



# CONTROLLING FLOW OF EXECUTION

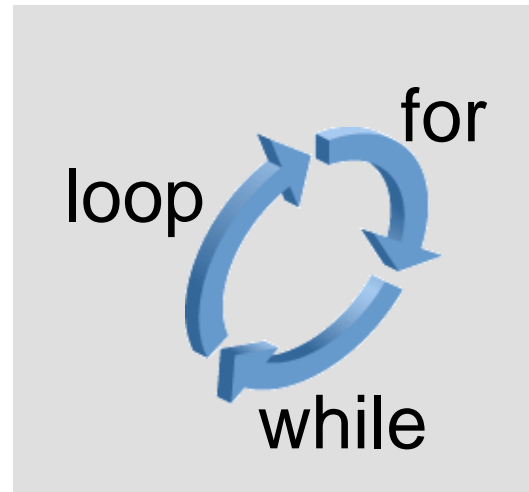
```
IF...  
THEN...  
END IF;
```

```
IF...  
THEN...  
ELSE...  
END IF;
```

```
IF...  
THEN...  
ELSIF...  
THEN...  
END IF;
```

```
IF...  
THEN...  
ELSIF...  
THEN...  
ELSE...  
END IF;
```

```
CASE  
WHEN... THEN..  
WHEN... THEN..  
WHEN... THEN..  
ELSE  
END CASE;
```



# IF STATEMENT

Syntax:

```
IF condition THEN
    statements;
[ELSIF condition THEN
    statements;]
[ELSE
    statements;]
END IF;
```

## SIMPLE IF STATEMENT

```
DECLARE
  v_myage  number:=31;
BEGIN
  IF v_myage < 11
  THEN
    DBMS_OUTPUT.PUT_LINE(' I am a child ');
  END IF;
END;
/
```

Output:

```
anonymous block completed
```

## IF THEN ELSE STATEMENT

```
DECLARE
v_myage  number:=31;
BEGIN
IF v_myage < 11
  THEN
    DBMS_OUTPUT.PUT_LINE(' I am a child ');
  ELSE
    DBMS_OUTPUT.PUT_LINE(' I am not a child ');
END IF;
END;
/
```

Output:

```
anonymous block completed
  I am not a child
```

## IF ELSIF ELSE CLAUSE

```
DECLARE
  v_myage number:=31;
BEGIN
  IF v_myage < 11 THEN
    DBMS_OUTPUT.PUT_LINE(' I am a child ');
  ELSIF v_myage < 20 THEN
    DBMS_OUTPUT.PUT_LINE(' I am young ');
  ELSIF v_myage < 30 THEN
    DBMS_OUTPUT.PUT_LINE(' I am in my twenties');
  ELSIF v_myage < 40 THEN
    DBMS_OUTPUT.PUT_LINE(' I am in my thirties');
  ELSE
    DBMS_OUTPUT.PUT_LINE(' I am always young ');
  END IF;
END;
/
```

```
anonymous block completed
  I am in my thirties
```

## NULL VALUE IN IF STATEMENT

```
DECLARE
  v_myage  number;
BEGIN
  IF v_myage < 11 THEN
    DBMS_OUTPUT.PUT_LINE(' I am a child ');
  ELSE
    DBMS_OUTPUT.PUT_LINE(' I am not a child ');
  END IF;
END;
/
```

Output:

```
anonymous block completed
  I am not a child
```

# CASE EXPRESSIONS

คำสั่งในการกำหนดทางเลือก CASE expression

```
CASE selector
  WHEN expression1 THEN result1
  WHEN expression2 THEN result2
  ...
  WHEN expressionN THEN resultN
  [ELSE resultN+1]
END;
/
```







# CASE STATEMENT

```
DECLARE
  v_deptid NUMBER;
  v_deptname VARCHAR2(20);
  v_emps NUMBER;
  v_mngid NUMBER:= 108;
BEGIN
  CASE v_mngid
  WHEN 108 THEN
    SELECT department_id, department_name
       INTO v_deptid, v_deptname FROM departments
    WHERE manager_id=108;
    SELECT count(*) INTO v_emps FROM employees
    WHERE department_id=v_deptid;
  WHEN 200 THEN
    ...
  END CASE;
  DBMS_OUTPUT.PUT_LINE ('You are working in the ' || deptname ||
' department. There are ' || v_emps || ' employees in this
department');
END;
/
```

