

# Operators & Control Flow

---

## Challenge 1

### Step 1

Create a variable `favNumber` and set it equal to your favorite number (or any number really, I'm not checking this 😊).

Write some code that will check if the `favNumber` variable is smaller than the number `22`. Save the result into a variable called `comp1`.

Write some code that will check if the `favNumber` variable is larger than the number `4`. Save the result into a variable called `comp2`.

Write some code that will check if the `favNumber` variable is larger or the same as the number `21`. Save the result into a variable called `comp3`.

Write some code that will check if the `favNumber` variable is smaller or the same as the number `8`. Save the result into a variable called `comp4`.

Log `comp1` `comp2` `comp3` and `comp4` to the console.

### Step 2

Create a variable `name` and set it equal to your first name (or any name really, I'm not STILL not checking this. Or am I? 🙄).

Create the variables `comp5` `comp6` `comp7` and `comp8` without setting them equal to anything.

### Step 3

On this next step we will set our `comp5` `comp6` `comp7` and `comp8` uninitialized variables on separate lines of code.

First, compare the string value of `5` and the number value of `5` using the loose equality operator. Save its evaluated result into the `comp5` variable.

Next, compare the `name` variable you created in Step 2 and the string value of "Maria" using the strict equality operator. Save its evaluated result into the `comp6` variable.

Next, compare the string value of `cat` and the number value of `6` using the loose inequality operator. Save its evaluated result into the `comp7` variable.

Next, compare the string value of `8` and the number value of `8` using the strict inequality operator. Save its evaluated result into the `comp8` variable.

### Step 4

Think of how you set the `comp5` `comp6` `comp7` and `comp8` variables and see if you can figure out what their values are without logging them to your console.

Log the `comp5` `comp6` `comp7` and `comp8` variables to your console. How did you do?

If you got any of them wrong, make sure to go back and figure out why you guessed wrong.

## Challenge 2

For these next couple of steps I've provided a Variable Glossary for you to use that I don't think anyone can disagree with. Or maybe you can... I don't care. This is my course 😊👩.

## Step 1

Create a variable `value1`. Think about how the Logical `AND` Operator works. Use the Logical `AND` Operator to compare two of the variables from the Variable Glossary in a way that will evaluate to a boolean value of `true`. Save that value in the `value1` variable.

Next, Create a variable `value2`. Think about how the Logical `AND` Operator works. Use the Logical `AND` Operator to compare two variables from the Variable Glossary that haven't been used yet in a way that will evaluate to a string value. Save that value in the `value2` variable.

## Step 2

Next, Create a variable `value3`. Think about how the Logical `OR` Operator works. Use the Logical `OR` Operator to compare two variables from the Variable Glossary that haven't been used yet in a way that will evaluate to a boolean value of `false`. Save that value in the `value3` variable.

Next, Create a variable `value4`. Think about how the Logical `OR` Operator works. Use the Logical `OR` Operator to compare two variables from the Variable Glossary that haven't been used yet in a way that will evaluate to a string value. Save that value in the `value4` variable.

## Step 3

Next, Create a variable `value5`. Think about how the Logical `NOT` Operator (aka the "Bang Operator") works. Use the Logical `NOT` Operator on a single variable from the Variable Glossary that will evaluate to a boolean value of `true`. Save that value in the `value5` variable.

Lastly, Create a variable `value6`. Think about how the Logical `NOT` Operator works. Use the Logical `NOT` Operator on a single variable from the Variable Glossary that will evaluate to a boolean value of `false`. Save that value in the `value6` variable.

Log `value1` `value2` `value3` `value4` `value5` and `value6` variables to the console.

## Step 4 (Bonus Step)

For this bonus step we are going to get a little more complex. We are going to write a small program that will help an event planning company. This program will declare some data about a guest, validate that data using some Javascript operators, and then depending on the data provided, we will initialize a variable with some useful information.

First, create a variable `guest` and set it equal to an empty object.

Next, use either dot or bracket notation to add some properties to the `guest` object. You should add a `name` key with a value of the some random name, an `age` key with the value of some random age, a `has identification` key with a value of some boolean value, an `identification is real` key with some boolean value, and finally a `face matches identification` key with a value of some boolean value.

This next part is the tricky stuff.

Write some code that will do the following:

- Check if the guest is 21 or older.
- Check if the guest has identification.
- Check if the guest's identification is real.
- Check if the guest's face matches their identification.

If all of these things are true, your code should save the guest's name into a `nameToBeAddedToTheGuestList` variable. If any of these things are not true (i.e. the guest is younger than 21 or the guest doesn't have identification, etc) you should save the string *"Sorry,(person's name) you're not getting into the event."* into the `nameToBeAddedToTheGuestList` variable.

The "(person's name)" should be filled in by the `name` property from our `guest` object.

Here is the kicker, you should be able to do this using only a combination of Comparison & Logical Operators 🤖.

If you run into blocks doing this, check out MDN's documentation on [Logical AND](#), [Logical OR](#), and [Comparison](#) Operators.

## Challenge 3

In this Challenge we are going to focus on proper control flow using Comparison Operators, Logical Operators, and Conditional Statements.

### Step 1

Create a variable `num` and set it equal to some random number.

### Step 2

Write a program using only conditional `if` statements to check whether that number is negative, positive, or the number `0`.

If the number is positive log to the console the string "it's positive", if it is negative log "it's negative" and if it is `0` log "it's zero".

### Step 3

Now, refactor (rewrite) your program from Step 2 to be a bit more efficient using just `if` and `else if` conditional statements.

### Step 4

Now, refactor your program from Step 2 and Step 3 further to be even more efficient using `if`, `else if`, and `else` conditional statements.

## Challenge 4

### Step 1

Create a variable `temperature` and set it equal to some random number value.

Create a variable `kindOfDay`, but leave it uninitialized.

### Step 2

Write a program using Comparison Operators and Conditional Statements to do the following.

Check if the `temperature` is hotter than 98 degrees. If it is, set the `kindOfDay` variable to the string *"super hot"* and log to the console the message *"I'm staying inside next to the AC"*.

Check if the `temperature` is between 84 and 99 degrees (not including 84 and 99). If it is, set the `kindOfDay` variable to the string *"a little warm"* and log to the console the message *"I'll head out for a bit, but I'm wearing shorts"*.

Check if the `temperature` is between 69 and 85 degrees (not including 69 and 85). If it is, set the `kindOfDay` variable to the string *"perfect"* and log to the console the message *"It's perfect outside. I think I'll go to the park"*.

Check if the `temperature` is between 54 and 70 degrees (not including 54 and 70). If it is, set the `kindOfDay` variable to the string *"chilly"* and log to the console the message *"I'd better bring a jacket"*.

Check if the `temperature` any temperature lower than 55 (not including 55). If it is, set the `kindOfDay` variable to the string *"freezing"* and log to the console the message *"Heck no! I'm staying inside where it's warm"*.

Finally, Check to see if the `kindOfDay` variable has been set (it is not `undefined`). If it has, log to the console the message *"Looks like it's going to be (type of day) today"*, but be sure to fill in the "(type of day)" with the value stored in the `kindOfDay` variable.

## Challenge 5 (Bonus Challenge 😬)

Being able to refactor code to make it cleaner and more efficient is a very important skill to develop as an engineer. We are going to dig into that for this Bonus Challenge.

I have given you some variables in this challenge and some conditional `if` statements that use these variables. These statements work, but they are not the most efficient.

For this challenge see if you can refactor some of this code that isn't great so that it is a bit cleaner and more efficient. Happy refactoring!!! 🍌 😊