Crafting Small Scripts, Converters, and Practical Tools in Python

By Doug Purcell

What We'll Learn

We will learn how to make scripts that asks the user a series of questions and then process the input to do various tasks such as converting the temperature to different units, computing auto loans, and handy translators.

Project: Your Biography

Let’s write a program that creates a biography for us. One way to do this is to ask a series of probing questions. Some questions that you may want to answer are things like:

- first name
- last name
- nationally
- birth place
- age
- height:
  - feet
  - inches
- weight (in pounds)
- favorite food
- favorite city

We won’t use this for a dating app, we just want to use this exercise as a means to explore some of the possibilities of python ;). You can add on any additional questions you want.

See How The Program Works in Action

```
$ python biography.py
```

Script Hints

Hint # 1

In PyCharm go to the directory where you'll place all of your programs. Right-click on the directory and select: New → Python File

Enter the name of the python file as *bio.py*.

Hint # 2

In the PyCharm editor add a function named `questions`. Inside the function is where all of the statements for your logic to go inside. You can use the `input` function to read in text form the terminal. If you need to read in an `int` or a `float` then you can pass the input function into the `int` or `float` functions.

```
weight = float(input('Enter weight in lbs: '))
```

To view a list of the built-in functions in python check out this url: https://docs.python.org/3/library/functions.html

Hint # 3: Include the following code snippet after the questions function

```
if __name__ == '__main__':
    # this is where your program starts
    questions()
```

This lets the python interpreter know where to start at. In every python file there's a `__name__` variable that's set equal to `__main__`. Therefore, if your file explicitly includes this then it will tell the python interpreter to start here.

A Template of How Things Should Look

```
def questions()
     """This is the part of the
     program that prompts the user"""
if __name__ == '__main__':
     questions()
```

How to read in multiple user input in a single statement?

Use the built in `split` function.

Solution

View full source here:

https://github.com/purcellconsult/Code-Cool-Stuff-With-Python/blob/master/sourcecode/ch 02/your bio.py

Project: Temperature Converter

Let’s code a temperature converter! There’s three commonly used scales for measuring temperature: Celsius (°C), Fahrenheit (°F), and Kelvin (K).

Demo the Script

```
$ python temperature_converter.py
```

Temperature Formulas

- F -> C = (F - 32) x 5/9
- F -> K = (F - 32)/1.8 + 273.15
- C -> F = (C x 9/5) + 32
- C -> K = C + 273.15
- K -> F = (K - 273.15) x 9/5 + 32
- K -> C = K - 273.15

Script Hints

`fahrenheit_to_celsius`: Converts the user input from Fahrenheit to Celsius.

`fahrenheit_to_kelvin`: Converts the user input from Fahrenheit to Kelvin.

`celsius_to_fahrenheit`: Converts the user input from Celsius to Fahrenheit.

`celsius_to_kelvin`: Converts the user input from Celsius to Kelvin.

`kelvin_to_fahrenheit`: Converts the user input from Kelvin to Fahrenheit.

`kelvin_to_celsius`: Converts the user input from Kelvin to Celsius.

`fahrenheit_to_celsius`: Converts the user input from Fahrenheit to Celsius.

fahrenheit_to_kelvin

```
def fahrenheit_to_kelvin():

	temp_in_fahren = float(input('Enter the temperature in
Kelvin '))

	kelvin = (temp_in_fahren - 32) / 1.8 + 273.15

	print(kelvin, 'K')
```

celsius_to_fahrenheit

```
def celsius_to_fahrenheit():

      temp_in_celsius = float(input('Enter the temperature
in Celsius '))

      celsius_to_fahren = (temp_in_celsius * 9 / 5) + 32

      print(celsius_to_fahren, '°F')
```

celsius_to_kelvin

```
def celsius_to_kelvin():

      temp_in_cel = float(input('Enter the temperature in
Celsius '))

      celsius_to_kel = (temp_in_cel + 273.15)

      print(celsius_to_kel, 'K')
```

