3.3 Object Types

The object types include references to class and interface types as well as anonymous object types created by a number of constructs such as object literals, function declarations, and module declarations. Object types are composed from properties, call signatures, construct signatures, and index signatures, collectively called members.

3.3.1 Named Type References

Type references (section 3.6.2) to class and interface types are classified as object types. Type references to generic class and interface types include type arguments that are substituted for the type parameters of the class or interface to produce an actual object type.

3.3.2 Array Types

Array types represent JavaScript arrays. Array types are type references (section 3.6.2) created from the generic interface type 'Array' in the global module. Array type literals (section 3.6.4) provide a shorthand notation for creating such references.

Array literals (section 4.6) may be used to create values of array types.

3.3.3 Anonymous Types

Several constructs in the TypeScript language introduce new anonymous object types:

- Function and constructor type literals (section 3.6.4).
- Object type literals (section 3.7).
- Object literals (section 4.5).
- Function expressions (section 4.9) and function declarations (6.1).
- Constructor function types created by class declarations (section 8.2.5).
- Module instance types created by module declarations (section 10.3).

3.3.4 Members

Every object type is composed from zero or more of the following kinds of members:

- ***Properties***, which define the names and types of the properties of objects of the given type. Property names are unique within their type.
- ***Call signatures***, which define the possible parameter lists and return types associated with applying call operations to objects of the given type.
- ***Construct signatures***, which define the possible parameter lists and return types associated with applying the `new` operator to objects of the given type.
- ***Index signatures***, which define type constraints for properties in the given type. An object type can have at most one string index signature and one numeric index signature.

Properties are either ***public*** or ***private*** and are either ***required*** or ***optional***: