TypeScript: The Big Picture

HOW TYPESCRIPT CAN HELP

Simon Allardice
STAFF AUTHOR, PLURALSIGHT

@allardice www.pluralsight.com

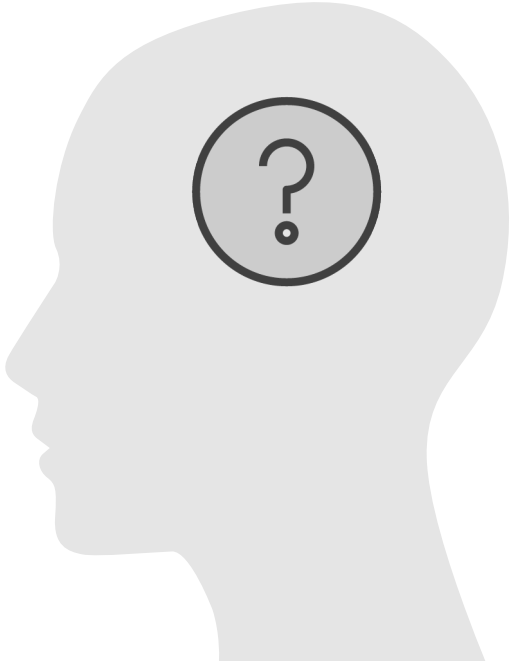
