MISSION READY

DARE TO

DEVELOP

Loops

Ewan Zhang

JEROME
ENGINEER AT MICROSOFT

Loops in JavaScript

- Loops are a fundamental concept in programming.
- They allow programmers to perform some action or run some code a specific number of times, or if needed, forever.
- The main types of loops in JavaScript include:
  - For loop
  - While loop

For loops

- Let’s say we wanted to log to the console the numbers 1 – 5.
- Without loops:

```
console.log(1);
console.log(2);
console.log(3);
console.log(4);
console.log(5);
```

- Violates the DRY principle (Don't Repeat Yourself)

Using a For loop

- A for loop in JavaScript looks like this:

```
for (let i = 0; i < 5; i++) {

}
```

(initialExpression; conditionExpression; incrementExpression)
Arg1 Arg2 Arg3

- We use the keyword "for" to start the for loop and then give it 3 arguments.
- Arg1: `let i = 0`; this is our initial statement which is executed once before loop begins. We are defining a variable i (short for index) and setting its value to 0
- Arg2: `i < 5`; This is our conditional statement. The loop will continue to run as long as the condition evaluates to true, and in our case: as long as `i` is less than 5
- Arg3: `i++`; this is our incrementing statement. The `++` operator increments our "`i`" variable (adds 1 to the value of `i`) at the end of each loop. `i++` is the same as writing `i = i + 1`

For loops continued…

- Let’s say we wanted to log to the console the numbers 1 – 5.
- Without loops:

```
console.log(1);
console.log(2);
console.log(3);
console.log(4);
console.log(5);
```

- With loops:

```
for (let i = 1; i <= 5; i++) {
    console.log(i);
}
```

Exercise 1

- Try running a loop that logs to the console the numbers 1 – 10

Exercise 2

- Try running a loop that logs to the console the numbers 10 – 1 on each console line (so in reverse this time).

Exercise 3

- We know that if we have a string we can use `myString[0]` to get the first letter, `myString[1]` to get the second letter etc.
- Use a loop to log to the console the each letter of “Responsiveness”

Exercise 4

1. Loop through the following foods object using the `for` loop. Log to the console each element of this Array

```
const foods = ["salad", "pie", "pickles", "scones"];
```

2. Challenge: Within the loop, log each element to the console inside a string which says "I like .....".

Iterating over elements of an array

- The **for...of** loop iterates over the elements of an array.

```
const cars = ["tesla", "jaguar", "ford"];

for (const car of cars) {
  console.log(car); // tesla, jaguar, ford
}
```

Exercise 5

1. Loop through the following foods object using the `for..of` loop.

```
const foods = ["salad", "pie", "pickles", "scones"];
```

2. Within the loop, log each element to the console inside a string which says "I like .....".

While loops

- Another loop we could use is called a while loop. It works in a similar way to the for loop but allows more freedom over the loop.
- Syntax:

```
while (condition) {
    console.log('something');
}
```

- The condition can be any condition that evaluates to either true or false, similar to the for loop.
- If the condition is true, the block of code will keep running, when the condition becomes false, the code will stop running.

While loop example

- Let’s take a look at an example of how we can use a while loop

```
let guess;
const secretNumber = 3;
while (guess != secretNumber) {
  guess = prompt();
  console.log(`It's a ${guess}`);
}
```

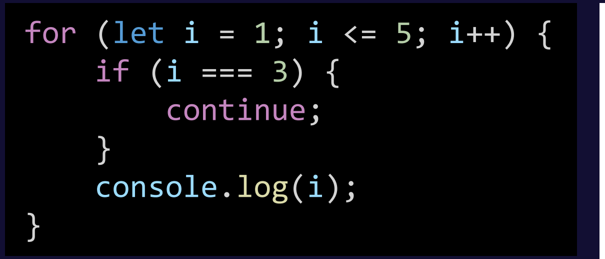
- We are declaring `guess` but not defining it
- We then start our loop and tell it to keep running as long as `guess` is not equal to 3
- Inside our loop we are defining guess with whatever the user inputs into a prompt()
- As long as the user doesn't type 3 into the prompt(), the loop continues
- When the user enters 3 into the prompt(), the loop ends

