JavaScript for App Development

By Mark Lassoff, Founder Framework Television

Section Four

Decisions... Decisions... Decisions...

At some point every app has to make a decision. Decisions in JavaScript are expressed as `if` statements or `switch...case` statements. Some decisions that might have to be made in the course of program execution include:

- Is the player's score greater than the high score?
- Is the password entered equivalent to the correct password stored in the database?
- Is the CTRL key depressed?
- Did the two sprite collide?

In this section you'll learn how to write the code that drives these programatic decisions in JavaScript.

Section 4 Goals

In this section of the course your goals are:

- [ ] Use Basic If Statements
- [ ] Create Compound Conditions
- [ ] Understand and Use Else Statements
- [ ] Use If...Else Patterns
- [ ] Create Switch/Case Statements

Watch This: Section 4 Video

As always your course videos are available on YouTube, Roku and other locations. However, only those officially enrolled have access to this course guide, are able to submit assignments, work with the instructor, and get this guide.

Watch this section video at: https://www.youtube.com/watch?v=o6mEwelYrS0

Basic If Statements

Most coding languages have some form of `if` statement. Simply put an `if` statement tests a condition, and, if that condition is true executes a block of code. For example (pseudocode):

```
if (the player's score is greater than the high score):
     Replace the high score with the player's score.
```

The part of the `if` statement inside the parentheses is the condition. The condition will contain a conditional operator. (If you've never heard of a conditional operator you should review the previous section!)

Here's a typical Javascript `if` statement:

```
if(age>=65)
{
 user = "Seasoned Citizen";
}
```

When the `if` statement is executed, the conditional statement is evaluated and is either `true` or `false`. If it's `true`, the associated code block, contained within the curly brackets, will be executed.

Let's take a look at a complete JavaScript example with a conditional:

```
<div id="output"></div>
<script>
    var age = prompt("How old are you?");
    if(age < 21)
        {
            document.getElementById('output').innerHTML =
            "<p>You are not old enough to consume alcohol.</p>";
        }
</script>
```

When you test this code you'll notice that if you enter a value under 21 the program responds with a message, but when you enter a value 21 or greater, there's no response at all.

Do This: Write a Conditional

Write a conditional that tests whether or not a value is an even number. Prompt the user to enter a number. If the number is even, output a message indicating so. *This is a good time to remember our friend modulus!*

Complex Conditionals with Compound Conditions

There are times you might need to test multiple conditions in a single if statement. There are two additional operators that allow you to create complex conditions. The AND operator (`&&`) allows you to join two conditions, both of which must be true. The OR operator (`||`) allows you to join two conditions, one of which must be true.

For example (in the USA) in order to vote someone must be 18 years old or older and a citizen. This would be expressed as an AND like this:

```
age >= 18 && citizen == true
```

In order to board a plane, you must have a valid passport or state driver's license. That would be expressed as an OR:

```
hasPassport == true || hasValidLicense == true
```

The compound conditional returns a single true or false when evaluated. Let's take one of these examples and expand it in to a full program.

```
<div id="output"></div>
<script>
    var age = prompt("How old are you?");
    var citizen = prompt("Are you a citizen [true/false]?");

    if(age >= 18 && citizen == "true")
        {
            document.getElementById('output').innerHTML =
            "<p>You are elgible to vote.</p>";
        }

</script>
```

In this example, both parts of the conditional have to be true-- which is why we used the AND operator. If either part of the conditional is false, the entire conditional is evaluated as false.

What ELSE? (Get it?)

In the previous examples, if a condition was evaluated as true, the associated code block is executed. If it is evaluated as false nothing happens. Nothing is not a good result for the user. If nothing happens they might wonder if the program crashed, or, if they did something wrong.

We should always provide feedback to the user, regardless of how our conditional test turned out.

The `else` statement allows us to react to the if statement if the condition is evaluated as false. Using pseudocode:

```
if(cats == dogs)
{
 Do this if it's true...
} else
{
 Do this if it's false...
}
```

The `else` statement essentially makes the `if` statement an either…or proposition. Either the true codeblock will be executed or the false codeblock will be executed depending on the evaluation of the original condition.

Let's build on the previous example and add an `else` clause:

```
<div id="output"></div>
<script>
    var age = prompt("How old are you?");
    var citizen = prompt("Are you a citizen [true/false]?");

    if(age >= 18 && citizen == "true")
        {
            document.getElementById('output').innerHTML =
            "<p>You are elgible to vote.</p>";
        } else
        {
            document.getElementById('output').innerHTML =
            "<p>Sorry, you are ineligible to vote</p>";
        }

</script>
```

Now we're reacting properly to the user regardless of how the initial conditional statement is evaluated.

