Running JavaScript

Chapter 18

Concatenation: Operating On Strings

- **string concatenation**: using the `+` operator between a string and another value to make a longer string

- Examples:
  - `'hello' + 42` is `'hello42`
  - `1 + "abc" + 2` is `"1abc2"`
  - `'abc' + 1 + 2` is `"abc12"`
  - `1 + 2 + "abc"` is `"3abc"
  - `"1" + 1` is `"11"

Popup Box

- Alert box syntax:
  ```javascript```
  ```javascript
  alert(<expression>);
  ```

- Examples:
  ```javascript```
  ```javascript
  alert("Hello, world!");
  ```

What Is In The Variable?

- Use alert boxes to reveal the value of variables.
  ```javascript```
  ```javascript
  var x = 100;
  alert(x);
  ```

- Use string concatenation to make alert messages even more useful.
  ```javascript```
  ```javascript
  alert("x = \[" + x + "]");
  ```

Linking JavaScript File To XHTML File

- Copy the `type` attribute and its corresponding value verbatim
- Use the `src` attribute to specify the location of a JavaScript file
  - Path location may be absolute or relative

Conditionals
Conditionals

"If button is clicked, then close the popup box."

"If Mario touches the flag, then end the level."

"If a correct password has been entered, then reveal the top secret documents, otherwise contact the FBI."

"If the coin collected brings the total to one hundred, make 1-up sound, otherwise make regular coin collection sound."

The if Statement

**if statement**: a control structure that executes a block of statements only if a certain condition is true

General syntax:

```javascript
if (<test>) {
  <statement(s)>
}
```

Example:

```javascript
var gpa = 3.25;
if (gpa >= 3.0) {
  alert("Good job! Have a cookie.");
}
```

The if/else Statement

**if/else statement**: A control structure that executes one block of statements if a certain condition is true, and a second block of statements if it is false. We refer to each block as a branch.

General syntax:

```javascript
if (<test>) {
  <statement(s)>
} else {
  <statement(s)>
}
```

Example:

```javascript
var gpa = 3.25;
if (gpa >= 3.0) {
  alert("Good job! Have a cookie.");
} else {
  alert("No cookie for you!");
}
```

Relational Expressions

The `<test>` used in an if or if/else statement must evaluate to a Boolean value (true or false).

Relational expressions evaluate to Boolean values and use the following relational operators:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
<th>Example 1</th>
<th>Example 2</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>==</td>
<td>equals</td>
<td>4 + 1 == 5</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>!=</td>
<td>does not equal</td>
<td>1.2 != 1.3</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>&lt;</td>
<td>less than</td>
<td>10 &lt; 5</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>&gt;</td>
<td>greater than</td>
<td>10 &gt; 5</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>&lt;=</td>
<td>less than or equal to</td>
<td>126 &lt;= 100</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>&gt;=</td>
<td>greater than or equal to</td>
<td>3.0 &gt;= 5.0</td>
<td>true</td>
<td></td>
</tr>
</tbody>
</table>
Evaluating Relational Expressions

- Relational operators have lower precedence than math operators.
  
  \[
  5 \times 7 \gg 3 + 5 \times (7 - 1) \\
  5 \times 7 \gg 3 + 5 \times 6 \\
  35 \gg 3 + 30 \\
  35 \gg 33
  \]

- Relational operators should not be “chained” as they can in algebra. **WARNING!** JavaScript will NOT complain if you do so and you may get unexpected results.

```
2 <= 1 <= 10
false <= 10
true
```

Errors In Coding

- Many students new to if/else write code like this:

  ```javascript
  var percent = 85;
  if (percent >= 90) {
    alert("You got an A!");
  } else if (percent >= 80) {
    alert("You got a B!");
  } else if (percent >= 70) {
    alert("You got a C!");
  } else {
    alert("You got an F!");
  }
  ```

  What will happen? What’s the problem?

  - You may get too many popup boxes. Try it out!

Nested if/else Statements

- **Nested if/else statement**: A chain of if/else that can select between many different outcomes based on several tests.

  - General syntax:
    ```javascript
    if (<test>) {
      <statement(s>);
    } else if (<test>) {
      <statement(s>);
    } else {
      <statement(s>);
    }
    ```

  - Example:
    ```javascript
    if (number > 0) {
      alert("Positive");
    } else if (number < 0) {
      alert("Negative");
    } else {
      alert("Zero");
    }
    ```

Nested if/else Variations

- A nested if/else can end with an if or an else.
  - If it ends with else, one of the branches must be taken.
  - If it ends with if, the program might not execute any branch.

  ```javascript
  if (<test>) {
    <statement(s>);
  } else if (<test>) {
    <statement(s>);
  } else if (<test>) {
    <statement(s>);
  } else {
    <statement(s>);
  }
  ```

Nested if/else Flow Chart

```
if (<test>) | <statement(s>);
| else if (<test>) | <statement(s>);
| else | <statement(s>);
```

```
if (<test>) | <statement(s>);
| else if (<test>) | <statement(s>);
| else if (<test>) | <statement(s>);
```
Nested if/else Variations

```javascript
if (place == 1) {
    alert("You win the gold medal!");
} else if (place == 2) {
    alert("You win a silver medal!");
} else if (place == 3) {
    alert("You earned a bronze medal.");
}
```

- Are there any cases where this code will not print a message?
  - Yes, if the `place` variable is not 1, 2, or 3.
- How could we modify it to print a message to non-medalists?
  - Add an `else` clause.

## Summary: if/else Structures

- Choose exactly 1 set of statements
  ```javascript
  if (<test>) {
      <statement(s>);
  }
  ```
- Choose 0, 1, or more set of statements
  ```javascript
  if (<test>) {
      <statement(s>);
  } else if (<test>) {
      <statement(s>);
  } else if (<test>) {
      <statement(s>);
  }
  ```
- Choose 0 or 1 set of statements
  ```javascript
  if (<test>) {
      <statement(s)>
  } else if (<test>) {
      <statement(s)>
  } else if (<test>) {
      <statement(s)>
  } else {
      <statement(s)>
  }
  ```

## Which if/else Construct To Use?

- Reading the user's GPA and printing whether the student is on the dean's list (3.8 to 4.0) or honor roll (3.5 to 3.7)
- Printing whether a number is even or odd
- Printing whether a user is lower-class, middle-class, or upper-class based on their income
- Determining whether a number is divisible by 2, 3, and/or 5
- Printing a user's grade of A, B, C, D, or F based on their percentage in the course

## That Thing Called Style

- As with HTML, you are required to indent your code properly.
  - Indent code within opening and closing curly braces.
- You should spend time on thinking or coding. You should **NOT** be wasting time looking for that missing closing brace.
- So code with style!