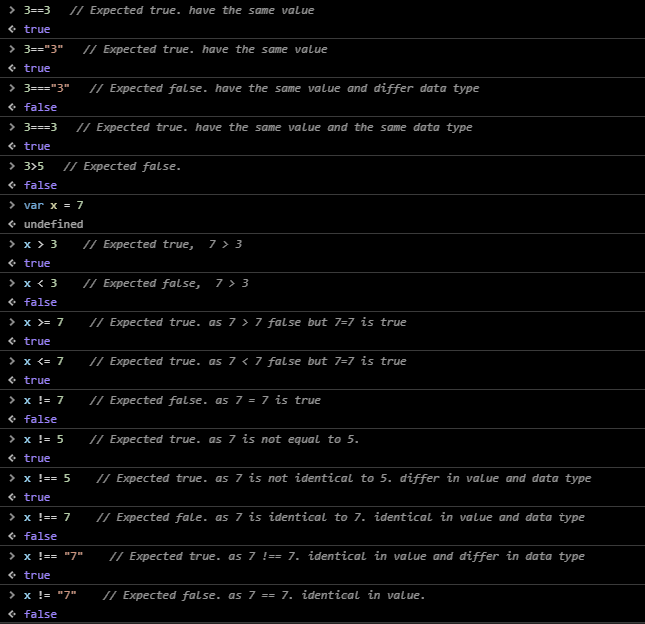
**Comparison operators**

Comparison operators are math signs used to **compare** between numbers and variables according to “**dictionary**” or “**lexicographical**” order (**Unicode** **order**). And return **true** if the condition is right else it return **false**. It also called “**Relational operators**”

|  |  |  |
| --- | --- | --- |
| **comparison Operators** | **sign** | **condition** |
| **Less than** | **<** | Is the first value smaller than the second |
| **Large than** | **>** | Is the first value greater than the second |
| **less than or equal to** | **<=** | Is the first value smaller or equal to the second |
| **greater than or equal to** | **>=** | Is the first value greater or equal to the second |
| **strict non-equality (loosely) operator** | **==** | Compare between two operators according to **value** not data type. |
| **not equal to** | **!=** | Is the first data not equal to the another in the **value** |
| **strict equality** | **===** | Compare between two operators according to **value** and **data type**. |
| **strictly not equal to** | **!==** | Is the first data not equal to the another in the **value & data type** |

**Note:** when compare between string value and number value, false will return

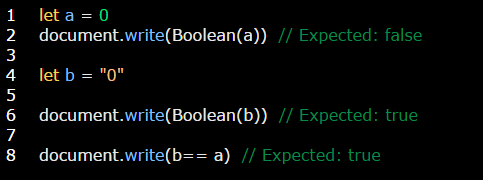
**Example:**



**Interview question:**

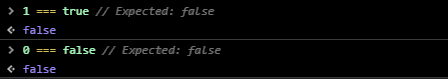
**Question:** **when Boolean true value loosely equality (==) to Boolean false?**

**Answer: by using Boolean constructor one with 0 and another one with “0”.**



**Question:** **in Boolean data type (false == 0, true == 1), how to differentiate between 1 and true or 0 and false?**

**Answer:** **By using strict equality operator.**



**Comparison between null and undefined values**

* Null is a special value doesn’t do mathematical operators with any operators except comparison operators.
* Undefined gets converted to NaN and NaN is a special numeric value which returns **false** for all comparisons.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Operator** | **Value** | **Result** |
| **null** | **>=** | **0** | **true** |
| **<=** | **0** | **true** |
| **!=** | **0** | **true** |
| **!==** | **0 and undefined** | **true** |
| **==** | **undefined** | **true** |
| **==** | **0 or any value** | **false** |
| **Any other operator** | **0 or any other value** | **false** |
| **undefined** | **Any operator** | **Any value except null** | **false** |

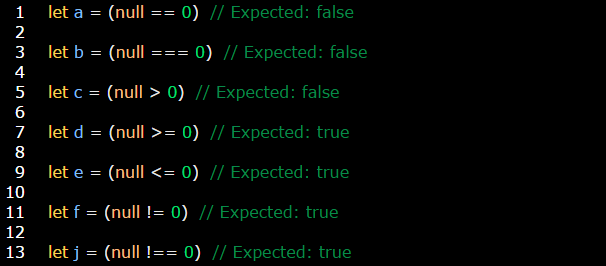
**Interview Question:**

**Question 1:** **Why null with all operators and values is false while with (>= and <=) comparison operators is true?**

**Answer:** null is a special value doesn’t enter in any mathematical equations but when we use (>= and <=) comparison operators it was converted to **0** number so in this case we can treats it as a number.

**Note:** conversion process occurs only with comparison operators.

**Question 2: Is all comparison operators give true value with null? Answer: False only >= and <= give true value.**

****