```
function getValue() { ... }

var s = getValue() + "";
```

The example above converts the result of 'getValue()' to a string if it isn't a string already. The type inferred for 's' is the String primitive type regardless of the return type of 'getValue'.

### 4.15.3 The <, >, <=, >=, ==, !=, ===, and !== operators

These operators require one operand type to be identical to or a subtype of the other operand type. The result is always of the Boolean primitive type.

| | Any | Boolean | Number | String | Object |
|---|---|---|---|---|---|
| Any | Boolean | Boolean | Boolean | Boolean | Boolean |
| Boolean | Boolean | Boolean | | | |
| Number | Boolean | | Boolean | | |
| String | Boolean | | | Boolean | |
| Object | Boolean | | | | Boolean |

### 4.15.4 The instanceof operator

The `instanceof` operator requires the left operand to be of type Any, an object type, or a type parameter type, and the right operand to be of type Any or a subtype of the 'Function' interface type. The result is always of the Boolean primitive type.

Note that object types containing one or more call or construct signatures are automatically subtypes of the 'Function' interface type, as described in section 3.3.

### 4.15.5 The in operator

The `in` operator requires the left operand to be of type Any, the String primitive type, or the Number primitive type, and the right operand to be of type Any, an object type, or a type parameter type. The result is always of the Boolean primitive type.

### 4.15.6 The && operator

The && operator permits the operands to be of any type and produces a result of the same type as the second operand.