Part 1:

1. it will send a false when it compares 3 with “3birds” because they are a different type. Logically, when you use the && symbol, if 1 statement is false. The rest will also be false.

2. The `&&` operator has higher precedence than `||`, so it is evaluated first. false && true evaluates to false. Now, the expression simplifies to true || false, which evaluates to true.

3. "Alberto" === “Beto" evaluates to false because it is strictly equal, and those types need to be the same. "Alberto" === "Alberto" evaluates to false. Since both are false, they are in parentheses. The first statement will be false. Similar to the second statement, but in this case, "Gonzalez" === "Gonzalez" evaluates to true. So, the second statement is true. Therefore, since the operator is && and one of them is false, the final answer is false.

4. The parrot has escaped. It was last seen on a bicycle.

5. The full name will be Jorge Conquistador. The first parentheses will check "Jorge Conquistador" === "Jorge," and it evaluates to false. Then, it will check for the next one where "Jorge" !== "George" is evaluated as true because if the first string type is not the same as the second string type, It should return true. "Jorge Conquistador" === "Conquistador" evaluates to false. True && false evaluates to false. For nickname = "Conquistador" === 42 evaluates to false. false || false evaluates to false.

6. 42 (number) is not strictly equal to “ 42” (string), so this evaluates to false. 42 (number) is loosely equal to “42” (string), so this evaluates to true. false && true evaluates to false. In lexicographical order, the number 42 is considered less than any string, so this evaluates to `true`. Then, it compares the number 42 with the string “234” by converting “234” to a number, which evaluates to false. true || false evaluates true.

7. The whole function inside the bracket is 504 + 6.5 + 22 = 532.5 % 2 = 0.5 and it is not equal to 1 since 0.5 less then 1 and the program is checking double with integer.

8. 3^3 = 27, so this is true. cos(pi) is -1, so this is false. The second equation evaluates to a value that is very close to 1, but due to floating-point precision, the exact result might not be exactly 1. Which is false. Since The first part is already true, then the rest is true because we are using operation OR.

9. sentence.substring(4,9) results in “world". "world" === "world" evaluates to true. The length of "The World is a Beauty" is 20, so this evaluates to true. sentence.substring(0,4) === "The world is round”.substring(0,4). Both substrings result in ‘The”. So it is true. true && true && true evaluates to true.

10. Since the smallCar is undefined, the whole string is false.