doctype html>
2. <!-- round.html
3. <!-- Web page that rounds a number to
4. <!-- =====
5.
6. <html>
7. <head>
8.   <title> Number Rounder </title>
9. </head>
10.
11. <body>
12.   <h2>Number Rounder</h2>
13.   <p>
14.     Enter a number: <input type="text" id="numberBox" size=12 value=3.14159>
15.   </p>
16.   <input type="button" value="Round It"
17.     onclick="number=parseFloat(document.getElementById('numberBox').value);
18.       rounded=Math.round(number*10)/10;
19.       document.getElementById('outputDiv').innerHTML=
20.         number + ' rounded to one decimal place is ' + rounded;">
21.   <hr>
22.   <div id="outputDiv"></div>
23. </body>
24. </html>
```



uses the `Math.round` function to round a number to 1 digit

$\text{Math.round}(3.14159 \times 10) / 10 \rightarrow \text{Math.round}(31.4159) / 10 \rightarrow 31 / 10 \rightarrow 3.1$

Math.random



`Math.random` generates a *pseudo-random* number in the range $[0...1)$

- *pseudo-random* refers to the fact that the numbers appear randomly distributed, but are in fact generated by a complex algorithm
- note: this function has no inputs; it returns a different number each call

`Math.random()` → 0.33008525626748814

`Math.random()` → 0.213335955823927

`Math.random()` → 0.8975001737758223

a call to `Math.random` can be placed in an expression to affect the range

`2*Math.random()` → $[0...2)$

`Math.random() + 1` → $[1...2)$

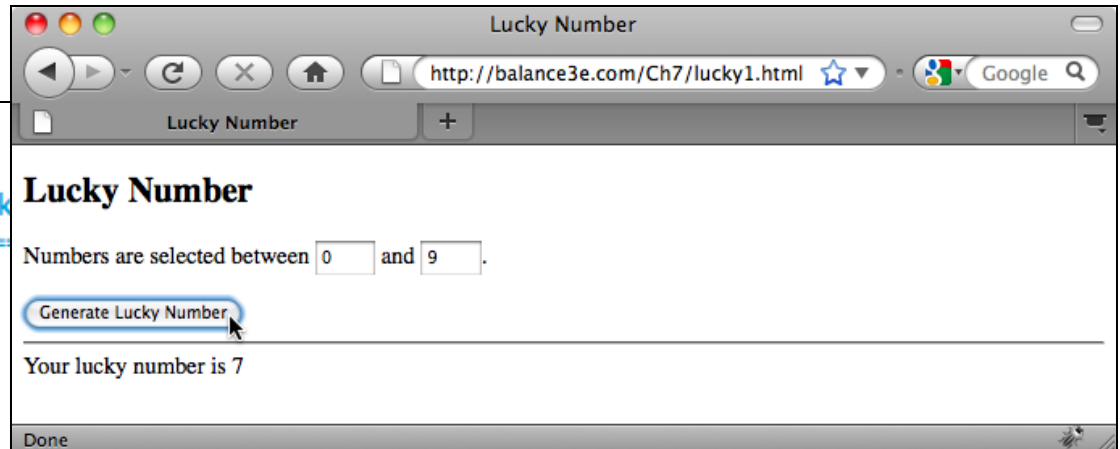
`9*Math.random() + 1` → $[1...10)$

`Math.floor(9*Math.random() + 1)` → 1, 2, 3, ..., 9

Lucky Number Page



```
1. <!doctype html>
2. <!-- lucky1.html
3. <!-- Web page that generates a lucky
4. <!-- =====
5.
6. <html>
7. <head>
8. <title> Lucky Number </title>
9. </head>
10.
11. <body>
12. <h2>Lucky Number</h2>
13. <p>
14.     Numbers are selected between <input type="text" id="minBox"
15.                                     size=3 value=0>
16.     and <input type="text" id="maxBox" size=3 value=9>.
17. </p>
18. <input type="button" value="Generate Lucky Number"
19.       onclick="min=parseFloat(document.getElementById('minBox').value);
20.               max=parseFloat(document.getElementById('maxBox').value);
21.               number=Math.floor(Math.random()*(max-min+1))+min;
22.               document.getElementById('outputDiv').innerHTML=
23.                 'Your lucky number is ' + number;">
24. <div id="outputDiv"></div>
25. </body>
26. </html>
```



displays a
random
number from
the range
specified by
the text boxes

Simplifying buttons



consider the button from lucky1.html:

