**Guess My Number**

Event Listener

An event listener, listens for a particular event to happen. Here is how to implement one;

A picture containing graphical user interface

Description automatically generated

In this case, we want to ‘listen’ for the point at which the ‘check’ button is clicked by the user.

First we need to select the button. We can do this by using the ‘class’ from our index.html file;

<section class="left">

<input type="number" class="guess" />

<button class="btn check">Check!</button>

</section>

Here we simply need the class ‘.check’. Notice that it is ‘btn check’ but we do not need the first btn.

To select the class we use the following syntax;

document.querySelector(".check")

Now we need to add the ‘event listener’. In this case, it will be when someone clicks on the ‘check’ button. The .addEventListener() is a method so we need to call it using parentheses. Within the parentheses we implement our argument.

document.querySelector(".check").addEventListener('click')

The first argument we pass into parentheses is ‘click’, which will simply listen for the user to click on the button. Now we need to tell the event Listener what to do. We need to specify the reaction to the event being met. We do this by defining a function. The function will be called the **event handler**. This is how the syntax should look;

document.querySelector(".check").addEventListener("click", *function* () {});

Now we need to specify what happens between the curly braces. In this instance, we want to log to the console what has been typed into the input field above the check button.

document.querySelector(".check").addEventListener("click", *function* () {

console.log(document.querySelector(".guess").value);

});

In the console log we select the document and then we select the class which is guess. We then want to print the value to the console, hence; ‘.value’. Remember the event handler is the function that prints to the console, in this instance.

Instead of logging to the console, we could save the value to a variable;

document.querySelector(".check").addEventListener("click", *function* () {

*const* guess = document.querySelector(".guess").value;

});

**Numbers entered by the user are saved as strings**

This is important to remember. Let’s say the user guessed the number 2. We can use the console.log function to show the ‘type’ of variable;

document.querySelector(".check").addEventListener("click", *function* () {

*const* guess = document.querySelector(".guess").value;

console.log(typeof guess);

});

Graphical user interface, text, application

Description automatically generatedAnd here in the console, we can see that 3 is ‘3’ the string and not the number. This is a number guessing game, not a string guessing game. So we will have to convert the string into a number. Here is how;

document.querySelector(".check").addEventListener("click", *function* () {

*const* guess = *Number*(document.querySelector(".guess").value);

console.log(typeof guess, guess);

});

Graphical user interface, application

Description automatically generatedAnd now we can see that we have converted the string to a number using the number() method. Remember that Javascript methods are actions that can be performed on objects. A Javascript method is a property containing a function definition.

**Check for a Value**

This is a number guessing game. Perhaps the user failed to enter a value before clicking check. We can programme a response accordingly. We can use an if statement.

If there is no value added we would get;

Printed to the console. Because 0 is a falsy value, we can use some simple logic to determine if anything has been written in the check box;

We can use the ‘!’ logical not operator;

if (!guess)

Here we have written if the guess value is ‘true’.

Here is how the code block will look;

if (!guess) {

document.querySelector(".message").textContent = `No Number! 🙈`;

}

});

The if statement will only execute if the ‘guess’ statement is evaluated to a truthy value. Because a value of 0 is falsy, in this case, it would evaluate to false. We use the ‘!’ logical not operator to invert the value; in this case, ‘falsy’ would be true. Hence if nothing is entered into the check number box before clicking check, we would end up with the ‘no number’ message.