

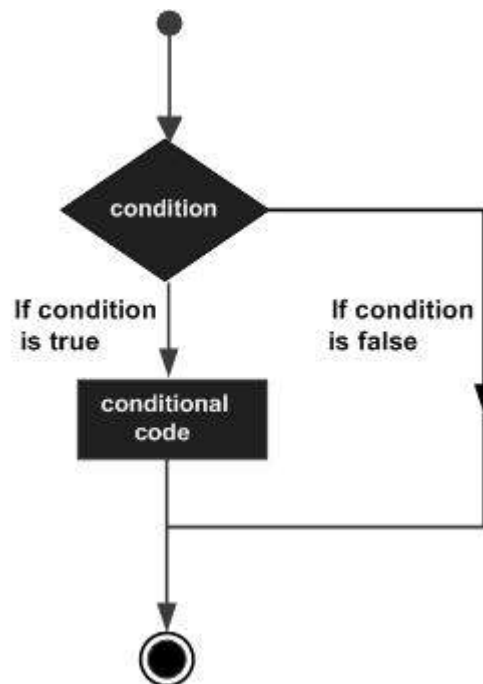
# PERL CONDITIONAL STATEMENTS - IF...ELSE

[http://www.tutorialspoint.com/perl/perl\\_conditions.htm](http://www.tutorialspoint.com/perl/perl_conditions.htm)

Copyright © tutorialspoint.com

Perl conditional statements helps in the decision making, which require that the programmer specifies one or more conditions to be evaluated or tested by the program, along with a statement or statements to be executed if the condition is determined to be true, and optionally, other statements to be executed if the condition is determined to be false.

Following is the general from of a typical decision making structure found in most of the programming languages –



*The number 0, the strings '0' and "", the empty list , and undef are all **false** in a boolean context and all other values are **true**. Negation of a true value by **!** or **not** returns a special false value.*

Perl programming language provides the following types of conditional statements.

Statement	Description
<a href="#">if statement</a>	An <b>if statement</b> consists of a boolean expression followed by one or more statements.
<a href="#">if...else statement</a>	An <b>if statement</b> can be followed by an optional <b>else statement</b> .
<a href="#">if...elsif...else statement</a>	An <b>if statement</b> can be followed by an optional <b>elsif statement</b> and then by an optional <b>else statement</b> .
<a href="#">unless statement</a>	An <b>unless statement</b> consists of a boolean expression followed by one or more statements.
<a href="#">unless...else statement</a>	An <b>unless statement</b> can be followed by an optional <b>else statement</b> .

[unless...elsif..else statement](#)

An **unless statement** can be followed by an optional **elsif statement** and then by an optional **else statement**.

[switch statement](#)

With the latest versions of Perl, you can make use of the **switch** statement. which allows a simple way of comparing a variable value against various conditions.

## The ? : Operator

Let's check the **conditional operator ? :** which can be used to replace **if...else** statements. It has the following general form –

```
Exp1 ? Exp2 : Exp3;
```

Where Exp1, Exp2, and Exp3 are expressions. Notice the use and placement of the colon.

The value of a ? expression is determined like this: Exp1 is evaluated. If it is true, then Exp2 is evaluated and becomes the value of the entire ? expression. If Exp1 is false, then Exp3 is evaluated and its value becomes the value of the expression. Below is a simple example making use of this operator –

```
#!/usr/local/bin/perl

$name = "Ali";
$age = 10;

$status = ($age > 60 )? "A senior citizen" : "Not a senior citizen";

print "$name is - $status\n";
```

This will produce the following result –

```
Ali is - Not a senior citizen
Loading [MathJax]/jax/output/HTML-CSS/jax.js
```