The If… Else Pattern

There are occasions where you'll need to run multiple tests in order to get the correct outcome. For example, if income tax rate is assigned by income. To a point, the higher your income, the higher your income tax rate. This type of pattern is common when you're trying to categorize data.

Let's take a look at an `if...else` pattern:

```
<div id="output"></div>
<script>
    var grade = prompt("Enter a numerical grade (0-100)");
    var letterGrade;

    if (grade >= 90)
        {
            letterGrade = "A";
        } else if (grade >= 80)
            {
                letterGrade = "B";
            } else if (grade >= 70)
                {
                    letterGrade = "C";
                } else if (grade >= 60)
                    {
                        letterGrade = "D";
                    } else
                        {
                            letterGrade = "F";
                        }
    document.getElementById('output').innerHTML =
    "<h2>Letter Grade: " + letterGrade + "</h2>";

</script>
```

The best way to understand these `if...else` blocks is to follow the logic. Let's say the user enters a numerical grade of 80. The first if condition is false so it goes to the next if condition which is if `(grade >= 80)`. This is evaluated as true so the letterGrade variable is assigned the string 'B' and then the *`if...else` statement block is exited.*

Do This: Modify an If…Else Block

Modify the if else block above to include '+' grades and '-' grades. Assign grades according to the following chart:

Numerical Grade	Letter Grade
97 or Above	A+
93-96	A
90-92	A-
87-89	B+
83-86	B
80-82	B-

Numerical Grade	Letter Grade
77-79	C+
73-76	C
70-72	C-
67-69	D+
63-66	D
60-62	D-
Below 60	F

That's a lot of conditions, so you should get plenty of practice! Good luck.

Switch... Case... Break!

In addition to the `if` statement there is another common way to evaluate conditionals: The `Switch...Case` Statement.

The `switch...case` statement is useful if evaluating a number of conditions at once. Here's an example:

```
<div id="output"></div>
<script>
    var airport = prompt("Originating from [LGA|JFK|EWR]");
    var out;

    switch (airport)
        {
            case "LGA":
                out = "LaGuardia Airport<br/>";
                out += "Queens, New York";
                break;

            case "JFK":
                out = "John F. Kennedy International Airport<br/>";
                out += "Queens, New York";
                break;

            case "EWR":
                out = "Newark International Airport<br/>";
                out += "Newark, New Jersey";
                break;

            default:
                out = "Can't identify airport code";
        }

    document.getElementById('output').innerHTML = out;

</script>
```

Again, following the logic is the easiest way to understand this code block. The user first enters one of three values (hopefully). Each case attempts to match the value entered with the string value in the `case` statement. If there is a match then the associated code block is run. When the JavaScript interpreter hits the `break` statement then entire `swith...case` block is exited.

The `default` statement is not required. It is executed only if none of the cases match the value being tested-- in this case the `airport` variable.

Do This: Debugging

Debug this till it works! Tax rates are fictional. Using the numbers provided assume the higher the salary, the higher the tax rate.

```
<div id="output"></div>
<script>
    var salary = prompt("Enter your salary.");
    var taxRate;

    if (salary  150000)
        {
            taxRate == .0155;
        } else if (salary  115000)
            {
                taxRate == .0125;
            } else if (salary  92500)
                {
                    taxRate == 0.11;
                } else if (salary  70000)
                    {
                        taxRate == .095;
                    } else
                        {
                            taxRate == .06;
                        }

    var taxBill = salary * taxRate;

    var out = "Tax Rate: " + taxRate + "<br/>";
    out ++ "Tax Bill: " + taxBill;

    document.getElementById("output") = out;

</script>
```

Good luck!

Submit This: Lab Exercise

Create a calculator. You will prompt the user three times. For the first prompt, ask the user for the first number to be calculated. For the second prompt, ask the user for the second number to be calculated. On the third prompt ask for an operation (+ , - , * , / , %).

Once the user entry to the prompts are stored, perform the operation requested and output the entire equation. For example if the user enters 4, 5 and + output “9 + 4 = 13”

Have fun!

Remember every exercise must be submitted in order for you to earn certification. Once you’ve completed this exercise, you’re ready to move on to Section 5.

Submit your exercises to: https://www.dropbox.com/request/Nl7bSaAe11Zl0PgLRonA