1) IF ---THEN

Syntax of IF --THEN

```
IF (condition )  THEN
     executable statement;

END IF;
```

Q. Write a Program To check number is positive
(Note: If number is greater than 0 then it is called positive
 number)

```
Declare
  n1 number:=10;
Begin

   IF (n1>0) THEN
        dbms_output.put_line(n1 || 'is positive number');
   END IF;
End;
```

CASE 2: IF --- THEN ELSE

```
IF (condition) THEN
    executable statements;
ELSE
    executable statements;
END IF;
```

Q Write a program to check number is positive or negative

```
Declare
  n1 number:=10;

Begin

   IF (n1>0) THEN
       DBMS_OUTPUT.PUT_LINE('Number is positive');
   ELSE
       DBMS_OUTPUT.PUT_LINE('Number is negative');
   END IF;

END;
```

Q. Write a program to check Candidate can vote or not

```
Declare
  age number:=17;
Begin
   IF (age>=18) THEN
        DBMS_OUTPUT.PUT_LINE('Can vote');
   ELSE
        DBMS_OUTPUT.PUT_LINE('cannot vote');
   END IF;
END;
```

Q. Write a program to check number is even or odd
Note: If number is divisible by 2 then it is called even number else it is called odd number.

```
Declar
   n1 number:=10;
   result number;
Begin
    select mod(n1,2) into result from Dual;

    IF( result =0) THEN
      dbms_output.put_line('Number is even');
    ELSE
      dbms_output.put_line('Number is Odd');

   END IF;
   END;
```

case 4

```
Case 3: IF ---THEN Elseif

IF condition THEN
    executable statement;

ELSE IF condition THEN
  executable statement;
END IF;
```

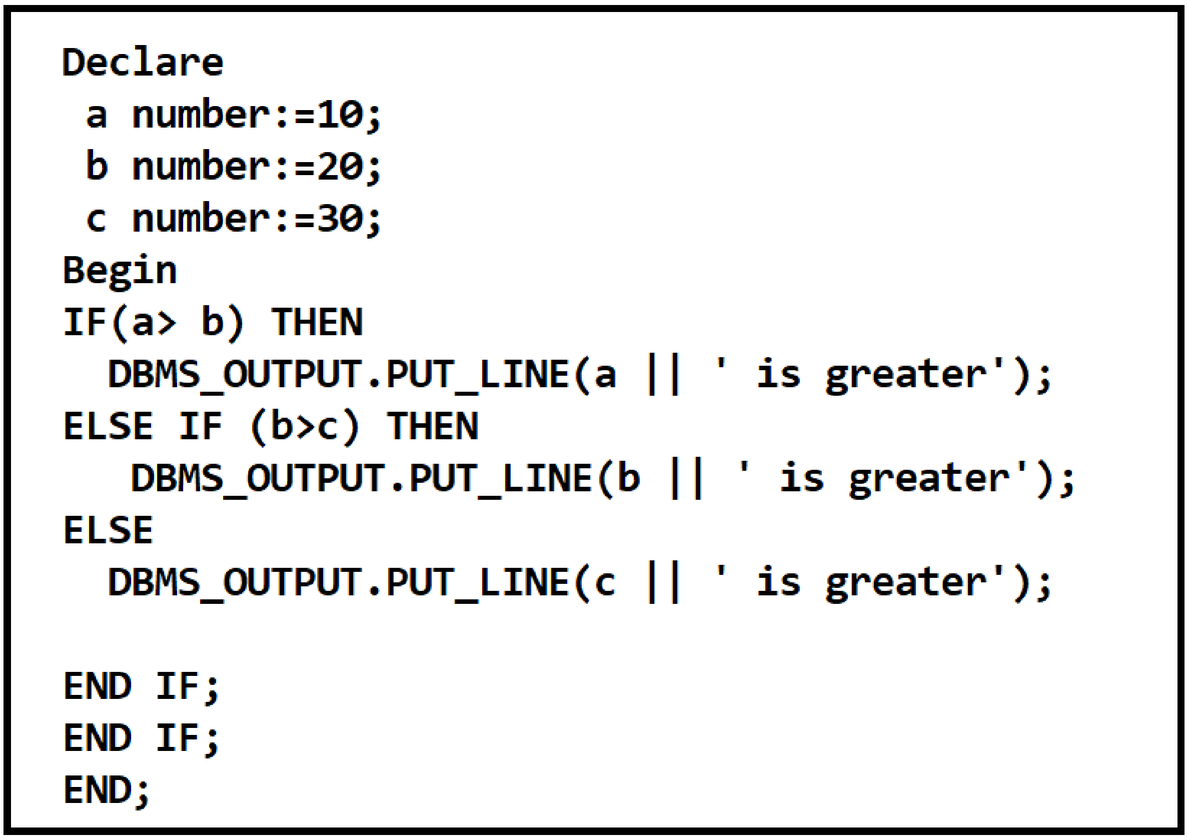
```
IF codition THEN

 executable statement;

ELSE IF CONDITION THEN
  executable statement;

ELSE
 executable statement;

 END IF;
END IF;
```

