5.6 Break Statements

A 'break' statement is required to be nested, directly or indirectly (but not crossing function boundaries), within an iteration ('do', 'while', 'for', or 'for-in') or 'switch' statement. When a 'break' statement includes a target label, that target label must appear in the label set of an enclosing (but not crossing function boundaries) statement.

5.7 Return Statements

It is an error for a 'return' statement to occur outside a function body. Specifically, 'return' statements are not permitted at the global level or in module bodies.

A 'return' statement without an expression returns the value 'undefined' and is permitted in the body of any function, regardless of the return type of the function.

When a 'return' statement includes an expression, if the containing function includes a return type annotation, the return expression is contextually typed (section 4.19) by that return type and must be of a type that is assignable to the return type. Otherwise, if the containing function is contextually typed by a type *T*, *Expr* is contextually typed by *T*'s return type.

In a function implementation without a return type annotation, the return type is inferred from the 'return' statements in the function body, as described in section 6.3.

In the example

```
function f(): (x: string) => number {
    return s => s.length;
}
```

the arrow expression in the 'return' statement is contextually typed by the return type of 'f', thus giving type 'string' to 's'.

5.8 With Statements

Use of the 'with' statement in TypeScript is an error, as is the case in ECMAScript 5's strict mode. Furthermore, within the body of a 'with' statement, TypeScript considers every identifier occurring in an expression (section 4.3) to be of the Any type regardless of its declared type. Because the 'with' statement puts a statically unknown set of identifiers in scope in front of those that are statically known, it is not possible to meaningfully assign a static type to any identifier.

5.9 Switch Statements

In a 'switch' statement, each 'case' expression must be of a type that is assignable to or from (section 3.8.4) the type of the 'switch' expression.