Fill in the rest of the functions :-)

Running the script part I

```
if __name__ == '__main__':

            message = input("""Select one of the following options:

Type 'fc' to convert from Fahrenheit to Celsius.

Type 'fk' to convert from Fahrenheit to Kelvin.

Type 'cf' to convert from Celsius to

Fahrenheit.

Type 'ck' to convert from Celsius to Kelvin.

Type 'kf' to convert from Kelvin to Fahrenheit.

Type 'kc' to convert from Kelvin to Celsius.

Enter input here:

    """)
```

```
if __name__ == '__main__':
    message = input("""Select one of the following options:
Type 'fc' to convert from Fahrenheit to Celsius.
Type 'fk' to convert from Fahrenheit to Kelvin.
Type 'cf' to convert from Celsius to
 Fahrenheit.
Type 'ck' to convert from Celsius to Kelvin.
Type 'kf' to convert from Kelvin to Fahrenheit.
Type 'kc' to convert from Kelvin to Celsius.
Enter input here:
    """)

    # casefold is for case-insensitive comparisons

    message = message.casefold()
    if message == 'fc':
        fahrenheit_to_celsius()
    elif message == 'fk':
        fahrenheit_to_kelvin()
    elif message == 'cf':
        celsius_to_fahrenheit()
    elif message == 'ck':
        celsius_to_kelvin()
    elif message == 'kf':
        kelvin_to_fahrenheit()
    elif message == 'kc':
        kelvin_to_celsius()
    else:
        print('Not a valid option pal!')
```

Full solution to Temperature Converter

https://github.com/purcellconsult/Code-Cool-Stuff-With-Python/blob/master/sourcecode/ch_02/temperature_converter.py

Project: Auto loan Calculator

Cars are incredible mechanical inventions. Something they can be driven for countless of miles and still function properly is quite amazing. Like many quality things in life it costs money so in this project we're going to code a useful script that will help us make better purchasing decisions when deciding on a new car. The script will have two parts: one, how many months it will take to pay off the loan and two, what's the total interest paid accumulated over that period of time.

Demo Script

```
$ python autoloan_calculator.py
```

Formulas needed

A = P x (1 + r) ^N / (1 + r)^N – 1

Formula for calculating the accrued interest:

A = P(1 + rt)

The first formula looks tricky but it's simple once you know what all of the variables represent. Here's a quick overview:

- P = Principal or the amount owned on a loan.
- A = Total accrued amount (principal + interest).
- r = Rate of interest per year as a decimal, or interest rate / 100.
- N = Number of months in the loan period.

Script Hints

Create a function that calculates the monthly cost. You can break down the formula in order to minimize the chances of making a mistake. This reduces syntax errors and makes debugging potential arithmetic errors more straightforward.

View Solution on GitHub:

https://github.com/purcellconsult/Code-Cool-Stuff-With-Python/blob/master/sourcecode/ch02/autoloan calculator.py

Project: Mortgage Calculator

Getting a house is the American dream, but to obtain a dream does cost. In this exercise we're going to create a script that calculates the monthly payment that someone owes on a house. A *mortgage* is the type of loan that one takes out for a house.

Demo Script

```
$ python mortgage_calculator.py
```

A Really Simplified Formula

Mortgage calculators can get pretty complex, so in this exercise we're going to *stupify* the process so that we just calculate the monthly payments and total mortgage that one would owe contingent on some variables. These variables are:

- **Mortgage period**: How long the mortgage will last in years.
- **Principal**: The amount of money owed on the loan.
- **Interest rate**: The percentage of principal charged by the lender for the use of their money.

