

## PL/SQL プログラミング命令 プログラミング・ステートメント

### CASE 文

```
a NUMBER ;
b NUMBER ;
a := 2 ;
b := 1 ;
CASE a
  WHEN 1 THEN ;
  WHEN 2 THEN ;
  ELSE ;
END CASE ;
```

-- WHEN 句の中では、a の値を評価対象に固定する  
-- 比較条件は、a = 1  
-- 比較条件は、a = 2  
-- 条件に一致しなかった時の処理

もしくは、

```
CASE a
  WHEN a = 1 THEN ;
  WHEN b = 2 THEN ;
  ELSE ;
END CASE ;
```

-- WHEN 句の中での評価対象を固定しない  
-- 比較条件は、a = 1  
-- 比較条件は、b = 2  
-- 条件に一致しなかった時の処理

### LOOP 文

```
LOOP
  処理 ;
  EXIT WHEN 条件 ;
END LOOP ;
```

-- LOOP を抜ける条件記述

もしくは、

```
LOOP
  処理 ;
  IF 条件 THEN
    EXIT ;
  END IF ;
END LOOP ;
```

-- 条件式の記述  
-- EXIT が実行されると LOOP から抜ける

もしくは、

```
WHILE 条件 LOOP
  処理 ;
END LOOP ;
```

-- 条件が成立 (true) の間、LOOP 処理が繰り返される

### FOR 文

```
FOR カウンタ変数 IN [REVERSE] 最小値 .. 最大値 LOOP
  /* カウンタ変数は、DECLARE で変数定義しない (FOR 関数による自動定義) */
  処理 ;
END LOOP ;
```

## ラベル文

```
<<OUT_LOOP>>
DECLARE
  a NUMBER ;
BEGIN
  LOOP                                -- LOOP_A のループ
  <<LOOP_A>>                            -- ループ抽出し用ラベルは、ループ内に記述
    . . . . .
    IF 条件1 THEN
      . . . . .
      EXIT LOOP ;                      -- LOOP_A からの抽出し
    END IF ;
END LOOP ;                             -- LOOP_A 用の END LOOP
END;
```

```
DECLARE
  a NUMBER ;
BEGIN
  LOOP                                -- LOOP_B のループ
  <<LOOP_B>>
    IF 条件2 THEN
      . . . . .
      EXIT LOOP ;                  -- LOOP_B からの抜け出し
    ELSEIF 条件3 THEN
      EXIT LOOP_A ;                 -- LOOP_A と B の両方の抜け
    ELSE
      . . . . .
    END IF ;
    LOOP_OUT.a = 99 ;               -- ループの外で定義した変数 a の使用
  END LOOP ;                          -- LOOP_B 用の END LOOP
END ;
```

※ ラベルは、FOR 変数 IN 初期値..終了値 LOOP 文の中にも記述できる