鸡皮疙瘩

[TypeScript Tutorial (w3schools.com)](https://www.w3schools.com/typescript/index.php)

目录

TS HOME

TS Introduction

[TS Get Started](https://www.w3schools.com/typescript/typescript_getstarted.php" \t "https://www.w3schools.com/typescript/_top)

[TS Simple Types](https://www.w3schools.com/typescript/typescript_simple_types.php" \t "https://www.w3schools.com/typescript/_top)

[TS Special Types](https://www.w3schools.com/typescript/typescript_special_types.php" \t "https://www.w3schools.com/typescript/_top)

TS Arrays

TS Tuples

[TS Object Types](https://www.w3schools.com/typescript/typescript_object_types.php" \t "https://www.w3schools.com/typescript/_top)

[TS Enums](https://www.w3schools.com/typescript/typescript_enums.php" \t "https://www.w3schools.com/typescript/_top)

[TS Aliases & Interfaces](https://www.w3schools.com/typescript/typescript_aliases_and_interfaces.php" \t "https://www.w3schools.com/typescript/_top)

[TS Union Types](https://www.w3schools.com/typescript/typescript_union_types.php" \t "https://www.w3schools.com/typescript/_top)

[TS Functions](https://www.w3schools.com/typescript/typescript_functions.php" \t "https://www.w3schools.com/typescript/_top)

[TS Casting](https://www.w3schools.com/typescript/typescript_casting.php" \t "https://www.w3schools.com/typescript/_top)

[TS Classes](https://www.w3schools.com/typescript/typescript_classes.php" \t "https://www.w3schools.com/typescript/_top)

[TS Basic Generics](https://www.w3schools.com/typescript/typescript_basic_generics.php" \t "https://www.w3schools.com/typescript/_top)

[TS Utility Types](https://www.w3schools.com/typescript/typescript_utility_types.php" \t "https://www.w3schools.com/typescript/_top)

[TS Keyof](https://www.w3schools.com/typescript/typescript_keyof.php" \t "https://www.w3schools.com/typescript/_top)

[TS Null](https://www.w3schools.com/typescript/typescript_null.php" \t "https://www.w3schools.com/typescript/_top)

[TS Definitely Typed](https://www.w3schools.com/typescript/typescript_definitely_typed.php" \t "https://www.w3schools.com/typescript/_top)

## TypeScript tutorial

[TS HOME](https://www.w3schools.com/typescript/index.php" \t "https://www.w3schools.com/typescript/_top)[TS Introduction](https://www.w3schools.com/typescript/typescript_intro.php" \t "https://www.w3schools.com/typescript/_top)[TS Arrays](https://www.w3schools.com/typescript/typescript_arrays.php" \t "https://www.w3schools.com/typescript/_top)

[TS Tuples](https://www.w3schools.com/typescript/typescript_tuples.php" \t "https://www.w3schools.com/typescript/_top)

# TypeScript tutorial

**TS HOME**

We recommend reading this tutorial(教程) in the sequence(顺序，次序) listed in the left menu.

**TS Introduction**

What is TypeScript?

TypeScript is a syntactic superset(语法超集) of JavaScript which adds **static typing**.

This basically means that TypeScript adds syntax on top of JavaScript, allowing developers to add **types**.

TypeScript being a "Syntactic Superset" means that it shares the same base syntax as JavaScript, but adds something to it.

TS Simple Types

Type Assignment

When creating a variable, there are two main ways TypeScript assigns a type:

