Javascript Circuit

UNIT 2 REVIEW GUIDE

KEY TERMS & DEFINITIONS

Boolean
A variable that can either equal true or false.

```
var isEven = true;
```

Condition
A mathematical statement that evaluates to true or false.

```
(x > 10)
```

Operator
A sign that allow us to make comparisons in a condition.

=, <, >

Statement
Code used to perform an action.

```
return y;
```

Collection
Groups of values and an array is an example of a collection

Type Coercion
If the values on both sides of the double equals sign (==) have a different type (i.e., the number 1 and the string "1"), JavaScript will try to change the type of both operands to check whether or not they are equal.

```
7 == "7" //returns true
```

Expression
Code that is written whenever a value is expected.

```
y+6
```

Logical Operators
Operators used to combine several boolean statements into a single statement (i.e., AND, NOT, OR).

If it is raining AND I have an umbrella, then I will go outside with an umbrella.

Undefined
A variable that is not assigned a value.

```
var x;
```

Null
A variable declared to have no value.

```
var playerScore = null;
```

Truthy
Something that evaluates to true, like strings, non-zero numbers, or true.

```
("Hello Dolly") //truthy
```

Falsey: Something that evaluates to false. This includes false, 0 (zero), " " (empty string), null, undefined, or NaN (a special Number value meaning Not-a-Number).

```
("") //falsey
```

if Statement
Allows us to check whether or not a condition is true and, if it is, runs our code.

else if Statement
Allows us to specify a second condition to test; this second conditions will only be tested if the first condition evaluates to false.

else Statement
Default action taken if none of the other conditions in a code block are true.

The following code contains examples for the last three definitions:

```
if (answer === 38) {
// Do something if first condition is true
} else if (answer === 30) {
// Do something if second condition is true
} else {
// Do something if all above conditions are false
}
```

Loop
A JS command that tells your program to repeat something.

while Loop
A loop that runs a block of code over and over again as long as our condition remains true (or at least truthy).

```
while (someCondition) {
// A block of code.
}
```

for Loop
A loop that runs for a fixed, controllable number of times, ensuring it will never get stuck in an infinite loop.

```
for(var i = 0; i < 100; i++) {
//Note: i is not syntax but stands for any variable
//Do something for a 100 times.
}
```

Switch Statement
A variable that gets evaluated; if there is a case listed for the value that it evaluates to, the code between case __: and break will be executed.

```
var grade = "B";
switch (grade) {
case "A":
console.log("Awesome Job");
break;
case "B":
console.log("Good Job");
break;
case "V":
console.log("Okay Job");
break;
case "D":
console.log("Bad Job");
break;
case "F":
console.log("Horrible Job");
break;
default:
console.log("Nonexistent");
}
```

Ternary Statement
The one-line shorthand for an if...else statement.

```
var goToWork = day === "Saturday" || day === "Sunday" ? false : true;
```

Truth Table
A table for determining whether something will evaluate to true or false.

A	B	A AND B	A OR B	NOT A
false	false	false	false	true
false	true	false	true	true
true	false	false	true	false
true	true	true	true	false

GUIDING QUESTIONS

1. How does an if/else statement control the logic of your program?
2. What is the difference between== and ===?
3. How does boolean logic manipulate conditionals?
4. What is a for loop?
5. What is a while loop?
6. Why would we use a switch statement?
7. What is a ternary statement?

LOGICAL OPERATORS AND COMMANDS

Command	Description
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
===	Strict equal to
==	Equal to
!==	Strict not equal to
!=	Not equal to
||	OR
&&	AND
!	NOT
break;	Exit function (necessary in switch statements)
null	Used to declar variable to have no value