

# PL/SQL - IF-THEN-ELSIF STATEMENT

[http://www.tutorialspoint.com/plsql/plsql\\_if\\_then\\_elsif.htm](http://www.tutorialspoint.com/plsql/plsql_if_then_elsif.htm)

Copyright © tutorialspoint.com

The **IF-THEN-ELSIF** statement allows you to choose between several alternatives. An **IF-THEN** statement can be followed by an optional **ELSIF...ELSE** statement. The **ELSIF** clause lets you add additional conditions.

When using **IF-THEN-ELSIF** statements there are few points to keep in mind.

- Its ELSIF not ELSEIF
- An IF-THEN statement can have zero or one ELSE's and it must come after any ELSEIF's.
- An IF-THEN statement can have zero to many ELSIF's and they must come before the ELSE.
- Once an ELSIF succeeds, none of the remaining ELSIF's or ELSE's will be tested.

## Syntax:

The syntax of an IF-THEN-ELSIF Statement in PL/SQL programming language is:

```
IF(boolean_expression 1) THEN
    S1; -- Executes when the boolean expression 1 is true
ELSIF( boolean_expression 2) THEN
    S2; -- Executes when the boolean expression 2 is true
ELSIF( boolean_expression 3) THEN
    S3; -- Executes when the boolean expression 3 is true
ELSE
    S4; -- executes when the none of the above condition is true
END IF;
```

## Example:

```
DECLARE
    a number(3) := 100;
BEGIN
    IF ( a = 10 ) THEN
        dbms_output.put_line('Value of a is 10' );
    ELSIF ( a = 20 ) THEN
        dbms_output.put_line('Value of a is 20' );
    ELSIF ( a = 30 ) THEN
        dbms_output.put_line('Value of a is 30' );
    ELSE
        dbms_output.put_line('None of the values is matching');
    END IF;
    dbms_output.put_line('Exact value of a is: ' || a );
END;
/
```

When the above code is executed at SQL prompt, it produces the following result:

```
None of the values is matching
Exact value of a is: 100

PL/SQL procedure successfully completed.
```