You **Don't** Need To Know

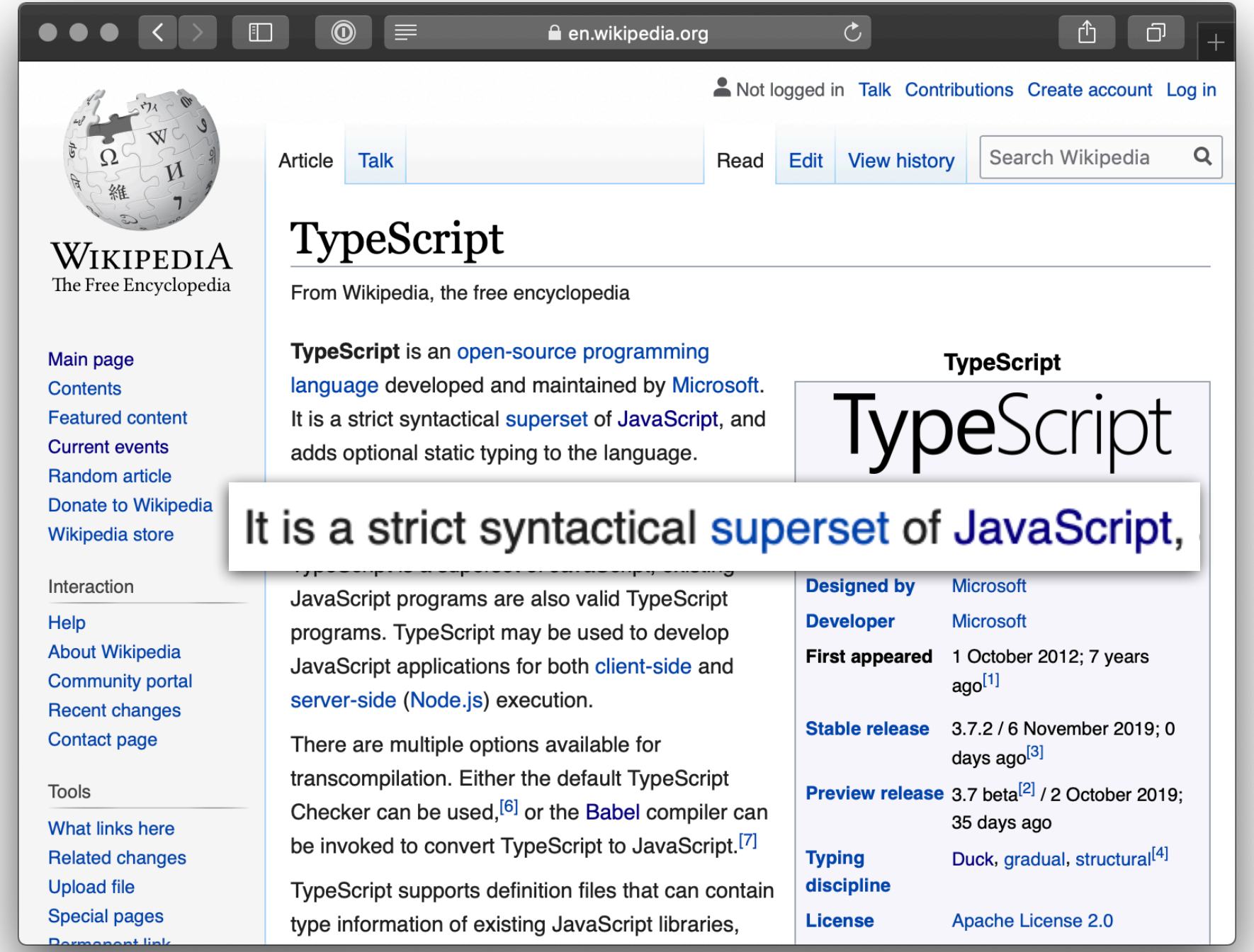
TypeScript

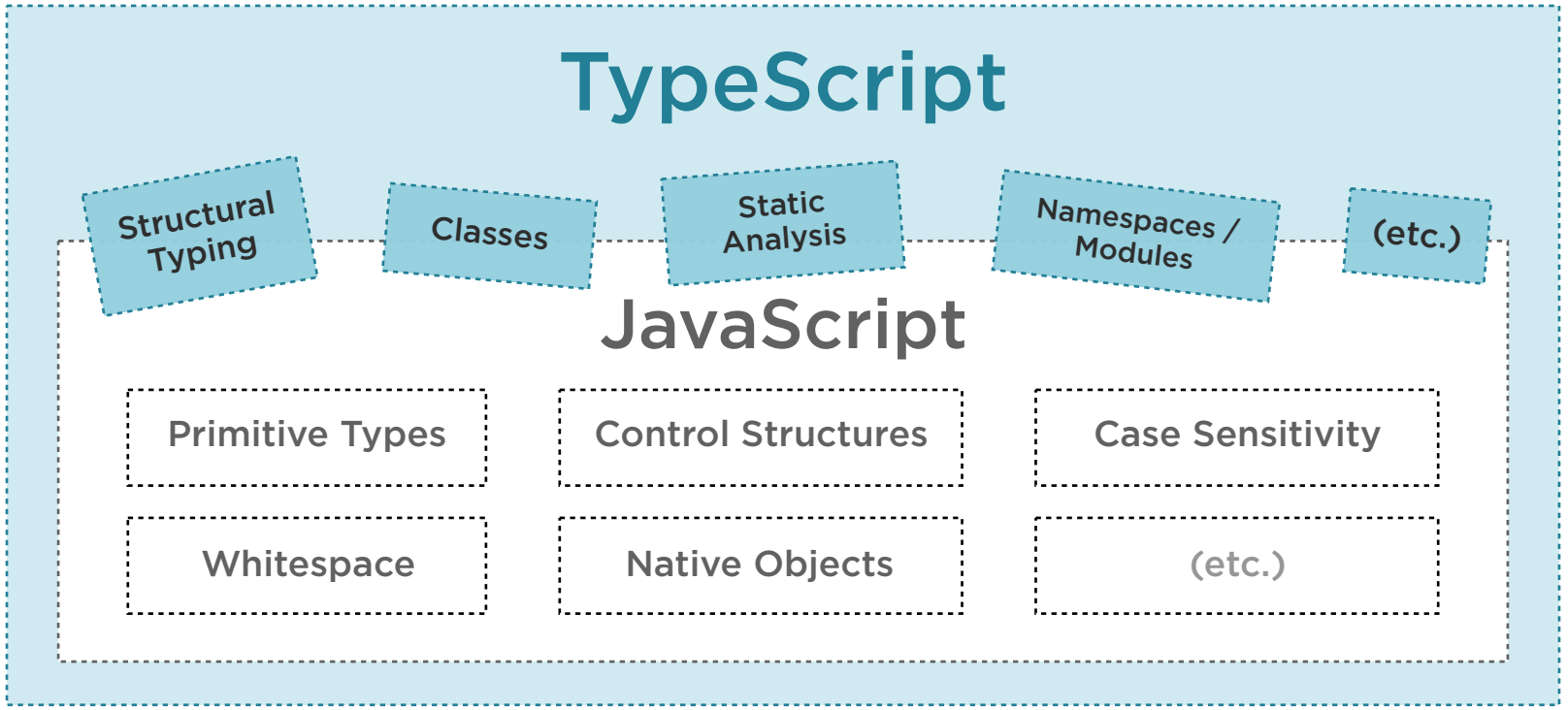
You Need To Know

JavaScript (basics)