Note
1) IF --THEN came in paire

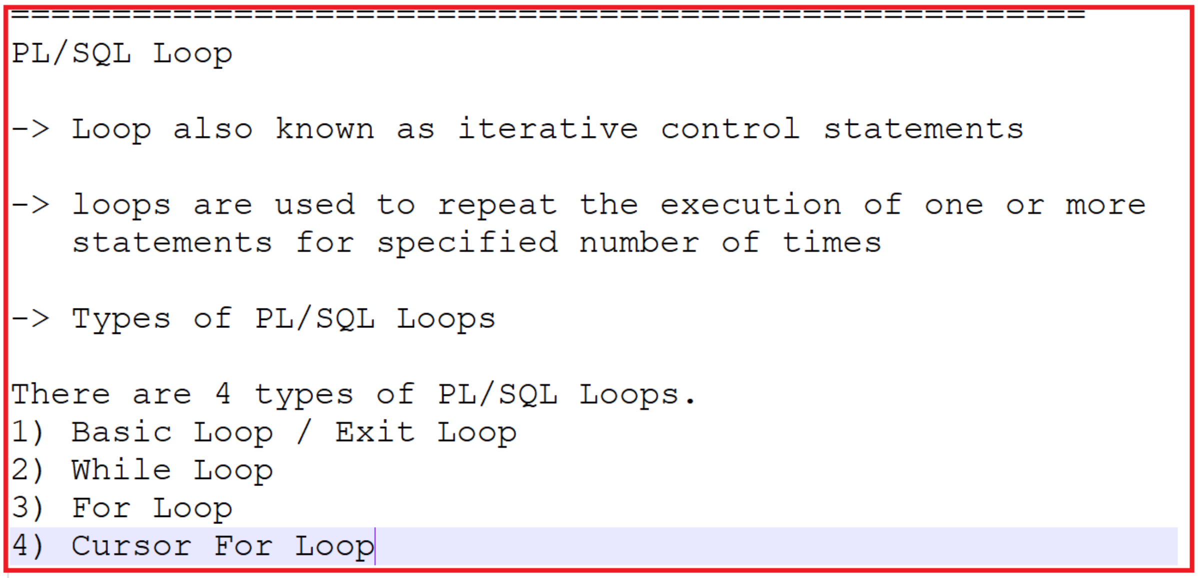
2) every if must be end with END IF;

3) In IF condition is required but for
   else condition is not required

4) Number of IF must be equal to number
 of END IF

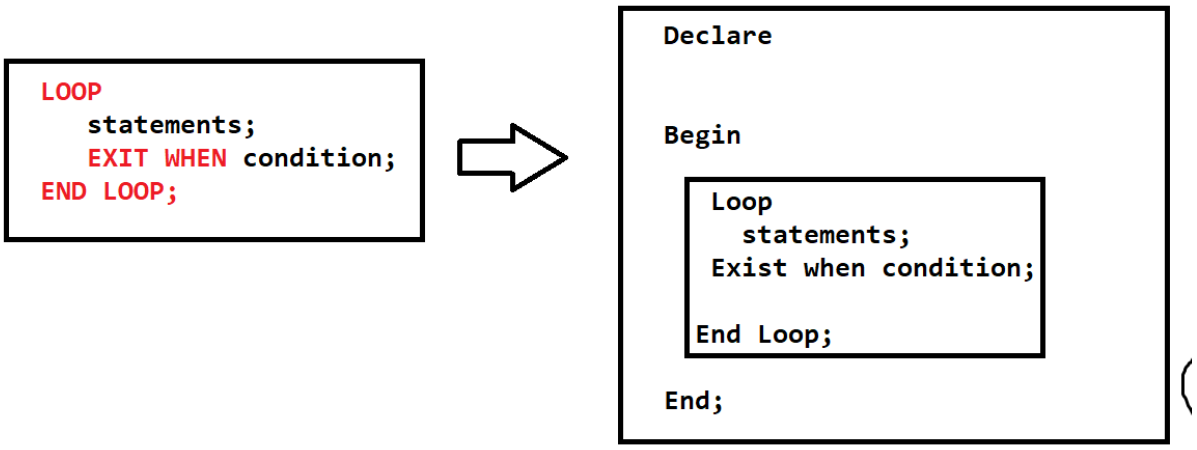


```
Declare
 a number:=10;
 b number:=20;
 c number:=30;
Begin
IF(a> b) THEN
  DBMS_OUTPUT.PUT_LINE(a || ' is greater');
ELSE IF (b>c) THEN
   DBMS_OUTPUT.PUT_LINE(b || ' is greater');
ELSE
  DBMS_OUTPUT.PUT_LINE(c || ' is greater');

END IF;
END IF;
END;
```