# HANDLING NULLS

การจัดการกับค่าว่างในโปรแกรม เพื่อจัดการข้อผิดพลาด ซึ่งควรคำนึงถึงตามกฎหมายดังนี้

- เปรียบเทียบตัวแปรกับค่าว่างเสมอ

```
x := 5;
```

```
y := NULL;
```

```
IF x != y THEN -- yields NULL, not TRUE
```

```
a := NULL;
```

```
b := NULL;
```

```
IF a = b THEN -- yields NULL, not TRUE
```

- ประยุกต์คำสั่ง NOT กับค่าว่าง

```
a := NULL
```

```
IF a IS NOT NULL THEN
```

- ถ้าเงื่อนไขที่เกิดค่าว่าง ในคำสั่งนั้น ๆ ชุดคำสั่งต่อจากนั้นจะไม่ถูกประเมินผล

# ITERATIVE CONTROL: LOOP STATEMENTS

- ชุดคำสั่งทำซ้ำ
- มี 3 ประเภทของการทำซ้ำ
  - Basic loop
  - FOR loop
  - WHILE loop



# BASIC LOOPS

Syntax:

```
LOOP  
  statement1;  
  . . .  
  EXIT [WHEN condition];  
END LOOP;
```

# BASIC LOOPS

Example:

```
DECLARE
  v_countryid      locations.country_id%TYPE := 'CA';
  v_loc_id         locations.location_id%TYPE;
  v_counter        NUMBER(2) := 1;
  v_new_city       locations.city%TYPE := 'Montreal';
BEGIN
  SELECT MAX(location_id) INTO v_loc_id FROM locations
  WHERE country_id = v_countryid;
  LOOP
    INSERT INTO locations(location_id, city, country_id)
    VALUES((v_loc_id + v_counter), v_new_city, v_countryid);
    v_counter := v_counter + 1;
    EXIT WHEN v_counter > 3;
  END LOOP;
END;
/
```

# WHILE LOOPS

Syntax:

```
WHILE condition LOOP
  statement1;
  statement2;
  . . .
END LOOP;
```

ทำซ้ำในคำสั่ง WHILE เมื่อเงื่อนไขเป็นจริง



# WHILE LOOPS

## Example

```
DECLARE
  v_countryid    locations.country_id%TYPE := 'CA';
  v_loc_id       locations.location_id%TYPE;
  v_new_city     locations.city%TYPE := 'Montreal';
  v_counter      NUMBER := 1;
BEGIN
  SELECT MAX(location_id) INTO v_loc_id FROM locations
  WHERE country_id = v_countryid;
  WHILE v_counter <= 3 LOOP
    INSERT INTO locations(location_id, city, country_id)
    VALUES((v_loc_id + v_counter), v_new_city, v_countryid);
    v_counter := v_counter + 1;
  END LOOP;
END;
/
```

# FOR LOOPS

- ใช้คำสั่ง FOR เพื่อทำการทำซ้ำที่รู้จำนวนรอบ
- ไม่จำเป็นต้องประกาศตัวแปรตัวนับ (counter)

```
FOR counter IN [REVERSE]
    lower_bound..upper_bound LOOP
    statement1;
    statement2;
    . . .
END LOOP;
```

# FOR LOOPS

Example:

```
DECLARE
  v_countryid  locations.country_id%TYPE := 'CA';
  v_loc_id     locations.location_id%TYPE;
  v_new_city   locations.city%TYPE := 'Montreal';
BEGIN
  SELECT MAX(location_id) INTO v_loc_id
    FROM locations
   WHERE country_id = v_countryid;
  FOR i IN 1..3 LOOP
    INSERT INTO locations(location_id, city, country_id)
      VALUES((v_loc_id + i), v_new_city, v_countryid );
  END LOOP;
END;
/
```

# FOR LOOPS

## ตัวชี้นำ

- อ้างอิงหรือเรียกใช้ตัวนับ (counter) เฉพาะภายในลูป ไม่สามารถอ้างอิงนอกลูปได้
- ไม่อ้างอิงตัวนับ ด้วยการกำหนดค่าให้
- ค่าขอบเขตของตัวนับจะต้องไม่เป็นค่าว่าง

```
DECLARE
```

```
    v_lower NUMBER := 1; //สามารถใช้ v_lower varchar2(1) := '1';
```

```
    v_upper NUMBER := 3;
```

```
BEGIN
```

```
    FOR i IN v_lower..v_upper LOOP
```

```
        ...
```

```
    END LOOP;
```

```
END;
```