

Operators

1. **Arithmetic operators** are operators that we use to perform arithmetic operations.
 - + (plus) operator is used to add
 - - (minus) operator is used to subtract
 - * (star or asterisk) operator is used to multiply
 - / (slash) operator is used to divide
 - ** (double star) operator is used for exponentiation/power
 - % (percentage) operator is used for modulus operation (the remainder left over after division)
2. **Assignment operators** are operators that are used to assign values to variables
= , += , -= , *= , /= , **= , %=
 - 3. **Comparison Operators** are used to compare values or variables against other values or variables
 - == operator checks if the operand on the left is of equal value to the operand on right
 - === operator checks if the operand on the left is of equal value *and equal type* to the operand on right
 - != operator checks if the operand on the left is not of equal value to the operand on right
 - > operator checks if the operand on the left is greater than that on the right
 - < operator checks if the operand on the left is lesser than that on the right
 - >= operator checks if the operand on the left is greater than or equal to that on the right
 - <= operator checks if the operand on the left is lesser than or equal to that on the right
 - 4. **Logical Operators** are used to combine more than one conditions.
 - && operator checks if the condition on left and right are true. Returns true only if both conditions are true. Else returns false.
 - || operators checked if either the condition on the left is true or right is true. Returns true even if one of the two conditions is true.
 - ! operator checks if the condition is not met.

Short-Circuit Evaluation

Short-circuit evaluation is a concept in which the compiler will skip checking sub-expressions in a compound statement (a statement with logical operators) once the value is determined.

- exp1 && exp2 will not evaluate exp2 if exp1 is **false** because if even one expression is false with an &&, the entire expression is false
- exp1 || exp2 will not evaluate exp2 if exp1 is **true** because if even one expression is true with an ||, the entire expression is true

This can be very useful when evaluating certain expressions, and should be taken advantage of where needed.

Condition statement :

5. **if - else if - else** Conditional statements are very useful to control the flow of your code.
6. **switch-case** statements are used to replace multiple if - else if conditions which check the same variable. After one of the conditions is satisfied and the block of code is executed, the control should explicitly **break** out of the switch block. Otherwise, all other conditions will be executed until either a **break** statement is found, or until there is no more code.
7. **Loops** can be used when the same block of code needs to be executed many times.

for loops have an initial value, condition based on which the loop is executed, and an incremental value.

while loops have only expression: a condition based on which a block of code is executed. This is the same type of expression as the second one in a **for loop**

Collection :

1. Array is an indexed collection. The index positions start from 0. The element in first position is at index 0, the second element is at position 1, and so on. The index of the last position will always be one less than the length of the array.
2. To iterate through arrays, there is a special type of for loop, **forEach**, which gets executed for each value in the given array.
3. To find the index position and the value, we can use the generic `Object.entries` method, which can be used with all collection objects. This maps each index position to the value.

```
let myArray = ["Jack","Jill",4,5,true,"John"]
for (const [index, value] of Object.entries(myArray)) {
  console.log(index, " - ", value);
}
```

4. **Map** object maps a key to a value. The keys have to be unique. The values can be string, int, float, or any other valid JavaScript datatype. An empty Map object can be create with the new keyword.

Module Summary

In this module, you learned that:

- JavaScript is a scripting language that enables developers to add dynamic content to web pages.
- JavaScript variables are declared using the 'let' or 'const' keywords and take their type from the value assigned.

- Program execution is controlled by statements like If...Then...Else, Switch, For loops, and While loops.
- JavaScript uses blocks of code, called functions, that can be called from anywhere in the script.
- New methods and properties can be added to an object by modifying the prototype for that object.
- Prototypes allow you to define properties and methods for all instances of a specific object.
- Client-side scripts are programs that accompany HTML documents and are used by developers to incorporate more interactive elements.
- The script tag can incorporate a script within an HTML document or call a script from an external file.
- The Document Object Model (DOM) is the programming interface between HTML or XHTML and JavaScript.
- Developers can access HTML DOM elements from JavaScript scripts using the correct DOM notation.
- APIs are often used to access HTML DOM elements in web pages.