What is TypeScript

Summary: in this tutorial, you'll understand what TypeScript is and its advantages over vanilla JavaScript.

Introduction to TypeScript

TypeScript is a superset of JavaScript.

TypeScript builds on top of JavaScript. First, you write the TypeScript code. Then, you compile the TypeScript code into plain JavaScript code using a TypeScript compiler.

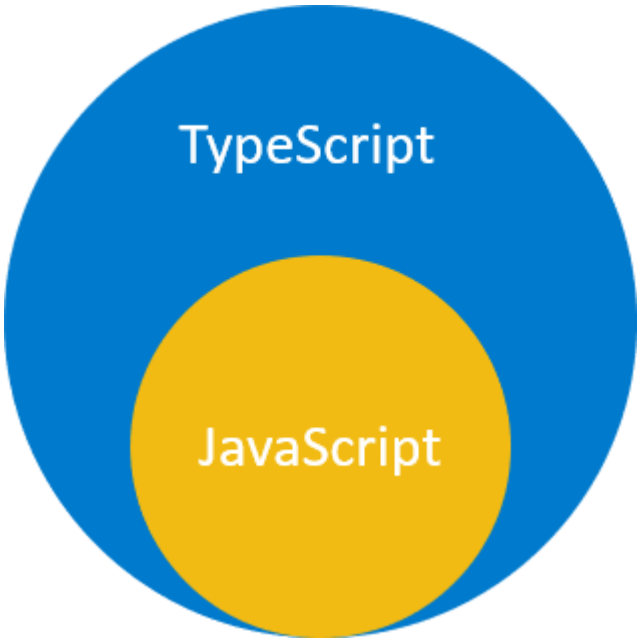
Once you have the plain JavaScript code, you can deploy it to any environment that JavaScript runs.

TypeScript files use the `.ts` extension rather than the `.js` extension of JavaScript files.

TypeScript uses the JavaScript syntaxes and adds additional syntaxes for supporting Types.

If you have a JavaScript program that doesn't have any syntax errors, it is also a TypeScript program. It means that all JavaScript programs are TypeScript programs. This is very helpful if you're migrating an existing JavaScript codebase to TypeScript.

The following diagram shows the relationship between TypeScript and JavaScript:



Why TypeScript

The main goals of TypeScript are:

- Introduce optional types to JavaScript.
- Implement planned features of future JavaScript, a.k.a. ECMAScript Next or ES Next to the current JavaScript.

1) TypeScript improves your productivity while helping avoid bugs

Types increase productivity by helping you avoid many mistakes. By using types, you can catch bugs at the compile time instead of having them occur at runtime.

For example, the following function adds two numbers `x` and `y`:

```
function add(x, y) {
   return x + y;
}
```

If you get the values from HTML input elements and pass them into the function, you may get an unexpected result:

```
let result = add(input1.value, input2.value);
```

```
console.log(result); // result of concatenating strings
```

For example, if users entered `10` and `20`, the `add()` function would return `1020`, instead of `30`.

The reason is that `input1.value` and `input2.value` are strings, not numbers. When you use the operator `+` to add two strings, it concatenates them into a single string.

When you use TypeScript to specify the type for the parameters like this explicitly:

```
function add(x: number, y: number) {
   return x + y;
}
```

In this function, we added the number types to the parameters. The function `add()` will accept only numbers, not any other values.

When you invoke the function as follows:

```
let result = add(input1.value, input2.value);
```

... the TypeScript compiler will issue an error if you compile the TypeScript code into JavaScript. Hence, you can prevent the error from happening at runtime.

2) TypeScript brings the future JavaScript to today

TypeScript supports the upcoming features planned in the ES Next for the current JavaScript engines. It means you can use the new JavaScript features before web browsers (or other environments) fully support them.

Every year, TC39 releases several new features for ECMAScript, which is the standard of JavaScript. The feature proposals typically go through five stages:

- Stage 0: Strawperson
- Stage 1: Proposal
- Stage 2: Draft
- Stage 3: Candidate
- Stage 4: Finished

And TypeScript generally supports features that are in stage 3.

Summary

- TypeScript is a superset of JavaScript.
- TypeScript adds type to the JavaScript and helps you avoid potential bugs that occur at runtime.
- TypeScript also implements the future features of JavaScript.