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

目录

TS HOME

TS Introduction

TS Get Started

TS Simple Types

TS Special Types

TS Arrays

TS Tuples

TS Object Types

TS Enums

TS Aliases & Interfaces

TS Union Types

TS Functions

TS Casting

TS Classes

TS Basic Generics

TS Utility Types

TS Keyof

TS Null

TS Definitely Typed

# 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

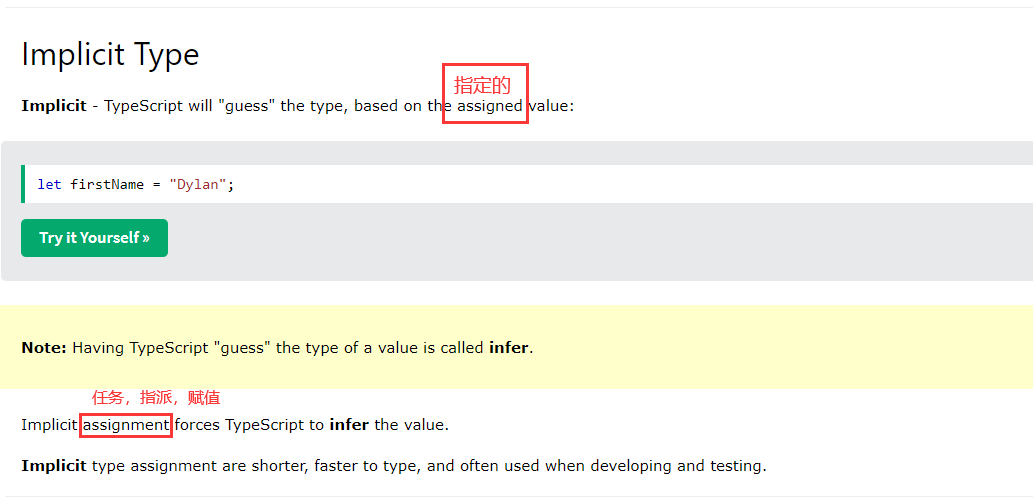
In both examples below **firstName**is of type **string**

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**

**TS Tuples**

**TS Object Types**

**TS Enums**

**TS Aliases & Interfaces**

**TS Union Types**

**TS Functions**

**TS Casting**

**TS Classes**

**TS Basic Generics**

**TS Tuples**

**TS Utility Types 实用类型**

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.

Example

Record 是一种便捷方式，用来定义一个对象的key和value的类型

const nameAgeMap: Record<string, number> = {  
  'Alice': 21,  
  'Bob': 25  
};

Record<string, number> is equivalent to { [key: string]: number }

**TS Keyof**

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

1. `The type of the value returned by the function can be explicitly defined.`
2. 1. TypeScript Type Aliases(**别名**) and Interfaces
3. Type Aliases allow defining types with a custom name (an Alias).
5. 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:

1. TypeScript Casting

**TS Null**

**TS Definitely Typed**