A formula that you can use to calculate the monthly payments on a house:

p x r (1 + r) ^ N / (1 + r) ^ N - 1

```
def mortgage():
        name = input('Enter your name ').capitalize()
        print(f"Time to calculate your mortgage payments {name}... ")
        principal = float(input('Enter in your principal '))
        interest_rate = float(input('Enter interest rate '))
        r = (interest_rate / 100) / 12
        n = int(input('Enter mortgage period (years) '))
        # get total number of months
        n = n * 12
        numerator = r * (1 + r) ** n
        deno = (1 + r) ** n - 1
        monthly_payment = principal * (numerator / deno)
        monthly_payment = round(monthly_payment)
        total_mortgage = monthly_payment * 30 * 12
        print(f'Monthly payment: ${monthly_payment}, total mortgage =
${total_mortgage}')

if __name__ == '__main__':
        mortgage()
```

Script Explanation

This script is very similar to the auto loan script except that the formula varies:

- The variables of principal, interest rate, r, and n are prompted from the user.
- The numerator and denominator of the function is calculated separately and then combined together at the end to emulate the formula: p x r (1 + r) ^ N / (1 + r) ^ N – 1
- Once the total monthly payment is calculated, the total mortgage is computed by taking the monthly payment and multiplying it by 30 and 12. The reason for this is because there's roughly 30 days in a month, and 12 months in a year.

View Full Source Code

https://github.com/purcellconsult/Code-Cool-Stuff-With-Python/blob/master/sourcecode/ch_02/mortgage_calculator.py

Project: Spanish Translator

Language translation is a complicated art. Even online translation tools created by teams of talented engineers at behemoth tech companies miss structural and linguistic elements in their translations. Therefore, to serve as a learning project we're going to simplify the process and focus on a narrow domain of words and phrases to transliterate.

What we’re going to do?

We’re going to create a python script that focuses on translating foods and general phrases.

Script Demo

```
$ python spanish_translator
```

Script Outline

```
def food():
    pass

def general_phrases():
    pass

if __name__ == '__main__':
    print('Bienvenidos! What phases would you like to translate?')
    print('1: Common foods ')

    print('2: General phases')
    your_choice = int(input("Enter your choice: '1' or '2'"))
        if your_choice == 1:
        food()
    elif your_choice == 2:
        general_phrases()
    else:
        print('Not a possible choice!')
```

```
def food():
    """Names for some popular mexican food items.
    Translate spanish food names to english.
    """
    print('\n')
    spanish_to_english = {
        'Birria': 'Spicy stew made with goat or mutton. ',
        'Quesadilla con carne': 'season steak strips. ',
        'Barbacoa': 'Slow cooked meat in soup. Beef, goat, or sheep.',
        'Burrito Banado': 'Wet Burrito.',
        'Huevos rancheros': '"Rancher\'s eggs." Corn tortillas, fried eggs, topped with warm salsa.',
        'Coctel de camarones': 'Shrimp cocktail served cold with tomato, onion, cucumber, and cilantro. ',
        'Huevos a la mexicana': 'Eggs, tomato, onion, and serrano chile. A classic.',
        'Huevo con Chorizo': 'Eggs and chorizo sausage.',
        'Burritos de Desayuno ': 'Breakfast burrito.',
        'Chilli con carne': 'Chili with meat.',
        'Lengua': 'Beef tongue, typically in tacos.',
        'Tripas': 'Small intestines of farm animals that\'s cleaned, boiled, and grilled. ',
        'Al pastor': 'Pork based taco based on shawarma',
        'Suadero': 'Tender slow cooked beef brisket. Typically served in tacos.',
        'Cabeza': 'Beef head/cheek meat. Served in soups or tacos.',
        'Sesos': 'Brains from either a goat or cow. Popular taco filling.'
    }

    print('Spanish phases available for translation:')
    for spanish, english in spanish_to_english.items():
        print(spanish)

    print()
    translate = input('Type in spanish phase you\'ll like to translate: ').capitalize()
    for english, spanish in spanish_to_english.items():
        if translate == english:
            print(spanish_to_english.get(translate))
            break
    else:
        print('Word is not available for translation')
```

Identical logic is used for `general_phrases` with the exception that the dictionary contains mappings of English to Spanish.

View the Full Source Code

https://github.com/purcellconsult/Code-Cool-Stuff-With-Python/blob/master/sourcecode/ch_02/spanish_translator.py