Because this page hit a limit on a GitHub wiki page size, it's now hosted on the website: https://www.typescriptlang.org/docs/handbook/release-notes/overview.html

TypeScript 3.6

See TypeScript 3.6

Stricter Generators

See Stricter Generators

More Accurate Array Spread

See More Accurate Array Spread

Improved UX Around Promises

See Improved UX Around Promises

Better Unicode Support for Identifiers

See Better Unicode Support for Identifiers

`import.meta` Support in SystemJS

See `import.meta` Support in SystemJS

`get` and `set` Accessors Are Allowed in Ambient Contexts

See `get` and `set` Accessors Are Allowed in Ambient Contexts

Ambient Classes and Functions Can Merge

See Ambient Classes and Functions Can Merge

APIs to Support `--build` and `--incremental`

See APIs to Support `--build` and `--incremental`

Semicolon-Aware Code Edits

See Semicolon-Aware Code Edits

Smarter Auto-Import Syntax

See Smarter Auto-Import Syntax

`await` Completions on Promises

See `await` Completions on Promises

New TypeScript Playground

See New TypeScript Playground

TypeScript 3.5

See TypeScript 3.5

Speed improvements

See Speed improvements

Type-checking speed-ups

See Type-checking speed-ups

`--incremental` improvements

See `--incremental` improvements

The `Omit` helper type

See The `Omit` helper type

Improved excess property checks in union types

See Improved excess property checks in union types

The `--allowUmdGlobalAccess` flag

See The `--allowUmdGlobalAccess` flag

Smarter union type checking

See Smarter union type checking

Higher order type inference from generic constructors

See Higher order type inference from generic constructors

TypeScript 3.4

See TypeScript 3.4

Faster subsequent builds with the `--incremental` flag

See Faster subsequent builds with the `--incremental` flag

Composite projects

See Composite projects

`outFile`

See `outFile`

Higher order type inference from generic functions

See Higher order type inference from generic functions

Improvements for `ReadonlyArray` and `readonly` tuples

See Improvements for `ReadonlyArray` and `readonly` tuples

A new syntax for `ReadonlyArray`

See A new syntax for `ReadonlyArray`

`readonly` tuples

See `readonly` tuples

`readonly` mapped type modifiers and `readonly` arrays

See `readonly` mapped type modifiers and `readonly` arrays

Caveats

See Caveats

`const` assertions

See `const` assertions

Caveats

See Caveats

Type-checking for `globalThis`

See Type-checking for `globalThis`

TypeScript 3.3

See TypeScript 3.3

Improved behavior for calling union types

See Improved behavior for calling union types

Caveats

See Caveats

Incremental file watching for composite projects in `--build --watch`

See Incremental file watching for composite projects in `--build --watch`

TypeScript 3.2

See TypeScript 3.2

`strictBindCallApply`

See `strictBindCallApply`

Caveats

See Caveats

Generic spread expressions in object literals

See Generic spread expressions in object literals

Generic object rest variables and parameters

See Generic object rest variables and parameters

BigInt

See BigInt

Caveats

See Caveats

Non-unit types as union discriminants

See Non-unit types as union discriminants

`tsconfig.json` inheritance via Node.js packages

See `tsconfig.json` inheritance via Node.js packages

The new `--showConfig` flag

See The new `--showConfig` flag

`Object.defineProperty` declarations in JavaScript

See `Object.defineProperty` declarations in JavaScript

TypeScript 3.1

See TypeScript 3.1

Mapped types on tuples and arrays

See Mapped types on tuples and arrays

Properties declarations on functions

See Properties declarations on functions

Version selection with `typesVersions`

See Version selection with `typesVersions`

Matching behavior

See Matching behavior

Multiple fields

See Multiple fields

TypeScript 3.0

See TypeScript 3.0

Tuples in rest parameters and spread expressions

See Tuples in rest parameters and spread expressions

Rest parameters with tuple types

See Rest parameters with tuple types

Spread expressions with tuple types

See Spread expressions with tuple types

