This page outlines specific features and fixes that are scheduled or planned for given releases. The 6-month roadmaps that outlines focus areas of work can be viewed for

- January to June 2021 (current)
- January to December 2020
- July to December 2019
- January to June 2019

☒ Feature already available in `typescript@next`

Future

- Investigate nominal typing support
- Flattening declarations
- Implement ES Decorator proposal
- Investigate Ambient, Deprecated, and Conditional decorators
- Investigate partial type argument inference
- Quick fix to Scaffold local `@types` packages
- Investigate error messages in haiku or iambic pentameter
- Decorators for function expressions/arrow functions

4.7 (May 2022)

See the 4.7 Iteration Plan for more details.

☒ Control Flow and Class Initialization Checks for Computed Properties
☒ Instantiation Expressions
☒ Improved Inference Between Function-Initialized Properties and Methods
☒ `extends` Constraints on `infer` Type Variables
☒ Optional Variance Annotations
☒ `moduleSuffixes` Option For Resolution
☒ `resolution-mode` for `/// <reference types="..." />` Directives
☒ `resolution-mode` for `import type`
☒ Allow `typeof` on `#private` Fields
☒ Object Method Snippet Completions
☒ Grouping-Aware Organize Imports
☐ Go to Source Definition/JavaScript Implementation
☐ Virtual File System Support for TSServer

4.6 (February 2022)

See the 4.6 Iteration Plan for more details.

☒ Narrowing for Variables Destructured from Discriminated Unions
☒ Relaxed Checks on Code Before `super()`
☒ Improved Recursion Depth Checks
☒ Indexed Access Inference Improvements
☒ `--target es2022`
☒ Syntax and Binding Errors in JavaScript Files
☒ Release of `@typescript/analyze-trace`

4.5 (November 2021)

See the 4.5 Iteration Plan for more details.

☒ `lib` Overrides from `node_modules`
☒ `Awaited` and `Promise` Function Improvements
☒ Tail Recursive Evaluation of Conditional Types
☒ Template String Discriminants
☒ `--module es2022`
☒ Controls for Import Elision via `--preserveValueImports`
☒ `type` Modifiers on Import Names
☒ `in` Checks on `#private` Fields
☒ Import Assertions
☒ Speed Improvements on Windows/Mac via `realpathSync.native`
☒ Snippet Completions for Subclass Methods
☒ Snippet Completions for JSX Attributes
☒ Better Editor Support for Unresolved Types # 4.4 (August 2021)

See the 4.4 Iteration Plan for more details.

☒ Control Flow Analysis of Aliased Conditions
☒ Generalized index signatures
☒ Strict optional property checks with `--strictOptionalProperties`
☒ Use `unknown` for `try`/`catch` variables with `--useUnknownInCatchVariables`
☒ Updated `--help` descriptions
☒ Inlay Hints
☒ High-Confidence Suggestions in Unchecked JavaScript
☒ Quick Fix to Declare Missing Properties on Objects

4.3 (May 2021)

See the 4.3 Iteration Plan for more details.

☒ Separate "write" types
☒ `--noImplicitOverride` and the `override` keyword
☒ Template string type inference and compatibility improvements
☒ Contextual Narrowing for Generics
☒ Infer template string types when contextually typed

☒ ECMAScript `#private` Methods/Statics/Accessors
☒ `static` Index Signatures
☒ Editor support for `@link` tags
- Investigate auto-import improvements
  ☒ Complete import statements through auto-imports
- Support Go-to-Definition on Relative Non-JS/TS Files

4.2 (February 2021)

See the 4.2 Iteration Plan for more details.

☒ Leading/middle rest elements in tuple types
☒ Smarter Type Alias Printing and Emit
☒ Template Literal Types for Template String Expressions
☒ `--noPropertyAccessFromIndexSignature`
☒ `abstract` constructor types
☒ Smarter unused local analysis in destructuring
☒ `--explainFiles` to understand why files have been included in a program
☒ Declare missing helper function

4.1. (November 2020)

See the 4.1 Iteration Plan for more details

☒ Template literal types and mapped type 'as' clauses
☒ Recursive Conditional Types
☒ Pedantic Index Signature Checks (`--noUncheckedIndexedAccess`)
☒ `export * as default`
☒ Aliased `this` in JavaScript functions
☒ Editor support for `/** @see */` tags
☐ Semantic Highlighting API

4.0 (August 2020)

See the 4.0 Iteration Plan for more details.