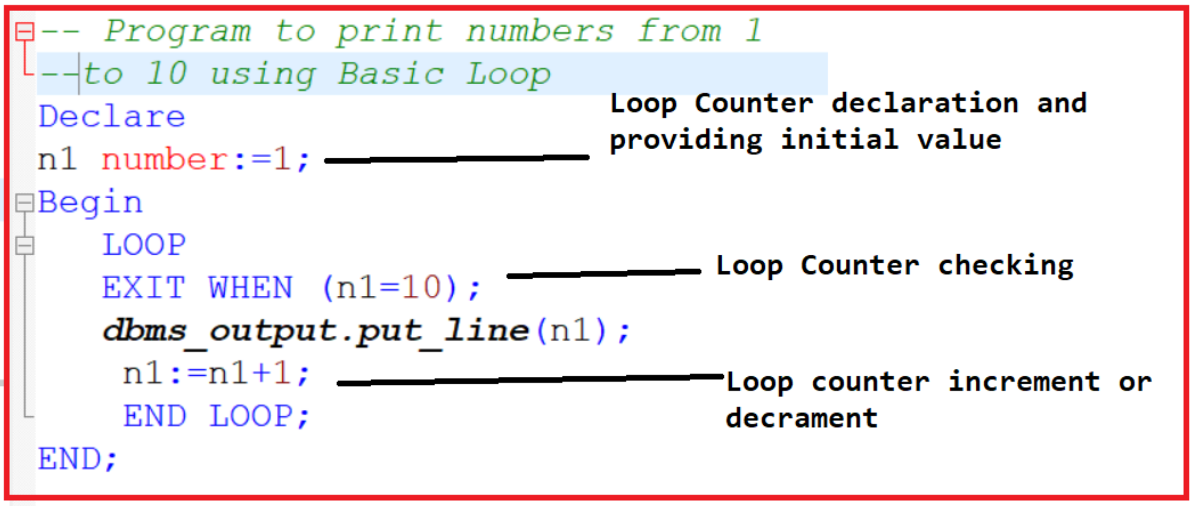
```
=====================================================
PL/SQL Loop

-> Loop also known as iterative control statements

-> loops are used to repeat the execution of one or more
   statements for specified number of times

-> Types of PL/SQL Loops

There are 4 types of PL/SQL Loops.
1) Basic Loop / Exit Loop
2) While Loop
3) For Loop
4) Cursor For Loop
```

Basic Loop



```
LOOP
    statements;
    EXIT WHEN condition;
END LOOP;
```

```
Declare


Begin

  Loop
    statements;
  Exist when condition;

 End Loop;

End;
```

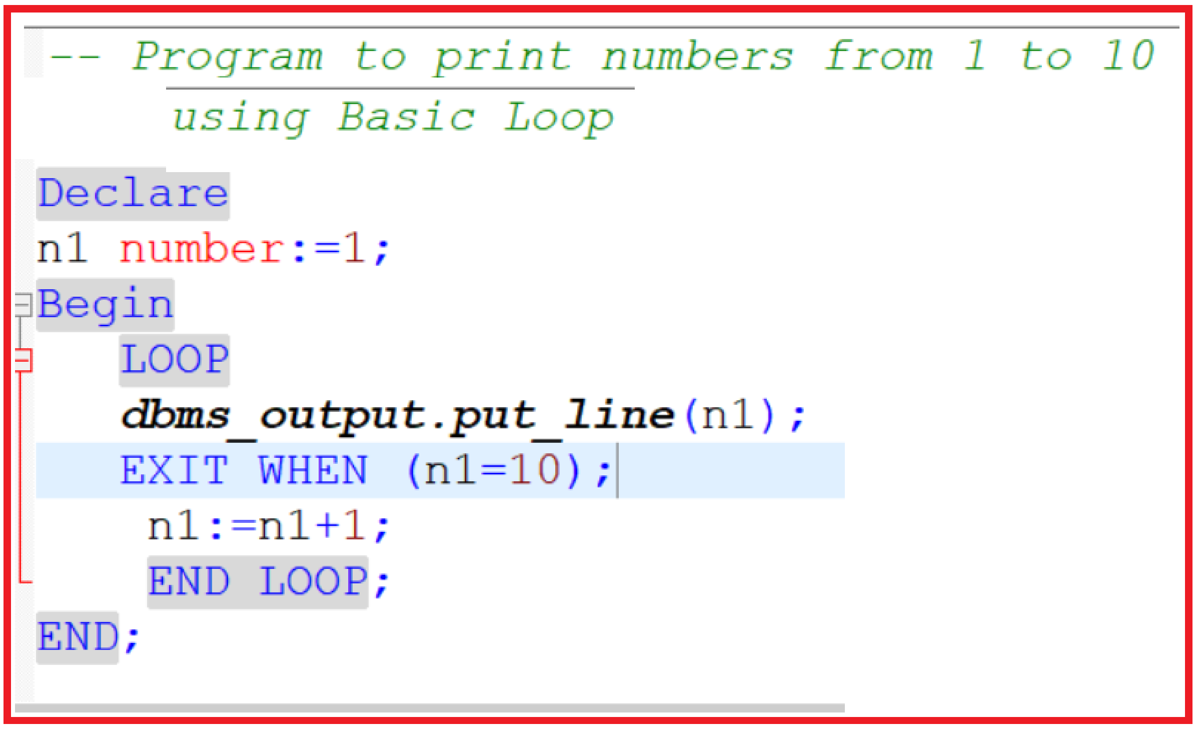


```
-- Program to print numbers from 1
--to 10 using Basic Loop
Declare
n1 number:=1;
Begin
    LOOP
    EXIT WHEN (n1=10);
    dbms_output.put_line(n1);
     n1:=n1+1;
     END LOOP;
END;
```

