Hi, welcome to Basic Program Structure.

☼Part C2: Boolean Data Type, Relational Operators, and Selection Basics in PYTHON.

☼At the end of this lesson, you should be able to explain all the following terminologies or syntax in Python. There is nothing special in Python for relational operators, so we will not explain them in details.

☼In this lesson, We will go through

Boolean

Statement

Comment

Interpreter error

User input

Print result

Whitespace

Indentation

one by one.

☼ Recall that Boolean data type has two values, either True or False. However, it is important to note that. Python is case-sensitive. The value true is defined as True, with capital T. The same rule applies for the value false. You will get an error if you fail to follow the rule.

☼A quick check. What do you think is the output of the following Python code?

☼The answer is B. The value 19 is assigned to a variable tom. “Greater than” is the relational operator to compare the value of tom, which is 19, and 18. 19 is greater than 18, so the result of the Boolean expression is True. There are two “trues” in the choices, A and B. In Python, the True value is capital T, so choice B is the correct answer.

☼Since this is the second lesson of Python programming, it is a good time to introduce some basic terms in Python. For example, Statement. Python differentiates its code into one of two categories: expressions and statements. Each line of code in a Python program is called a statement. Recall what we have learnt in 3B1. The concept of an expression is consistent with the mathematical definition, a combination of values and operations that creates a new value that we call a return value. A statement does not return a value but performs some tasks. These two statements conduct some calculations and this statement generates the output. Python interprets and runs the statements one by one.

☼For a long statement, the symbol "\" is used to continue a statement with the next line so that the two lines can be joined as one statement. This improves readability in the text editor.

☼Intelligent Guess. What do you think is the output of the following Python code?

☼The answer is A. The area. Why was the statement "print(" 18")" ignored?

☼Because it is a comment line. The pound sign "#" in Python indicates a comment. Anything after pound sign "#" is ignored during interpretation.

Comments provide information to improve code readability. We will say many times, in different ways, that a program is more than "just some code that does something." A program is a document that describes the thought process of its writer. Good programs can be read, just like any other essay. Comments contribute nothing to the running of the program because Python ignores them. However, comments are critical to the readability of the program.

☼There are no universally agreed rules for the right style and number of comments. Here are some useful guidelines. Good comments do not repeat the code but explain what you're trying to do at a higher level of abstraction than the code.

☼As we know, Python interprets and runs the statements one by one. The interpreter translates the Python code into machine language. The first stage of the process is to determine whether the syntax of the code is valid or not.

If the code is somehow invalid or malformed, Python cannot run the code and an interpreter error is received. Syntax error is the most common type of interpreter error. Here is a sample Python program that illustrates syntax errors that we are familiar with.

☼Besides syntax error, there are other error types. Intelligent Guess again. What do you think is the output of the following Python code if the user inputs "25"?

☼There is an error. Why?

☼To know why, let's learn about the details of a built-in function. Input. An input is a built-in function provided by Python to get an input It prints the message string on the screen and waits until the user types anything and presses "Enter".

No matter what is given, even a number, the input function returns a string.

For this program, we want to work with numbers, not characters, so we must convert the user's response from a string of characters to numbers. The float function takes the value associated with the variable radiusString and returns the float value of radiusString. In other words, it converts the string to a float.

☼(click the arrow to go back to 16)

Now, we know the reason for the error clearly. The variable "date" is holding a string, which cannot perform numerical operation.

☼ (click the "Answer" arrow button to jump to 18)

Quick Check. What do you think is the output of the following modified Python code if the user inputs "25"?

☼The answer is 27. The "int" function converts those characters to the integer 25, plus 2 , that is 27, the new value is assigned to the variable "thedayaftertomorrow" and display the result. That is 27.

☼ Like the input statement, Print is another built-in function, provided by Python, that displays related message and data on the shell screen.

We have illustrated basic usages of Python print statement to print results since our first program. The print statement can print strings bracketed by quotes and a value associated with a variable. Now, let's explore more.

The function "**print()**" makes an empty line.

If you have more than one object to be printed, each object that is to be printed is separated from other objects by commas.

In this example, the print statement outputs a string, a variable value, another string, and finally a variable value. The backslash character indicates that the statement continues onto the next line--in this case, the two-line print statement behaves as if it were a one long line. Stretching a statement across two lines can enhance readability-particularly on a narrow screen or page.

☼Up to this point, all we have been able to do is to write Python code that is executed sequentially; that is, one statement after another. We have not yet shown a way to incorporate decisions into our programs. So, let's get into that now.

The simplest decision presents a choice of doing one thing or not. Let's recall the structure of if statement and

☼look into the syntax of if statement in Python. It starts with the keyword if, followed by a Boolean expression, A colon **must** be used to mark the start of a block.

one or more indented statements form the entire true block. An indentation must be used for the entire true block.

Let's look at some examples,

In the first example, a single statement in the true block

In the second example, two statements with the same indentation belong to the same true block and will be executed as a group.

Indentation is a Python method for associating or grouping statements. Details will be given later.

☼Before we talk about the unique indentation usage in Python, let's look at whitespace. Python counts the "Tab", "Spacebar" and "Enter" as whitespaces. Normally, the purpose of a whitespace is to separate words in a statement. For the most part, you can place whitespaces anywhere in your program to make the code more readable.

☼Then, how about this case? Is the following Python code correct?

☼No. Error message: unexpected indent. What is the meaning?

☼Leading whitespace at the beginning of a line-defines indentation. Indentation plays a special role in Python.

Python requires it for grouping when a set of statements needs to be grouped together. Python does so by consistent indentation. That is, to use the same number of spaces for indentation **consistently**.

A suite of statements is a set of statements that has the same indentation, and these statements are executed as a group. This group is called a compound statement, which is considered to be one logical statement.

In short, Indentation has a dual purpose in Python, besides making the code more readable, more importantly, for grouping, particularly for control flow such as branching and looping.

☼Based on the numbers of spaces for indentation, we can see that for these three statements, all are One level of indentation. For this statement, it has two levels of indentation.

If N <= 1: NFact = 1 will be executed and the compound statement in the false block will be skipped.

The power of an IF statement is that, the program can **selectively run** the block based on the runtime condition.

Sometimes the program runs it, sometimes the program skips it!

☼Quick check. **What is the output of the following Python code?**

☼Answer is 8. The value of a is 3, which is smaller than 5, so the return value of the Boolean expression is false. We skip the compound statement of the true path

and directly execute the statement after If statement, that is, **print(a + 5), which is 8.**

☼Let's recap. A colon marks the start of a block.

In Python, an indentation is not just for readability but also for defining the scope of a block.

The number of whitespace for indentation is flexible but should be consistent for the same level in the same program.

☼Recalling scenario 4, we can implement the selection in Python now.

☼simply add if statement to check if the travel_dist is greater than 3000meters or not.

If yes, suggest a taxi,

☼ or ese, do nothing for that part.

☼quick summary of this lesson.

- Read slides.