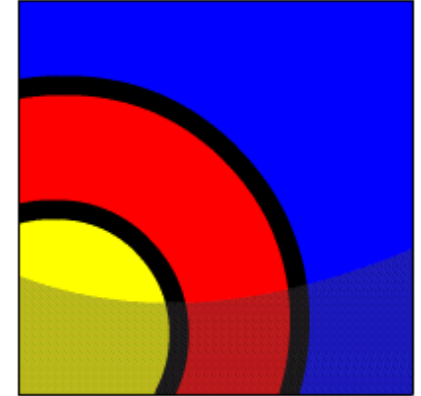# Good Programming Practices

# What Will I Learn?

In this lesson, you will learn to:

- List examples of good programming practices
- Accurately insert comments into PL/SQL code
- Create PL/SQL code that follows formatting guidelines to produce readable code

Why Learn It?

Good programming practices are techniques that you can follow to create the best code possible. Programming practices cover everything from making code more readable to creating code with faster performance.
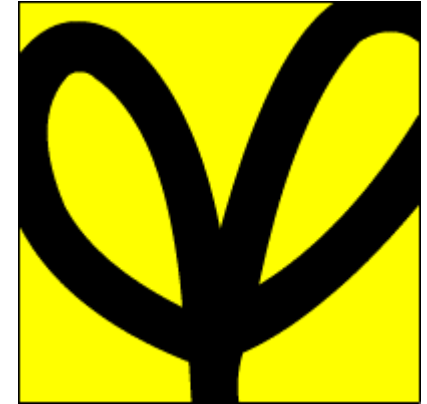
Software engineering teams often follow a style guide so that everyone on the team uses the same techniques. This makes it easier to read and modify code written by others.

Tell Me/Show Me

Programming Practices

You've already learned several good programming practices in this course:

- Conversions:
  - Do not rely on implicit data type conversions because they can be slower and the rules can change in later software releases.

- Declaring and initializing PL/SQL variables:
  - Use meaningful names
  - Declare one identifier per line for better readability and code maintenance.
  - Use the `NOT NULL` constraint when the variable must hold a value.
  - Avoid using column names as identifiers.
  - Use the `%TYPE` attribute to declare a variable according to another previously declared variable or database column.

Tell Me/Show Me

Programming Guidelines

Other programming guidelines include:

- Documenting code with comments
- Developing a case convention for the code
- Developing naming conventions for identifiers and other objects
- Enhancing readability by indenting
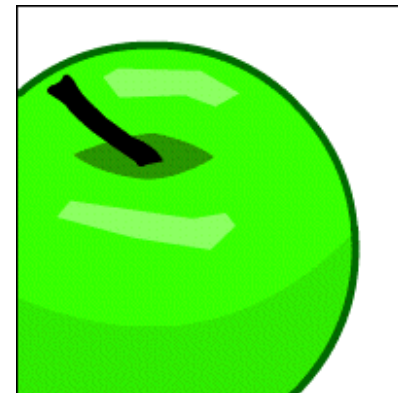
Tell Me/Show Me

Commenting Code

- Prefix single-line comments with two dashes (--).
- Place multiple-line comments between the symbols "/*" and "*/".

Example:

```
DECLARE
...
  v_annual_sal NUMBER (9,2);
BEGIN     -- Begin the executable section

/* Compute the annual salary based on the
   monthly salary input from the user */
  v_annual_sal := v_monthly_sal * 12;
END;      -- This is the end of the block
```

Tell Me/Show Me

Case Conventions

The following table provides guidelines for writing code in uppercase or lowercase to help you distinguish keywords from named objects.

Category	Case Convention	Examples
SQL keywords	Uppercase	`SELECT, INSERT`
PL/SQL keywords	Uppercase	`DECLARE, BEGIN, IF`
Data types	Uppercase	`VARCHAR2, BOOLEAN`
Identifiers and parameters	Lowercase	`v_sal, emp_cursor, g_sal, p_empno`
Database tables and columns	Lowercase	`employees, employee_id, department_id`

Tell Me/Show Me

Naming Conventions

The naming of identifiers should be clear, consistent, and unambiguous. One commonly-used convention is to name:

- Variables starting with `v_`
- Constants starting with `c_`
- Parameters (passed to procedures and functions) starting with `p_`

Examples: `v_date_of_birth`; `c_tax_rate`; `p_empno`;

Tell Me/Show Me

Indenting Code

For clarity, indent each level of code.

Examples:

```
BEGIN
   IF x=0 THEN
      y:=1;
   END IF;
END;
```

```
DECLARE
   v_deptno        NUMBER(4);
   v_location_id   NUMBER(4);
BEGIN
   SELECT  department_id,
           location_id
      INTO  v_deptno,
            v_location_id
      FROM  departments
      WHERE department_name = 'Sales';
...
END;
```
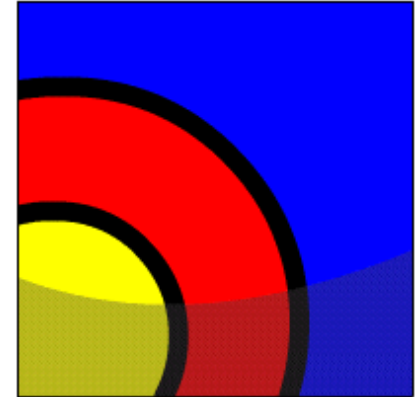
ORACLE Academy

◎ **Summary**

In this lesson, you have learned how to:

- List examples of good programming practices
- Accurately insert comments into PL/SQL code
- Create PL/SQL code that follows formatting guidelines to produce readable code

Try It/Solve It

The exercises for this lesson cover the following topics:

- Listing examples of good programming practices

- Inserting comments into PL/SQL code

- Creating PL/SQL code that follows formatting guidelines to produce readable code