

# TypeScript Boolean

If this site saves you hours of work, please  
**whitelist it in your ad blocker** 🙏 to  
**support us** ❤️  
in creating more helpful and free content  
in the future.

**Summary:** in this tutorial, you will learn about the TypeScript `boolean` data type and how to use the boolean keyword.

## Introduction to the TypeScript boolean

The TypeScript `boolean` type has two values: `true` and `false`. The `boolean` type is one of the primitive types in TypeScript.

### Declaring boolean variables

In TypeScript, you can declare a boolean variable using the `boolean` keyword. For example:

```
let pending: boolean;  
pending = true;  
// after a while  
// ..  
pending = false;
```

### Boolean operator

To manipulate boolean values, you use the boolean operators. TypeScript supports common boolean operators:

| Operator | Meaning              |
|----------|----------------------|
| &&       | Logical AND operator |

| Operator        | Meaning              |
|-----------------|----------------------|
| <code>  </code> | Logical OR operator  |
| <code>!</code>  | Logical NOT operator |

For example:

```
// NOT operator
const pending: boolean = true;
const notPending = !pending; // false
console.log(result); // false

const hasError: boolean = false;
const completed: boolean = true;

// AND operator
let result = completed && hasError;
console.log(result); // false

// OR operator
result = completed || hasError;
console.log(result); // true
```

## Type annotations for boolean

As seen in previous examples, you can use the `boolean` keyword to annotate the types for the boolean variables:

```
let completed: boolean = true;
```

However, TypeScript often [infers types](#) automatically, so type annotations may not be necessary:

```
let completed = true;
```

Like a variable, you can [annotate](#) boolean parameters or return the type of a function using the `boolean` keyword:

```
function changeStatus(status: boolean): boolean {  
    //...  
}
```

## Boolean Type

JavaScript has the `Boolean` type that refers to the non-primitive boxed object. The `Boolean` type has the letter `B` in uppercase, which is different from the `boolean` type.

It's a good practice to avoid using the `Boolean` type.

## Summary

- TypeScript `boolean` type has two values `true` and `false`.
- Use the `boolean` keyword to declare boolean variables.
- Do not use `Boolean` type unless you have a good reason to do so.