Rylee Leavitt

Dr. Tammy Robertson

COSC1350

10/09/24

Let's Practice JavaScript Variables

**Answer the following questions:**

**Did any of the type of outputs surprise you?**

No, based on the reading, I understand why strings and integers output differently based on the function its performing. for example: console.log("five" \* 2); multiplies a string by an integer = Not-a-Number

**What is happening in step 5.1 that is causing the line break?**

\n is a newline character; this will add the text "Let's get started!" in a new line, causing the line break.

**What do you think is happening in 5.2? You might need to research this one. :)**

based on previous conversations between Dr. T and I, in console.log(`half of 100 is ${100 / 2}`); ${100 / 2} is a template literal which is meant to embed the result of 100 / 2 in the string. This results in = "half of 100 is 50"

**Tell me why the output is what it is for 5.2 - 5.8.?**

* 5.1 =\n is a newline character; this will add the text "Let's get started!" in a new line resulting in:
  + Hi Dr. T!
  + Let’s get started!
* 5.2 = uses a template literal to embed the result of 100 / 2 in the string. Resulting in:
  + half of 100 is 50
* 5.3 = adds the values of the string together = concatenate: (def) links things together in a chain or series. This log function results in:
  + Concatenate
* 5.4 = Multiplies an integer (8) with no data (null) which equals 0; basically 8\*0 resulting in:
  + 0
* 5.5 = Subtracts an integer from a string. - turning “5” into a number 5, so 5-1 =4 resulting in:
  + 4
* 5.6 = Adds a string and an integer. ("5" + 1) does NOT result in 6 because 5 is still a string in this context. The result of this console.log function is:
  + 51
* 5.7 = Multiplies a string by an integer. ("five" \* 2) is impossible without five being an integer, thus the result is:
  + Not-a-Number
* 5.8 = == is the equality operator; (false == true); false does not equal true. Thus, the output is:
  + false

*Output: Rylee Leavitt*