Theory: Conditional operators

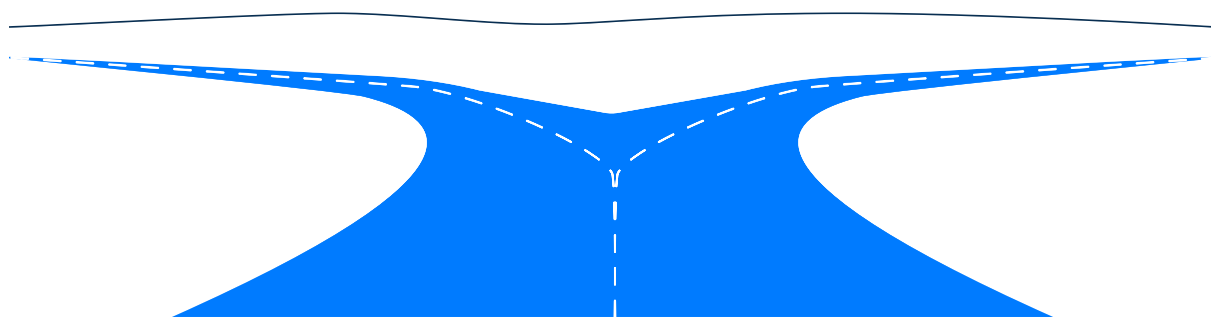
11 minutes 0 / 5 problems solved

Skip this topic Start practicing

1178 users solved this topic. Latest completion was about 1 hour ago.

§1. Conditional operators

Until now, we've only considered programs that execute commands line by line. But in real programming, they may look like a crossroad:



In life, we sometimes find ourselves at a crossroads where our path forks and splits in two. This happens in programming too: programs may fork. In these cases we use **conditional operators.**

§2. The "if" statement

Often we need to make a decision based on some conditions. In programming and JavaScript particularly, this concept is realized with the help of the `if` statement. It's called **a conditional operator**. For example, imagine you have a cat that meows when it gets hungry after 6 am. We can write a statement:

```
function meow() {
  return "Meow!";
}

let time = 10;
let sound;

if (time >= 6) {
  sound = meow();
  console.log(sound);
}
```

In the function `meow()` we describe what the cat does every time it is after 6 am.

Note that when we work with the `if` statement, the condition inside the parentheses is converted to a boolean. The code inside `if` is executed if the boolean is true. So every boolean could be a condition:

```
let condition = true;

if (condition) {
  console.log("True!");
}
```

§3. The "else" block

When the condition is false, the `else` block can be used instead of `if`. For example:

Current topic:

Conditional operators ···

Topic depends on:

× Comparison operators ···

Topic is required for:

Switch ···

JavaScript Code Style ···

For loops ···

While loops ···

Introduction to Promises ···

Keyboard events handling ···

Table of contents:

```
function meow() {
  return "Meow!";
}

function sleep() {
  return "Zzzzz...";
}

let time = 5;

let sound;



if (time >= 6) {

  sound = meow();

} else {

  sound = sleep();

}

console.log(sound);
```

Here, our cat meows when the time is later than or equal to 6 am; otherwise, it sleeps.

§4. Several conditions: "else if" block

There are situations when we have not just two but several possible conditions. For this purpose we use the `else if` block:

```
function meow() {
  return "Meow!";
}

function sleep() {
  return "Zzzzz...";
}

function play() {

  return "Bang bang!";

}



let time = 9;

let sound;



if (time < 6) {

  sound = sleep();

} else if (time < 8) {

  sound = meow();

} else {

  sound = play();

}

console.log(sound);
```

As you can see, here we check three possible variants: the cat sleeps when time is before 6 am, meows when it's between 6 and 8 am, and plays otherwise. We've defined three functions for three possible sounds the cat makes. What sound will be issued as a result?

§5. The ternary operator "? :"

When the aim of the program is to assign a variable depending on a condition, we can use the short version of the `if...else` block: the **ternary operator** `? :` . It works like this:

```
let time = 11;
let isHungry = (time >= 6) ? true : false;
```

In the example, you can see the condition. Then goes `?` and two possible values of the set variable: first, what we set when the condition is true, and after `:` comes whatever we set for false. A condition for visibility can be placed in brackets, but this action is not necessary. The same code, but without brackets, will be executed in the same way:

```
let time = 11;
let isHungry = time >= 6 ? true : false;
```

The condition can be as simple as in the example above, as well as more complex. For example, by using logical operators:

```
let time = 11;
let isHungry = !(time <= 6) ? true : false;
```

Let's take a closer look at our condition. The `!` means "not", so we can interpret the statement as "isn't the time after or equal to 6?".

§6. Conclusion

In this topic, we've considered what to do if our program should work according to some conditions. Conditional operators are an important part of programming. Remember: when we work with `if...else` operators or the ternary operator, the condition must be boolean.

Report a typo

107 users liked this theory. 5 didn't like it. **What about you?**

Start practicing

Comments (3) Hints (0) Useful links (0) Show discussion