☒ Variadic Tuple Types
☒ Labeled Tuple Elements
☒ Class Property Inference from Constructors
☒ Short-Circuiting Assignment Operators
☒ `unknown` on `catch` Clause Bindings
☒ Lightweight Editor Mode
☒ Improved Auto-Import
☒ Editor support for `/** @deprecated */` tags
☒ Refactoring Discoverability (ideas)

- ☒ Refactorings
    - ☒ Convert to Optional Chaining and Coalescing

3.9 (May 2020)

See the 3.9 Iteration Plan for more details.

- ☒ Fix Inference in `Promise.all`
- ☒ Type-Checking Speedups
    - https://github.com/microsoft/TypeScript/pull/36576
    - https://github.com/microsoft/TypeScript/pull/36590
    - https://github.com/microsoft/TypeScript/pull/36607
    - https://github.com/microsoft/TypeScript/pull/36622
    - https://github.com/microsoft/TypeScript/pull/36754
    - https://github.com/microsoft/TypeScript/pull/36696
- ☒ `// @ts-expect-error` Comments
- ☒ Uncalled Function Checks in Conditionals
- ☒ Support for CommonJS Auto-Imports
- ☒ Preserved Newlines in Refactorings/Quick Fixes
- ☒ Editor Support for Solution-Style `tsconfig.json` Files
- ☒ Editor Support for Multi-Project Symbol Navigation
- ☒ Quick Fixes to
    - ☒ Correct Cases of Missing Return Expressions
    - ☐ Declare a `private` method
    - ☒ Declare a `private` property

3.8 (February 2020)

See the 3.8 Iteration Plan for more details.

- ☒ Type-Only Imports and Exports
- ☒ New `export * as ns` Syntax
- ☒ Top-Level Await
- ☒ Private Fields
- ☒ JSDoc `@public`/`@private`/`@protected` Modifiers
- ☒ JSDoc `@readonly` Modifier
- ☒ Improved File Watching and `watchOptions`
- ☒ "Fast and Loose" Incremental Checking
- ☒ Editor Support for Call Hierarchy View

3.7 (November 2019)

See the 3.7 Iteration Plan for more details.

- ☒ Optional Chaining

- ☒ Nullish Coalescing
- ☒ Assertions in Control Flow Analysis
- ☒ Recursive Type References
- ☒ Allow `allowJs` and `declaration` to be used together
- ☒ Uncalled Function Checks
- ☒ `--declaration` and `--isolatedModules`
- ☒ Flattened Property/Method Reporting
- ☒ `--declaration` and `--allowJs`
- ☒ Editor Functionality/APIs
  - ☒ Build-Free Editing for Project References
  - ☒ Auto-inserted `await` for completions
  - ☒ Quick fix to convert `const` to `let`
  - ☒ Semicolon Formatting Preferences

3.6 (August 2019)

See the 3.6 Iteration Plan for more details.

- ☒ Strongly typed iterators and generators
- ☒ More accurate array spreads
- ☒ Improved UX around Promises
- ☒ Semicolon-aware auto-imports

3.5 (May 2019)

See the 3.5 Iteration Plan for more details.

- ☒ Higher order inferences for constructor functions
- ☒ Easier assignment to discriminated unions
- ☒ Change default constraint for type parameters from `{}` to `unknown`
- ☒ Improved checking of indexed access types
- ☒ The `Omit` helper type
- ☒ APIs for composite projects (`--build` mode)
- ☒ `--allowUmdGlobalAccess` to use UMD globals from modules
- ☒ `--noEmitOnError` with `--isolatedModules`
- ☒ Speed improvements in `--incremental` and `--build`
  - ☒ Caching module resolution results
  - ☒ Caching results of parsing `tsconfig.json`
- ☒ Sorted/prioritized completions
- ☒ Smart Select API
- ☒ Refactoring to extract type

3.4 (March 2019)

See the 3.4 Iteration Plan for more details.