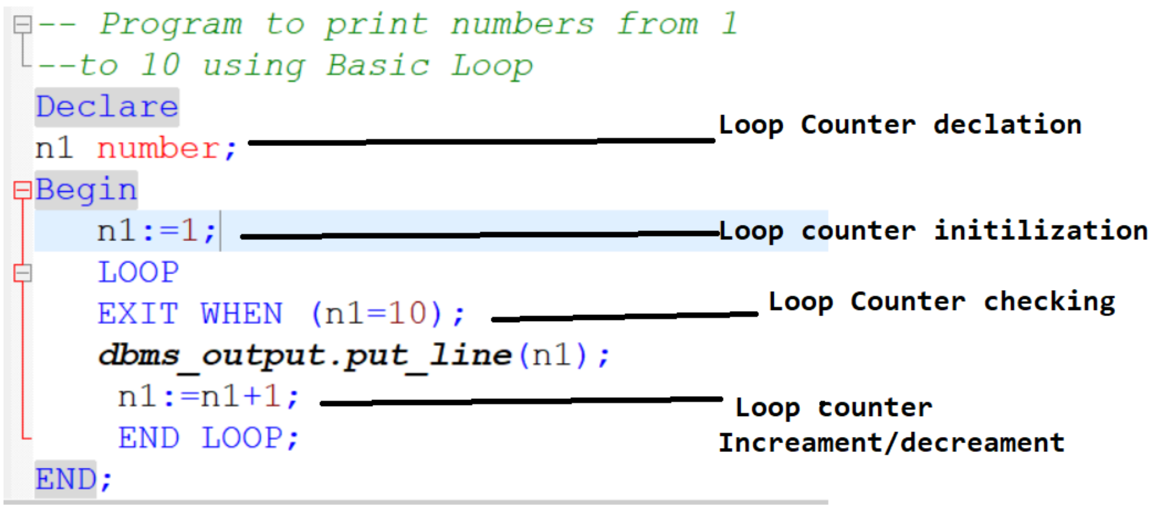
Loop Counter declaration and providing initial value