Generic rest parameters

See Generic rest parameters

Example See Example

Optional elements in tuple types

See Optional elements in tuple types

Example See Example

Rest elements in tuple types

See Rest elements in tuple types

Example See Example

New `unknown` top type

See New `unknown` top type

Example See Example

Support for `defaultProps` in JSX

See Support for `defaultProps` in JSX

Caveats

See Caveats

Explicit types on `defaultProps` See Explicit types on `defaultProps`

Changes to `@types/React` See Changes to `@types/React`

`/// <reference lib="..." />` reference directives

See `/// <reference lib="..." />` reference directives

Example See Example

TypeScript 2.9

See TypeScript 2.9

Support `number` and `symbol` named properties with `keyof` and mapped types

See Support `number` and `symbol` named properties with `keyof` and mapped types

Example See Example

Example See Example

Recommendations See Recommendations

Generic type arguments in JSX elements

See Generic type arguments in JSX elements

Example See Example

Generic type arguments in generic tagged templates

See Generic type arguments in generic tagged templates

Example See Example

`import` types

See `import` types

Example See Example

Relaxing declaration emit visiblity rules

See Relaxing declaration emit visiblity rules

Support for `import.meta`

See Support for `import.meta`

Example See Example

New `--resolveJsonModule`

See New `--resolveJsonModule`

Example See Example

`--pretty` output by default

See `--pretty` output by default

New `--declarationMap`

See New `--declarationMap`

TypeScript 2.8

See TypeScript 2.8

Conditional Types

See Conditional Types

Example See Example

Distributive conditional types

See Distributive conditional types

Example See Example

Example See Example

Example See Example

Type inference in conditional types

See Type inference in conditional types

Predefined conditional types

See Predefined conditional types

Example See Example

Improved control over mapped type modifiers

See Improved control over mapped type modifiers

Example See Example

Example See Example

Example See Example

Improved `keyof` with intersection types

See Improved `keyof` with intersection types

Example See Example

Better handling for namespace patterns in `.js` files

See Better handling for namespace patterns in `.js` files

IIFEs as namespace declarations

See IIFEs as namespace declarations

Defaulted declarations

See Defaulted declarations

Prototype assignment

See Prototype assignment

Nested and merged declarations

See Nested and merged declarations

Per-file JSX factories

See Per-file JSX factories

Example See Example

Locally scoped JSX namespaces

See Locally scoped JSX namespaces

New `--emitDeclarationsOnly`

See New `--emitDeclarationsOnly`

TypeScript 2.7

See TypeScript 2.7

Constant-named properties

See Constant-named properties

Example See Example

Example See Example

`unique symbol`

See `unique symbol`

Example See Example

Example See Example

Strict Class Initialization

See Strict Class Initialization

Definite Assignment Assertions

See Definite Assignment Assertions

Fixed Length Tuples

See Fixed Length Tuples

Improved type inference for object literals

See Improved type inference for object literals

Example See Example

Improved handling of structurally identical classes and `instanceof` expressions

See Improved handling of structurally identical classes and `instanceof` expressions

Example: See Example:

Type guards inferred from `in` operator

See Type guards inferred from `in` operator

Example See Example

Support for `import d from "cjs"` form CommonJS modules with `--esModuleInterop`

See Support for `import d from "cjs"` form CommonJS modules with `--esModuleInterop`

Example See Example

Numeric separators

See Numeric separators

Example See Example

Cleaner output in `--watch` mode

See Cleaner output in `--watch` mode

Prettier `--pretty` output

See Prettier `--pretty` output

TypeScript 2.6

See TypeScript 2.6

Strict function types

See Strict function types

Example See Example

Note: See Note:

