Conditionals and Logical Operators

# Week3 Day 1 15/5/23

# Conditionals

* Conditionals in JavaScript are used to decide which block of code to run depending on some “condition”.
  + There are three fundamental conditional statements that we can make use of in JavaScript:
  + if
  + else
  + if else
  + “If” statements can be used on their own to run some code that passes a condition, but “else” and “if else” statements will always come after an “if” statement to run some other block of code if the initial condition is false.

# If Statements.

* + Conditionals syntax:

A picture containing text, font, screenshot, line

Description automatically generated

* + First, we write the if statement, then we include our conditional statement between paratheses. In this case the statement is 1 === 1.
  + The conditional statement should always evaluate to either true or false.
  + If the condition is true, the code within the “if” statement is executed, otherwise the program continues.

# If statements continued…

* We can also use variables in our conditional statement.

1. Try it yourself, create your own if statement.

A picture containing text, font, screenshot

Description automatically generated

# Else statements

* Conditionals syntax:

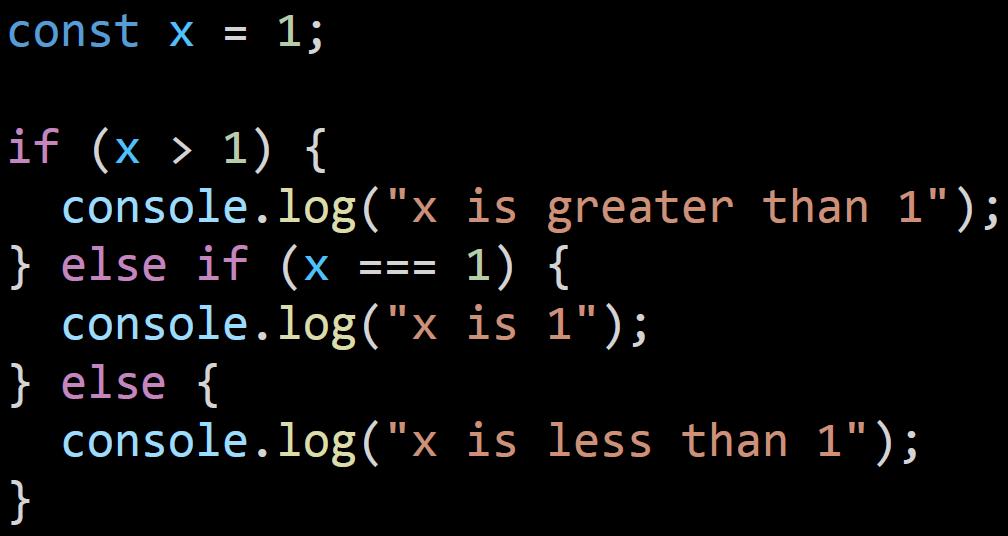
A picture containing text, font, screenshot, number

Description automatically generated

* In this example we’ve added an “else” statement.
* If the condition next to the “if” statement is true then it will console.log that x is greater than 1, but if the condition is false then it will console.log that x is less than 1.

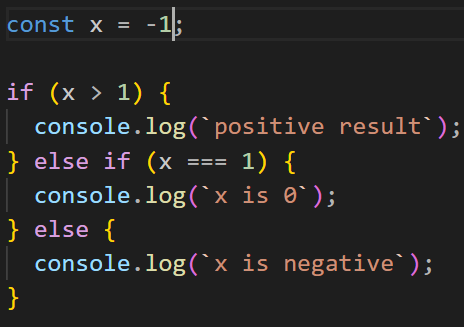
# If else statements

* You might’ve noticed something wrong with the last if statement
* Changing the value of x to 1 will give us “x is less than 1” in the console log. But that isn’t true. How can we fix that?
* We can add an if else statement.
* We can add another condition that will check to see if x is === to 1 and will console.log a different string.



# Exercise 2

* Write a JavaScript conditional statement that checks whether a variable is positive, negative or 0 and logs an appropriate message to the console.



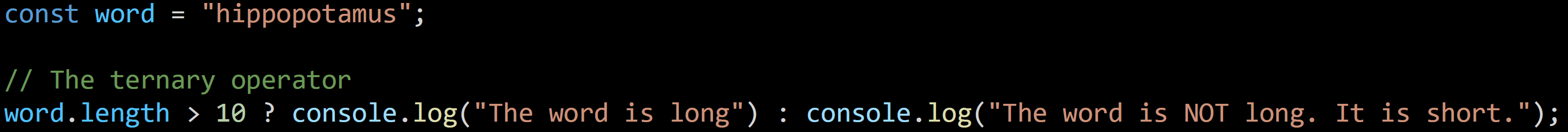
Exercise 3

* + If we take a string like 
    - * + If we run 
  + It will give us  in the console.
  + If I wanted to get just the first letter I could run
  + This would give me 
  + Use this knowledge to create an if statement that will log to the console True if the string starts with an “a” and will log to the console False if it does not.

1. A screen shot of a computer code

   Description automatically generated with low confidence

# Ternary Operator



* + In this example we can see an alternative conditional statement called the ternary operator. This statement can be broken into 3 parts.
  + The first part is the condition that will either be true or false (word.length> 10)
  + Then we have the “?” section. Anything that comes after the question mark will execute if the condition is true.
  + Lastly, we have the “:” section. Anything after the colon will execute if the condition is false.

# Ternary vs Traditional

A picture containing text, screenshot, font

Description automatically generated

* The reason for using the *ternary operator* over a traditional *if* statement is to keep things simple and concise (takes up only one line of code instead of 5).
* A general rule of thumb is that if there are only two possible outcomes e.g true or false, then a ternary is ok to use. But if there are multiple possible outcomes then using the traditional if else statement is preferred

# Ternary exercise

* Convert the following if else statement into a ternary

A screenshot of a computer screen

Description automatically generated with medium confidence