Loop Counter checking

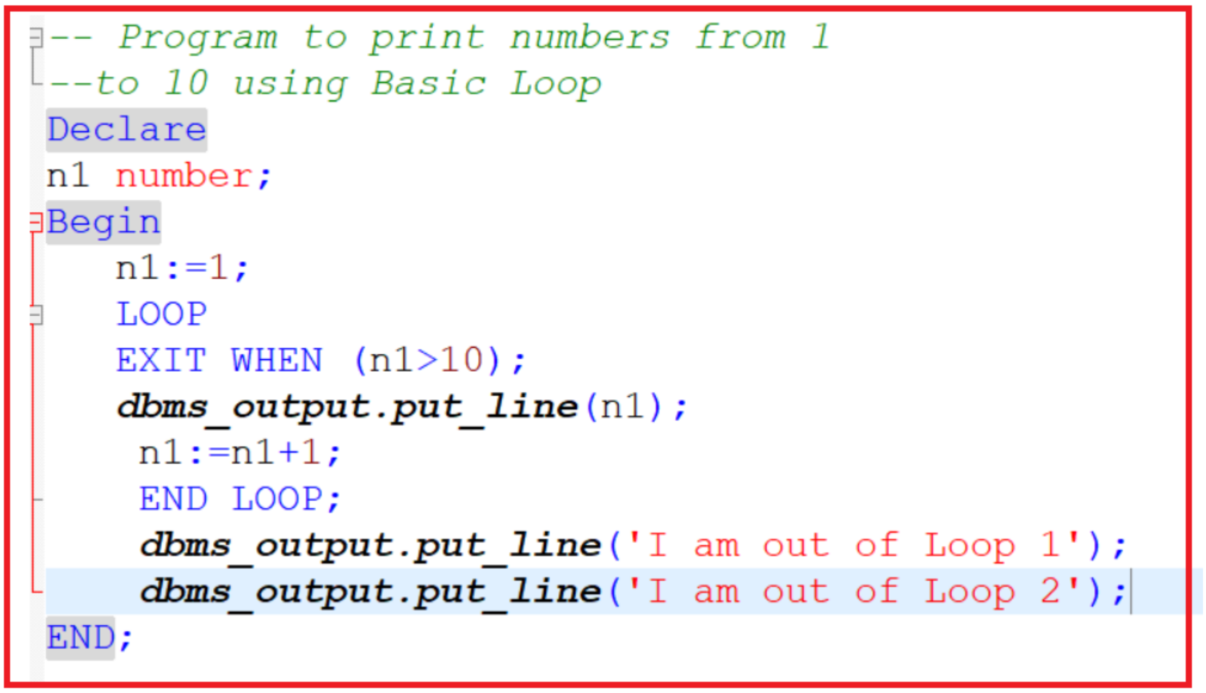
Loop counter increment or decrament



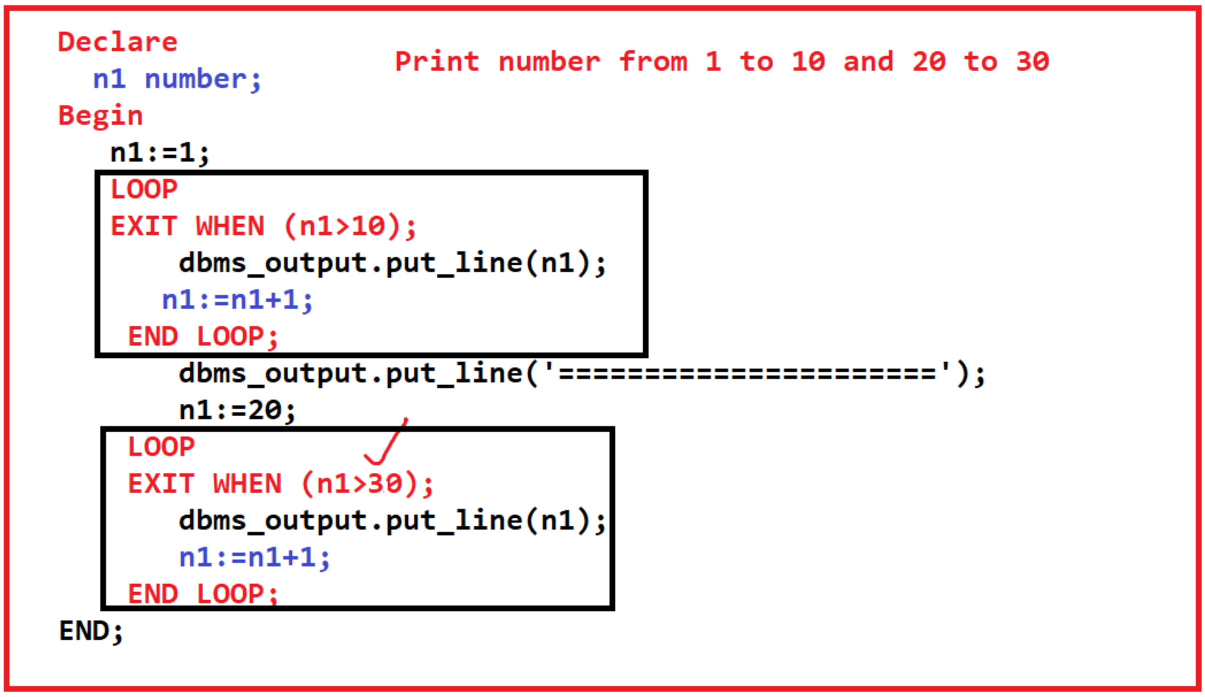
```
-- Program to print numbers from 1 to 10
     using Basic Loop
Declare
n1 number:=1;
Begin
    LOOP
    dbms_output.put_line(n1);
    EXIT WHEN (n1=10);
     n1:=n1+1;
     END LOOP;
END;
```



```
-- Program to print numbers from 1
--to 10 using Basic Loop
Declare
n1 number;
Begin
    n1:=1;
    LOOP
    EXIT WHEN (n1=10);
    dbms_output.put_line(n1);
     n1:=n1+1;
     END LOOP;
END;
```

