1. var addit = 6 + “5”;

**It will be a concatenation not addition.**

**It should return “65”**

1. var addit = “6” + “5”;

**It will be a concatenation not addition.**

**It should return “65”**

1. var x=5; x += "2";

**x+ = “2” should be evaluated as x = x + “2”**

**since x = 5 therefore it should evaluate x = 5 + “2”**

**Now, I should expect it will return a concatenation not addition.**

**The final output should be “52”**

**Graphical user interface, text, application, email

Description automatically generated**

1. 1 == “1”

**== operator won’t consider the data type**

**It will evaluate 1 == 1 (convert “1” to 1)**

**It should return true**

1. 1 === “1”

**=== operator is a strict equality operator so it will consider the data type**

**It won’t convert the “1” to 1.**

**So the result/ output should be false.**

1. “42” === “42”

**=== is a strict equality operator so data type is considered.**

**String of 42 the same as string of 42 seems truthy 😊**

**So the final result/ output should be true.**

1. 42 !== “42”

**There is no data type conversion in this scenario 42 !== “42”**

**The final result should be true because number 42 is not equal to string 42**

1. c = Math.sqrt(-9);

**As mentioned in the lecture, this will return an imaginary number.**

**Imaginary numbers cannot be represented in JavaScript.**

**Thus, the final output/ result will be NaN**

1. var sub = “1” - “1”;

**With minus, the “1” will be converted to number 1 (stated in page 286)**

**It will evaluate 1 – 1**

**So the final output/ result will be 0**

1. if ( [ ] ) { //code }; //assume that there is code in the braces and the array is defined.

**This is like saying if true.**

**The code block w/n the if statement will be executed since [] is always true.**

1. if ( 0 ) { //code }; //assume that there is code in the braces.

**0 is falsey just like (undefined, null, empty string, false, NaN)**

**Therefore, I should expect that the code block w/n the if statement won’t be executed because the condition will always return false**

1. 8675309 == ""

**It will evaluate 8675309 == false (since I have an empty string here)**

**8675309 == false is false**

**So the return/ output will be false.**

1. false == (5==="5")

**First, the logic between the parenthesis will be executed**

**It is a strict equality operator between the parenthesis so no data type conversion.**

**5===”5” should return false**

**Second, we will have false == false.**

**So the final output/ result for this will be true**