The if, while, and do-while Statements

In this lesson, you will have an overview of all flow-control statements provided by the JavaScript language.

WE'LL COVER THE FOLLOWING

📜**NOTE:** Control-flow statements have similar syntax and semantics to their pairs in C, C++, Java, and C#. This section gives you a very concise

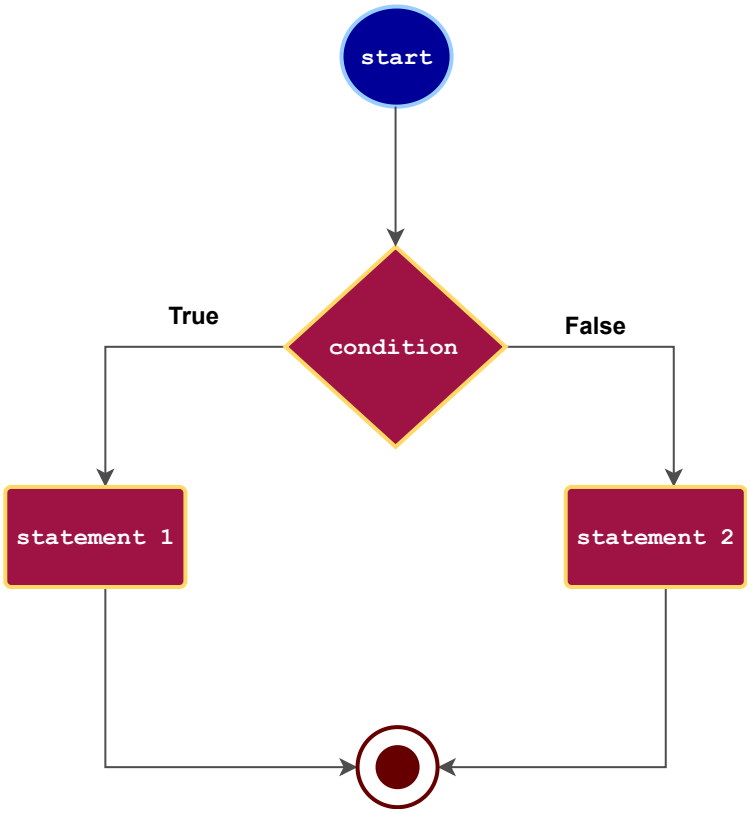
description of these statements and explains more details only upon uncommon constructs.

The `if` statement

The most common flow-control statement in almost every programming language is the `if` statement.

Illustration

Here is the concept explained in the form of an illustration:



Syntax

The syntax for the for loop is given below:

```
if (condition) true_statement [ else false_statement ]
```

if statement syntax in JavaScript

The condition can be any expression; it will be automatically converted to a Boolean value with the `Boolean()` casting function. When this expression is true, `true_statement` is executed; otherwise `false_statement`.

As shown by the syntax description, the `else` branch with the

`false_statement` is optional.

Examples

Here are a few examples:

```
if (wheel > 4) {
    console.log("This must be a big car!");
}

// ...
if (horsepower < 200) {
    console.log("It's an ordinary car...");
} else {
    console.log("This car must be powerful!");
}
```

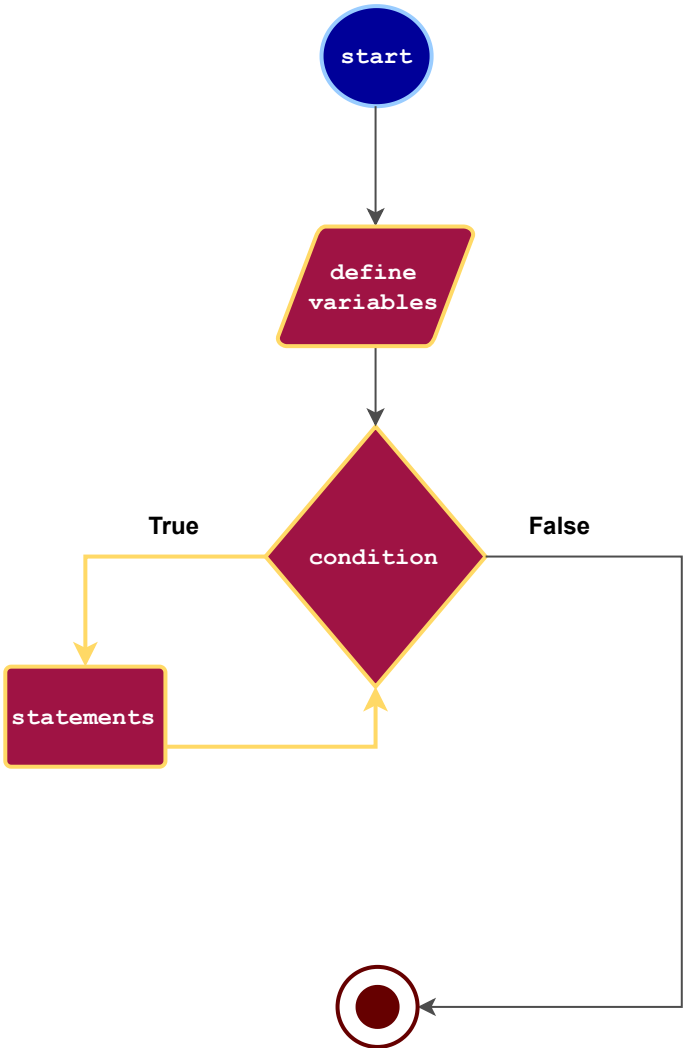
Show Useful Info

The `while` statement

The `while` statement provides a loop that can be executed zero, one, or more times depending on the loop condition:

Illustration

Here is the concept explained in the form of an illustration:



Syntax

The syntax for the while loop is given below:

```
while (condition) loop_statement
```

while statement syntax in JavaScript

The condition is evaluated before the `loop_statement` is executed. If the condition results true, the `loop_statement` is executed and the execution goes back to check the condition in a loop. This goes on while the condition results `false` (the condition fails), and then the loop is aborted.

Examples

Here is an example that displays numbers between 0 and 9:

```
var index = 0;
while (index < 10) {
```

```
    console.log(index++);
}
```

The `do-while` statement

In contrast to the `while` statement that is a pretest loop, the `do-while` statement is a post-test loop, so the loop body executes at least once:

```
do {
  loop_statement
} while (condition)
```

do-while statement syntax in JavaScript

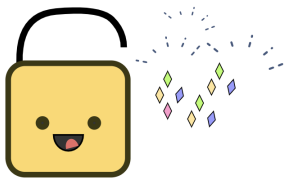
If the condition evaluates to true, the loop continues; otherwise, it is aborted.

Example

Take a look at this example that displays numbers between 0 and 9:

```
var index = 0;
do {
    console.log(index++);
}
while (index < 10)
```

Achievement unlocked!



Congratulations! You've learned how to use the `if`, `while`, and `do-while` statements in JavaScript.

Great work! Give yourself a round of applause! :)

In the *next lesson*, we will study the `for` and `for-in` loops in JavaScript.

See you there! :)