Support for JSX Fragment Syntax

See Support for JSX Fragment Syntax

Cache tagged template objects in modules

See Cache tagged template objects in modules

Example See Example

Localized diagnostics on the command line

See Localized diagnostics on the command line

Example See Example

Suppress errors in .ts files using '// @ts-ignore' comments

See Suppress errors in .ts files using '// @ts-ignore' comments

Example See Example

Faster `tsc --watch`

See Faster `tsc --watch`

Write-only references now flagged as unused

See Write-only references now flagged as unused

Example See Example

Example See Example

TypeScript 2.5

See TypeScript 2.5

Optional `catch` clause variables

See Optional `catch` clause variables

Type assertion/cast syntax in `checkJs`/`@ts-check` mode

See Type assertion/cast syntax in `checkJs`/`@ts-check` mode

Deduplicated and redirected packages

See Deduplicated and redirected packages

The `--preserveSymlinks` compiler flag

See The `--preserveSymlinks` compiler flag

TypeScript 2.4

See TypeScript 2.4

Dynamic Import Expressions

See Dynamic Import Expressions

String Enums

See String Enums

Improved inference for generics

See Improved inference for generics

Return types as inference targets

See Return types as inference targets

Type parameter inference from contextual types

See Type parameter inference from contextual types

Stricter checking for generic functions

See Stricter checking for generic functions

Strict contravariance for callback parameters

See Strict contravariance for callback parameters

Weak Type Detection

See Weak Type Detection

TypeScript 2.3

See TypeScript 2.3

Generators and Iteration for ES5/ES3

See Generators and Iteration for ES5/ES3

Iterators See Iterators

Generators See Generators

New `--downlevelIteration` See New `--downlevelIteration`

Async Iteration

See Async Iteration

Async iterators See Async iterators

Async Generators See Async Generators

The `for-await-of` Statement See The `for-await-of` Statement

Caveats See Caveats

Generic parameter defaults

See Generic parameter defaults

Example See Example

New `--strict` master option

See New `--strict` master option

Enhanced `--init` output

See Enhanced `--init` output

Errors in .js files with `--checkJs`

See Errors in .js files with `--checkJs`

TypeScript 2.2

See TypeScript 2.2

Support for Mix-in classes

See Support for Mix-in classes

First some terminology: See First some terminology:

Putting all of the above rules together in an example: See Putting all of the above rules together in an example:

`object` type

See `object` type

Support for `new.target`

See Support for `new.target`

Example See Example

Better checking for `null`/`undefined` in operands of expressions

See Better checking for `null`/`undefined` in operands of expressions

Dotted property for types with string index signatures

See Dotted property for types with string index signatures

Support for spread operator on JSX element children

See Support for spread operator on JSX element children

Example See Example

New `jsx: react-native`

See New `jsx: react-native`

TypeScript 2.1

See TypeScript 2.1

`keyof` and Lookup Types

See `keyof` and Lookup Types

Example See Example

Example See Example

Mapped Types

See Mapped Types

`Partial`, `Readonly`, `Record`, and `Pick`

See `Partial`, `Readonly`, `Record`, and `Pick`

Object Spread and Rest

See Object Spread and Rest

Downlevel Async Functions

See Downlevel Async Functions

Example See Example

tsconfig.json See tsconfig.json

dramaticWelcome.ts See dramaticWelcome.ts

Support for external helpers library (`tslib`)

See Support for external helpers library (`tslib`)

Untyped imports

See Untyped imports

Example See Example

Support for `--target ES2016`, `--target ES2017` and `--target ESNext`

See Support for `--target ES2016`, `--target ES2017` and `--target ESNext`

Improved `any` Inference

See Improved `any` Inference

Example See Example

Implicit any errors

See Implicit any errors

Example See Example

Better inference for literal types

See Better inference for literal types

Example See Example

Example See Example

Use returned values from super calls as 'this'

See Use returned values from super calls as 'this'

