# Booleans

# In JavaScript, a boolean value is one that can either be TRUE or FALSE. If you need to know “yes” or “no” about something, then you would want to use the boolean function. It sounds extremely simple, but booleans are used all the time in JavaScript programming, and they are extremely useful. Anything that needs to be “on” or “off”, “yes” or “no”, “true” or “false”, or which just has a temporary purpose, is usually a good fit for booleans.

# Below is an example of how to use booleans in JavaScript:

|  |
| --- |
| EXAMPLE |
| var kitchenLights = false; kitchenLights = true; kitchenLights; |
| OUTPUT |
| true |

# In this example, the variable "kitchenLights" being set to "true" would indicate that the lights are on. If it was set to "false" then that would mean they are off.

# It’s useful to store booleans in variables to keep track of their values and change them over time. Booleans are used as functions to get the values of variables, objects, conditions, and expressions. In fact, Booleans are critical for conditionals to work.

# In your code, when you need to know whether or not a condition is being met before proceeding to the next step, the boolean function becomes your best friend. If *such and such* is TRUE, then do this. If it is FALSE, then do something else.

# Consider something simple like making a sandwich. In your head, you will actually be going through a series of booleans and conditionals:

# Do I have bread? TRUE or FALSE. If TRUE, then I can proceed to the next step of whether or not I have mustard, ham, etc. If FALSE, then I’ll need to go to the store to get some bread. And so on and so forth.

# You can never assume anything when programming - you have to write the logic explicitly, and a large part of that is simply knowing whether or not a condition is being met. Hence the importance of booleans in JavaScript.