Announcement

- Gradebook has been down
- So we haven’t been able to transfer your quiz scores over to MyUW
- Supposed to be fixed this morning
Announcement

• Marc’s Friday office hour
  * Has miraculously transformed into
  • 2 hours
    - In a computer classroom
    » MGH 030 from 4-6pm on Fridays
Announcement

• Project 2A
  ∗ Available Friday
  ∗ Due on Wednesday
  • 2 paragraph story
  • 2 images
  • Copyright information
  • Choose words in story to replace
Keepin’ on with the Program:

Fundamental Programming Concepts Expressed in JavaScript (continued)
Exercise Part 4,

• You’ll understand more as we work through the next few slides.
Right side in the assignment statement

EXPRESSIONS
Expression and its Syntax

• Algebra-like formula called an expression
  * Describe the means of performing the actual computation
  * Built out of values and operators
    • Standard arithmetic operators are symbols of basic arithmetic
Arithmetic Operators

- Multiplication requires an asterisk ( * ), the multiply operator
- Multiply and divide are performed before add and subtract
  * Anything within parentheses is done first
  * Any multiplication or division within parentheses is performed first
- No operator for exponents
- Modulus or mod ( % ) divides two integers and returns the remainder
Relational Operators

- Make comparisons between numeric values
- Outcome is a **Boolean** value, true or false
- <  less than
- <= less than or equal to
- == equal to
- != not equal to
- => greater than or equal to
- > greater than

(Note difference between = and ==)
Logical Operators

- To test two or more relationships together
  - Teenagers are older than 12 and younger than 20

- Logical **AND**
  - Operator is `&&`
  - Outcome of `a && b` is true if both `a` and `b` are true; otherwise it is false

- Logical **OR**
  - Operator is `||`
  - Outcome of `a || b` is true if either `a` is true or `b` is true

- Logical **NOT**
  - Operator is `!`
  - Outcome is opposite of value of comparison

Teenagers are older than 12 and younger than 20
Operators (cont'd)

• Operator Overload
  * Use of an operator with different data types
  * Case of interest in JavaScript is +

• Addition
  * When used with numbers, + adds
    • 4 + 5 produces 9

• Concatenation
  * When + is used with strings, + concatenates or joins the strings together
    • "four" + "five" produces "fourfive"
A Conditional Statement

```javascript
if ( <Boolean expression> )

<then-statement>;
```

- **Boolean expression is a relational expression**
  
  * Evaluates as either True or False

- **then-statement is any JavaScript statement**
If Statements and Their Flow of Control

- The Boolean statement, called a predicate, is evaluated, producing a true or false outcome.
- If the outcome is true, the then-statement is performed.
- If the outcome is false, the then-statement is skipped.
- Then-statement can be written on the same line as the Boolean or on the next line.
Compound Statements

• Sometimes we need to perform more than one statement on a true outcome of the predicate test

• You can have a sequence of statements in the then clause

• Group these statements using curly braces 
  
* They are collected as a compound statement
if/else Statements

- To execute statements if a condition is false

```
if ( <Boolean expression> )
{
    <then-statements>;
}
else
{
    <else-statements>;
}
```

- The Boolean expression is evaluated first
  * If the outcome is true, the then-statements are executed and the else-statements are skipped
  * If the outcome is false, the then-statements are skipped and the else-statements are executed
Nested if/else Statements

• The then-statement and the else-statement can contain an if/else

• The else is associated with the immediately preceding if

• Correct use of curly braces ensures that the else matches with its if
Nested if/else Statements

```plaintext
if (<Boolean exp1>)
{
    if (<Boolean exp2>)
    {
        <then-stmts for exp2>;
    }
    else
    {
        <else-stmts for exp2>;
    }
}
else
{
    <else-stmts for exp1>;
}
```
## Exercise Part 4

<table>
<thead>
<tr>
<th>Expressions or conditions</th>
<th>Replacing Variables with values</th>
<th>Result</th>
<th>Number</th>
<th>String</th>
<th>Boolean</th>
</tr>
</thead>
<tbody>
<tr>
<td>e + f</td>
<td>&quot;Donald&quot; + &quot;Duck&quot;</td>
<td>&quot;DonaldDuck&quot;</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td><em>Better:</em> &quot;Donald &quot; + &quot;Duck&quot;</td>
<td>&quot;Donald Duck&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e + g</td>
<td>&quot;Donald&quot; + 10</td>
<td>&quot;Donald10&quot;</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>( (b / c) &lt; a )</td>
<td>75 / 25 &lt; 100</td>
<td>true</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>= 3 &lt; 100</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( ( c &gt; a )</td>
<td></td>
<td>( b &lt; a ))</td>
<td>(25 &gt; 100) OR (75 &lt; 100)</td>
<td>If either is true, it’s true</td>
<td></td>
</tr>
<tr>
<td>( h = b )</td>
<td>h = 75</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>( h == b )</td>
<td>75 == 75?</td>
<td>True</td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
Working Together

**HTML, CSS, AND JAVASCRIPT**
Purposes of Each

- Three separate types of coding
  - HTML—for content
  - CSS—for appearance
  - JavaScript—for action
Examples

- HTML—static page
- CSS—add styling to the page
- JavaScript—adds action!
JavaScripts and HTML

Types of JavaScripts are based on location in the HTML page:

* Body scripts—body section
* Header scripts—head section
* External scripts—links to a .js page
  • Similar to .css pages
<html>
  <head>
    <title>Name of Page</title>
  </head>
  <body>
    <script type="text/javascript">
      // JavaScript goes here
    </script>
  </body>
</html>
<html>
<head>
  <title>Name of Page</title>
</head>
<body>
<script type="text/javascript">
  // JavaScript goes here
</script>
</body>
</html>
<html>
  <head>
    <title>Name of Page</title>
    <script type="text/javascript">
      // JavaScript goes here
    </script>
  </head>
  <body>
    Body content goes here
  </body>
</html>
External Script

- Linked in the `<head>`
- `<script>` tag with `src` gives path name

```html
<html>
  <head>
    <title>Name of Page</title>
    <script type="text/javascript"
      src="basic.js"></script>
  </head>
  <body>
    Body content goes here
  </body>
</html>
```
External JavaScripts

- Make changes to scripts in one place
- Reusable
  * Can be linked to any page, every page in a site, or many sites
Summary

- Programming is the exact specification of an algorithm
- JavaScript is typical ... with many rules

* Learning strategy
  - Do the reading first
  - Practicing is better than memorizing for learning the rules
  - Use the program-save-reload-test plan
  - Precision is your best friend