

LOOP

基本的循環結構封裝在LOOP和END LOOP語句之間語句序列。

隨著每次迭代，語句順序被執行，然後在循環的頂部控制過程。

=====

範例1.：

IF-THEN

```
DECLARE
  x number := 10;
BEGIN
  LOOP
    dbms_output.put_line(x);
    x := x + 10;
    IF x > 50 THEN
      exit;
    END IF;
  END LOOP;
  -- after exit, control resumes here
  dbms_output.put_line('After Exit x is: ' || x);
END;
/
```

db<>fiddle

By using db<>fiddle, you agree to license everything you submit by [Creative Commons CC0](https://creativecommons.org/licenses/by/4.0/).

<pre>1 DECLARE 2 x number := 10; 3 BEGIN 4 LOOP 5 dbms_output.put_line(x); 6 x := x + 10; 7 IF x > 50 THEN 8 exit; 9 END IF; 10 END LOOP; 11 -- after exit, control resumes here 12 dbms_output.put_line('After Exit x is: ' x); 13 END; 14 /</pre>	<pre>1 rows affected dbms_output: 10 20 30 40 50 After Exit x is: 60</pre>
--	---

範例2.：

WHEN

```
DECLARE
  x number := 10;
BEGIN
  LOOP
    dbms_output.put_line(x);
    x := x + 10;
    exit WHEN x > 50;
  END LOOP;
  -- after exit, control resumes here
  dbms_output.put_line('After Exit x is: ' || x);
END;
/
```

db<>fiddle

By using db<>fiddle, you agree to license everything you submit by [Creative Commons CC0](https://creativecommons.org/licenses/by/4.0/).

<pre>1 DECLARE 2 x number := 10; 3 BEGIN 4 LOOP 5 dbms_output.put_line(x); 6 x := x + 10; 7 exit WHEN x > 50; 8 END LOOP; 9 -- after exit, control resumes here 10 dbms_output.put_line('After Exit x is: ' x); 11 END; 12 /</pre>	<pre>1 rows affected dbms_output: 10 20 30 40 50 After Exit x is: 60</pre>
---	--