IF

| English (en) | français (fr) | 日本語 (ja) | 中文（中国大陆）(zh_CN) |

◄ ▲ ►

navigation
- Main Page
- Documentation
- FAQ
- Downloads
- Glossary
- Index
- Recent changes
- Random page
- Help

search
Go Search

tools
- What links here
- Related changes
- Special pages
- Printable version
- Permanent link
- Page information

3Ca - IF (author: Tao Yue, state: unchanged)

The IF statement allows you to branch based on the result of a Boolean operation. The one-way branch format is:

```
if BooleanExpression then
  StatementIfTrue;
```

If the Boolean expression evaluates to true, the statement executes. Otherwise, it is skipped.

The IF statement accepts only one statement. If you would like to branch to a compound statement, you must use a begin-end block to enclose the statements:

```
if BooleanExpression then
begin
  Statement1;
  Statement2
end;
```

There is also a two-way selection:

```
if BooleanExpression then
  StatementIfTrue
else
  StatementIfFalse;
```

If the Boolean expression evaluates to FALSE, the statement following the else will be performed. Note that you may not use a semicolon after the statement preceding the else. That causes the computer to treat it as a one-way selection, leaving it to wonder where the else came from.

If you need multi-way selection, simply nest if statements:

```
if Condition1 then
  Statement1
else
  if Condition2 then
    Statement2
  else
    Statement3;
```

Be careful with nesting. Sometimes the computer won't do what you want it to do:

```
if Condition1 then
  if Condition2 then
    Statement2
else
  Statement1;
```

The else is always matched with the most recent if, so the computer interprets the preceding block of code as:

```
if Condition1 then
  if Condition2 then
    Statement2
  else
    Statement1;
```

You can get by with a null statement:

```
if Condition1 then
  if Condition2 then
    Statement2
  else
else
  Statement1;
```

Or you could use a begin-end block. But the best way to clean up the code would be to rewrite the condition.

```
if not Condition1 then
  Statement1
else
  if Condition2 then
    Statement2;
```

This example illustrates where the not operator comes in very handy. If Condition1 had been a Boolean like: (not(a < b) or (c + 3 > 6)) and g, reversing the expression would be more difficult than NOTting it.

Also notice how important indentation is to convey the logic of program code to a human, but the compiler ignores the indentation.

◄ ▲ ►

Category: Object Pascal Introduction