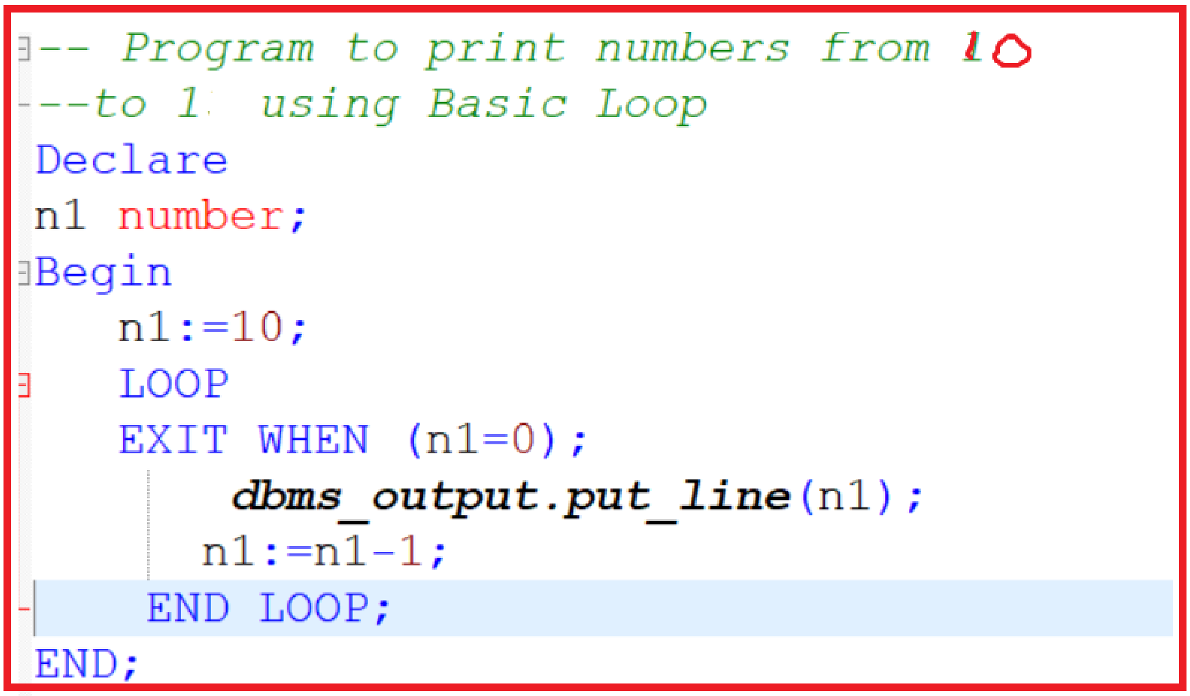


```
-- Program to print numbers from 1
--to 10 using Basic Loop
Declare
n1 number;
Begin
    n1:=1;
    LOOP
    EXIT WHEN (n1>10);
    dbms_output.put_line(n1);
     n1:=n1+1;
     END LOOP;
     dbms_output.put_line('I am out of Loop 1');
     dbms_output.put_line('I am out of Loop 2');
END;
```



Print number from 1 to 10 and 20 to 30

```
Declare
  n1 number;
Begin
   n1:=1;
   LOOP
   EXIT WHEN (n1>10);
       dbms_output.put_line(n1);
     n1:=n1+1;
    END LOOP;
       dbms_output.put_line('======================');
       n1:=20;
    LOOP
    EXIT WHEN (n1>30);
       dbms_output.put_line(n1);
       n1:=n1+1;
    END LOOP;
END;
```



```
-- Program to print numbers from 10
--to 1  using Basic Loop
Declare
n1 number;
Begin
    n1:=10;
    LOOP
    EXIT WHEN (n1=0);
        dbms_output.put_line(n1);
      n1:=n1-1;
     END LOOP;
END;
```

```
LOOP
   statements;
   EXIT WHEN condition;
END LOOP;
```