* Explicit
* Implicit

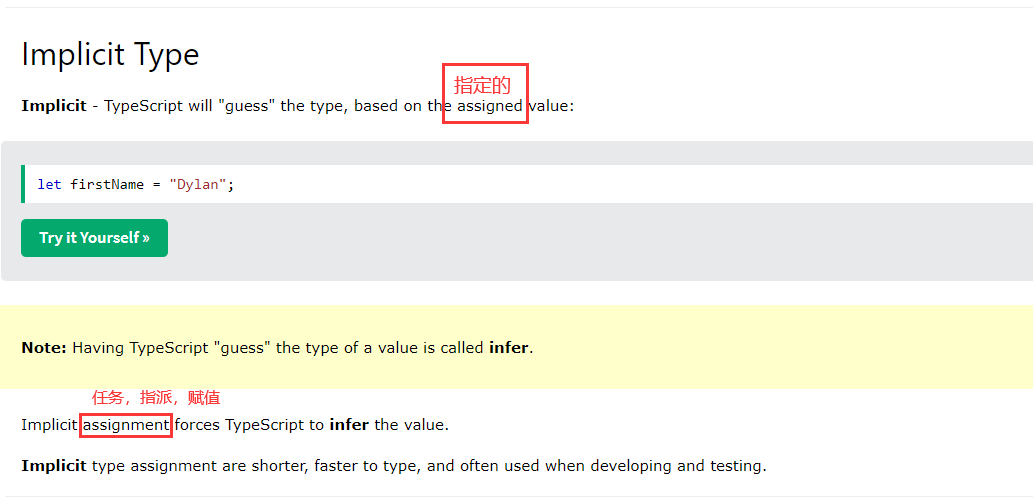
In both examples below **firstName**is of type **string**

Explict Type

**Explicit**- writing out the type:

let firstName: string = "Dylan";

**Explicit**type assignment are easier to read and more intentional(故意的).



developing and testing(开发和测试).

**TS Special Types**

**TS Arrays**

Sequences(有序列)

**TS Tuples**

**TS Object Types**

**TS Enums**

**[TS Aliases(别名) & Interfaces](https://www.w3schools.com/typescript/typescript_aliases_and_interfaces.php)**

1. Type Aliases allow defining types with a custom name (an Alias).
2. Type Aliases can be used for primitives(原始事物；基本体) like string or more complex types such as objects and arrays:

|  |  |  |
| --- | --- | --- |
|  |  | diff |
| Type Aliases | Type Aliases allow defining types with a custom name (an Alias).  Type Aliases | example |
| Interfaces |  | **Only** apply to object types |
|  |  |  |

Type Aliases can be used for primitives like string or more complex types such as objects and arrays.

**Interfaces** are similar to type aliases, except they **only apply to object types.**

**Interfaces 定义对象类型**

Type Aliases

**TS Union Types**

Union types(联合类型) are used when a value can be more than a single type.

Union | (OR)

Using the | we are saying our parameter is a string or number:

**Example**

**TS Functions**

**TS Casting**(类型转换)

1. **Casting with as**

A straightforward way to cast a variable is using the as keyword, which will directly change the type of the given variable.

1. **Casting with <>**

Using <> works(从事……工作,起到...作用) the same as casting with as.

1. **Force casting**

To override(覆盖) type errors that TypeScript may throw when casting, first cast to unknown, then to the target type.

**TS Classes**

**TS Basic Generics**

**TS Tuples(固定有序列)**

**TS Utility Types 实用类型: Partial, Required,Record**

TypeScript comes with a large number of types that can help with **some common type manipulatio**(一些常见类型操作), usually referred to as utility types.

This chapter **covers**(涵盖) the most popular utility types

**#Partial**

Partial changes all the properties in an object to be optional.

**#Required**

Required changes all the properties in an object to be required.

**#Record**

Record is a shortcut to defining an object type with a specific key type and value type.

Record 是一种便捷方式，用来定义一个对象的key的类型和value的类型

Record<string, number> is equivalent(等效的) to { [key: string]: number }

**TS Keyof**

keyof with explicit keys

When used on an object type with explicit keys, keyof creates a union type with those keys.

Example1

**注释**：keyof 把对象的属性name作为类型，用来创建一种类型，这个类型是一个联合类型，是取对象属性的类型创建的，从某方面来说他限定了类型的范围是对象的所有属性name

// `keyof Person` here creates a union type of "name" and "age", other strings will not be allowed

Example2

**TS Null**

**TS Definitely Typed**

1. number doesn't have \*access\* to it.

1. `The type of the value returned by the function can be explicitly defined.`