Elvis operator

In certain computer programming languages, the **Elvis operator**, often written `?:`, `or` or `||`, is a binary operator that returns its first operand if that operand evaluates to a true value, and otherwise evaluates and returns its second operand. This is identical to a short-circuit *or* with "last value" semantics. The notation of the Elvis operator was inspired by the ternary conditional operator, `? :` since the Elvis operator expression `A ?: B` is approximately equivalent to the ternary conditional `A ? A : B`.

The name "Elvis operator" refers to the fact that when its common notation, `?:`, is viewed sideways, it resembles an emoticon of Elvis Presley with his quiff.[1]

A similar operator is the null coalescing operator, where the check for boolean truthiness is replaced with a check for non-null instead. This is usually written `??`, and can be seen in languages like C#.[2]

Contents

Example

Boolean variant

In a language that supports the Elvis operator, something like this:

```
x = f() ?: g()
```

will set `x` equal to the result of `f()` if that result is a true value, and to the result of `g()` otherwise.

It is equivalent to this example, using the conditional ternary operator:

```
x = f() ? f() : g()
```

except that it does not evaluate the `f()` twice if it is true.

Object reference variant

This code will result in a reference to an object that is guaranteed to not be null. Function `f()` returns an object reference instead of a boolean, and may return null:

```
x = f() ?: "default value"
```

Languages supporting the Elvis operator

- In GNU C and C++ (that is: in C and C++ with GCC extensions), the second operand of the ternary operator is optional.[3] This has been the case since at least GCC 2.95.3 (March 2001), and seems to be *the* original elvis operator.[4]
- In Apache Groovy, the "Elvis operator" `?:` is documented as a distinct operator;[5] this feature was added in Groovy 1.5[6] (December 2007). Groovy, unlike GNU C and PHP, does *not* simply allow the second operand of ternary `?:` to be omitted; rather, binary `?:` must be written as a single operator, with no whitespace in between.
- In PHP, it is possible to leave out the middle part of the ternary operator since PHP 5.3.[7] (June 2009).
- The Fantom programming language has the `?:` binary operator that compares its first operand with `null`.
- In Kotlin, the Elvis operator returns its left-hand side if it is not null, and its right-hand side otherwise.[8] A common pattern is to use it with `return`, like this: `val foo = bar() ?: return`
- In Gosu, the `?:` operator returns the right operand if the left is null as well.
- In C#, the null-conditional operator, `?.` is referred to as the "Elvis operator",[9] but it does not perform the same function. Instead, the null-coalescing operator `??` does.
- In ColdFusion and CFML, the Elvis operator was introduced using the `?:` syntax.
- The Xtend programming language has an Elvis operator.[10]
- In Google's Closure Templates, the Elvis operator is a null coalescing operator, equivalent to `isNonnull($a) ? $a : $b`.[11]
- Swift supports this concept with its Nil-coalescing operator `??`,[12] e.g. `(a ?? b)`.
- SQL supports this concept with its COALESCE function, e.g. `COALESCE(a, b)`.
- In Ballerina, the Elvis operator `L ?: R` returns the value of `L` if it's not nil. Otherwise, return the value of `R`.[13]
- Clojure supports this concept with the `or`[14] macro, e.g. `(or a b)`. In the case of Clojure, it is var-arg, and not binary, e.g. `(or a b c d e)` will return the first non false value.
- Dart language provides ?? operator which returns right side value if left side value is null
- TypeScript supports this concept with its nullish-coalescing operator `??`, e.g. `(a ?? b)`, since v3.7.[15]
- JavaScript supports this concept, same as TypeScript[16]
- Lua supports this concept with the `or`[17] logical operator, e.g. `(a or b)`.

Analogous use of the short-circuiting OR operator

In several languages, such as Common Lisp, Clojure, Lua, Object Pascal, Perl, Python, Ruby, and JavaScript, the OR operator (typically `||` or `or`) has the same behavior as the above: returning its first operand if it would evaluate to true in a boolean environment, and otherwise evaluating and returning its second operand. When

the left hand side is true, the right hand side is not even evaluated; it is "short-circuited." This is different than the behavior in other languages such as C/C++, where the result of `||` will always be a boolean.

See also

- **? :** or conditional operator, when used as a ternary operator
- Safe navigation operator, often **?.**
- Spaceship operator <=>
- Option type

References

1. Joyce Farrell. *Java Programming*. p. 276. ISBN 978-1285081953. "The new operator is called Elvis operator because it uses a question mark and a colon together (?:); if you view it sideways, it reminds you of Elvis Presley."
2. "?? Operator" (https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/operators/null-coalescing-operator). *C# Reference*. Microsoft. Retrieved 5 December 2018.
3. "Using the GNU Compiler Collection (GCC): Conditionals" (https://gcc.gnu.org/onlinedocs/gcc/Conditionals.html#Conditionals). *gcc.gnu.org*.
4. "Using and Porting the GNU Compiler Collection (GCC): C Extensions" (https://gcc.gnu.org/onlinedocs/gcc-2.95.3/gcc_4.html#SEC70). *gcc.gnu.org*.
5. "Elvis Operator (?:)" (http://docs.groovy-lang.org/latest/html/documentation/index.html#_elvis_operator).
6. "The Apache Groovy programming language - Groovy 1.5 release notes" (http://groovy-lang.org/releasenotes/groovy-1.5.html). *groovy-lang.org*.
7. "PHP: Comparison Operators - Manual" (http://php.net/manual/en/language.operators.comparison.php#language.operators.comparison.ternary). PHP website. Retrieved 2014-02-17.
8. "Null Safety - Kotlin Programming Language" (https://kotlinlang.org/docs/reference/null-safety.html#elvis-operator). *Kotlin*.
9. Albahari, Joseph; Albahari, Ben (2015). *C# 6.0 in a Nutshell* (6 ed.). O'Reilly Media. p. 59. ISBN 978-1491927069.
10. Efftinge, Sven. "Xtend - Expressions" (https://eclipse.org/xtend/documentation/203_xtend_expressions.html#elvis-operator). *eclipse.org*.
11. "Closure Templates - Expressions" (https://github.com/google/closure-templates/blob/master/documentation/reference/expressions.md#null-coalescing-operator--null-coalescing-operator). *GitHub*.
12. "The Swift Programming Language (Swift 4.1): Basic Operators" (https://developer.apple.com/library/content/documentation/Swift/Conceptual/Swift_Programming_Language/BasicOperators.html). *developer.apple.com*.
13. "Elvis Operator - Ballerina Programming Language" (https://ballerina.io/learn/by-example/elvis-operator.html). *Ballerina*.
14. "clojure.core or macro API reference" (https://clojure.github.io/clojure/clojure.core-api.html#clojure.core/or).
15. "nullish coalescing commit by Kingwl · Pull Request #32883 · microsoft/TypeScript" (https://github.com/microsoft/TypeScript/pull/32883). *GitHub*. Retrieved 2019-10-08.
16. "Nullish coalescing operator (??)" (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Nullish_coalescing_operator). *mozilla*. Retrieved 2021-01-28.
17. "Lua or operator Reference" (https://www.lua.org/manual/5.4/manual.html#3.4.5).