3.2.2 The Boolean Type

The Boolean primitive type corresponds to the similarly named JavaScript primitive type and represents logical values that are either true or false.

The `boolean` keyword references the Boolean primitive type and the `true` and `false` literals reference the two Boolean truth values.

For purposes of determining type relationships (section 3.8) and accessing properties (section 4.10), the Boolean primitive type behaves as an object type with the same properties as the global interface type 'Boolean'.

Some examples:

```
var b: boolean;         // Explicitly typed
var yes = true;         // Same as yes: boolean = true
var no = false;         // Same as no: boolean = false
```

3.2.3 The String Type

The String primitive type corresponds to the similarly named JavaScript primitive type and represents sequences of characters stored as Unicode UTF-16 code units.

The `string` keyword references the String primitive type and string literals may be used to write values of the String primitive type.

For purposes of determining type relationships (section 3.8) and accessing properties (section 4.10), the String primitive type behaves as an object type with the same properties as the global interface type 'String'.

Some examples:

```
var s: string;          // Explicitly typed
var empty = "";         // Same as empty: string = ""
var abc = 'abc';        // Same as abc: string = "abc"
var c = abc.charAt(2);  // Property of String interface
```

3.2.4 The Void Type

The Void type, referenced by the `void` keyword, represents the absence of a value and is used as the return type of functions with no return value.

The only possible values for the Void type are `null` and `undefined`. The Void type is a subtype of the Any type and a supertype of the Null and Undefined types, but otherwise Void is unrelated to all other types.

NOTE: We might consider disallowing declaring variables of type Void as they serve no useful purpose. However, because Void is permitted as a type argument to a generic type or function it is not feasible to disallow Void properties or parameters.