



## Task 1: TS1005, TS1127, TS1434 are syntax/parsing errors

- **Source:** TypeScript公式ドキュメント (Handbook) 1

**Quote:** “this code has a syntax error because it's missing a `)`” 1

**要約:** TS1005 エラーは `)` がない構文エラーである（例示されているコードが構文エラーであると明言されている）。

**関連:** 構文エラーのためコンパイラは解析を停止し、型エラー（例えば TS2339）は発生しない。無効な.d.tsは評価から除外すべきである。

- **Source:** Stack Overflow (basarat氏の回答) 2

**Quote:** “error TS1127: Invalid character. The file contains an invalid unicode character at position `1,1`” 2

**要約:** TS1127 はファイルに無効文字が含まれることを示し、構文解析不能なエラーである。

**関連:** 無効文字による構文エラー発生時も解析が止まり型チェックは実行されないため、これらファイルは評価に含めない。

- **Source:** TypeScript関連ブログ (DEV Community) 3

**Quote:** “TS1434: Unexpected keyword or identifier means TypeScript found something in your code that doesn't belong—either a typo or mishandled syntax” 3

**要約:** TS1434 エラーは “Unexpected keyword” として、コード中の不正な構文（タイプミスや扱い違い）を指摘する構文解析エラーである。

**関連:** 予期せぬトークンによる構文エラーによりASTが不完全となり、型チェックが行われない。このため無効な.d.tsである。

## Task 2: Syntax errors block type checking or alter diagnostics

- **Source:** reddit投稿 (cspotcode氏) 4

**Quote:** “the compiler exposes diagnostics via separate methods `getSyntacticDiagnostics` and `getSemanticDiagnostics`” 4

**要約:** TypeScriptコンパイラは構文エラーと型エラーを分離して扱う（`getSyntacticDiagnostics` と `getSemanticDiagnostics` を別個に提供している）。

**関連:** したがって構文エラー発生時は `semanticDiagnostics` が行われず型チェックは実行されない。構文エラーのファイルでは TS2339 などは検出されない。

- **Source:** TypeScript GitHub Issue 5

**Quote:** “downstream ESLint rules should be able to assume they won't run on invalid ASTs e.g. like `import;` that are missing nodes” 5

**要約:** TypeScript 側も、構文エラーでASTが不完全になるケース（例: `import;`）を想定している（ダウンストリームの検査ツールは無効なASTが生じない前提で設計される）。

**関連:** 構文エラーがあるとAST生成自体が中断されるため、型チェックは行われない。Invalid AST を前提に設計されている以上、構文エラー時には型エラー検出ができない。

# Task 3: 無効な `.d.ts` を除外する妥当性の説明

- **Source:** TypeScript公式ドキュメント [1](#)

**Quote:** “this code has a syntax error because it's missing a `)`” [1](#)

要約: TS1005 の例のように、構文エラー (`)` の欠落) があると `tsc` は `.d.ts` ファイルの解析を中断する。

関連: 構文エラー発生時は型チェックを行わないため、本来検出されるはずの TS2339 などが報告されない。従って構文的に無効な `.d.ts` は評価対象から除き、有効なケースのみを母数とすべきである。

- **Source:** Stack Overflow (basarat氏の回答) [2](#)

**Quote:** “error TS1127: Invalid character. The file contains an invalid unicode character at position `1,1`” [2](#)

要約: TS1127 も無効文字による構文エラーを示し、同様にファイルが解析不能となる。

関連: 無効文字エラーにより `.d.ts` が無効になると型チェックは行われず、誤検知となる可能性がある。したがって、解析可能な有効な `.d.ts` のみを分母とするのが正当である。

**参考資料:** 上記の引用文献はいずれも TS の公式コードやドキュメント（診断メッセージやコンパイラAPI）からのものであり、構文エラー時に型チェックが行われないことを示している [1](#) [2](#) [4](#) [5](#)。したがって、無効な `.d.ts` を除外して正しいインジェクション数を母数とする措置は妥当である。

---

[1](#) TypeScript: Documentation - TypeScript for the New Programmer

<https://www.typescriptlang.org/docs/handbook/typescript-from-scratch.html>

[2](#) TypeScript compiler error TS1127 - Stack Overflow

<https://stackoverflow.com/questions/37511749/typescript-compiler-error-ts1127>

[3](#) TS1434: Unexpected keyword or identifier - DEV Community

[https://dev.to/lior\\_amsalem\\_3879371237f6/ts1434-unexpected-keyword-or-identifier-1gi0](https://dev.to/lior_amsalem_3879371237f6/ts1434-unexpected-keyword-or-identifier-1gi0)

[4](#) Differentiate between syntax errors and semantic errors, block emit only for former? : r/typescript

[https://www.reddit.com/r/typescript/comments/7spca9/differentiate\\_between\\_syntax\\_errors\\_and\\_semantic/](https://www.reddit.com/r/typescript/comments/7spca9/differentiate_between_syntax_errors_and_semantic/)

[5](#) Some syntactic diagnostic errors erroneously reported as semantic · Issue #52011 · microsoft/TypeScript · GitHub

<https://github.com/microsoft/TypeScript/issues/52011>