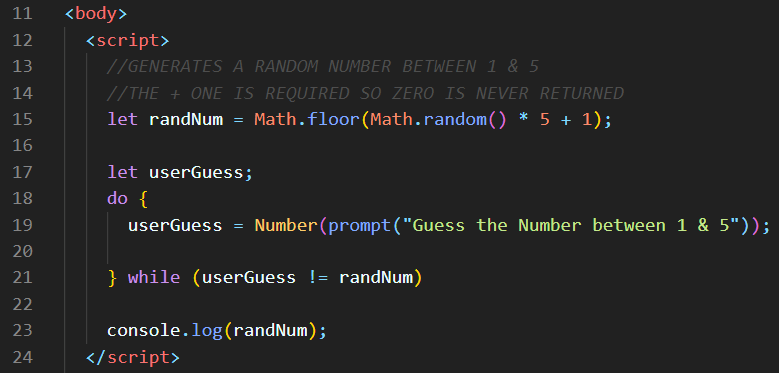
Create a new file named ***guess.html*** and inside the opening **<body>** tag insert the code below:

1. **Math**.**random** \* **5** generates random number between **1** & **5** - The **+ 1** has been added at the end in order that ***zero*** cannot be returned
2. **Math**.**floor** rounds the number down to the next whole number (*e.g.****3.14*** *would be rounded to* ***3***)
3. The variable **userGuess** needs to be declared outside the ***do…while…***loop
4. **! =** is a logical operator meaning **NOT EQAUL TO**

****

Check in the browser

You can also open

The console by pressing

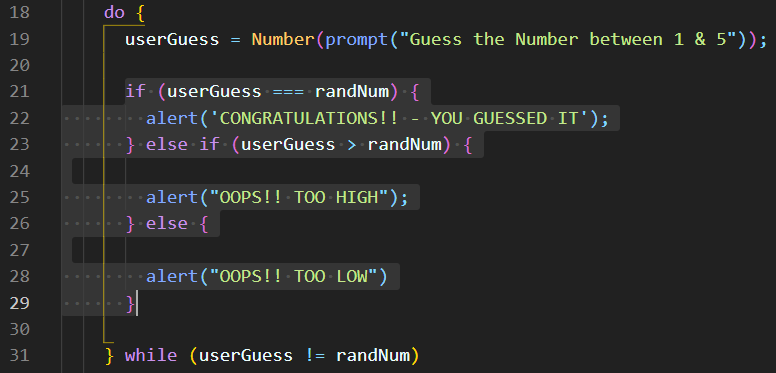
**F12** – which will reveal

The random number

Using **console.log ()**

Inside the **do…while…**loop add the following **IF…else…IF** conditions

The user is now receiving feedback as to how close they are to guessing the number

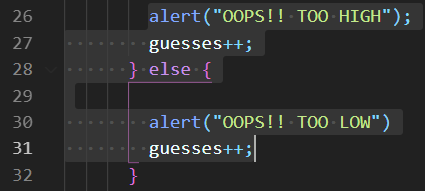


Keep track of how many guesses the user takes



Declare a new variable named **guesses** to store the number of guesses taken

**NOTE:** The **guesses** variable is assigned an initial value of **1** in case they guess the number correctly first time



Now increment the **guesses** variable by one using

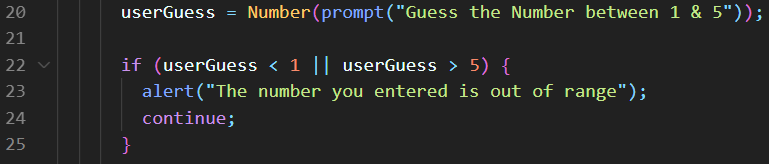
the **++ increment operator** each time they get it wrong

Test in the browser/console

Let’s do some validation if the user enters a number that is below **1** or above **5**

Add an **IF condition** that will check to see if the **userGuess** is in the required range – the **2** pipes now added **||** means if **ANY** of the **2** conditions return **true –** run the code below

The **continue** keyword stops the loop from running the **2** messages (“*TOO HIGH, TOO LOW*”) and jumps back to the start again and runs the prompt box – if you leave the **continue** keyword out it will run the messages

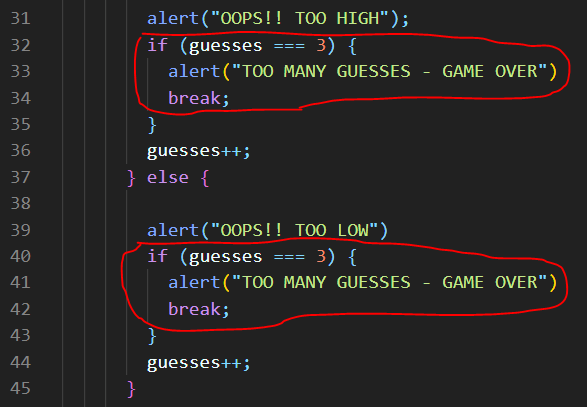


**NOTE - THE WHILE LOOP CONDITION MUST ALSO BE UPDATED -** Thepipes(**||**) in the **while** condition will keep the loop running if **ANY** of the conditions return **true**



**End the game if they guess too many times** – it’s set to **3** here just for convenience when testing - you can change to suit

**NOTE** – The **break** keyword – this will break out of the loop and end the game after **3** guesses



**TASKS**

1. Add **IF…ELSE…**validation to show a **unique**

message when the value entered is too

high or too low

1. Add validation to only allow numbers to be

Entered – **isNaN()**

1. Add an exit message when the game ends

when they’ve had too many guesses

e.g.

**“THANKS FOR PLAYING RANDOM GUESS….”**