 



Database Programming with PL/SQL 2-1: Using Variables in PL/SQL Practice Activities

**Vocabulary**

Identify the vocabulary word for each definition below:

|  |  |
| --- | --- |
| Variabile | Used for storage of data and manipulation of stored values. |
| Parametri | Values passed to a program by a user or by another program to customize the program. |

Try It / Solve It

1. Fill in the blanks.
   1. Variables can be assigned to the output of a function.
   2. Variables can be assigned values in the declarative section of a PL/SQL block.
   3. Variables can be passed as parameters to subprograms.
2. Identify valid and invalid variable declaration and initialization:

number\_of\_copies PLS\_INTEGER;

printer\_name CONSTANT VARCHAR2(10); deliver\_to VARCHAR2(10) := Johnson;

by\_when DATE := SYSDATE+1;

1. Examine the following anonymous block and choose the appropriate statement.

DECLARE

fname VARCHAR2(25);

lname VARCHAR2(25) DEFAULT 'fernandez'; BEGIN

DBMS\_OUTPUT.PUT\_LINE(fname || ' ' || lname); END;

* 1. The block will execute successfully and print ‘ fernandez’. ->True
  2. The block will give an error because the fname variable is used without initializing.->False
  3. The block will execute successfully and print ‘null fernandez’.->False
  4. The block will give an error because you cannot use the DEFAULT keyword to initialize a variable of the VARCHAR2 type.->False
  5. The block will give an error because the FNAME variable is not declared.->False

1. In Application Express:
   1. Create the following function:

CREATE FUNCTION num\_characters (p\_string IN VARCHAR2) RETURN INTEGER AS

v\_num\_characters INTEGER; BEGIN

SELECT LENGTH(p\_string) INTO v\_num\_characters FROM dual;

RETURN v\_num\_characters; END;

* 1. Create and execute the following anonymous block:

DECLARE

v\_length\_of\_string INTEGER; BEGIN

v\_length\_of\_string := num\_characters('Oracle Corporation'); DBMS\_OUTPUT.PUT\_LINE(v\_length\_of\_string);

END;

1. Write an anonymous block that uses a country name as input and prints the highest and lowest elevations for that country. Use the COUNTRIES table. Execute your block three times using United States of America, French Republic, and Japan.

DECLARE

v\_country VARCHAR2(30) :='United States of America';

v\_high VARCHAR2(30);

v\_low VARCHAR2(30);

BEGIN

SELECT lowest\_elevation, highest\_elevation INTO v\_low, v\_high

FROM countries

WHERE country\_name = v\_country;

DBMS\_OUTPUT.PUT\_LINE(v\_high);

DBMS\_OUTPUT.PUT\_LINE(v\_low);

END;

DECLARE

v\_country VARCHAR2(30) :='French Republic';

v\_high VARCHAR2(30);

v\_low VARCHAR2(30);

BEGIN

SELECT lowest\_elevation, highest\_elevation INTO v\_low, v\_high

FROM countries

WHERE country\_name = v\_country;

DBMS\_OUTPUT.PUT\_LINE(v\_high);

DBMS\_OUTPUT.PUT\_LINE(v\_low);

END;

DECLARE

v\_country VARCHAR2(30) :='Japan';

v\_high VARCHAR2(30);

v\_low VARCHAR2(30);

BEGIN

SELECT lowest\_elevation, highest\_elevation INTO v\_low, v\_high

FROM countries

WHERE country\_name = v\_country;

DBMS\_OUTPUT.PUT\_LINE(v\_high);

DBMS\_OUTPUT.PUT\_LINE(v\_low);

END;

Copyright © 2020, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.