# Section 5 - Do...while loop

The third kind of loop we are going to be looking at is the **do-while loop**.

Do-while loops are similar to while loops, with only one real difference.

**Do-while loops will run the code at least once.**

Now to understand this behaviour, let’s look at a while loop.

The JavaScript engine reads this code from top to bottom, so in order, the while loop will:

1) Check the condition.

2) If the condition is true, run the statement and

3) Increase i by 1.

//While loop printing out odd numbers

let i = 0;

while(i <= 5){

if(i % 2 !== 0){

console.log(i);

}

i++;

}

In the **do-while loop** below, the exact same logic as the while loop is implemented, however it will read in this order:

1) execute this code.

2) Check if the condition is true, then run the code again.

//Do-while loop printing out odd numbers

let i = 0;

do{

if(i % 2 !== 0){

console.log(i);

}

i++;

} while(i <= 5)

So do-while loops run the code *before checking if the condition is true*.

To test this, we can set i = 9;

The while loop will check if 9 is less than 5, see it as false, then not run the code.

The do-while loop will run the code, see that 9 is an odd number, then print it out. **Only then** will it check if 9 is smaller than 5, it will return false, then break out of the do-while loop.

# Section 6 - Infinite Loops

One thing to beware of when writing loops, is creating an infinite loop.

As the name implies, the loop executes forever, however if you create one, **you are going to crash your browser or computer**.

Write out the following and check the console.

let i = 0;

while(i < 5){

console.log(“Hello World”);

// i++;

}

Now, because we don’t increase i (as it is commented out, and commented code will not run), it will forever be less than 5, creating and infinite loop.

\* Note that the browser also shows a spinning loader, showing that it is constantly loading. The only way to stop this is to close the page or force close the browser.

More examples of an infinite loops:

while(true){

//code to execute

}

do{

//code to execute

}while(true)

Because true will always be true, these will run forever. (Unless there is logic within the code to break out of them)

If you forget to increment in a for loop, it will also run forever:

for(let i = 0; i < 10;){

//code to execute but runs forever

}

Also if you increment the wrong way:

for(let i = 0; i < 10; i--){

//code to execute but runs forever

}

Be mindful when creating loops, and don’t create infinite loops. You will just end up breaking your page, making it completely unresponsive.