HTML, CSS




en.wikipedia.org
Not logged in Talk Contributions Create account Log in
Article Talk Read Edit View history Search Wikipedia
WIKIPEDIA
The Free Encyclopedia
Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store
Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page
Tools
What links here
Related changes
Upload file
Special pages
TypeScript
From Wikipedia, the free encyclopedia
TypeScript is an open-source programming language developed and maintained by Microsoft. It is a strict syntactical superset of JavaScript, and adds optional static typing to the language.
It is a strict syntactical superset of JavaScript,
JavaScript programs are also valid TypeScript programs. TypeScript may be used to develop JavaScript applications for both client-side and server-side (Node.js) execution.
There are multiple options available for transcompilation. Either the default TypeScript Checker can be used,[6] or the Babel compiler can be invoked to convert TypeScript to JavaScript.[7]
TypeScript supports definition files that can contain type information of existing JavaScript libraries,
TypeScript
TypeScript
Designed by Microsoft
Developer Microsoft
First appeared 1 October 2012; 7 years ago[1]
Stable release 3.7.2 / 6 November 2019; 0 days ago[3]
Preview release 3.7 beta[2] / 2 October 2019; 35 days ago
Typing discipline Duck, gradual, structural[4]
License Apache License 2.0


TypeScript
Structural Typing
Classes
Static Analysis
Namespaces / Modules
(etc.)
JavaScript
Primitive Types
Control Structures
Case Sensitivity
Whitespace
Native Objects
(etc.)

TypeScript Compiles to JavaScript

TypeScript
Source Code

JavaScript
Source Code

Benefits of TypeScript: Overview

Static Types

variables, parameters,
return types, etc.

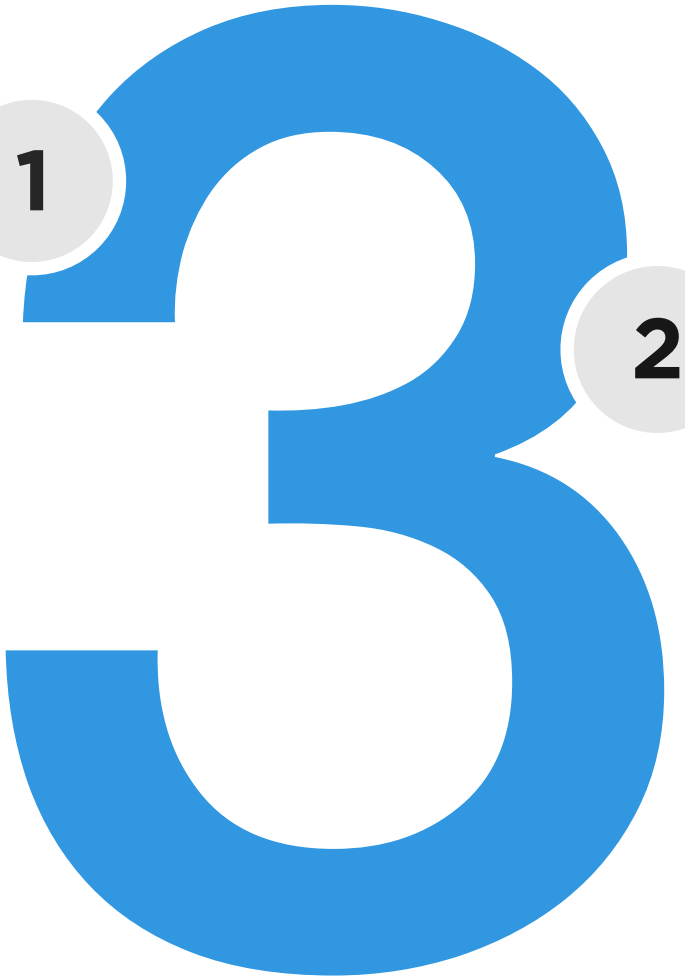
1

Benefits of TypeScript: Overview

Static Types

variables, parameters,
return types, etc.

1



2

Organization

2

Organization

Making it easier to manage a large codebase

classes

namespaces

modules

interfaces

TypeScript Classes

```
class Customer {

    name: string;
    isActive: boolean;


    announce() {
        return "Hello, my name is " + this.name;
    }
}
```

Classes

On this Page

- Defining classes
- Class body and method de
- Sub classing with `extend`
- Species
- Super class calls with `super`
- Mix-ins
- Specifications
- Browser compatibility
- Running in Scratchpad
- See also

JavaScript classes, introduced in ECMAScript 2015, are primarily syntactical sugar over JavaScript's existing prototype-based inheritance. The class syntax *does not* introduce a new object-oriented inheritance model to JavaScript.

Defining classes

Classes are in fact "special functions", and just as you can define function expressions and function declarations, the class syntax has two components: class expressions and class declarations.