☒ Higher order function type inference
☒ `--incremental` builds with `.tsbuildinfo` files
☒ Improved support for read-only arrays and tuples
☒ Const contexts for literal expressions
☒ `globalThis`
☒ Refactorings to…
  ☒ Convert to “named” parameters

3.3 (January 2019)

☒ Relaxed rules on methods of union types
☒ File-incremental builds in `--build --watch` mode for composite projects

3.2 (November 2018)

☒ Strict bind, call, and apply methods on functions
☒ Allow non-unit types in union discriminants
☒ Generic spread expressions in object literals
☒ Generic object rest variables and parameters
☒ Type-checking support for `BigInt`
☒ Configuration inheritance through node packages
☒ Supporting `Object.defineProperty` property assignments in JS
☒ Support printing the implied configuration object to the console with `--showConfig`
☒ Improved formatting and indentation for lists and chained calls
☒ Quick fixes to…
  ☒ Add intermediate `unknown` type assertions
  ☒ Add missing `new` keyword
  ☒ Infer types from usage outside of `noImplicitAny`
  ☒ Infer JSDoc types from usage

3.1 (September 2018)

☒ Mapped tuple and array types
☒ Property assignments on function declarations
☒ `typesVersions` redirects
☒ Error UX improvements
☒ Refactorings to…
  ☒ Rename files from import/export paths
  ☒ Convert from `Promise#then/catch` to `async/await`

3.0 (July 2018)

- ☒ Support for project references/composite projects
- ☒ Tuples in rest parameters and spread expressions
- ☒ New `unknown` top type
- ☒ `/// <reference lib="..." />` reference directives
- ☒ Support for `defaultProps` and `propTypes` in React JSX elements
- ☒ Error UX improvements
  - ☒ Related error spans
  - ☒ Improved message quality and suggestions
- ☒ Refactors to…
  - ☒ Convert named imports to namespace imports and back
  - ☒ Add or remove braces from arrow function
- ☒ Quick fixes to…
  - ☒ Remove unused labels
  - ☒ Remove unreachable code
- ☒ Outlining spans for JSX expressions
- ☒ Auto-closing JSX tags

2.9 (May 2018)

- ☒ Support number and symbol named properties with keyof and mapped types
- ☒ Support for passing generics to JSX elements
- ☒ Support for passing generics to tagged template calls
- ☒ Allow `import(...)`-ing types at any location
- ☒ `--pretty` error output by default
- ☒ Support for `import.meta`
- ☒ New `--resolveJsonModule`
- ☒ Declaration source maps and code navigation via them
- ☒ Show suggestion diagnostics for open files
- ☒ Show unused declarations as suggestions
- ☒ Support setting quote style in quick fixes and refactorings
- ☒ New Rename File command
- ☒ Refactors to…
  - ☒ Move to a new file
  - ☒ Convert property to getter/setter
- ☒ Quick fix to…
  - ☒ Convert `require` to `import` in `.ts` files
  - ☒ Install `@types/node` for built-in node modules

2.8 (March 2018)

- ☒ Conditional types
- ☒ Type inference in conditional types

☒ Predefined conditional types in `lib.d.ts`
☒ Improved control over mapped type modifiers
☒ Improved `keyof` with intersection types
☒ New `--emitDeclarationOnly`
☒ Better handling for namespace-patterns in .js files
☒ Support for per-file JSX pragmas
☒ New Organize imports command
☒ Quick fix for uninitialized class properties

2.7 (January 2018)