Example See Example

Configuration inheritance

See Configuration inheritance

Example See Example

New `--alwaysStrict`

See New `--alwaysStrict`

TypeScript 2.0

See TypeScript 2.0

Null- and undefined-aware types

See Null- and undefined-aware types

`--strictNullChecks`

See `--strictNullChecks`

Example See Example

Assigned-before-use checking

See Assigned-before-use checking

Example See Example

Optional parameters and properties

See Optional parameters and properties

Non-null and non-undefined type guards

See Non-null and non-undefined type guards

Example See Example

Dotted names in type guards

See Dotted names in type guards

Example See Example

Expression operators

See Expression operators

Type widening

See Type widening

Non-null assertion operator

See Non-null assertion operator

Compatibility

See Compatibility

Control flow based type analysis

See Control flow based type analysis

Example See Example

Tagged union types

See Tagged union types

Example See Example

The `never` type

See The `never` type

Read-only properties and index signatures

See Read-only properties and index signatures

Example See Example

Specifying the type of `this` for functions

See Specifying the type of `this` for functions

`this` parameters in callbacks

See `this` parameters in callbacks

Example See Example

`--noImplicitThis`

See `--noImplicitThis`

Glob support in `tsconfig.json`

See Glob support in `tsconfig.json`

Example See Example

Module resolution enhancements: BaseUrl, Path mapping, rootDirs and tracing

See Module resolution enhancements: BaseUrl, Path mapping, rootDirs and tracing

Base URL

See Base URL

Example See Example

Path mapping

See Path mapping

Example See Example

Virtual Directories with `rootDirs`

See Virtual Directories with `rootDirs`

Example See Example

Tracing module resolution

See Tracing module resolution

Shorthand ambient module declarations

See Shorthand ambient module declarations

declarations.d.ts See declarations.d.ts

Wildcard character in module names

See Wildcard character in module names

Example See Example

Example See Example

Support for UMD module definitions

See Support for UMD module definitions

math-lib.d.ts See math-lib.d.ts

Optional class properties

See Optional class properties

Example See Example

Private and Protected Constructors

See Private and Protected Constructors

Example See Example

Abstract properties and accessors

See Abstract properties and accessors

Example See Example

Implicit index signatures

See Implicit index signatures

Including built-in type declarations with `--lib`

See Including built-in type declarations with `--lib`

Example See Example

Flag unused declarations with `--noUnusedParameters` and `--noUnusedLocals`

See Flag unused declarations with `--noUnusedParameters` and `--noUnusedLocals`

Example See Example

Module identifiers allow for `.js` extension

See Module identifiers allow for `.js` extension

Support 'target : es5' with 'module: es6'

See Support 'target : es5' with 'module: es6'

Trailing commas in function parameter and argument lists

See Trailing commas in function parameter and argument lists

Example See Example

New `--skipLibCheck`

See New `--skipLibCheck`

Allow duplicate identifiers across declarations

See Allow duplicate identifiers across declarations

Example See Example

New `--declarationDir`

See New `--declarationDir`

TypeScript 1.8

See TypeScript 1.8

Type parameters as constraints

See Type parameters as constraints

Example See Example

Control flow analysis errors

See Control flow analysis errors

Unreachable code

See Unreachable code

Example See Example

Unused labels

See Unused labels

Example See Example

Implicit returns

See Implicit returns

Example See Example

Case clause fall-throughs

See Case clause fall-throughs

Example See Example

Stateless Function Components in React

See Stateless Function Components in React

Simplified `props` type management in React

See Simplified `props` type management in React

Augmenting global/module scope from modules

See Augmenting global/module scope from modules

Example See Example

Example See Example

String literal types

See String literal types

Improved union/intersection type inference

See Improved union/intersection type inference

Example See Example

Concatenate `AMD` and `System` modules with `--outFile`

See Concatenate `AMD` and `System` modules with `--outFile`

Example See Example

Support for `default` import interop with SystemJS

See Support for `default` import interop with SystemJS

Allow captured `let`/`const` in loops

See Allow captured `let`/`const` in loops

Example See Example

Improved checking for `for..in` statements

See Improved checking for `for..in` statements

Example See Example

Modules are now emitted with a `"use strict";` prologue

See Modules are now emitted with a `"use strict";` prologue

Including `.js` files with `--allowJs`

See Including `.js` files with `--allowJs`

Custom JSX factories using `--reactNamespace`

See Custom JSX factories using `--reactNamespace`

Example See Example

`this`-based type guards

See `this`-based type guards

Example See Example

Official TypeScript NuGet package

See Official TypeScript NuGet package

Prettier error messages from `tsc`

See Prettier error messages from `tsc`

Colorization of JSX code in VS 2015

See Colorization of JSX code in VS 2015

The `--project` (`-p`) flag can now take any file path

See The `--project` (`-p`) flag can now take any file path

Allow comments in tsconfig.json

See Allow comments in tsconfig.json

Support output to IPC-driven files

See Support output to IPC-driven files

Improved support for `tsconfig.json` in Visual Studio 2015

See Improved support for `tsconfig.json` in Visual Studio 2015

A couple of limitations:

