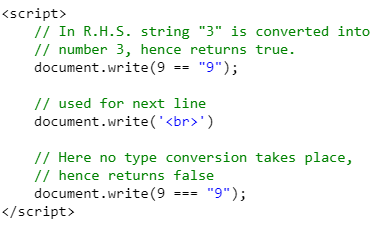
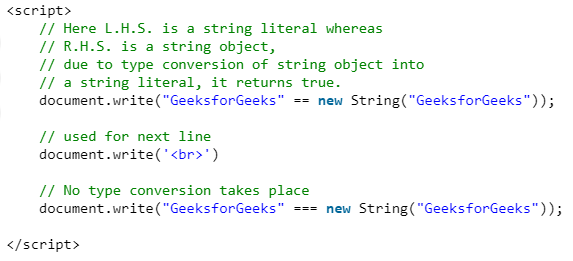
* The **+** converts the variables to strings, while the **–**, **\*,** /or otheroperator converts the variables to numbers.

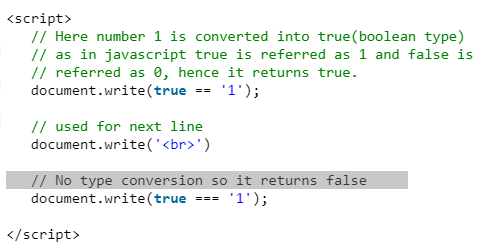
 🡺 

 🡺

* **Equality operators:**
  + **==** operator test for abstract equality and does the necessary type conversion before the equality operation.
  + **===** operator test for strict equality and does not the necessary type conversion before the equality operation.



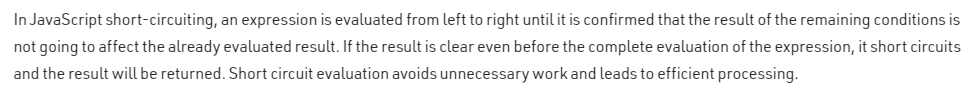


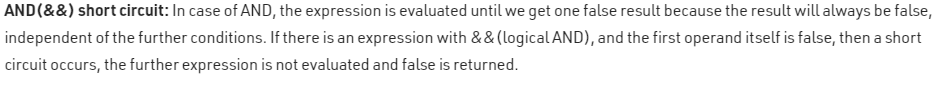


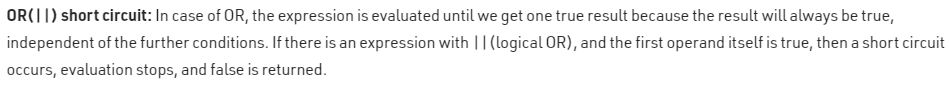
**All cases output:**



**Short Circuiting**



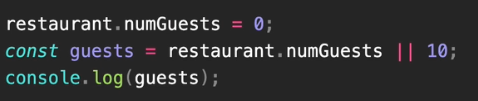




**Nullish Coalesric Operator**

This operator is used for a specific case. Since 0 and “” is a falsy value, in turn is treated as false in conditional operators. So, if we try to process a value of zero or empty string, we would get false and, in many evaluations, it will create bug. To solve this problem, we can use nullish coalesric operator. This only treats null and undefined as false.

For example. Using normal OR operator,



OutPut: 

Using nullish coalesric operator,



OutPut: 

From the example, we can see the problem that can arise.