Debugging

Finding your errors!

D.A. Clements

11/20/2009

Rear Admiral Grace Hopper

- Computing pioneer (1906-1992)
- Invented the compiler

The first computer bug

FINDING SYNTAX ERRORS

Finding Syntax Errors

- Firefox Error Console
  - Gives line number
  - Specifies error
  - Points to error

Common error messages

- Undefined
  - The variable was never assigned a value
- Undeclared variable
  - Variable was never declared
- Null
  - document.getElementById("x") couldn’t find an element with that id
  - Check spelling and capitalization of the id
- Unterminated string literal
  - Make sure the string has matching quote marks
  - Delete any lengthy spaces in your string
Setup & Installation Videos

- Install Web Developer Toolbar
- Troubleshooting the Web Developer Toolbar—what to do when you can’t see it
- Disable caching
- Set up Error Console
- Turn off Error Console when you’re done

FINDING LOGICAL ERRORS

What to do after you’ve fixed all the errors in the Firefox Error Console and it still doesn’t work!

What is a logical error?

- Your code has the right syntax but the output is wrong.

How do we find logical errors?

- Testing is the only method
  - Test values at both ends of the range
  - Test anything a user would enter
    - Text instead of numbers
    - Numbers instead of text
  - Expect the unexpected
  - Users will always confound your expectations!

Common causes

- Statements in the wrong order
- Working with the wrong variable
- Misplaced curly brace messes up an if/else statement

How to find the problem?

- Tools
  - Alerts
  - NotePad2 / TextWrangler
  - Taking a break!
  - Asking for help!
**Alerts**

- Alerts can help you to
  - Verify control passes to a function after it's called
  - Verify your user data is being picked up by the program
  - Show you the current value in a variable at different points in your program

**A Useful Alert**

```javascript
alert("Line 67 variablename: " + variablename);
```

- `line number` tells you which alert was fired
- `variablename` in the string tells you which variable
- `variablename` gives you its current value

**Viking Names—Checking with Alerts**

```javascript
function fixOutput() {
    alert("Running function fixOutput('I have some interesting news for you.')
    var firstName;
    // get the value from the text field
    firstName = document.getElementById("firstNameField").value;
    // check that it's actually pulling the textfield's value
    alert("Your name: " + firstName);
    // find the new paragraph with id of "output"
    var elementToChange = document.getElementById("output");
    // set the innerHTML for the paragraph
    elementToChange.innerHTML = "If you were a <em>real Viking</em>, your son's last name would be " + firstName + "sson " + "and your daughter's last name would be " + firstName + "sdottir!";
}
```

**Finding Misplaced Curly Braces**

- If statements can throw your logic off if a curly brace is in the wrong place
- Watch the video!

**Video demonstration**

**When all else fails**

- Walk the dog around the block
- Hug your spouse or partner
- Do the laundry
- Take some really slow deep breaths
- Stretch
- Jog
- Escape!
Come back refreshed….

- Don’t give up!
- The solution will pop out at you!

If not…

Where to go when you’re out of ideas….

OTHER SOURCES OF HELP

Sources of Help

- JavaScript Phrasebook
- Friends and classmates
- W3 Schools tutorials and reference

Sources of Help

- Class GoPost
  - Post questions and answers
- TA’s and CLUE tutor
  - Drop-in labs
  - Instructor

Help

- When you email the Listserve or the instructor
  1. Specify the problem you are having
  2. Specify what steps you have tried
  3. Include the complete URL for your file on your Web space:
     - http://students.washington.edu/yourUWnetID/
     - And the path to the particular page or folder

End papers…

- The road to wisdom?
  Well it’s plain and simple to express:
  Err and err and err again,
  but less and less and less.
  ~ Piet Hein

- Mistakes are the portals of discovery.
  ~James Joyce (1882 - 1941)
Functions and Methods

Write it once, run it often.

D.A. Clements

Packaging Code

Conditionals
if (today == 'Friday')
{
    alert("Attend lecture!");
    alert("Read the book");
}
else
{
    alert("Go to lab");
}

Loops, or Iterations
for (i=0; i<n; i++)
{
    alert(books[i]);
}

Packaging Code

JavaScript Functions

- Functions package code
- Write it once
- Use it often
- Functions are named
- Call it by name to run it
- Pass values to it easily

Function Definition

Functions
function convertToCelsius(temp)
{
    //code goes here
}

Function call

- Calls the function when you want to run it
- Passes values to the function
Demo of building a function

JavaScript Methods

- Commonly used functions are pre-defined
  - So you don't have to!
  - No need to write a function definition

Familiar Methods

- `document.write()`
- `toLowerCase()`
- `toString()`
- `Number()`
- `Math.random()` and `prompt()`
- `document.getElementById(id)`

JavaScript Methods

- DOM methods (on Monday)
- Math methods
  - `round()`
- String methods
  - treats strings like arrays
- Array methods
  - push()
  - sort()
  - reverse()

Applying JavaScript Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document</td>
<td><code>document.getElementById(&quot;tiger&quot;).value</code></td>
</tr>
<tr>
<td>Math</td>
<td><code>Math.round(x)</code>, <code>Math.sqrt(16)</code></td>
</tr>
<tr>
<td>String</td>
<td><code>word.charAt(0)</code>, <code>word.toLowerCase()</code></td>
</tr>
<tr>
<td>Array</td>
<td><code>presidents.push()</code>, <code>presidents.sort()</code>, <code>presidents.reverse()</code></td>
</tr>
</tbody>
</table>