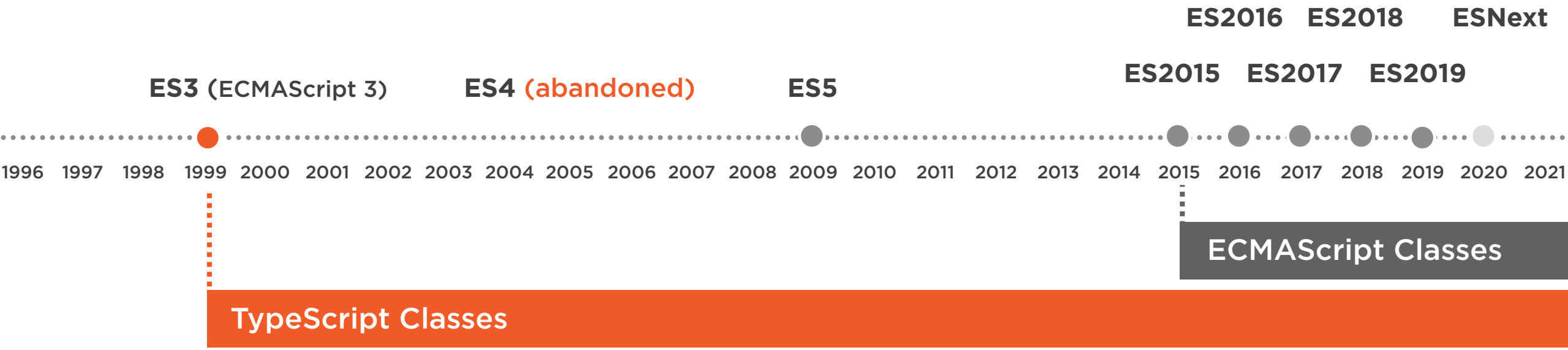
Class declarations

One way to define a class is using a **class declaration**. To declare a class, you use the `class`

Related Topics

JavaScript

TypeScript vs ECMAScript Class Compatibility

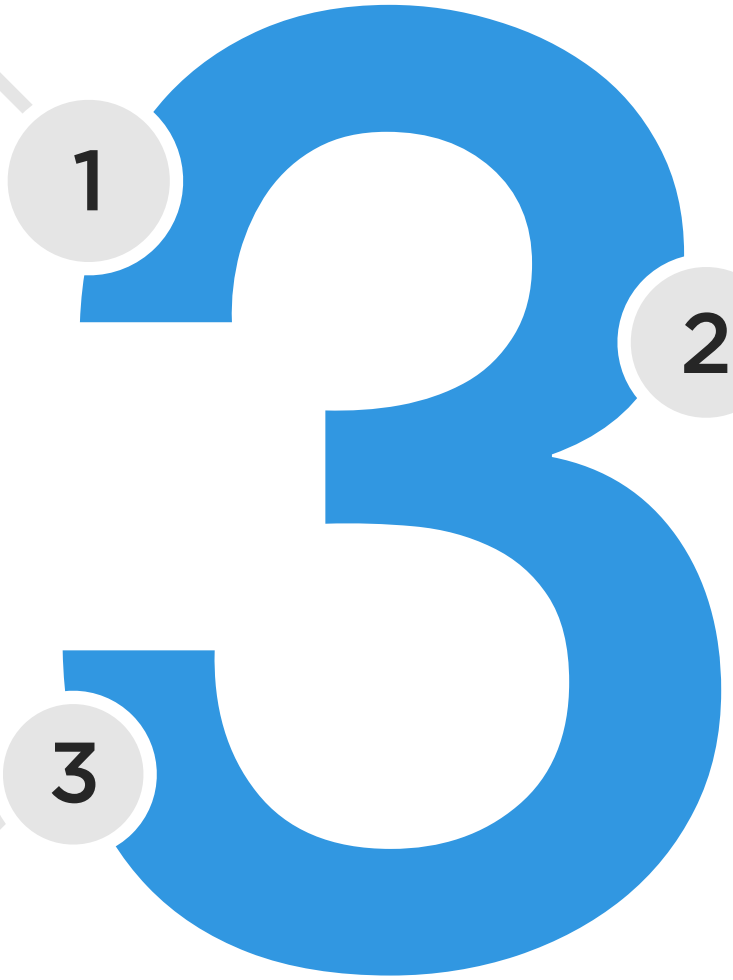


TypeScript Interfaces

```
interface MediaAsset {
    name: string;
    size: number;
    location: string;

    play(): void;
}
```

Benefits of TypeScript: Overview



1. **Static Types**
   variables, parameters, return types, etc.
2. **Organization**
   classes, modules, namespaces, interfaces
3. **Tooling**

3

Tooling

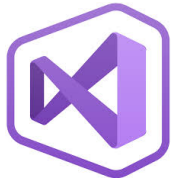
static type analysis

many "instant" errors

detect unused data / unreachable code

source maps — debug directly in TypeScript

TypeScript IDE Support



Visual Studio

Eclipse

Atom

Sublime Text

Emacs

Vim