☒ Support for properties named with `const`-declared symbols
☒ Strict property initialization checks in classes
☒ Definite assignment assertions
☒ Improved type inference for object literals
☒ Improved handling of structurally identical classes
☒ Fixed length tuples
☒ Type guards inferred from `in` operator
☒ Numeric separators
☒ Support for `import d from "cjs"` form CommonJS modules with `--esModuleInterop`
☒ Apply all quick fixes in a file
☒ Refactors to convert CommonJS module to ES6 module
☒ Support for incremental builder compiler API
☒ Quick fix to add missing `async` keyword
☒ Completion list preselects suggested item based on context
☒ Completion list includes `this.`, brackets, and curlies for JSX

2.6 (October 2017)

☒ Strict function types
☒ Support for JSX Fragments
☒ Cached tagged template objects in modules
☒ Include localized diagnostics in npm
☒ Suppress errors in .ts files using `// @ts-ignore` comments
☒ Speed improvements to `--watch`
☒ Automatic imports from completion lists
☒ Refactors to…
    ☒ Extract constants/locals in refactorings
    ☒ Convert JSDoc type annotations to TypeScript
☒ Quick fixes to…
    ☒ Infer from usage on `noImplicitAny` errors
    ☒ Invoke uncalled decorators
    ☒ Install from `@types`

- ☒ Support outlining for `// #regions`

2.5 (August 2017)

- ☒ Assertion/cast JSDoc syntax in `checkJs` mode
- ☒ Refactoring to extract functions & methods
- ☒ Optional `catch` binding
- ☒ Quick fixes to…
    - ☒ correct references to types of properties
    - ☒ correct uses of JSDoc types to TypeScript types

2.4 (June 2017)

- ☒ Infer from generic function return types
- ☒ Contextual generic function types
- ☒ Stricter generic signature checks
- ☒ Covariant checking for callback parameters
- ☒ String valued members in enums
- ☒ Weak type detection
- ☒ ES dynamic `import()` expressions
- ☒ Refactoring support in Language Service API
    - – Refactor ES5 function to ES6 class in .js files
- ☒ Suggestions and quick fixes for spelling corrections in error messages

2.3 (April 2017)

- ☒ Generator support for ES3/ES5
- ☒ Asynchronous iterators
- ☒ Generic defaults
- ☒ Controlling `this` in methods of object literals through contextual type
- ☒ JSX stateless components overload resolution
- ☒ JSX `children` checking
- ☒ New `--strict` master option
- ☒ Report errors in .js files with new `--checkJs`
- ☒ Enhanced `tsc --init` output
- ☒ Language Service Extensibility

2.2 (February 2017)

- ☒ Mixin classes
- ☒ Allow deriving from object and intersection types
- ☒ Support for `new.target`
- ☒ Improved checking of nullable operands in expressions

☒ Update `__extends` to use `Object.setPrototypeOf`
☒ Allow property (dotted) access for types with string index signatures
☒ Support for JSX spread children
☒ New `--jsx react-native`
☒ Support for the `object` type.
☒ More Quick Fixes!
  - Add missing imports
  - Implement interface/abstract class members
  - Remove unused declarations
  - Add missing `this.`
  - Add missing property declaration

2.1 (December 2016)

☒ Switch to a transformation-based emitter
☒ `async`/`await` support for ES5/ES3
☒ Support for external helpers library
☒ Static types for dynamically named properties (`keyof T` and `T[K]`)
☒ Mapped types (e.g. `{ [P in K]: T[P] }`)
☒ Support ESNext object property spread and rest
☒ Better inference for literal types
☒ Use returned values from super calls as 'this'
☒ Control flow analysis for implicit any variables
☒ Control flow analysis for array construction
☒ Narrow string and number types in literal equality checks
☒ Contextual typing of partially annotated signatures
☒ Normalize union/intersection type combinations
☒ New `--jsxFactory`
☒ New `--alwaysStrict`
☒ Support for `--target ES2016`, `--target ES2017` and `--target ESNext`
☒ Configuration inheritance
☒ Go to implementation support
☒ Completions in imports and triple-slash reference paths
☒ Quick fixes support in language service API
☒ Untyped (implicit-any) imports

2.0

☒ Non-nullable types
☒ Control flow based type analysis
☒ Discriminated union types
☒ Improved definition file acquisition
☒ Specifying `this` types for functions
☒ Type guards on property access

