* const arr = ['A', 'N', 'U'];  
  arr[10] = 10;  
  console.log(arr.length);

*Explanation:*When you set arr[10] = 10, JavaScript automatically extends the length of the array to accommodate the new element at index 10. The elements between the existing indices and the new index are filled with undefined

**Output:11**

* let array = [1, 2, 3, 4, 5];  
  array.length = 0;  
  console.log(array)

*Explanation:*Initially, array is [1, 2, 3, 4, 5].

When you set array.length = 0;, the array is truncated, and all elements are removed.

The resulting array is now [].

Logging array to the console will output [].

**Output:[]**

* What will be the output of the following code?

console.log(2 + true);

Explanation:

When JavaScript encounters a + operator with a number and a boolean, it automatically converts the boolean to its numeric equivalent (true becomes 1). Therefore, the expression becomes 2 + 1, which evaluates to 3.Where as false returns 0

**Output:3**

* What will be the output of the following code?

console.log([] == ![]);

![] evaluates to false because the logical NOT operator (!) converts the operand to a boolean value and negates it. An empty array [] is truthy in JavaScript, so ![] is false.

**Output true.**

* What will be the output of the following code?

console.log(NaN === NaN);

*Explanation:*NaN (Not a Number) is a special numeric value in JavaScript, and interestingly, NaN is not equal to itself. Therefore, the expression NaN === NaN evaluates to false.

**Output :false.**

* What will be the output of the following code?

console.log('5' - - '3');

*Explanation:*The unary - operator can be used to convert a string to a number. In this case, - '3' converts the string '3' to the number -3. Then, the expression becomes '5' - (-3), which is equivalent to '5' + 3. JavaScript performs string concatenation since one of the operands is a string, resulting in the string '53'. Finally, the result is converted to a number, yielding 8.

**Output: 8.**

What will be the output of the following code?

console.log(null == undefined);

*Explanation:*In JavaScript, null and undefined are considered equal only when using loose equality (==). Therefore, null == undefined evaluates to true.

**Output: true.**