



## Session 3

### Modifying Program Flow

In this activity you will create a new program, insert a documentation string, show a menu to the user, ask the user to select a menu option, and then run code according to what menu option was selected.

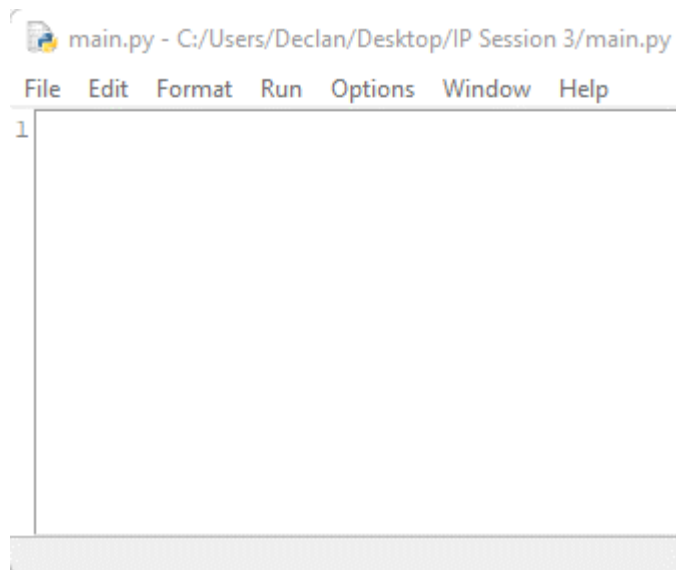
If at any point you encounter an error that you cannot resolve yourself, please ask your instructor for assistance. A copy of the completed exercise, including the answers to the challenge questions, are available on eLearn.

### Table of Contents

<b>main.py</b>	<b>2</b>
<b>Inserting the Docstring</b>	<b>2</b>
<b>Control Structures: Iterative Structure</b>	<b>2</b>
Displaying the Menu	3
<b>Control Structures: Selective Structure</b>	<b>3</b>
Energy Conversion	4
Invalid Input	5
Testing the formulae	5
<b>Challenge questions</b>	<b>6</b>

## main.py

1. Open **IDLE**.
2. Create a **new file** (CTRL + N or File → New File).
3. Open the **Save** dialog (CTRL + S or File → Save).
4. Create a new folder on your Desktop called IP Session 3.
5. Save your file in IP Session 3. Your file should be called main.py.
6. Go to the Options menu and click Show Line Numbers. Your interface should look like the image on the right.



## Inserting the Docstring

This program will be used to perform unit conversions (kilojoules to calories, Celsius to Fahrenheit and megabytes to gigabytes). Remember that your docstring should be the first thing in main.py and should follow the format below.

```
"""
Program Summary

Created by: <Your Name>
Created on: <Date Created>
Last edited: <Date Last Edited>
"""
```

## Control Structures: Iterative Structure

The first thing that our program is going to do is present a menu to the user so that they can select what type of conversion they want to perform, or so that they can quit the program. To do this, we'll use an infinite while loop. An infinite while loop will execute all the statements in its code block until the loop is broken with the break statement.

### While loop syntax

```
while condition:
    statements
```

1. Make sure there is a **blank line after your docstring**.
2. Insert a **while** loop. The *condition* for your loop should be **True**.  
**Note:** This causes an infinite loop because conditions are resolved as True or False. We're basically telling Python to run this block of code while True is True. After pressing Enter you should see that IDLE will automatically move your text insertion cursor (the blinking black bar) four spaces in from the left. If it hasn't, double check to make sure that you've included the colon after your condition, then try again.

## Displaying the Menu

We're going to display a simple menu to our user so that they can pick what they want to do in the application. We will be building the menu pictured on the right of this paragraph.

The menu will let the user know where in the program, and it will inform the user of the options that they can chose from, and it will tell the user how to proceed.

```
=====
Conversion Categories
=====
1. Energy
2. Temperature
3. Storage
q. Quit
Select a category>
```

1. Make sure that you're in the code block for your while loop. Your text insertion cursor should be on the line underneath where you inserted your while loop, and four spaces in from the left.
2. Use the **print** function to show **21 equals signs** on the screen. A lazy way of doing this is to use the statement `print('=' * 21)`
3. Use the **print** function to show '**Conversion Categories**' on the screen.
4. Use the **print** function to show **21 equals signs** on the screen.
5. Use the **print** function to show '**1. Energy**' on the screen.
6. Use the **print** function to show '**2. Temperature**' on the screen.
7. Use the **print** function to show '**3. Storage**' on the screen.
8. Use the **print** function to show '**q. Quit**' on the screen.
9. Use the **input** function to request that the user select a category. Store their response in a variable called **response**.

## Control Structures: Selective Structure

Now that the user has informed us where they want to go in the application, we can create a selective structure that uses that information. This will let us run a specific set of statements that apply only to the option that the user picked. To do this, we can use an if/elif/else structure.

**if/elif/else structure:**

```
if condition:
    statements
elif condition:
    statements
else:
    statements
```

1. Leave a blank line under your last line of code.
2. Insert an **if** statement. The condition will be **response == '1'**.  
**Note:** This makes Python check to see if the input stored in response is equal to '1'. Also take note that '1' is a string literal. This is because input gathered by the input function is stored as a string.
3. On the next line, insert the comment **# TODO: Energy conversion**.

4. Insert an **elif** statement. The condition will be **response == '2'**.
5. On the next line, insert the comment **# TODO: Temperature conversion**.
6. Insert an **elif** statement. The condition will be **response == '3'**.
7. On the next line, insert the comment **# TODO: Storage conversion**.
8. Insert an **elif** statement. The condition will be **response == 'q'**.
9. On the next line, insert the statement **break**. Remember from our learning materials that this statement terminates the loop, forcing us to exit it completely.
10. Insert an **else** statement.
11. On the next line, insert the comment **# TODO: Invalid input**.