☒ Readonly properties and index signatures
☒ Use path mappings in module resolution
☒ Shorthand ambient module declarations and wildcard matching in module names
☒ Implicit index signatures
☒ Support private and protected constructors
☒ Support `abstract` properties
☒ Optional properties in classes
☒ The `never` type
☒ `--skipLibCheck` compiler option
☒ Support for declaration output folder using `--declarationDir`
☒ Glob support in tsconfig.json
☒ Improve lib.d.ts modularity and new `--lib` support
☒ Support for UMD module definitions
☒ Trailing Commas in Function Param Lists
☒ Support for jsdoc `@typedef` for JS files
☒ Completion lists for string literals
☒ Module name in imports allow .js extension
☒ Support 'target:es5' with 'module:es6'
☒ Flag unused declarations with `--noUnusedLocals` and `--noUnusedParameters`
☒ Number, boolean, and Enum literal types

1.8

☒ Allow JavaScript in TypeScript compilations with `--allowjs`
☒ Allow captured `let/const` in loops
☒ Flag unreachable code
☒ Concatenate module output with `--outFile`
☒ Accept comments in tsconfig.json
☒ Stylize error messages in terminal output with `--pretty`
☒ Support for `--outFile` with named pipes, sockets, and special devices
☒ Support computed property with literal names
☒ String literal types
☒ Stateless Functional Components in JSX
☒ Improved union/intersection type inference
☒ Support for F-Bounded Polymorphism
☒ Support full path for `-project/-p` parameter
☒ Extract type information from JSDoc in js files
☒ Support for `default` import interop with SystemJS using `--allowSyntheticDefaultImports`
☒ Recognize prototype assignments in JavaScript files
☒ Augmenting global/module scope from other modules
☒ Use tsconfig.json as higher priority source of configuration in Visual Studio
☒ `this`-based type guards
☒ Support for custom JSX factories using `--reactNamespace`
☒ Improved checking of for-in statements

☒ Colorization of JSX code in VS 2015
☒ Publish official TypeScript NuGet packages

1.7

☒ ES7 exponentiation operator
☒ Polymorphic `this` type
☒ Support `--module` with `--target es6`
☒ Support for decorators when targeting ES3
☒ `async`/`await` support for ES6 (Node v4)
☒ Improved checking of destructuring with literal initializers

1.6

☒ ES6 Generators
☒ Local types
☒ Generic type aliases
☒ Expressions in class extends clauses
☒ Class expressions
☒ `exclude` property in tsconfig.json
☒ User defined type guard functions
☒ External module resolution enhancements
☒ JSX support
☒ Intersection types
☒ `abstract` classes and methods
☒ Strict object literal assignment checking
☒ Declaration merging for classes and interfaces
☒ New –init

1.5

☒ Support for Destructuring
☒ Support for Spread Operator
☒ Support for ES6 Modules
☒ Support for for..of
☒ Support for ES6 Unicode specification
☒ Support for Symbols
☒ Support for Computed properties
☒ Support for tsconfig.json files
☒ Support for let and const in ES3/ES5
☒ Support for tagged templates in ES3/ES5
☒ Expose a new editor interface through TS Server
☒ Support for ES7 Decorators proposal
☒ Support for Decorator type metadata

☒ New –rootDir
☒ New ts.transpile API
☒ Support –module umd
☒ Support –module system
☒ New –noEmitHelpers
☒ New –inlineSourceMap
☒ New –inlineSources
☒ New –newLine
☒ New –isolatedModules
☒ Support for new `namespace` keyword
☒ Support for tsconfig.json in Visual Studio 2015
☒ Improved template literal highlighting in Visual Studio 2013

1.4

☒ Support for Union Types and Type Guards
☒ New –noEmitOnError
☒ New –target ES6
☒ Support for Let and Const
☒ Support for Template Literals
☒ Library typings for ES6
☒ Support for Const enums
☒ Export Language Service public API

1.3

☒ Language service re-write to target new compiler
☒ Support for protected members in classes
☒ Support for Tuple Types