TS TypeScript TUTORIAL

TypeScript while

Summary: in this tutorial, you will learn how to create a loop using the TypeScript `while` statement.

Introduction to the TypeScript while statement

The `while` statement allows you to create a loop that executes a block of code as long as a condition is `true` .

The following shows the syntax of the TypeScript `while` statement:

```
while(condition) {
    // do something
}
```

The `while` statement evaluates the condition **before** each loop iteration.

If the `condition` evaluates to `true` , the `while` statement executes the code in the body surrounded by the curly braces (`{}`).

When the `condition` evaluates to `false` , the execution continues with the statement after the `while` statement.

Since the `while` statement evaluates the `condition` before its body is executed, a `while` loop is also called a pretest loop.

To break the loop immaturely based on another condition, you use the `if` and `break` statements:

```
while(condition) {
    // do something
    // ...

    if(anotherCondition)
        break;
}
```

If you want to run a loop a number of times, you should use the TypeScript `for` statement.

TypeScript while statement examples

Let's take some examples of using the TypeScript `while` statement.

TypeScript while: a simple example

The following example uses the `while` statement to output a number to the console as long as it is less than 5:

```
let counter = 0;

while (counter < 5) {
    console.log(counter);
    counter++;
}
```

Output:

```
0
1
2
3
4
```

How it works:

- First, declare a `counter` variable and initialize it to zero.

- Then, check if the `counter` is less than `5` before entering the loop. If it is, output the `counter` to the console and increment it by one.
- Finally, repeat the above step as long as `counter` is less than 5.

TypeScript while loop practical example

Let's say you have the following list element on an HTML document:

```
<ul id="list">
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
    <li>Item 4</li>
</ul>
```

The following example shows how to use the `while` statement to remove all `<li>` element of the `<ul>` element:

```
let list = document.querySelector('#list');

while (list.firstChild) {
    list.removeChild(list.firstChild);
}
```

How it works:

- First, select the `<ul>` element by its id using the `querySelector()` method.
- Then, check if `firstChild` of the `list` is available and remove it. Once the first child node is removed, the next child node is automatically promoted as the first child node. Therefore, the `while` statement removes all child nodes of the `list` element.

Summary

- Use the TypeScript `while` statement to create a loop that will run as long as a condition is `true`.