

# TypeScript Type Aliases

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 how to define new names for types using type aliases.

## Introduction to TypeScript type aliases

In TypeScript, a type alias allows you to create a new name for an existing type.

Type aliases can be useful for:

- Simplifying complex types.
- Making code more readable.
- Creating reusable types that can be used in many places in the codebase.

To define a type alias, you use the `type` keyword followed by the alias name and the type it represents.

Here's the syntax for defining a type alias

```
type alias = existingType;
```

The existing type can be any valid TypeScript type including primitive type, [object type](#), [union type](#), [intersection type](#), and [function type](#).

## Type alias examples

### 1) Primitive types

The following example uses the type alias `chars` for the string type:

```
type Name: string;

let firstName: Name;
let lastName: Name;
```

In this example, we create the `Name` as a type alias for the `string` type and use it to declare two variables `firstName` and `lastName`.

## 2) Object types

The following example defines a type alias `Person` for an object that has two properties `name` and `age`:

```
type Person = {
  name: string;
  age: number;
};

let person: Person = {
  name: 'John',
  age: 25
};
```

## 3) Union Types

The following example shows how to define a type alias for the union type `string | number`:

```
type alphanumeric = string | number;

let input: alphanumeric;
input = 100; // valid
input = 'Hi'; // valid
input = false; // Compiler error
```

## 4) Intersection Types

The following example shows how to create a type alias for the intersection type `Person & Contact`:

```
type Personal = {  
  name: string;  
  age: number;  
};  
  
type Contact = {  
  email: string;  
  phone: string;  
};  
  
type Candidate = Personal & Contact;  
  
let candidate: Candidate = {  
  name: "Joe",  
  age: 25,  
  email: "joe@example.com",  
  phone: "(408)-123-4567"  
};
```

## Summary

- Use type aliases to define new names for existing types.