BYU IDAHO CSE 111 | Programming with Functions ☼

01 Prepare: Review Python

During lesson 1 of CSE 111, you will review programming concepts that you learned in CSE 110.

Concepts

Here is a list of the Python programming concepts and topics from CSE 110 that you should review during this lesson.

1. We use the `input()` function to get input from a user in a terminal window.

2. A variable is a location in a computer's memory where a program stores a value. A variable has a name, a data type, and a value. In Python, the value and data type of a variable may change while the computer is executing a program.

3. In Python, we assign a value to a variable by using the assignment operator, which is the equals symbol (=).

4. Python has many data types including `str`, `bool`, `int`, `float`, `list`, `dict`, `function`, and `type`. This program demonstrates many of the Python data types.

5. It is possible to convert between many of the data types. For example, to convert from any data type to a string, we use the `str()` function. To convert from a string to an integer, we use the `int()` function, and to convert from a string to a float, we use the `float()` function. The `int()` and `float()` functions are especially useful to convert user input, which is always a string, to a number. See the program in example 1 below.

6. Python has many arithmetic operators including power (**), negation (-), multiplication (*), division (/), floor division (//), modulo (%), addition (+), and subtraction (-).

7. When we write an arithmetic expression that contains more than one operator, the computer executes the operators according to their precedence, also known as the order of operations. This table shows the precedence for the arithmetic operators.

Operators	Description	Precedence
( )	parentheses	highest
**	exponentiation (power)	↑
-	negation	|
* / // %	multiplication, division, floor division, modulo	|
+ -	addition, subtraction	↓
=	assignment	lowest

   When an arithmetic expression includes two operators with the same precedence, the computer evaluates the operators from left to right. For example, in the arithmetic expression x / y * c the computer will first divide *x* by *y* then multiply that result by *c*. If you need the computer to evaluate a lower precedence operator before a higher precedence one, you can add parentheses to the expression to change the evaluation order. The computer will always evaluate arithmetic that is inside parentheses first because parentheses have the highest precedence of all the arithmetic operators.

8. We use the `print()` function to display output to a user.

   The Python program in example 1 gets input from the user, converts the input to two numbers, computes the length of a cable from the two numbers, and prints the length rounded to two places after the decimal point.

```
 1   # Example 1
 2
 3   # Given the distance that a cable will span and the distance
 4   # it will sag or dip in the middle, this program computes
 5   # the length of the cable.
 6
 7   # Get user input and convert it from
 8   # strings to floating point numbers.
 9   span = float(input("Distance the cable must span in meters: "))
10   dip = float(input("Distance the cable will sag in meters: "))
11
12   # Use the numbers to compute the cable length.
13   length = span + (8 * dip**2) / (3 * span)
14
15   # Print the cable length in the
16   # console window for the user to see.
17   print(f"Length of cable in meters: {length:.2f}")
```

```
> python cable.py
Distance the cable must span in meters: 500
Distance the cable will sag or dip in meters: 18.5
Length of cable in meters: 501.83
```

   In example 1, the arithmetic that is written on line 13 comes from a well known formula. Given the distance that a cable must span and the vertical distance that the cable will be allowed to sag or dip in the middle of the cable, the formula for calculating the length of the cable is:

$$length = span + \frac{8\ dip^2}{3\ span}$$

9. In Python, we use `if` statements to cause the computer to make decisions; `if` statements are also called selection statements because the computer selects one group of statements to execute and skips the other group of statements.

10. There are six comparison operators that we can use in an `if` statement:

<	less than
<=	less than or equal
>	greater than
>=	greater than or equal
==	equal to
!=	not equal to

   Example 2 contains Python code that checks if a number is greater than 500.

```
# Example 2

# Get an account balance as a number from the user.
balance = float(input("Please enter the account balance: "))
```

```
 5
 6   # If the balance is greater than 500, then
 7   # compute and add interest to the balance.
 8   if balance > 500:
 9       interest = balance * 0.03
10       balance += interest
11
12   # Print the balance.
13   print(balance)
```

If you have written programs in other programming languages such as JavaScript, Java, or C++, you always used curly braces to mark the start and end of the body of an `if` statement. However, notice in example 2 that `if` statements in Python do not use curly braces. Instead, we type a colon (:) after the comparison of the `if` statement as shown on line 8. Then we indent all the statements that are in the body of the `if` statement as shown on lines 9 and 10. The body of the `if` statement ends with the first line of code that is not indented, like line 13.

It may seem strange to not use curly braces to mark the start and end of the body of an `if` statement. However, the Python way forces us to write code where the indentation matches the functionality or in other words, where the way we indent the code matches the way that the computer will execute the code.

11. Each `if` statement may have an `else` statement. We can combine `else` and `if` into the keyword `elif` as shown in example 3 on lines 9 and 11.

```
 1   # Example 3
 2
 3   # Get the cost of an item from the user.
 4   cost = float(input("Please enter the cost: "))
 5
 6   # Determine a discount rate based on the cost.
 7   if cost < 100:
 8       rate = 0.10
 9   elif cost < 250:
10       rate = 0.15
11   elif cost < 400:
12       rate = 0.18
13   else:
14       rate = 0.20
15
16   # Compute the discount amount and the discounted cost.
17   discount = cost * rate
18   cost -= discount
19
20   # Print the discounted cost for the user to see.
21   print(f"After the discount, you will pay {cost}")
```

12. Python includes two logic operators which are the keywords `and`, `or` that we can use to combine two comparisons. Python also includes the logical `not` operator. Notice in Python that the logical operators are literally the words: `and`, `or`, `not` and not symbols as in other programming languages:

```
if driver >= 54 or (driver >= 32 and passenger >= 54):
    message = "Enjoy the ride!"
```

Videos

If any of the concepts or topics in the previous list seem unfamiliar to you, you should review them. To review the unfamiliar concepts, you could rewatch some of the Microsoft videos about Python that you watched for CSE 110:

- Introducing Python
- Getting Started
- Configuring VS Code
- Input and print functions
- Demo of print function
- Comments
- String data type
- Numeric data types
- Conditional Logic
- Handling Multiple Conditions
- Complex Conditions

Documentation

Reading these tutorials may help you recall programming concepts from CSE 110.

- Why Choose Python?
- Interacting with Python
- Basic Data Types in Python
- Variables in Python
- Operators and Expressions in Python
- Conditional Statements in Python

You could also read some of the Python tutorials at W3 Schools. Or you could search for "Python" in the online catalog of the BYU-Idaho library and read one of the online books from the result set.