Basic Loop
Syntax

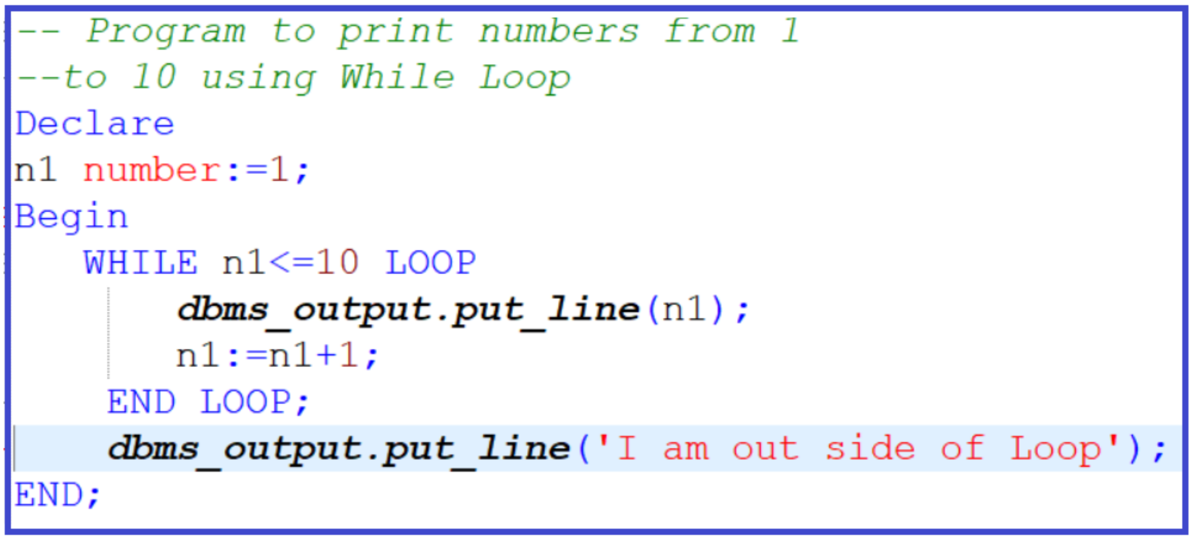
```
WHILE <condition> LOOP
   statements;
END LOOP;
```

While Loop
Syntax

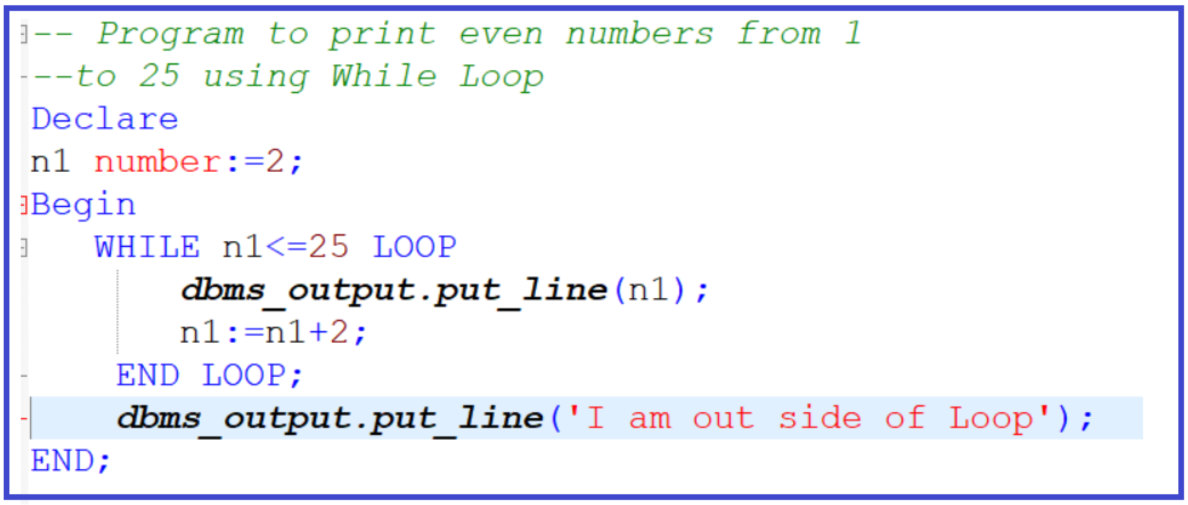
```
Declare
   n1 number:=1;

Begin

   While (n1 <=10) LOOP

     dbms_output.put_line(n1);
     n1:=n1+1;

   End Loop;

End;
```

If condition is
true then only
loop statement get
executed;