See A couple of limitations:

TypeScript 1.7

See TypeScript 1.7

`async`/`await` support in ES6 targets (Node v4+)

See `async`/`await` support in ES6 targets (Node v4+)

Example See Example

Support for `--target ES6` with `--module`

See Support for `--target ES6` with `--module`

Example See Example

`this`-typing

See `this`-typing

ES7 exponentiation operator

See ES7 exponentiation operator

Example See Example

Improved checking for destructuring object literal

See Improved checking for destructuring object literal

Example See Example

Support for decorators when targeting ES3

See Support for decorators when targeting ES3

TypeScript 1.6

See TypeScript 1.6

JSX support

See JSX support

New `.tsx` file extension and `as` operator See New `.tsx` file extension and `as` operator

Using React See Using React

Using other JSX frameworks See Using other JSX frameworks

Output generation See Output generation

Intersection types

See Intersection types

Example See Example

Local type declarations

See Local type declarations

Class expressions

See Class expressions

Extending expressions

See Extending expressions

`abstract` classes and methods

See `abstract` classes and methods

Examples See Examples

Generic type aliases

See Generic type aliases

Stricter object literal assignment checks

See Stricter object literal assignment checks

Examples See Examples

ES6 generators

See ES6 generators

Experimental support for `async` functions

See Experimental support for `async` functions

Example See Example

Nightly builds

See Nightly builds

Adjustments in module resolution logic

See Adjustments in module resolution logic

Merging ambient class and interface declaration

See Merging ambient class and interface declaration

User-defined type guard functions

See User-defined type guard functions

Examples See Examples

`exclude` property support in tsconfig.json

See `exclude` property support in tsconfig.json

`--init` command line option

See `--init` command line option

TypeScript 1.5

See TypeScript 1.5

ES6 Modules

See ES6 Modules ##

Export Declarations See Export Declarations

Re-exporting See Re-exporting

Default Export See Default Export

Bare Import See Bare Import

Destructuring in declarations and assignments

See Destructuring in declarations and assignments

Declarations See Declarations

Assignments See Assignments

`namespace` keyword

See `namespace` keyword

`let` and `const` support

See `let` and `const` support

Const See Const

Block scoped See Block scoped

for..of support

See for..of support

Example See Example

Decorators

See Decorators

Example See Example

Computed properties

See Computed properties

Support for `UMD` and `System` module output

See Support for `UMD` and `System` module output

Unicode codepoint escapes in strings

See Unicode codepoint escapes in strings

Tagged template strings in ES3/ES5

See Tagged template strings in ES3/ES5

AMD-dependency optional names

See AMD-dependency optional names

Project support through `tsconfig.json`

See Project support through `tsconfig.json`

Example See Example

`--rootDir` command line option

See `--rootDir` command line option

`--noEmitHelpers` command line option

See `--noEmitHelpers` command line option

`--newLine` command line option

See `--newLine` command line option

`--inlineSourceMap` and `inlineSources` command line options

See `--inlineSourceMap` and `inlineSources` command line options

TypeScript 1.4

See TypeScript 1.4

Union types

See Union types

Overview

See Overview

Stricter Generics

See Stricter Generics

Better Type Inference

See Better Type Inference

`let` declarations

See `let` declarations

`const` declarations

See `const` declarations

Template strings

See Template strings

Type Guards

See Type Guards

Type Aliases

See Type Aliases

`const enum` (completely inlined enums)

See `const enum` (completely inlined enums)

`-noEmitOnError` commandline option

See `-noEmitOnError` commandline option

AMD Module names

See AMD Module names

TypeScript 1.3

See TypeScript 1.3

Protected

See Protected

Tuple types

See Tuple types

TypeScript 1.1

See TypeScript 1.1

Performance Improvements

See Performance Improvements

Better Module Visibility Rules

See Better Module Visibility Rules