Our if/elif/else selection structure should now look like this.

If the response is 1, do the energy conversion block. Else, if the response is 2, do the temperature conversion block. Else, if the response is 3, do the storage conversion block. Else, if the response is q, quit the application.

If no match was found, the user has provided an invalid input.

```
if response=='1':
    # TODO: Energy conversion.
elif response=='2':
    # TODO: Temperature conversion.
elif response=='3':
    # TODO: Storage conversion.
elif response=='q':
    break
else:
    # TODO: Invalid input.
```

## Energy Conversion

In this code block, we're going to ask the user to pick if they want to convert kJ to calories, or calories to kJ.

1. Find the comment **# TODO: Energy conversion**, and **remove TODO:**
2. Underneath the comment, use the **print** function to inform the user which section of the program they're in and the conversions available to them.
3. Underneath your print functions use the **input** function to request the user select an option, then store that response in a variable called **conversion\_response**.
4. Leave a blank line under your input function.
5. On the next line, enter an **if** statement to check if **conversion\_response == '1'**.
6. Use the **input** function again to request the user enter the value to convert, then store that response in a variable called **conversion\_value**.
7. The formula to convert kJ to calories is to divide kJ by 4.184. Create a variable called **converted\_value** and assign it a value of **float(conversion\_value)/4.184**.

**Note:** We've used the **float()** function to convert the text response provided by the user into a number with decimal places so that we can use it in mathematical equations. If we don't convert it into a number first our program will crash. We could also use **int()** if we were sure that the user would only use whole numbers.

8. Insert a **print** statement that reminds the user how many kJs they're converting, and how many calories that converts to (for example: 1 kJ is 0.239006 calories.)
9. On the next line, press **backspace** so that your text insertion cursor is at the same level as your if statement.
10. Enter an **elif** statement to check if **conversion\_response == '2'**.
11. Use the **input** function again to request the user enter the value to convert, then store that response in a variable called **conversion\_value**.
12. The formula to convert calories to kJ is to multiply calories by 4.184. Create a variable called **converted\_value** and store the converted value.
13. Insert a **print** statement that reminds the user how many calories they're converting, and how many kJ that converts to (for example: 1 calorie is 4.184 kJ).

Your energy conversion code block should look like the below.

```
if response=='1':
    # Energy conversion.
    print('Energy Conversions.')
    print('1. kJ to cal')
    print('2. cal to kJ')
    conversion_response = input('Select a conversion> ')

    if conversion_response=='1':
        conversion_value = input('Enter the value to convert> ')
        converted_value = float(conversion_value)/4.184
        print(conversion_value, 'kJ is', converted_value, 'calories.')
    elif conversion_response=='2':
        conversion_value = input('Enter the value to convert> ')
        converted_value = float(conversion_value)*4.184
        print(conversion_value, 'calories is', converted_value, 'kJ.')
```

## Invalid Input

1. Find the comment **# TODO: Invalid input**, and **remove TODO:**
2. Underneath the comment, use the **print** function to tell the user that they've made an invalid selection and then ask the user to try again.

Because invalid inputs have been kept as the last option to be evaluated in our if/elif/else structure, we don't need to add any further code. If the user provides an invalid input, our code will tell the user they made a mistake, and then the menu will be displayed again.

## Testing the formulae

1. **Run** your program (F5 or Run → Run Module).
2. If told that Source Must be Saved, click **OK** to save your file.
3. When asked to **select a category**, type **1**, then press **Enter**.
4. When asked to **select a conversion**, type **1**, then press **Enter**.

5. When asked to **enter a value**, type **1388** then press **Enter**.

What did your program tell you 1388 kJ is in calories? \_\_\_\_\_

1388 kJ is 331.7399617590822 calories.

6. When asked to **select a category**, type **1**, then press **Enter**.
7. When asked to **select a conversion**, type **2**, then press **Enter**.
8. When asked to **enter a value**, type **720**, then press **Enter**.

What did your program tell you 720 calories is in kJ? \_\_\_\_\_

720 calories is 3012.48 kJ.

## Challenge questions

Each of your practical activities will have challenge questions where you can test the skills and knowledge that you have learned this session and in the previous sessions while being given minimal direction.

**Challenge 1:** Using the instructions on completing the energy conversion code block to assist, complete the temperature conversion block. The user should be able to convert Fahrenheit to Celsius and Celsius to Fahrenheit.

**Formula to convert F to C:**  $(F - 32) \times (5/9)$

**Formula to convert C to F:**  $(C * (9/5)) + 32$

**Challenge 2:** Using the instructions on completing the energy conversion block to assist, complete the storage conversion block. The user should be able to convert megabytes to gigabytes and gigabytes to megabytes.

**Formula to convert MB to GB:**  $MB / 1024$

**Formula to convert GB to MB:**  $GB * 1024$

**Challenge 3:** Using the instructions on testing the formulae you've made; test the code you wrote for Challenges 1 and 2 to make sure your conversions are correct.

**F to C:** 89.6 F is 32.0 C

**C to F:** -4 C is 24.8 F

**MB to GB:** 8192 MB = 8.0 GB

**GB to MB:** 2 GB = 2048.0 MB

**Challenge 4:** Using the syntax provided in this week's learning materials, convert the if structure used by the looping menu into a match-case structure.

**Note:** To complete this challenge you must be using Python 3.10.0 or newer. You can see which version you're using by looking at the end of the IDLE title bars. For example, the screenshot below tells me that I'm using 3.10.0 rc1.