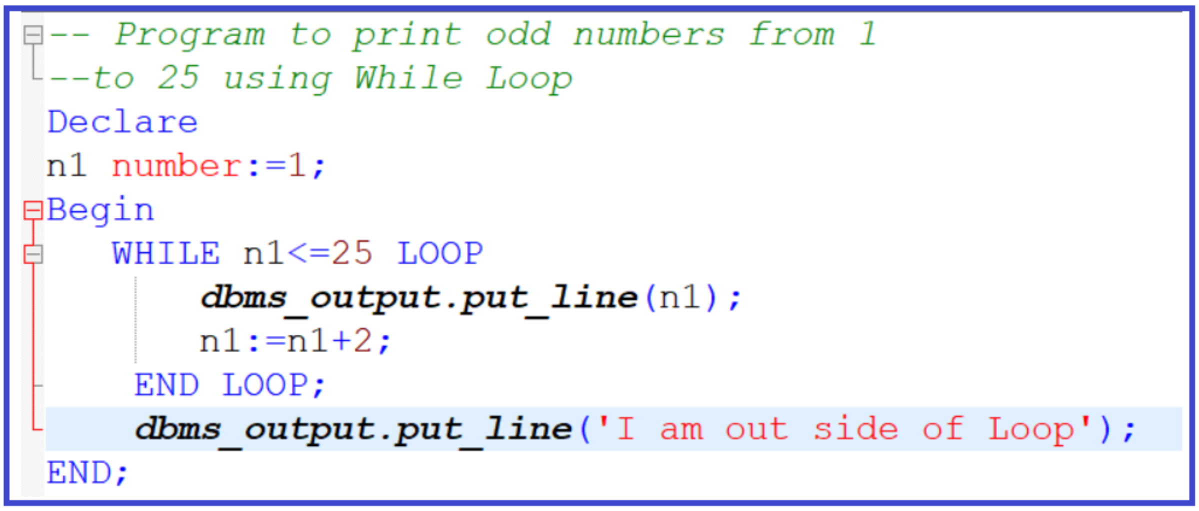
Program to Print number
from 1 to 10



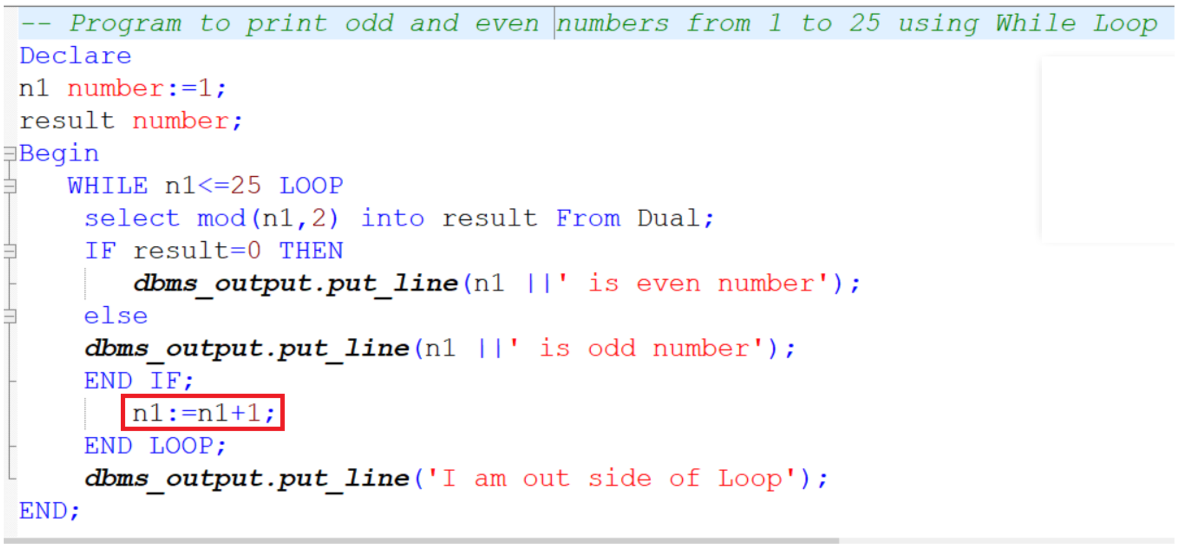
```
-- Program to print numbers from 1
--to 10 using While Loop
Declare
n1 number:=1;
Begin
   WHILE n1<=10 LOOP
        dbms_output.put_line(n1);
        n1:=n1+1;
    END LOOP;
    dbms_output.put_line('I am out side of Loop');
END;
```



```
-- Program to print even numbers from 1
--to 25 using While Loop
Declare
n1 number:=2;
Begin
   WHILE n1<=25 LOOP
        dbms_output.put_line(n1);
        n1:=n1+2;
    END LOOP;
    dbms_output.put_line('I am out side of Loop');
END;
```



```
-- Program to print odd numbers from 1
--to 25 using While Loop
Declare
n1 number:=1;
Begin
   WHILE n1<=25 LOOP
        dbms_output.put_line(n1);
        n1:=n1+2;
    END LOOP;
    dbms_output.put_line('I am out side of Loop');
END;
```



```
-- Program to print odd and even numbers from 1 to 25 using While Loop
Declare
n1 number:=1;
result number;
Begin
   WHILE n1<=25 LOOP
    select mod(n1,2) into result From Dual;
    IF result=0 THEN
       dbms_output.put_line(n1 ||' is even number');
    else
    dbms_output.put_line(n1 ||' is odd number');
    END IF;
       n1:=n1+1;
    END LOOP;
    dbms_output.put_line('I am out side of Loop');
END;
```