loops:

Loops can execute a block of code a number of times.

for loop:

The JavaScript for loop is used to execute a block of code repeteatedly, until a specified condition evaluates to false. It can be used for iteration if the number of iteration is fixed and known.

The JavaScript loops are used to execute the particular block of code repeatedly. The 'for' loop is the most compact form of looping. It includes the following three important parts −

Initialization − The loop initialization expression is where we initialize our counter to a starting value. The initialization statement is executed before the loop begins.

Condition − The condition expression which will test if a given condition is true or not. If the condition is true, then the code given inside the loop will be executed. Otherwise, the control will come out of the loop.

Iteration − The iteration expression is where you can increase or decrease your counter.

JavaScript while Loop

The most basic loop in JavaScript is the while loop which would be discussed in this chapter. The while loop is an entry-controlled loop.

The purpose of a while loop is to execute a statement or code block repeatedly as long as an expression is true. Once the expression becomes false, the loop terminates.

do while loop:

The do...while statements combo defines a code block to be executed once, and repeated as long as a condition is true.

The do...while is used when you want to run a code block at least one time.

JavaScript Variable Names (Identifiers)

In JavaScript, a unique character sequence is used to name the variables called identifiers.

Here are some rules for the naming of the identifiers in JavaScript −

Valid characters − In JavaScript, a variable name can contain digits, alphabetical characters, and special characters like underscore (_) and dollar sign ($). JavaScript variable names should not start with a numeral (0-9). They must begin with a letter or an underscore character. For example, 123test is an invalid variable name but _123test is a valid one..

Case sensitivity − Variable names are case sensitive. It means Name and name are different identifiers.

Unicode support − The identifiers can also contain the Unicode. So, developers may define variables in any language.

Reserve keywords − You should not use any of the JavaScript reserved keywords as a variable name. For example, break or boolean variable names are not valid. Here, we have given a full list of the JavaScript revered keywords.