Exercise 6

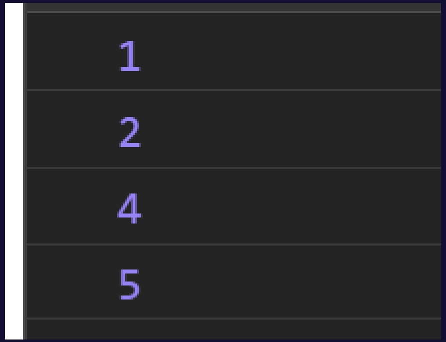
- Create a while loop that logs to the console the user's guess. But add some conditional logic where if the user types in the correct number, it logs some kind of success message to the console.

Breaks and Continues

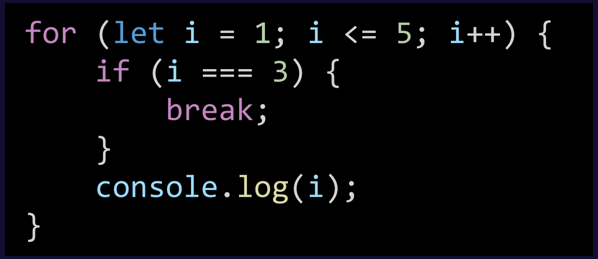
- When we are in a loop there may be times where we want to skip an iteration of the loop or stop the loop completely based on some condition.
- We can do this with `break` and `continue`
- To use a `break` or a `continue` all we need to do is add the key word into our loop



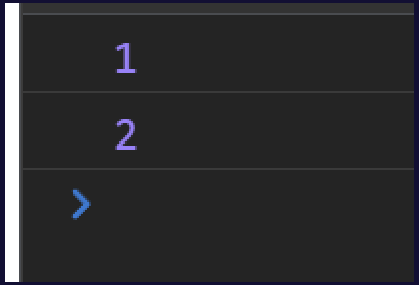
```
for (let i = 1; i <= 5; i++) {
    if (i === 3) {
        continue;
    }
    console.log(i);
}
```



```
1
2
4
5
```



```
for (let i = 1; i <= 5; i++) {
    if (i === 3) {
        break;
    }
    console.log(i);
}
```



```
1
2
>
```

Breaks in while loops

- At times we may not know exactly how many loops we want to go through but we may have an end goal in mind.
- Let’s look at this example

```
let x = 0;

while (true) {
    if (x === 3) {
        console.log('x is now 3')
    }
    if (x === 10) {
        console.log('x is now 10')
    }
    if (x === 15) {
        console.log('x is now 15, goodbye...')
        break;
    }
    x++;
}
```

- In this example, we want the value of x to be 15 but we want to hit some milestones along the way and log a message for them.
- We can set the condition of the while statement to true, this allows it to run on a loop forever.
- But when we get to our goal of 15 we can set a break and stop the loop.

Exercise 7

- Use a for loop to loop through a string: "Hell@ the#e".
- If you find a symbol in the string i.e. !, @, #, $ then log to the console an error message and break the loop

Iterating over properties of an <u>object</u>

- The `for...in` loop iterates over the properties of an object.

```
const student = {
  name: "Rob",
  age: 5,
  isAdmin: true,
};

for (const key in student) {
  console.log(key); // name, age, isAdmin
  console.log(student[key]); // Rob, 5, true
}
```

Exercise 8

1. Loop through the following user object using the `for..in` loop.

```
const user = {
   name: "John",
   age: 5,
   isAdmin: true
};
```

2. Within the loop, log the value of the key `age`.

Bonus: Exercise 9

1. Create an object `myFavNumbers` with three of your favourite numbers as values (any names for keys).
2. Loop through the object using `for..in` and find the sum of the numbers.
3. Log the sum to the console.

MISSION READY

www.missionreadyhq.com

DARE TO
DEVELOP

Thank you | Ewan Zhang