```
<input type="button" value="Generate Lucky Number"
  onclick="min=parseFloat(document.getElementById('minBox').value);
  max=parseFloat(document.getElementById('maxBox').value);
  number=Math.floor(Math.random()*(max-min+1))+min;
  document.getElementById('outputDiv').innerHTML=
    'Your lucky number is ' + number;">
```

- the size of ONCLICK attribute makes the button complex and difficult to read
- plus, must be careful with nested quotes ("..." vs. '...')

functions provide a mechanism for simplifying complex buttons such as this

recall:

- functions minimize the amount of detail that has to be considered
 - ▣ e.g., can use `Math.sqrt` without worrying about how it works
- functions reduce the length and complexity of code
 - ▣ e.g., a single call to `Math.sqrt` replaces the underlying complex algorithm

Simple user-defined functions



in addition to JavaScript's predefined functions, the user can define new functions in the HEAD section and call them within the page

we will explore user-defined functions fully in Chapter 9

- for now, the following simple form suffices for simplifying buttons

```
function FUNCTION_NAME()  
// Assumes: DESCRIPTION OF ANY ASSUMPTIONS ABOUT THE PAGE  
// Results: DESCRIPTION OF THE ACTION PERFORMED BY THE FUNCTION  
{  
    STATEMENTS_TO_BE_EXECUTED  
}
```

- a function definition begins with the word `function` followed by its name and `()`
 - ▣ a function name should be descriptive of the task being performed
- lines beginning with `//` are comments that describe the function's behavior
 - ▣ comments are ignored by the interpreter, but make code more user-readable
- the statements to be executed when the function is called are placed between the curly braces

Lucky Number Revisited



```
1. <!doctype html>
2. <!-- lucky2.html                                Dave Reed -->
3. <!-- Web page that generates a lucky number from a range. -->
4. <!-- ===== -->
5.
6. <html>
7. <head>
8.   <title> Lucky Number </title>
9.   <script type="text/javascript">
10.     function GenerateNumber()
11.     // Assumes: minBox and maxBox define the range for the value
12.     // Results: picks a random number and displays it in outputDiv
13.     {
14.       min=parseFloat(document.getElementById('minBox').value);
15.       max=parseFloat(document.getElementById('maxBox').value);
16.       number=Math.floor(Math.random()*(max-min+1))+min;
17.       document.getElementById('outputDiv').innerHTML=
18.         'Your lucky number is ' + number;
19.     }
20.   </script>
21. </head>
22.
23. <body>
24.   <h2>Lucky Number</h2>
25.   <p>
26.     Numbers are selected between
27.     <input type="text" id="minBox" size=3 value=0>
28.     and <input type="text" id="maxBox" size=3 value=9>.
29.   </p>
30.   <input type="button" value="Generate Lucky Number"
31.     onclick="GenerateNumber();">
32.   <hr>
33.   <div id="outputDiv"></div>
34. </body>
35. </html>
```

the code from the button is moved to the user-defined GenerateNumber function

SCRIPT tags enclose the function definition in the HEAD

as a result, the button is greatly simplified

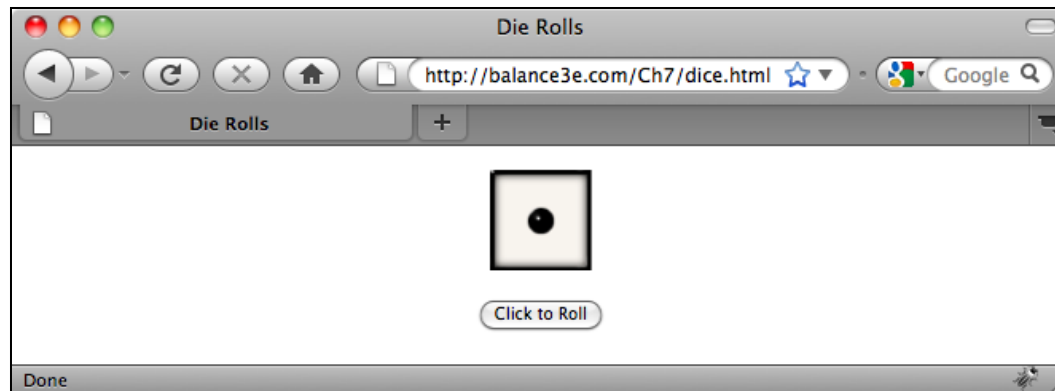
GENERAL ADVICE: if more than one statement is to be associated with a button, define a separate function

Example: Dice Simulation



suppose we wanted to simulate the roll of a 6-sided die

- at the click of a button, see a randomly selected roll of a die



can use `Math.random` and `Math.floor` to generate a random roll between 1 & 6

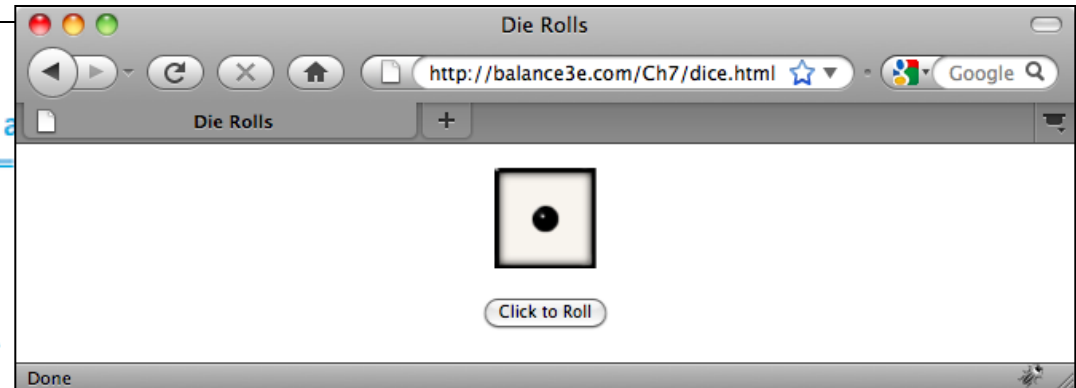
```
roll = Math.floor(Math.random()*6) + 1;
```

die images are stored as `http://balance3e.com/Images/die1.gif`,
`http://balance3e.com/Images/die2.gif`, ...,
`http://balance3e.com/Images/die6.gif`

Example: Dice Simulation



