Theory: While loops

10 minutes 0 / 5 problems solved

Skip this topic Start practicing

600 users solved this topic. Latest completion was **1 day ago**.

Perhaps you had time to make yourself a cup of coffee before starting this topic, which means that you are drinking your coffee while reading this. Meanwhile, the weather is changing outside; maybe it started to rain after a cloudy morning. While it is raining, the pavement is getting wet, and the puddles swell. These processes, all based on the construction "While...(condition)", illustrate the idea of `while` loops.

A **loop** is an action (a part of code) that is repeated a certain amount of times. The `while` loops keep going while a certain condition is true, for example, "while you are reading the topic", or "while it is raining". Some conditions are always true. For instance, if you write `while (5 < 6)`, you will get an **infinite loop**, since `5` is always less than `6`. Other conditions can be run only a finite number of iterations.

§1. While syntax

The `while` loop has the following syntax:

```
while (condition) {
...
}
```

It runs while the `condition == true`. Once the value in parentheses becomes *false*, the loop terminates.

Look at the following example:

```
let n = 2;
while (n < 100) {
    n = n * n;
    console.log(n);
}
```

Eventually, we get this answer:

```
4
16
256
```

How did it happen? Let's consider each step of the iteration in detail.

In the beginning, the variable `n = 2`. When we enter the while loop, we check straight away if the `n < 100`. The condition is still true, that is why we enter the loop. Afterwards, we square `n` and the product is displayed, which is how we get `4` in the first line.

At the second iteration, we again check the condition before entering the loop. `4 * 4 = 16`: that's the second line. Since `16 < 100`, we enter the loop for the third time, multiply `16` by `16` and get the result `256`. Now, before we try to enter the loop for the fourth time, we notice that `256` is not less than `100`, so the loop ends.

§2. Do...while syntax

There is another kind of loop which runs while the conditions are true: it is called the `do...while` loop. It has the following syntax:

```
do {
...
} while (condition);
```

The example above can be rewritten using the `do...while` construction:

Current topic:

While loops ···

Topic depends on:

✕ Conditional operators ···

✕ Increment and decrement ···

Topic is required for:

Break and continue ···

Table of contents:

```
let n = 2;
do {
    n = n * n;
    console.log(n);
} while (n < 100);
```

It is not hard to see that the conclusion will be just the same as it was with `while`:

```
4
16
256
```

So what is the difference between these, and why would we need two types of loops that work the same?

§3. While *Vs* do...while loops

Actually, there is a difference. The example below will help us spot it.

```
let n = 256;
while (n < 100) {
    n = n * n;
    console.log(n);
}
```

What will be displayed? That's right: nothing! The accuracy of the condition is checked before entering the loop. So what about `do...while`? Run the following code in the console:

```
let n = 256;
do {
    n = n * n;
    console.log(n);
} while (n < 100);
```

You'll see for yourself that the screen will display `65536`!

So why does it happen? The thing is, in the case of `do...while`, the condition is checked *after* entering the loop and running the first iteration. This guarantees that the loop runs at least once, even if the condition is false from the beginning.

§4. Conclusion

In this topic, we have learned a new term: loops. We looked at two types of loops, `while` and `do...while`, and saw the difference between them. We also figured out that if the set condition is always true, we get an infinite loop.

Now it's time to challenge yourself and put this knowledge into practice.

Report a typo

54 users liked this theory. 0 didn't like it. **What about you?**

😍 🙂 😐 🙁 😡

Start practicing

Comments (0) Hints (0) Useful links (0) Show discussion