TS TypeScript TUTORIAL

TypeScript if else

Summary: in this tutorial, you will learn about the TypeScript `if...else` statement.

TypeScript if statement

An `if` statement executes a statement based on a condition. If the condition is truthy, the `if` statement will execute the statements inside its body:

```
if(condition) {
   // if-statement
}
```

For example, the following statement illustrates how to use the `if` statement to increase the `counter` variable if its value is less than the value of the `max` constant:

```
const max = 100;
let counter = 0;

if (counter < max) {
    counter++;
}

console.log(counter); // 1
```

Output:

```
1
```

In this example, because the `counter` variable starts at zero, it is less than the `max` constant. The expression `counter < max` evaluates to `true` therefore the `if` statement executes the statement `counter++` .

Let's initialize the `counter` variable to `100` :

```
const max = 100;
let counter = 100;

if (counter < max) {
    counter++;
}

console.log(counter); // 100
```

Output:

```
100
```

In this example, the expression `counter < max` evaluates to `false` . The `if` statement doesn't execute the statement `counter++` . Therefore, the output is 100.

TypeScript if...else statement

If you want to execute other statements when the condition in the if statement evaluates to `false` , you can use the `if...else` statement:

```
if(condition) {
   // if-statements
} else {
  // else statements;
}
```

The following illustrates an example of using the `if..else` statement:

```
const max = 100;
let counter = 100;

if (counter < max) {
    counter++;
} else {
    counter = 1;
}

console.log(counter);
```

Output:

```
1
```

In this example, the expression `counter < max` evaluates to `false` therefore the statement in the `else` branch executes that resets the `counter` variable to `1` .

Ternary operator ?:

In practice, if you have a simple condition, you can use the ternary operator `?:` rather than the `if...else` statement to make code shorter like this:

```
const max = 100;
let counter = 100;

counter < max ? counter++ : counter = 1;

console.log(counter);
```

TypeScript if…else if…else statement

When you want to execute code based on multiple conditions, you can use the `if...else if...else` statement.

The if…else if…else statement can have one or more `else if` branches but only one `else` branch.

For example:

```
let discount: number;
let itemCount = 11;

if (itemCount > 0 && itemCount <= 5) {
    discount = 5;  // 5% discount
} else if (itemCount > 5 && itemCount <= 10) {
    discount = 10; // 10% discount
} else {
    discount = 15; // 15%
}

console.log(`You got ${discount}% discount. `)
```

Output:

```
You got 15% discount.
```

This example used the `if...else if...else` statement to determine the discount based on the number of items.

If the number of items is less than or equal to 5, the discount is 5%. The statement in the `if` branch executes.

If the number of items is less than or equal to 10, the discount is 10%. The statement in the `else if` branch executes.

When the number of items is greater than 10, the discount is 15%. The statement in the `else` branch executes.

In this example, the assumption is that the number of items is always greater than zero. However, if the number of items is less than zero or greater than 10, the discount is 15%.

To make the code more robust, you can use another `else if` instead of the `else` branch like this:

```
let discount: number;
let itemCount = 11;

if (itemCount > 0 && itemCount <= 5) {
  discount = 5; // 5% discount
} else if (itemCount > 5 && itemCount <= 10) {
  discount = 10; // 10% discount
} else if (itemCount > 10) {
  discount = 15; // 15%
} else {
  throw new Error('The number of items cannot be negative!');
}

console.log(`You got ${discount}% discount. `);
```

Output:

```
You got 15% discount.
```

In this example, when the number of items is greater than 10, the discount is 15%. The statement in the second else if branch executes.

If the number of items is less than zero, the statement in the `else` branch executes.

Summary

- Use the `if` statement to execute code based on a condition.
- Use the `else` branch if you want to execute code when the condition is false. It’s good practice to use the ternary operator `?:` instead of a simple if…else statement.
- Use `if else if...else` statement to execute code based on multiple conditions.