```
1. <!doctype html>
2. <!-- dice.html
3. <!-- This page simulates and displays a dice roll
4. <!-- =====
5.
6. <html>
7. <head>
8. <title> Die Rolls </title>
9. <script type="text/javascript">
10. function SelectImage()
11. // Assumes: die images are in balance3e.com/Images
12. // Results: displays a randomly selected image of a 6-sided die
13. {
14.     roll=Math.floor(Math.random()*6) + 1;
15.     imgName='http://balance3e.com/Images/die' + roll + '.gif';
16.     document.getElementById('dieImg').src = imgName;
17. }
18. </script>
19. </head>
20.
21. <body>
22. <div style="text-align:center">
23. <p>
24. 
26. </p>
27. <input type="button" value="Click to Roll" onclick="SelectImage();">
28. </div>
29. </body>
30. </html>
```



the desired die image can be selected using the roll variable

'.../die' +
roll +
' .gif'

Example: Slide Show



the dice simulation page can be generalized into a random slide show

- name the slide images using a consistent naming scheme

slide1.jpg, slide2.jpg, slide3.jpg, ...

each time the button is clicked, the `SelectImage` function is called to randomly change the image

to select a random slide at the start, make use of the `ONLOAD` attribute of the `BODY` tag

```
<body onload="CODE_TO_EXECUTE_AFTER_PAGE_LOADS">
```

here, call `SelectImage` after the page loads in order to start with a random image

```
<body onload="SelectImage();">
```

Example: Banner Ads



the random slide show page can be generalized into random banner ads

- name the ad images using a consistent naming scheme

ad0.jpg, ad1.jpg, ad2.jpg, ...

the `selectAd` function changes the ad to a random image

instead of calling the function at the click of a button, can automate using the predefined `setInterval` function

```
setInterval('JAVASCRIPT_FUNCTION_CALL', INTERVAL_IN_MSEC)
```

sets a timer and repeatedly executes the specified function at set intervals

```
<body onload="setInterval('selectAd()', 5000);">
```

will call the function to change the ad image every half second

Example: Banner Ads



```
1. <!doctype html>
2. <!-- bannerads.html
3. <!-- This page displays banner ads tha
4. <!-- =====
5.
6. <html>
7. <head>
8. <title> Random Banner Ads </title>
9. <script type="text/javascript">
10.     function SelectAd()
11.     // Assumes: the banners ad0.gif, ad1.gif, ad2.gif, and ad3.gif are
12.     // stored in http://balance3e.com/Images
13.     // Results: displays the next banner ad image in the page
14.     {
15.         adNum = Math.floor(4*Math.random());
16.         document.getElementById('adImg').src=
17.             'http://balance3e.com/Images/ad' + adNum + '.gif';
18.     }
19. </script>
20. </head>
21.
22. <body onload="setInterval('SelectAd()', 5000);">
23.     <div style="text-align:center">
24.         
26.     </div>
27.     <p>
28.         Contents of the page.
29.     </p>
30. </body>
31. </html>
```

