A property assignment of the form

PropertyName *CallSignature* `{` *FunctionBody* `}`

is simply shorthand for

PropertyName `:` `function` *CallSignature* `{` *FunctionBody* `}`

Each property assignment in an object literal is processed as follows:

- If the object literal is contextually typed and the contextual type contains a property with a matching name, the property assignment is contextually typed by the type of that property.
- Otherwise, if the object literal is contextually typed, the contextual type contains a numeric index signature, and the property assignment specifies a numeric property name, the property assignment is contextually typed by the type of the numeric index signature.
- Otherwise, if the object literal is contextually typed and the contextual type contains a string index signature, the property assignment is contextually typed by the type of the string index signature.
- Otherwise, the property assignment is processed without a contextual type.

The type of a property introduced by a property assignment of the form *Name* `:` *Expr* is the widened form (section 3.9) of the type of *Expr*.

A get accessor declaration is processed in the same manner as an ordinary function declaration (section 6.1) with no parameters. A set accessor declaration is processed in the same manner as an ordinary function declaration with a single parameter and a Void return type. When both a get and set accessor is declared for a property:

- If both accessors include type annotations, the specified types must be identical.
- If only one accessor includes a type annotation, the other behaves as if it had the same type annotation.
- If neither accessor includes a type annotation, the inferred return type of the get accessor becomes the parameter type of the set accessor.

If a get accessor is declared for a property, the return type of the get accessor becomes the type of the property. If only a set accessor is declared for a property, the parameter type (which may be type Any if no type annotation is present) of the set accessor becomes the type of the property.

When an object literal is contextually typed by a type that includes a string index signature of type *T*, the resulting type of the object literal includes a string index signature with the widened form of the best common type of *T* and the types of the properties declared in the object literal. Likewise, when an object literal is contextually typed by a type that includes a numeric index signature of type *T*, the resulting type of the object literal includes a numeric index signature with the widened form of the best common type of *T* and the types of the numerically named properties (section 3.7.4) declared in the object literal.