

Selective Execution



Dr. Stephen Blythe
CSC I 0022, Fall 2020

The `if` Statement

- JavaScript's `if` statement lets you optionally execute the next statement.
 - the `if` statement itself does not have a semicolon after it
 - considered standard (but optional) practice to indent next statement
- For example:

```
<script type=text/javascript>
  var outdoorTemperature=prompt("What is the temperature outside?");
  console.log("It is" + outdoorTemperature + " degrees<br>");
  if (outdoorTemperature>80)
    console.log("That is pretty hot!");
</script>
```

- Note that the `()`'s surrounding the "test condition" are *required*!

The `if` Statement ...

- By default, only one statement is “selectable” by the `if`
 - indenting any additional statements will not automatically fix this
 - Can be “fixed” by surrounding a code sequence in a `{ ... }` pair
- The following:

```
// still need previous code above here ...  
if (outdoorTemperature>80)  
    console.log("That is pretty hot!");  
    // the next line will always be done!  
    console.log("A/C is needed!");
```

- Should probably be:

```
// still need previous code above here ...  
if (outdoorTemperature>80)  
{    // note the opening curly brace  
    console.log("That is pretty hot!");  
    console.log("A/C is needed!");  
}    // note the closing curly brace
```

Comparison

Yes, you can do more than compare with “less than”:

Comparator	Meaning	Examples
<	“less than”	$x < 3$ $3 < x$ $a < b$
>	“greater than”	$x > 3$ $3 > x$ $a > b$
<=	“less than or equal to”	$x \leq 3$ $3 \leq x$ $a \leq b$
>=	“greater than or equal to”	$x \geq 3$ $3 \geq x$ $a \geq b$
==	“equal to”	$x == 3$ $3 == x$ $a == b$
!=	“not equal to”	$x != 3$ $3 != x$ $a != b$

Warning: single equal sign (i.e. just =) does *not* compare two things!

- what does it do?

Compound Conditions

Consider the following:

```
// attempt to see if x is between 0 and 10?  
if ( 0 < x < 10 )  
    alert("x is in range!");
```

frustratingly, this will always say `x is in range!`

- Instead, use compound conditionals:

Operator	Meaning	Example
&&	“and”	<code>ab!=0 && hits<10</code>
	“or”	<code>answer=='y' answer=='n'</code>
!	“not”	<code>!(answer=='y' && ab==0)</code>

- For example:

```
// working code to see if x is between 0 and 10  
if ( 0 < x && x < 10 )  
    alert("x is in range!");
```

The `if...else` Statement

Consider the following:

```
if (outdoorTemperature>80)
  console.log("Turn on the A/C!");
else
  console.log("Save some money - turn off the A/C.");
```

- should the if condition be true, the first “block” of statements is done
- otherwise (the condition is false), the second “block” is done.

Nested if Statement

You can have (one or more) of the statements inside an `if` be another `if`:

```
if (outdoorTemperature>80)
{
    console.log("Turn on the A/C!");
    if (outdoorTemperature > 95)
        alert("Pour me a drink!!");
}
else
    console.log("Save some money - turn off the A/C.");
```

Nested if...else Statement

You can link together a sequence of nested if/else statements:

```
if (outdoorTemperature>90)
  console.log("Turn on the A/C!");
else
  if (outdoorTemperature > 80)
    console.log("A bit rough !!");
  else
    if (outdoorTemperature > 60)
      console.log("Nice - have a picnic?");
    else // generally, this may not be needed, essentially a "catch all" here
      console.log("A bit cold - grab a coat!");
console.log("Thank you for listening."); // not part of if/else, so always printed!
```


Whitespace in JavaScript

- Tabs, newlines (returns), and extra spaces are ignored by Javascript.
- So, the following two code sections perform *identically !!!*

```
if ( grade >= 60 )  
    alert( "P" );  
else  
    alert( "F" );  
    alert( "Done" );
```

```
if(grade>=60)alert("P");else alert("F");alert("Done");
```

- Obviously (I hope), the one on the left is better
 - it is much more readable!!!
- Note that extra spaces, tabs, and/or newlines between quotes do matter!

Nested if...else Revisited

Our previous if ... else can thus be equivalently rewritten as:

```
if (outdoorTemperature > 90)
    console.log("Turn on the A/C!");
else if (outdoorTemperature > 80)
    console.log("A bit rough !!");
else if (outdoorTemperature > 60)
    console.log("Nice - have a picnic?");
else // generally, this may not be needed, essentially a "catch all" here
    console.log("A bit cold - grab a coat!");
console.log("Thank you for listening."); // not part of if/else, so always printed!
```

Most people argue that this is much easier to read.